COMSOL MULTIPHYSICS

MEMS MODULE

USER'S GUIDE

VERSION 3.4

COMSOL

How to contact COMSOL:

Benelux
COMSOL BV
Röntgenlaan 19
2719 DX Zoetermeer
The Netherlands
Phone: +31 (0) 79 363 4230
Fax: +31 (0) 79 361 4212
info@femlab.nl
www.femlab.nl

Denmark
COMSOL A/S
Diplomvej 376
2800 Kgs. Lyngby
Phone: +45 88 70 82 00
Fax: +45 88 70 80 90
info@comsol.dk
www.comsol.dk

Finland
COMSOL OY
Arabianranta 6
FIN-00560 Helsinki
Phone: +358 9 2510 400
Fax: +358 9 2510 4010
info@comsol.fi
www.comsol.fi

France
COMSOL France
WTC, 5 pl. Robert Schuman
F-38000 Grenoble
Phone: +33 (0)4 76 46 49 01
Fax: +33 (0)4 76 46 07 42
info@comsol.fr
www.comsol.fr

Germany
FEMLAB GmbH
Berliner Str. 4
D-37073 Göttingen
Phone: +49-551-99721-0
Fax: +49-551-99721-29
info@femlab.de
www.femlab.de

Italy
COMSOL S.r.l.
Via Vittorio Emanuele II, 22
25122 Brescia
Phone: +39-030-3793800
Fax: +39-030-3793899
info.it@comsol.com
www.it.comsol.com

Norway
COMSOL AS
Søndre gate 7
NO-7485 Trondheim
Phone: +47 73 84 24 00
Fax: +47 73 84 24 01
info@comsol.no
www.comsol.no

Sweden
COMSOL AB
Tegnérgatan 23
SE-111 40 Stockholm
Phone: +46 8 412 95 00
Fax: +46 8 412 95 10
info@comsol.se
www.comsol.se

Switzerland
FEMLAB GmbH
Technoparkstrasse 1
CH-8005 Zürich
Phone: +41 (0)44 445 2140
Fax: +41 (0)44 445 2141
info@femlab.ch
www.femlab.ch

United Kingdom
COMSOL Ltd.
UH Innovation Centre
College Lane
Hatfield
Hertfordshire AL10 9AB
Phone:+44-(0)-1707 284747
Fax: +44-(0)-1707 284746
info.uk@comsol.com
www.uk.comsol.com

United States
COMSOL, Inc.
1 New England Executive Park
Suite 350
Burlington, MA 01803
Phone: +1-781-273-3322
Fax: +1-781-273-6603

COMSOL, Inc.
10850 Wilshire Boulevard
Suite 800
Los Angeles, CA 90024
Phone: +1-310-441-4800
Fax: +1-310-441-0868

COMSOL, Inc.
744 Cowper Street
Palo Alto, CA 94301
Phone: +1-650-324-9935
Fax: +1-650-324-9936

info@comsol.com
www.comsol.com

For a complete list of international representatives, visit
www.comsol.com/contact

Company home page
www.comsol.com

COMSOL user forums
www.comsol.com/support/forums

MEMS Module User's Guide

Version: October 2007 COMSOL 3.4

CONTENTS

Chapter 1: Introduction

References 3
Typographical Conventions 3

Overview **5**
What is the MEMS Module? 5
New Features in MEMS Module 3.4 7

Chapter 2: Creating and Analyzing MEMS Models

Basic Modeling Procedures **10**
Model Navigator 10
Options and Settings 11
Drawing the Geometry 12
Physics Settings 12
Coupling Application Modes 13
Meshing 13
Solving a Model 14
Analyzing Results Using the Postprocessing Menu 15

MEMS-Specific Modeling Techniques **17**
Moving Structures and ALE 17
Modeling Large Deformations 18
Modeling Electrostatic Fields 19
Thermal Effects in MEMS 21
Microfluidics 22
References 28

Example—Microresistor Beam **30**
Introduction 30
Model Definition 30
Results and Discussion 31

Modeling in COMSOL Multiphysics 32
Modeling Using the Graphical User Interface 33

Chapter 3: The MEMS Module Application Modes

Overview of Application Modes **52**
Application Mode Guide 52

Chapter 4: Structural Mechanics Application Modes

Solid, Stress-Strain **60**

Plane Stress **61**

Plane Strain **62**

Axial Symmetry, Stress-Strain **63**

Theory Background **64**
Strain-Displacement Relationship. 64
Stress-Strain Relationship. 66
Thermal Strain. 78
Initial Stress and Strain. 80
Follower Loads 81
Implementation 81
Contact Modeling 86

Application Mode Description **89**
Properties 90
Scalar Variables 91
Material Properties 93
Constraints 101
Load 105
Thermal Coupling 108
Damping 110

Initial Stress and Strain 111
Perfectly Matched Layers (PMLs) 112
Contact Modeling 117

Fluid-Structure Interaction **126**
Theory Background 126
Application Mode Description 127
Example Model 130

Thermal-Structure Interaction **131**
Theory Background 131
Application Mode Description 131
Example Model 132

Contact Modeling **133**
Constraints 133
Contact Pairs 133
Boundary Settings for Contact Pairs 134
Multiphysics Contact 135

Coordinate Systems **136**
Global Coordinate System 137
Local Geometrical Coordinate System 137
User-Defined Coordinate System 138

Damping **145**
Rayleigh Damping 145
Loss Factor Damping 146
Equivalent Viscous Damping 147
Explicit Damping 147
No Damping 148

Symbols for Loads and Constraints **149**
Load Symbols 149
Constraint Symbols 150

Solver Settings **152**
Symmetric Matrices 152

Selecting Iterative Solvers 153
Specifying the Absolute Tolerance 154
Solver Settings for Contact Modeling 155

Chapter 5: Piezoelectric Application Modes

Theory Background **158**
The Piezoelectric Effect 158
Piezoelectric Constitutive Relations 158
Material Models 160
Electrical Formulations 161

The Piezoelectric Application Modes **164**
Application Mode Properties 164
Scalar Variables 165
Material Properties 166
Electric Boundary Conditions 176
Constraints 180
Loads and Charges 182
Structural Damping 184
The Piezo Solid Application Mode 186
The Piezo Plane Stress Application Mode 186
The Piezo Plane Strain Application Mode 187
The Piezo Axial Symmetry Application Mode 187

Chapter 6: Film Damping Application Modes

Theory Background **193**
Conditions for Film Damping 193
The Modified Reynolds Equation 194
Structural Loads 195
Gas Outflow Conditions 196
Rarefaction and Slip Effects 197
Modified Reynolds Equation for Perforated Structures 198
Geometry Orientations 199

References 200

The Film Damping Application Mode **201**
Variables and Space Dimensions 201
Properties 202
Scalar Variables 203
Boundary Settings 203
Edge and Point Settings 206
Application Mode Variables 207

Predefined Structural-Film Damping Multiphysics Couplings **208**
Solid, Stress-Strain with Film Damping 208
Plane Strain with Film Damping 209
Axial Symmetry, Stress-Strain with Film Damping. 210

Chapter 7: Electrostatics Application Modes

Electrostatic Fields **212**
Conductive Media DC Application Mode. 212
Electrostatics Application Mode 217

Floating Potentials and Electric Shielding **222**
Floating Potentials 222
Electric Shielding 223
Example Model—Floating Potential 224
Model Definition 224
Results and Discussion. 224
Modeling Using the Graphical User Interface 225

Periodic Boundary Conditions **230**
User Interface for Periodic Conditions 230

Infinite Elements **232**
Known Issues When Modeling Using Infinite Elements. 235
Reference 236

Force and Torque Computations **237**
Computing Electromagnetic Forces and Torques 237

Lumped Parameters **241**
Calculating Lumped Parameters with Ohm's Law 241
Calculating Lumped Parameters Using the Energy Method 242
Lumped Parameters in the MEMS Module 243

Chapter 8: Microfluidics Application Modes

Laminar Flow Theory **249**
PDE Formulations 249
Boundary Conditions 253

Application Modes for Laminar Flow **264**
Variables and Space Dimension 264
Application Mode Properties 264
Scalar Variables 265
Subdomain Settings 265
Boundary Conditions Settings 270
Point Settings 276
Application Mode Variables 277

General Laminar Flow **278**

Incompressible Navier-Stokes **279**

Weakly Compressible Navier-Stokes **280**

Stokes Flow **281**

Weakly Compressible Stokes Flow **282**

The Level Set Method **283**
Subdomain Equations 283
Boundary Conditions 287

Reference 288

The Level Set Method for Two-Phase Flow **289**
Subdomain Settings 289
Boundary Conditions 291
Example Model 292
Numerical Stabilization 292
Reference 293
Application Mode Variables 293

Convection and Diffusion **294**
The Convection and Diffusion Application Mode 294

Electrokinetic Flow **299**
Transport Balances 299
The Electrokinetic Flow Application Mode 300

Predefined Multiphysics Couplings **305**
Flow with Species Transport 305
Electroosmotic Flow 305

Chapter 9: The Materials/Coefficients Library

Using the MEMS Material Libraries **310**
Loading Material Properties from the Library 311
The Material Libraries 313
The Liquids and Gases Library and Material Property Functions 313
Using Material Property Functions 314
Editing Material Properties 319
Piezoelectric Material Properties Library 321
MEMS Material Properties Library 322

Chapter 10: Glossary

Glossary of Terms **326**

INDEX **331**

1

Introduction

The term MEMS is an acronym for MicroElectroMechanical Systems. In this context,

- "Micro" means that the devices are extremely tiny, in the order of micrometers or smaller.
- "Electro" means that some electrical component is involved.
- "Mechanical" means that the system performs some mechanical motion.
- "Systems" refers to the fact that all these features are combined in one package.

MEMS technology exploits the existing microelectronics infrastructure to create complex machines on a micrometer scale. Extensive applications for these devices exist in both commercial and industrial systems. Well-known components such as integrated silicon pressure sensors, accelerometers, and motion detectors have found use for several years in automotive and industrial applications. Research activity in microfluidics is changing medical-diagnostic processes such as DNA analysis, and it is spurring the development of successful commercial products.

It is common knowledge that MEMS are made of tiny electromechanical components, but some engineers do not appreciate the fact that almost all MEMS devices involve multiple areas of physics—multiphysics. At a minimum, MEMS

devices involve at least the electrical and mechanical sciences. It is also common that the electronic and mechanic elements are coupled through thermal or electrochemical effects, thereby adding a third or fourth physical phenomenon to the system.

This multiphysics nature of MEMS devices requires that the system designer has a vast understanding and knowledge of these various branches of physics. Because some microscale effects are totally new or behave differently than at the macroscale, engineers require new system-design philosophies. They likely find it difficult to split one design into parts, which is common for macroscale device, where one engineer can fully study the mechanics while others concentrate, for instance, on the electrical or thermal aspects. Thus the MEMS engineer is a true systems designer, handling several physical phenomena simultaneously—and COMSOL Multiphysics and the MEMS Module can do the same.

Most MEMS devices are manufactured using lithography-based microfabrication, a technology that the microelectronics industry has refined for highly integrated circuits. Thanks to these efforts there are excellent methodologies and facilities for mass production. Suppliers can thus set the price of microsystems at a totally different level compared to their macro-scale counterparts.

Lithographic fabrication techniques, however, do present some limitations on geometrical structures in MEMS devices. Microfabrication is based on planar technology where the components are usually flat. From the modeling point of view, a flat structure presents some challenges, specifically in mesh generation and in finding numerical solutions. To get accurate solutions, the shape of a single triangle (2D) or tetrahedron (3D) in the mesh should be as regular as possible. In flat structures you can achieve regularity by decreasing the mesh size to accommodate the shortest distances, but doing so increases memory requirements. Fortunately, advances in modeling techniques such as mesh rescaling, mesh mapping, and mesh extrusion can reduce the mesh size and relieve memory demands. In addition, certain industries have moved away from the use of silicon in favor of glasses and plastics, and we are now seeing the emergence of chips in biotechnology that include microfluidic systems that can be regarded as true MEMS devices (Ref. 1).

MEMS pioneers in both research and industrial organizations have solved several challenges in the modeling and manufacturing areas. They have been able to apply existing tools to help analyze the behavior of MEMS devices through numerical methods. Meanwhile, it is apparent that the time is right to introduce new tools that demonstrate how well finite-element-based numerical solvers perform in this area. We are confident that this product, the MEMS Module, can help researchers, designers,

and instructors explore the electro-thermal-chemical-mechanical behavior of microsystem components and further develop this fascinating new area.

The documentation set for the MEMS Module consists of three books. The one in your hands, the *MEMS Module Users Guide*, introduces the basic modeling process. The tutorial section discusses the modeling strategies in MEMS devices and demonstrates their use in an illustrative example. It summarizes the MEMS-specific techniques used in the model library, and finally it presents the various application modes available in the module including structural analysis, predefined piezoelectric couplings, electrostatics and DC currents in conductive media, and microfluidics.

The second book in the set, the *MEMS Module Model Library*, includes models showing how to use the MEMS Module for the simulation of sensors, actuators, microfluidics devices, and other MEMS devices. Each model comes with theoretical background as well as step-by-step instructions that illustrate how to set it up. Further, they all come as Model MPH-files so you can import them into COMSOL Multiphysics for immediate access. This way you can follow along with the printed discussion as well as use them as a starting point for your own modeling needs.

There ia also a *MEMS Module Reference Guide*, which provide in-depth information about the application modes and their variables.

We hope the MEMS Module becomes a valuable tool in your modeling work, and we are convinced that the effort you put into understanding COMSOL Multiphysics will be repaid several times over. If you have any feedback on the models in this set, please let us know. Likewise, if you have any ideas for additional models that we could add to the library, we welcome your suggestions. Finally, if in your work you have developed a model you think would be a good candidate for inclusion in this model set, please let us hear about it. In any case, feel free to contact us at `info@comsol.com`.

References

1. Julian Gardner and others, *Microsensors, MEMS and Smart Devices*, John Wiley & Sons, 2001.

2. System Planning Corporation, Market Survey, 1999.

Typographical Conventions

All COMSOL manuals use a set of consistent typographical conventions that should make it easy for you to follow the discussion, realize what you can expect to see on the

screen, and know which data you must enter into various data-entry fields. In particular, you should be aware of these conventions:

- A **boldface** font of the shown size and style indicates that the given word(s) appear exactly that way on the COMSOL graphical user interface (for toolbar buttons in the corresponding tooltip). For instance, we often refer to the **Model Navigator**, which is the window that appears when you start a new modeling session in COMSOL; the corresponding window on the screen has the title **Model Navigator**. As another example, the instructions might say to click the **Multiphysics** button, and the boldface font indicates that you can expect to see a button with that exact label on the COMSOL user interface.
- The names of other items on the graphical user interface that do not have direct labels contain a leading uppercase letter. For instance, we often refer to the Draw toolbar; this vertical bar containing many icons appears on the left side of the user interface during geometry modeling. However, nowhere on the screen will you see the term "Draw" referring to this toolbar (if it were on the screen, we would print it in this manual as the **Draw** menu).
- The symbol **>** indicates a menu item or an item in a folder in the **Model Navigator**. For example, **Physics>Equation System>Subdomain Settings** is equivalent to: On the **Physics** menu, point to **Equation System** and then click **Subdomain Settings**. **COMSOL Multiphysics>Heat Transfer>Conduction** means: Open the **COMSOL Multiphysics** folder, open the **Heat Transfer** folder, and select **Conduction**.
- A `Code` (monospace) font indicates keyboard entries in the user interface. You might see an instruction such as "Type `1.25` in the **Current density** edit field." The monospace font also indicates COMSOL Script codes.
- An *italic* font indicates the introduction of important terminology. Expect to find an explanation in the same paragraph or in the Glossary. The names of books in the COMSOL documentation set also appear using an italic font.

Overview

This manual describes the MEMS Module version 3.4. It is intended to give you an introduction to the modeling stages in COMSOL Multiphysics™ and the MEMS Module, provide a detailed example of how to work with the models in this set, as well as serve as a reference for more advanced modeling techniques and available application modes.

Chapter 2, "Creating and Analyzing MEMS Models," gives you the knowledge necessary to start using the MEMS Module. It also explains how to model moving structures with the ALE (arbitrary Lagrangian-Eulerian) formulation, how to handle electric and thermal effects, when to use the large-deformation formulation in structural analysis, and how to describe fluid flow in microfluidic applications. Chapter 3, "The MEMS Module Application Modes," summarizes the available application modes. Chapters 4 through 8 review in detail the contents of the Structural Mechanics, Piezoelectric, Film Damping, Electrostatics, and Microfluidics application modes. Chapter 9, "The Materials/Coefficients Library," gives an overview of the most common materials and their properties, which can serve as realistic input data in your MEMS models.

What is the MEMS Module?

The MEMS Module is a collection of application modes and models for COMSOL Multiphysics with which you can model various MEMS devices and applications. It includes application modes for modeling of electrostatics, structural mechanics, piezoelectricity, film damping, and microfluidics. The MEMS Module Model Library shows how to use these application modes to model actuators, sensors, and microfluidic devices. The comprehensive documentation outlines the theoretical basis of MEMS-specific phenomena and the application modes in the module, and it also contains step-by-step instructions for creating the models.

The MEMS Module seamlessly connects to COMSOL Multiphysics and the other add-on modules in the COMSOL Multiphysics product line. Therefore, Heat Transfer by Conduction, Convection and Conduction, and other application modes that might prove useful for a particular application are available for MEMS modeling. You can also view and modify the models in terms of the underlying PDEs. The software thus offers a unique transparency because the model equations are always visible, and you have

complete freedom in the definition of phenomena that are not predefined in the module.

In MEMS, the operation of devices are often affected by several physical phenomena that are coupled by some mechanism. Traditionally, these couplings are described pairwise using, for example, electro-structural, fluid-structural, and thermo-mechanical couplings. Although COMSOL Multiphysics does not pose any limitations on the number or type of such couplings, these pairwise couplings are important building blocks in complex multiphysics models containing any number of coupled physical phenomena.

The following table summarizes the most important MEMS couplings and some common devices you can model using the MEMS Module. It also outlines the structure of the MEMS Module. The first column of the table lists phenomena, couplings, and devices that are often associated with the word electromechanical in the narrow and literal meaning of MEMS. The devices in this category are usually various kinds of actuators and sensors. The microfluidic devices, although using some of the same manufacturing and miniaturizing techniques, form a totally different application area. The "Fluid-Structure Interaction" column lists various techniques and phenomena that are useful for both electromechanical and microfluidic applications, where movement and deformation of solids are of concern. The last column, "Microfluidics," lists transport phenomena that are key issues in, for example, lab-on-a-chip type devices.

This table, however, shows only the tip of the iceberg—our view of the most important applications where you can use the MEMS Module. In your hands, the multiphysics combinations and applications of the MEMS Module are unlimited.

TABLE 1-1: EXAMPLES OF COUPLED PHENOMENA AND DEVICES YOU CAN MODEL USING MEMS MODULE

	ELECTROMECHANICAL	FLUID-STRUCTURE INTERACTION	MICROFLUIDICS
PHENOMENA / COUPLING	Electro-structural Electro-thermal Thermo-mechanical Piezoelectric Piezoresistive Prestress modal analysis Stress stiffening	Moving boundary using ALE technique Squeezed-film damping	Pressure-driven flow Electroosmotic flow Electrophoresis Dielectrophoresis Electrothermal flow Mass transport using diffusion, migration, and convection
DEVICES	Cantilever beams Comb drives Resonators Micromirrors Thermomechanical actuators Inertial sensors Pressure sensors	Mechanical pumps and valves	Lab-on-a-chip devices Microfluidic channels Microreactors Micromixers Nonmechanical pumps and valves MEMS heat exchangers

New Features in MEMS Module 3.4

- Improved application modes for piezoelectric modeling. You can now model piezoelectric devices that consist of piezoelectric, dielectric, and structural parts within the piezoelectric application mode. There is also improved compatibility with other application modes for electrostatics and electric currents and a new boundary condition for floating potentials. See "Piezoelectric Application Modes" on page 157.
- A new material library with about 25 common piezoelectric materials. See "Piezoelectric Material Properties Library" on page 321.
- Support for exporting reaction models from the COMSOL Reaction Engineering Lab to the MEMS Module for inclusion in microfluidics models. See "Including Chemical Reactions in Microfluidics Models" on page 27
- An application mode for multispecies mass transport by convection and diffusion. See "Convection and Diffusion" on page 294.

- Improved application mode for film damping, including slide film damping. See "Film Damping Application Modes" on page 191.
- New and updated models, including models of a SAW gas sensor, an elevator button using piezoresistivity, a MEMS gyroscope, and a tortuous reactor. See the *MEMS Module Model Library* for full documentation of these and all other models in the MEMS Module Model Library.

2

Creating and Analyzing MEMS Models

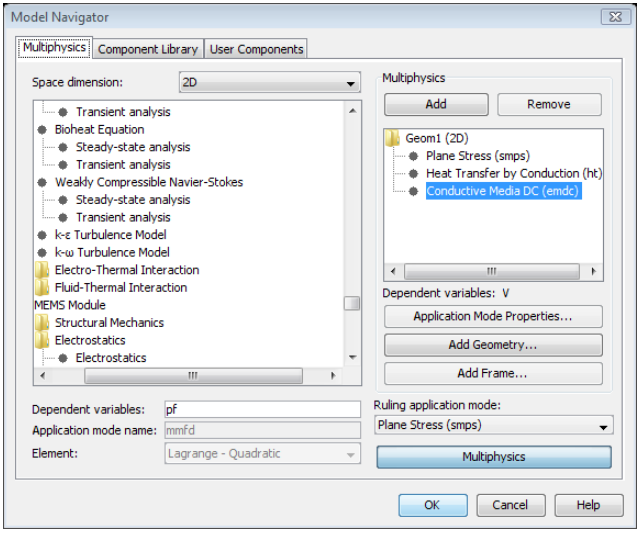
This chapter familiarizes you with modeling procedures useful when working with the MEMS Module. Because the module is fully integrated with COMSOL Multiphysics, the design and analysis process is very similar to the one you use in the base environment. Separate sections cover generic and MEMS-specific tricks and tips that should prove valuable during the modeling phase.

Basic Modeling Procedures

In COMSOL Multiphysics, the toolbar buttons and menus mirror the basic procedural flow during a modeling session. You work from left to right in the process of defining, drawing, meshing, modeling, solving, and postprocessing a problem. This manual maintains a consistent style when describing the introductory model in this chapter as well as the models in the MEMS Model Library. The format includes headlines corresponding to each major step in the modeling process. More details about basic modeling techniques appear in the *COMSOL Multiphysics User's Guide*.

Model Navigator

To begin setting up a model, activate the **Model Navigator**, which appears when you start COMSOL Multiphysics—or, if you are already in the COMSOL Multiphysics work area, select **New** from the **File** menu or click on the **New** button on the Main toolbar. The **Model Navigator** opens a page where you specify the application mode, names of dependent variables, and the nature of the problem (static, transient, eigenfrequency, frequency response, or parametric).



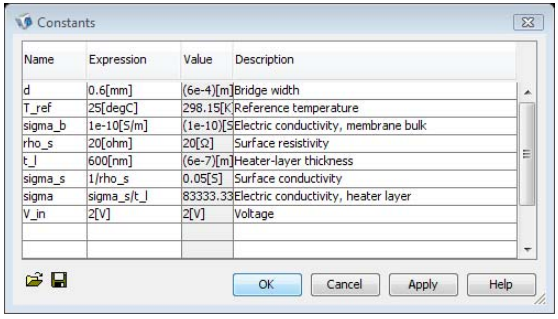
In the Model Navigator you select and configure one or more application modes.

By clicking the **Multiphysics** button you can set up a combination of application modes taken either from the MEMS Module, from other optional discipline-specific modules, or those in the core COMSOL Multiphysics package. It is also possible to add or remove an application mode at any time by opening the **Model Navigator** from the **Multiphysics** menu. In addition, you can change an application mode's properties by selecting the **Physics>Properties** menu. However, you cannot change the dependent variable directly; to do so, you must remove the application mode, add it back again, and enter another name for the variable.

Options and Settings

COMSOL Multiphysics provides a number of dialog boxes where you can configure the model. These settings are all accessible from the **Options** menu, and a few are accessible by double-clicking on the Status bar; for example, in that fashion you can turn the grid on and off. Basic settings—for example, those for the axes and grid spacing—are available from the **Options** menu.

If you want to enter constant parameters for the model, open the **Constants** dialog box.



In the Constants dialog box you enter model parameters and can also evaluate simple formulas.

The dialog-box choices under the **Options>Expressions** path recognize a number of basic functions including trigonometric operators (sin, cos, tan), exponentials (exp, log), and logical expressions (<, >, ==). In addition, user-defined expressions can contain custom variables calculated from existing dependent and application-specific derived variables.

The several selections for coupling variables under the **Options** menu (for instance, **Options>Integration Coupling Variables**) allow you to select a variable that can integrate, extrude, or project data among multiple application modes and geometries.

Another choice under **Options**, the **Material Library** dialog box, allows you to access a number of materials and their properties supplied with the software as well as maintain libraries of your own, user-defined materials.

Drawing the Geometry

To set up a model's geometry requires knowledge of how to use the **Draw** menu and the Draw toolbar. It further helps to be aware of some general aspects of drawing a geometry:

- To get the most accurate geometry possible, enter dimensions and coordinates using dialog boxes instead of drawing it. Creating and modifying objects graphically might seem easier, but in some cases that method can introduce small variations in numerical values. To help you avoid geometric conflicts, we suggest entering parameters directly in dialog boxes. You can later correct and modify these and other geometrical values by selecting the menu sequence **Draw>Object Properties**.
- Use the correct scale from the start. It is tempting to first create a geometry using long integers numbers (for instance, 42.5 instead of 42.5e-6 for a dimension that is 42.5 micrometer) to reduce the amount of keyboard entry and then later apply a scaling factor. However, in some cases—in particular when you create objects graphically rather than using the geometry object dialog boxes—mismatch problems can arise due to numerical rounding when later downscaling the geometry.
- Use work planes when creating 3D geometries. Several options are available to create 2D work planes that in turn generate 3D geometries; for instance, you can take a 2D geometry and embed, extrude, or revolve it into a full 3D geometry.

Physics Settings

After drawing the geometry, the next step is to specify the model's physics. You normally define material properties in the **Subdomain Settings** dialog box, whereas you typically go to the **Boundary Settings** dialog box to specify constrains, external forces and effects (such as mechanical constraints or loads), voltages, and temperatures.

To achieve a deeper understanding of the underlying physics of an application mode, study its **Equation System Settings** dialog boxes. They summarize how the software translates the material properties and boundary conditions into the Coefficient form, General form, or Weak form partial differential equations. They also list variables available for coupling and analysis purposes.

Coupling Application Modes

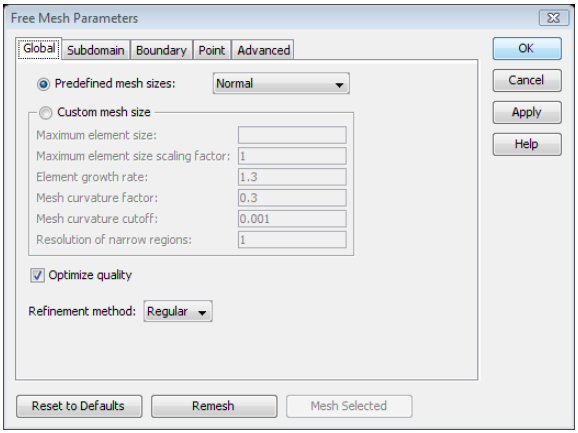
The easiest and computationally most efficient way to create couplings between application modes is through material properties and boundary settings. Dependent and derived variables can couple the solution from one application mode to another. If the coupling is unidirectional, you can solve the application modes sequentially using the **Solver Manager** to specify which variables to solve for and also create a solver script for automating the solution process.

If the dependent or derived variables are insufficient to describe the physical coupling, several extra couplings are available under the **Options** menu. You can integrate over subdomains or boundaries as well as extrude or project data among application modes and geometries. Note, however, that these couplings introduce intermediate geometries and mappings that add to the calculation load, so the model takes longer to compute.

Meshing

At this stage the software is ready to mesh the defined geometry. In typical macroscale cases you simply click one of the **Mesh** buttons on the Main toolbar. However, in MEMS models it is often necessary to work with specialized meshes. Specifically, with the flat structures typical in MEMS designs, you can define appropriate mesh scale factors by going to the **Main** menu and selecting **Mesh>Free Mesh Parameters**. In the resulting dialog box, click the **Advanced** tab where you can decrease the required mesh size yet obtain results that provide a quick understanding of the modeled phenomena. You can also use mesh-mapping and mesh-extrusion techniques for best operation

with MEMS devices. A more detailed description of meshing appears in the *COMSOL Multiphysics User's Guide*.



The Global page in the Free Mesh Parameters dialog box gives you control of the mesh size.

In MEMS models you often want to apply advanced techniques such as mesh scaling. In such cases, however, mesh quality (its regularity) can be poor, dropping to less than 0.1; an acceptable value is >0.1 in 3D geometries and >0.3 in 2D geometries. You should be aware of this effect because it might dramatically influence the results.

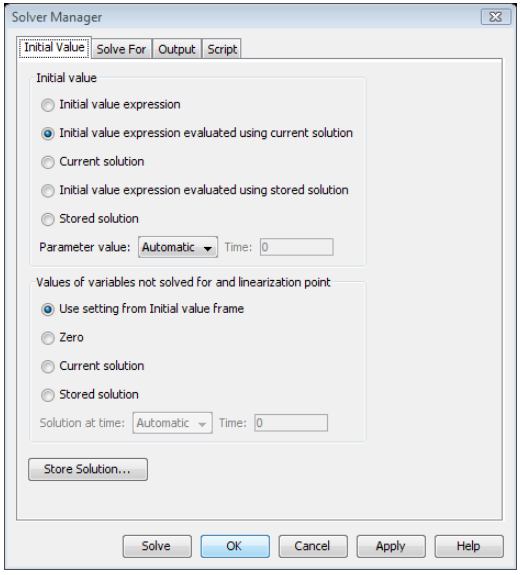
On the other hand, structured meshes, i.e. rectangular or brick meshes do not suffer from this problem, and they behave well even their aspect ratio is large.

Solving a Model

Most MEMS models are coupled-multiphysics problems, so pay attention to the choice of solver. In most cases, the auto-selected solver parameters are well tuned for performance and accuracy in the solution process. If the auto-selected solver does not converge and you have double-checked the material and boundary settings as well as the mesh quality, it is time to try another solver. Unfortunately there are no universal solver settings that always work well with the tightly coupled multiphysics models. It is thus a good idea to study already solved examples and listen to the experience of senior engineers to expand your own solution knowledge base. You can also find more information about solving and analyzing the model convergence in the "Solving the Model" chapter in the *COMSOL Multiphysics User's Guide* and in "Modeling Guidelines" on page 7 in the *COMSOL Multiphysics Modeling Guide*.

A parametric analysis can solve difficult nonlinear static problems. By adding advisory loads you can guide the solver to find the solution from the proper direction. The "Thermomechanical Microvalve" on page 122 in the *MEMS Module Model Library* illustrates this technique.

The **Solver Manager** dialog box contains settings for initial values, stored solutions, and which application modes or dependent variables to solve simultaneously. If the model involves only unidirectional couplings, you can split the solution process into sequential parts. A script language automates sequential solutions.



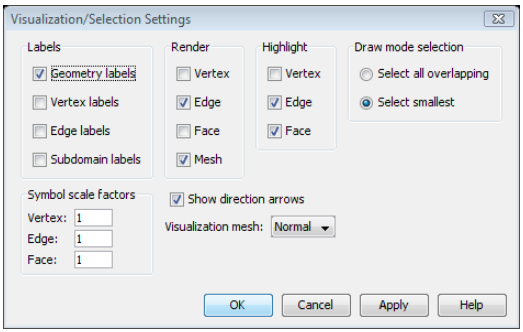
In the Solver Manager dialog box, go to the Initial Value page to make initial-value and linearization-point settings for your model.

Analyzing Results Using the Postprocessing Menu

The **Postprocessing** menu provides tools for postprocessing and visualizing model quantities. You are not limited to working with predefined quantities because you can calculate any function of variables by entering the appropriate expression.

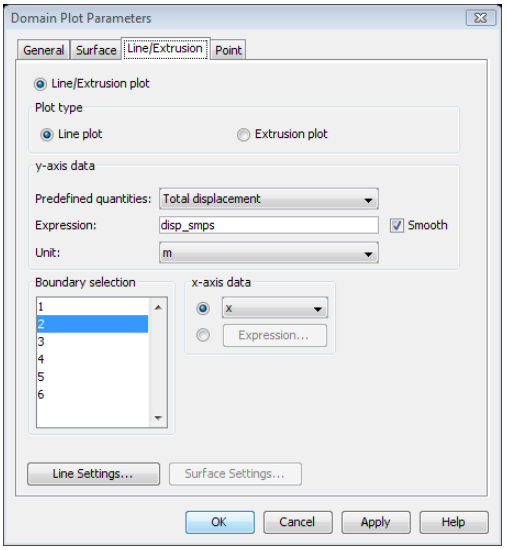
To access certain label and axis settings for postprocessing plots, choose **Options>Visualization/Selection Settings**. Another set of tools lets you filter the

visualizations by suppressing the rendering of subdomains or boundaries; find these tools in the **Options>Suppress** menu item.



This dialog box, which you open from the Options menu, lets you control the rendering of edges and faces.

In addition to the available postprocessing tools in COMSOL Multiphysics, you can export the results from a domain or cross-section plot to an ASCII file for further analysis with other tools.



Use the Domain Plot Parameters dialog box, accessible from the Postprocessing menu, to create plots of a quantity's restriction to a set of domains that you select.

MEMS-Specific Modeling Techniques

An important issue to consider when modeling microscopic devices is that they deform easily, and in many cases you must account for this deformation. The arbitrary Lagrangian-Eulerian (ALE) method has proven useful for modeling deforming structures that have electro-structural couplings or fluid-structural interactions. COMSOL Multiphysics' Moving Mesh Application Mode lets you study both static deformations and time-dependent deformations where the geometry changes its shape due to the dynamics of the problem. Further, you can consider most electric fields in MEMS designs as static, and you generally model conductivity only at boundaries. Other important phenomena that deserve careful consideration in MEMS modeling include thermal effects such as expansion and residual stress stiffening due to manufacturing processes. Finally, pressure-driven and electrokinetic fluid flow are standard techniques in microfluidic applications. This section takes a brief look at all these areas with MEMS designs in mind.

Moving Structures and ALE

The ALE (arbitrary Lagrangian-Eulerian) description is a common method for simulating deforming computational domains (Ref. 1). It found its first use in fluid mechanics and has since been successfully applied to finite element analysis in solid mechanics. In MEMS design, it has several applications for modeling structural deformations such as in electrostatically actuated comb drives, microswitches, microvalves, and microfluidic applications.

An example where the ALE method proves quite useful is the electrostatically actuated cantilever beam depicted in Figure 2-1. An air gap exists between the thin conductive layers on the beam's surface and the substrate; an electric potential difference across the gap generates a force that bends the beam. When the gap between the conductive layers decreases, that motion changes the electrostatic field and the attracting force. If this force is strong enough, the structure could even collapse. To simulate the coupling between the mechanical deformations and the electric potential, the model must account for the change in the geometry and distortions in the calculation mesh.

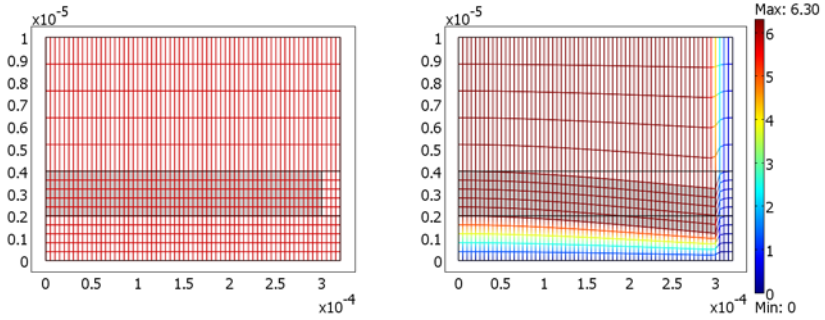


Figure 2-1: An electrostatically actuated cantilever beam (the grayed area) deforms due to an electrostatic force. The original geometry and mesh appears on the left, while the right image illustrates the geometry and mesh after deformation. In the right image, colors on the actual computer display show the electrostatic potential. Note also, that the vertical axis has been rescaled to improve visualization: the full width of the beam is 320 μm but the height is only 10 μm.

Normally COMSOL Multiphysics determines and solves the PDEs in a fixed mesh (reference frame). By adding a Moving Mesh (ALE) application mode to your model, you introduce a moving mesh (and a corresponding frame) for the ALE implementation. Then you can add and formulate the physics equations requiring a moving mesh in this frame; the software automatically transforms the equations, and uses them to calculate results in the reference and moving frames. For examples of using the Moving Mesh (ALE), see the models Cantilever Beam on page 8, Comb Drive on page 28, and ALE Fluid-Structure Interaction on page 294 of the *MEMS Module Model Library*.

Modeling Large Deformations

In models that include the Plane Stress; Plane Strain; Axial Symmetry, Stress-Strain; and Solid, Stress-Strain application modes, you must consider whether or not to activate the large-deformation option (refer to "Theory Background" in Chapter 4 for details); the default value is Off. For instance, calculations for large deformations can handle stress-stiffening effects. A special case requiring large deformations involves eigenfrequency calculations with residual stress. Without this option, the solver omits any static stresses you specified in the load settings. If you are unsure whether or not to use this option instead of the normal stress formulation, try to solve the problem both with and without it and check for differences. The advantage of using the normal stress formulation is in calculation speed—the model might solve several times faster.

In deciding whether to include large deformations, consider the following guidelines.

The first rule of thumb says that if the deformation is of the same rough magnitude as the structure's dimension, then the large-deformation option should be active. In practice this means using it when the deformation is approximately 10% of the structure's dimension. This effect, where large deformations have the restriction of small strains, is also sometimes referred to as a nonlinear geometric effect. In using the large-deformation option, the continuum mechanical application mode replaces the normal strain with the Green strain. It also replaces the stress with the second Piola-Kirchhoff stress, and then solves the problem using a total Lagrangian formulation.

Another phenomenon for which you should use the large-deformation option is stress stiffening. Stiffness normally arises when the structure bends and stretches. In MEMS devices the structure might contain built-in residual stresses due to the manufacturing processes. The large-deformation option takes these effects into account.

When solving a static problem with the large-deformation option set, be sure to use the nonlinear solver. In eigenvalue problems, such as in the thin-film resonator model, a more accurate solution results when you first solve the static stress value, which then serves as a linearization point for the eigenvalue problem.

The large-deformation option imposes a larger computational load than the normal formulation and is not always needed. In unclear cases it is easy to perform a quick comparison test. For example, the results for the 3D Accelerometer model described on page 146 of the *MEMS Module Model Library* are identical with and without the large-deformation option. However, the solution time with the normal formulation is many times shorter (in this case 20 times).

Several models in the *MEMS Module Model Library* use the large-deformation analysis type for the continuum mechanical application modes. This includes models in the chapter "MEMS Actuator Models" on page 7 of the *MEMS Module Model Library*—the Cantilever Beam, Comb Drive, Micromirror, Thin-Film Resonator, and Thermomechanical Microvalve models—as well as the Pressure Sensor and Fluid-Structure Interaction models.

Modeling Electrostatic Fields

Most MEMS devices meet the electroquasistatic criterion (Ref. 2) in which the wavelength of the applied fields is substantially larger than the modeled device. Thus

you can treat modeled electric fields as static and you create your model with the Electrostatics application mode or the Conductive Media DC application mode.

ELECTROSTATICS

The Electrostatics application mode solves for the electric potential V using the equation

$$-\nabla \cdot (\varepsilon \nabla V) = \rho .$$

In MEMS designs you rarely need to consider the details of the charge distribution, ρ, inside materials. Therefore charge exists only as a thin layer on a conductor's surface, and the electric field near its surface is always perpendicular to that surface.

To calculate electrostatic forces in the MEMS Module, Maxwell's stress tensor method is available. In the Electrostatics application mode the software calculates the force by integrating

$$\mathbf{n}_1\mathbf{T}_2 = -\frac{1}{2}(\mathbf{E} \cdot \mathbf{D}) + (\mathbf{n}_1 \cdot \mathbf{E})\mathbf{D}^T$$

on the surface of the object that the force acts on. $\mathbf{E}$ is the electric field, $\mathbf{D}$ is the electric displacement, and $\mathbf{n}_1$ is the outward normal from the object. In practice you model the electric fields not only in the air but also inside the solid materials. By entering the electromagnetic force variable name into the subdomain settings you activate force calculations, and the application mode automatically generates new subdomain and boundary variables.

For examples of electrostatic forces, you can study the Cantilever Beam and Comb Drive models. These models include electrostatic forces coupled to the boundaries of moving structures.

To derive a system's capacitance, as seen in the Capacitive 3D Comb Drive and Pressure Sensor models, the integral of the electric energy density (W_e) is calculated

$$C_{ii} = 2 \int_{\Omega_{mov}} W_e d\Omega$$

where i is the port number. To study capacitances, use the Port boundary condition. You can also model multiport system and calculate the capacitances between them. In that case each port must have an unique port number. On the port where the value of the input parameter (electric potential) should be forced to one, select the **Use port as inport** check box.

Keep in mind that the MEMS Module's Electrostatics application mode also contains electric-shielding and floating-potential boundary conditions, all of which are valuable tools for modeling MEMS devices.

CONDUCTIVE MEDIA

The Conductive Media DC application mode uses the equation

$$-\nabla \cdot (\sigma \nabla V - \mathbf{J}^e) = Q_j$$

where σ is the conductivity, $\mathbf{J}^e$ is an externally generated current density, and Q_j is the volume current source, which in almost all MEMS applications you can set to zero.

Conductors in MEMS designs are usually thin sheets. Thus, from a modeling standpoint it is unnecessary to describe a conductor as a subdomain, and you can instead account for it with the electric-shielding boundary condition

$$-\nabla_t \cdot (d\sigma \nabla V) = 0$$

where d is the thickness of the conductive layer. You can manually enter the voltage constraints in the point and edge settings. An example of using this technique appears in the model Thermomechanical Microvalve on page 122 of the *MEMS Module Model Library*.

You can use port boundary conditions to calculate resistances or conductances of the system. If you, for example, have two ports, Port 1 and Port 2, and use Port 1 as an input port you get two variables that are globally available:

- The conductance matrix elements `G11_emdc` and `G21_emdc` if voltage is the input property.
- The resistance matrix elements `R11_emdc` and `R21_emdc` if current or current density is the input property.

Thermal Effects in MEMS

HEAT TRANSFER

Heat transfer is characterized by three mechanisms: *conduction*, *convection*, and *radiation*. Conduction—heat transfer by diffusion in a solid or fluid medium due to a temperature gradient—is the most common mechanism in solid MEMS actuators and sensor models. In microfluidics, a model must account for convection—heat transfer within a moving fluid. From the modeling standpoint, heat transfer in MEMS

is similar to macro-scale heat-transfer problems, which means that you can employ the Heat Transfer by Conduction application mode and the Heat Transfer by Convection and Conduction application mode in the core COMSOL Multiphysics package.

THERMAL EXPANSION

Thermal-expansion coupling is an important phenomenon in MEMS modeling. Microactuators, such as seen in the thermomechanical microvalve or microresistor beam models, use thermal expansion as an acting force. The fabrication processes for MEMS devices leave thermal residual stresses in the structures, causing them to bend or change their mechanical characteristics as exemplified in the resonator model.

Thermal-expansion coupling is available as an option in the Plane Strain; Plane Stress; Axial Symmetry, Stress-Strain; and Solid, Stress-Strain application modes. In them you can specify the thermal-expansion coefficient and two temperature fields, T and T_{ref}. These temperatures can be any mathematical expression and are typically variables solved for in another application mode, for instance the Heat Transfer application modes. Example models that illustrate the use of thermal expansion are Residual Stress in a Thin-Film Resonator on page 73, Thermomechanical Microvalve on page 122, and Capacitive Pressure Sensor on page 194 of the *MEMS Module Model Library*, and Microresistor Beam described on page 30 in the current chapter.

THERMOELASTIC COUPLING

When a product of the thermal expansion coefficient and the temperature of the structure at rest is large, mechanical vibrations cause significant oscillations of the temperature field. The resulting energy transfer from the mechanical into thermal domain is called thermoelastic damping. For an example model (in both 2D and 3D) that illustrates the used of full thermoelastic coupling, see Thermoelastic Damping in a MEMS Resonator on page 106 in the *MEMS Module Model Library*.

Microfluidics

From the modeling standpoint, the first questions in dealing with fluid flow in microfluidic devices are which application modes to use and which boundary conditions to apply. For models in the MEMS Module, the General Laminar Flow application modes cover most situations. An exception is the description of electrophoretic transport of charged species in dilute solutions, where the Electrokinetic Flow application mode is applicable. See the chapter "Microfluidics Application Modes" on page 247 for details. Also, the Convection and Diffusion application mode is useful for modeling of mass transport.

The design and optimization of microfluidic devices (pumps, valves, mixers, reactors, and sensors) is a process that must take into account the multiphysics couplings between fluid mechanics and other phenomena, such as mass and heat transport, electromagnetics, or structural mechanics. This section focuses mainly on pressure-driven and electrokinetic flow.

PRESSURE-DRIVEN FLOW

In most cases, flow models in the MEMS Module are based on the incompressible Navier-Stokes equations

$$\rho\frac{\partial \mathbf{u}}{\partial t} - \eta\nabla^2\mathbf{u} + \rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{F}$$

$$\nabla\cdot\mathbf{u} = 0.$$

One exception is the squeezed-film damping expression, which is a simplified description of pressure-driven flow.

For pressure-driven flow the modeling processes are identical in the macro- and microscales. You generally apply a driving force at the boundaries by defining an inlet velocity or a pressure drop over the system. In such cases, the body force, $\mathbf{F}$, in the Navier-Stokes equations is zero.

One outstanding feature in the General Laminar Flow application mode is the simplicity of defining a velocity profile at an inlet or outlet. The application mode contains predefined settings for fully developed laminar flow.

The slip boundary condition required in microfluidics models depends on the Knudsen number

$$K_n = \frac{\lambda_m}{d},$$

which is the ratio of the molecular mean free path, λ_m, to the gap width, d. In practice, rarefied gases where $10^{-3} < K_n < 10^{-1}$ require a slip-velocity boundary condition (for example, 30% no-slip and 70% slip).

You can model fluid-structure interactions using the ALE technique and the Moving Mesh (ALE) application mode, and an example appears in the model ALE Fluid-Structure Interaction on page 294 of the *MEMS Module Model Library*.

ELECTROKINETIC FLOW

Electrokinetics is a general term describing phenomena that involve the interaction between solid surfaces, ionic solutions, and applied electric fields. Two important classes of electrokinetics are electrophoresis and electroosmosis, where the motions of solid bodies and fluids, respectively, occur when an external electric field is applied to a system. In microfluidic devices you apply electrokinetics to manipulate fluids and move particles for sample handling and chemical separation.

Electrokinetics fall broadly into two classifications, DC and AC electrokinetics, as the following table shows. DC electrokinetics includes both electrophoresis and electroosmosis. AC electrokinetics breaks down into AC electroosmosis, dielectrophoresis, and electrothermal phenomena.

TYPE OF FORCE	DC ELECTROKINETICS	AC ELECTROKINETICS
Surface force on fluid	Electroosmosis	AC electroosmosis
Force on suspended particles	Electrophoresis	Dielectrophoresis
Body force on fluid	-	Electrothermal

Electroosmosis finds wide use for sample injection and transport in microchannels. Electrophoresis is applicable to capillary gel electrophoresis for the fractionation of DNA, while capillary-zone electrophoresis is suitable for the separation of chemical species.

Electroosmosis

When a polar liquid (such as water) and a solid surface (such as glass or a polymer-based substrate) come into contact, they form an electric double layer (EDL). Its characteristic thickness comes from the Debye length

$$\lambda_D = \sqrt{\frac{\varepsilon k T}{2 z^2 F^2 c_\infty}}$$

where k is Boltzmann's constant, T is the temperature, z is the ion's valance number, F is Faraday's constant, and c_∞ is the ion's molar concentration in the bulk solution. Due to the generated charge distribution, an external electric field causes ions in the EDL to move. Their collective movement induces fluid motion in the channel, creating electroosmotic flow. If the relative thickness of the EDL compared to the channel diameter is small, you can model the electroosmotic flow velocity with the Helmholtz-Smoluchowski equation (Ref. 3)

$$u_{\text{eof}} = -\frac{\varepsilon\zeta \boldsymbol{E}}{\eta}$$

where $\boldsymbol{E}$ is the applied electric field, η is the liquid's dynamic viscosity, and ζ is the zeta potential you determine empirically from electroosmotic flow measurements. In the MEMS Module you model electroosmotic flow using the predefined Electroosmotic Flow multiphysics coupling, which solves this electric field using the Conductive Media DC application mode, and then applies this as a boundary condition in an application mode for the laminar flow. For a practical example of this technique, see Low-Voltage Electroosmotic Micropump on page 335 in the *MEMS Module Model Library*.

Electrophoresis

Electrophoresis is closely related to electroosmosis. However, whereas in electroosmosis the EDL arises between the fluid and the channel surface, in electrophoresis the EDL exists between ions and the fluid. Thus you calculate the electrophoretic velocity relative to the suspended fluid for the ion particles within the subdomain and, if necessary, add it to the velocity field obtained from the Incompressible Navier-Stokes application mode. For that procedure, use the following expressions for the electrophoretic velocity of the charged molecules (Ref. 3)

$$u_{\text{ep}} = \frac{q\boldsymbol{E}}{6\pi\eta r_0} = \frac{2}{3}\frac{\varepsilon\zeta \boldsymbol{E}}{\eta}.$$

The first form is suitable when the total charge, q, and the radius of the molecule are known; the second form is more appropriate for very small particles where you know the zeta potential ζ. If the relative thickness of the EDL layer is small compared to particle radius, the dynamics reduce to

$$u_{\text{ep}} = \frac{\varepsilon\zeta \boldsymbol{E}}{\eta},$$

which is the same Helmholtz-Smoluchowski equation used for electroosmosis flow except it now calculates the movements of the ions within the fluid.

AC Electroosmosis

Because an alternating electric field does not generate a net force in the EDL, AC electroosmosis is not appropriate for fluid transport in microfluidics. However, the fast back-and-forth movements an AC field generates are useful for mixing purposes. To model AC electroosmotic flow when the frequency of the electric field is sufficiently

low, use the same Helmholtz-Smoluchowski equation as in DC electroosmosis. With increasing frequency, AC electroosmosis becomes less important.

Dielectrophoresis

A dielectrophoresis (DEP) force comes about when a non-uniform electric field acts on polarizable particles suspended in a fluid. For a spherical particle you can calculate the relative velocity from

$$u_{\mathrm{DEP}} = \frac{\varepsilon_m r_0 \mathrm{Re}(K) \nabla E^2_{\mathrm{rms}}}{3\eta}$$

where ε_m is the medium's complex permittivity, r_0 is the radius of the particle's equivalent homogeneous sphere, $\mathrm{Re}(K)$ is the dielectrophoretic mobility, and η is the dynamic viscosity (Ref. 5). The complex permittivity, ε^*, for an isotropic homogeneous dielectric is

$$\varepsilon^* = \varepsilon - i\sigma/\omega$$

where ε is the electric permittivity, σ is the electric conductivity, and ω is the angular field frequency. The dielectrophoretic mobility is the real part of the Clausius-Mosotti factor

$$K = \frac{\varepsilon_p - \varepsilon_m}{\varepsilon_p + 2\varepsilon_m},$$

which depends on the particle's complex permittivity, ε_p, and that of the medium, ε_m. To solve for the gradient of the squared rms electric field, use this equation:

$$\nabla E^2_{\mathrm{rms}} = \nabla(\mathbf{E}_{\mathrm{rms}} \cdot \mathbf{E}_{\mathrm{rms}})$$

where you obtain $\mathbf{E}_{\mathrm{rms}}$ from the Electrostatics application mode and apply the rms value for the boundary voltages.

ELECTROTHERMALLY-DRIVEN FLOW

Electrothermal body forces arise from electrically generated nonuniform heating, also called Joule heating, which creates variations in conductivity and permittivity and thus Coulomb and dielectric body forces. It is possible to determine the resulting fluid motion by solving the Navier-Stokes equations with the electrothermal body force. You can also simulate electrothermally driven flow by solving the quasi-static electric field for a specific geometry. The electrothermal force equals

$$\mathbf{F} = -0.5\left[\left(\frac{\nabla\sigma}{\sigma} + \frac{\nabla\varepsilon}{\varepsilon}\right)\cdot\mathbf{E}\frac{\varepsilon\mathbf{E}}{1+(\omega\tau)^2} + 0.5|\mathbf{E}|^2\nabla\varepsilon\right]$$

where σ is the conductivity, ε is the fluid's permittivity, ω is the angular frequency of the electric field, and $\tau = \varepsilon/\sigma$ is the fluid's charge-relaxation time. The electric-field vector $\mathbf{E}$ contains the amplitude and direction of the AC electric field but not its instantaneous value.

As a result of the heating, ε and σ are temperature dependent, and you can write their gradients as function of the temperature gradient, $\nabla\varepsilon = (\partial\varepsilon/\partial T)\nabla T$ and $\nabla\sigma = (\partial\sigma/\partial T)\nabla T$. With water, for example, the relative change rate for the permittivity and conductivity are $(1/\varepsilon)(\partial\varepsilon/\partial T) = -0.004$ 1/K and $(1/\sigma)(\partial\sigma/\partial T) = 0.02$ 1/K, respectively.

For a practical implementation of electrothermally-driven flow, see the model AC Electrokinetically Enhanced Surface Reactions on page 242 of the *MEMS Module Model Library*.

INCLUDING CHEMICAL REACTIONS IN MICROFLUIDICS MODELS

With the COMSOL Reaction Engineering Lab you can model chemical reactions. You can extend these models to space-dependent microfluidics models in the MEMS Module by exporting reaction engineering models from the Reaction Engineering Lab to the following application modes:

- General Laminar Flow
- Convection and Diffusion
- Electrokinetic Flow

For more information, see the *COMSOL Reaction Engineering Lab User's Guide*. For an example model, see Hydrocarbon Dehalogenation in a Tortuous Microreactor on page 418 of the *MEMS Module Model Library*.

SQUEEZED FILM DAMPING

Figure 2-2 illustrates a situation that arises in many MEMS devices (Ref. 4): A narrow gap, h, formed by two horizontal plates, is filled with a gas such as air. The lower plate is fixed, while the upper plate is movable. When the upper plate moves down, it squeezes the gap, and gas flows out from its edges. The narrow pathway restricts the flow, which causes the gas pressure to increase. The increased pressure interacts with

the plates' surfaces and decelerates their movement. If the plate moves up, the situation reverses. This dissipative force is called *squeezed film damping*.

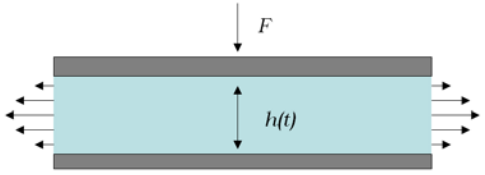


Figure 2-2: The basic concept behind squeezed film damping.

To model squeezed film damping in the MEMS Module, use the Film Damping application mode, which implements film damping on the boundaries of the model. This application mode, which is described in the chapter "Film Damping Application Modes" on page 191, also handles the related phenomenon of slide film damping.

The MEMS Module also provides predefined multiphysics couplings that define a structural mechanics application mode (Plane Strain in 2D, Axial Symmetry, Stress-Strain in 2D axial symmetry, and Solid, Stress-Strain in 3D) and a Film Damping application mode. The structural deformation from the structural mechanics application mode then automatically defines the boundary deformation in the Film Damping application mode. The Film Damping application mode also provides the ability to constrain the pressure p to 0 at the edges of the boundary as well as other gas film end conditions.

See the model Squeezed-Film Gas Damping in an Accelerometer on page 146 of the *MEMS Module Model Library* for a 2D and a 3D example of how to use the Film Damping application mode.

References

1. J. Donea and others, "Arbitrary Lagrangian-Eulerian Methods," *Encyclopedia of Computational Mechanics*, John Wiley & Sons, 2004.

2. S.D. Senturia, *Microsystem Design*, Kluwer Academic Publishers, 2001, Appendix B: Electromagnetic Fields.

3. N. Nguyen and S.T. Wereley, *Fundamentals and Applications of Microfluidics*, Artech House, 2002.

4. S.D. Senturia, *Microsystem Design*, Kluwer Academic Publishers, 2001, Chapter 13.4.

5. C. Meinhart, D. Wang, and K. Turner, "Measurement of AC Electrokinetic Flows," *Biomedical Microdevices*, vol. 5, pp. 139–145, 2003.

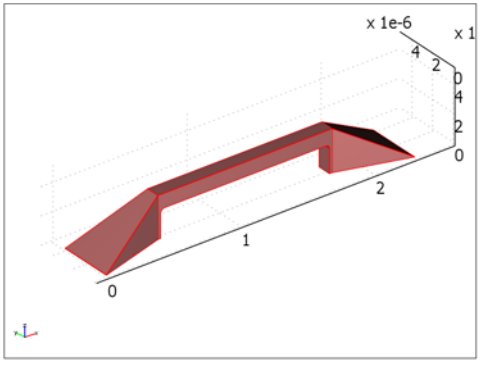
Example—Microresistor Beam

Introduction

To help you understand how to create models with the MEMS Module, this section walks through one in detail. The example illustrates the ability to couple thermal, electrical, and structural analysis in one model. This particular application moves a beam by passing a current through it; the current generates heat, and the temperature increase leads to displacement through thermal expansion. The model estimates how much current and increase in temperature are necessary to displace the beam.

Although the model involves a rather simple 3D geometry and straightforward physics, it provides a good example of multiphysics modeling because it contains several application modes added incrementally to the model. Note that this model of a microresistor beam also appears in the in the MEMS Module Model Library's Actuator Models folder under the name `microresistor_beam`.

Model Definition



Microbeam geometry.

A copper microbeam has a length of 13 µm with a height and width of 1 µm. Feet at both end bond it rigidly to a substrate. An electric potential of 0.2 V applied between the feet induces an electric current. Due to the material's resistivity, the current heats up the structure. Because the beam operates in the open, the generated heat dissipates into the air. The thermally induced stress loads the material and deforms the beam.

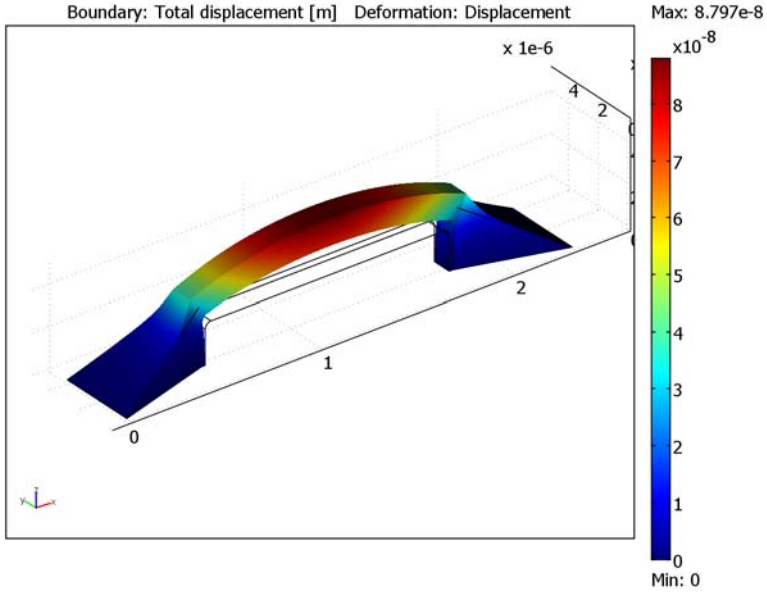
As a first approximation, assume the electric conductivity is constant. However, a conductor's resistivity increases with temperature. In the case of copper, the relationship between resistivity and temperature is approximately linear over a wide range of temperatures

$$\rho = \rho_0(1 + \alpha(T - T_0)) . \tag{2-1}$$

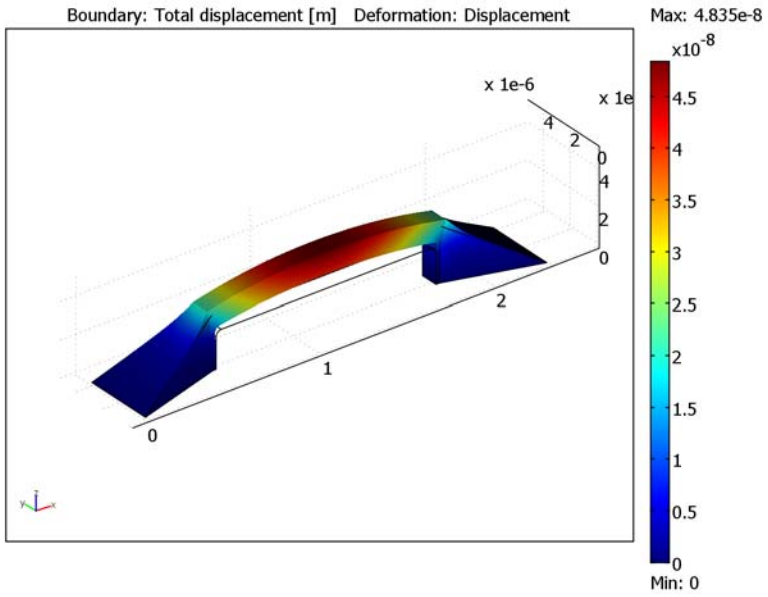
You obtain the conductor's temperature dependency from the relationship that defines electric resistivity; conductivity is simply its reciprocal ($\sigma = 1/\rho$).

Results and Discussion

The next two figures compare the microbeam's deformation given constant and then temperature-dependent conductivity, respectively. The maximum displacement for constant conductivity is 88 nm ($8.79 \cdot 10^{-8}$ m), whereas for the temperature-dependent case the displacement is 48 nm ($4.83 \cdot 10^{-8}$ m). The plots scale the deformation by 20 to emphasize the difference.



Microbeam deformation with constant electric conductivity.



Microbeam deformation with temperature-dependent electric conductivity.

Modeling in COMSOL Multiphysics

This example creates the 3D geometry from two 2D work planes; the first one views the geometry from above, and the second does so from the side. To draw the 2D work geometries you use the line tool with specific axes and grid settings, then extrude the work geometries into 3D. The final step is to create a composite geometry of the extruded objects. You can also skip the step-by-step instructions for the geometry creation and import the ready-made geometry directly from the Model Library.

In this model, three application modes describe the physics: Heat Transfer by Conduction from COMSOL Multiphysics, and two from the MEMS Module: the Solid, Stress-Strain and Conductive Media DC application modes. For modeling the material, this example takes copper from one of the MEMS Module's material libraries.

The overall modeling approach is as follows: In the Conductive Media DC application mode, all boundaries—except the two bases where you apply the potential difference—are insulated. You then enter the resistive-heating variable, `Q_emdc`, into the heat-source term of the Heat Transfer by Conduction application mode. Next set the

base boundaries facing the substrate to a constant temperature of 323 K. You model the convective air cooling in other boundaries using heat flux boundary conditions with a heat transfer coefficient, h, of 5 W/(m^2·K) and external temperature, T_{inf}, of 298 K. You then obtain the thermally induced stress by including a thermal-expansion term in the Solid, Stress-Strain application mode. Standard constraints handle the bases' rigid connection to the substrate.

The following procedure constructs the model sequentially: First enter only the Conductive Media DC application mode's physical settings and solve for the electric potential. Next add the Heat Transfer by Conduction settings and analyze the temperature distribution. Then calculate the deformations with the Solid, Stress-Strain application mode. Finally model the temperature dependency of the electric conductivity using Equation 2-1. COMSOL Multiphysics solves both the Conductive Media DC and Heat Transfer by Conduction application modes simultaneously and then, using the stored solution, it solves the Solid, Stress-Strain application mode.

Model Library path: `MEMS_Module/Actuator_Models/microresistor_beam`

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

1. Begin a new COMSOL Multiphysics session by selecting **File>New** or by clicking the **New** button. Doing so invokes the **Model Navigator**, in which you choose the application modes required for the model.
2. In the **Model Navigator** click the **New** tab, and in the **Space dimension** list select **3D**. You can now see all the application modes available in 3D.
3. Still on the **New** page, click the **Multiphysics** button. It opens the **Multiphysics** area where you specify the geometries and associated application modes.
4. In the same window from the list of **Application modes** on the left, select the **MEMS Module>Electrostatics>Conductive Media DC** application mode.
5. Click the **Add** button. Doing so gives the model a new geometry **Geom1 (3D)** with the associated **Conductive Media DC** application mode.

   Next add two more application modes: Heat Transfer by Conduction and Solid, Stress-Strain.

6 Within the list of application modes on the left side of the dialog box, select **COMSOL Multiphysics>Heat Transfer>Conduction** and then click the **Add** button.

7 From the same list of application modes, select **MEMS Module>Structural Mechanics>Solid, Stress-Strain** and click **Add** once again.

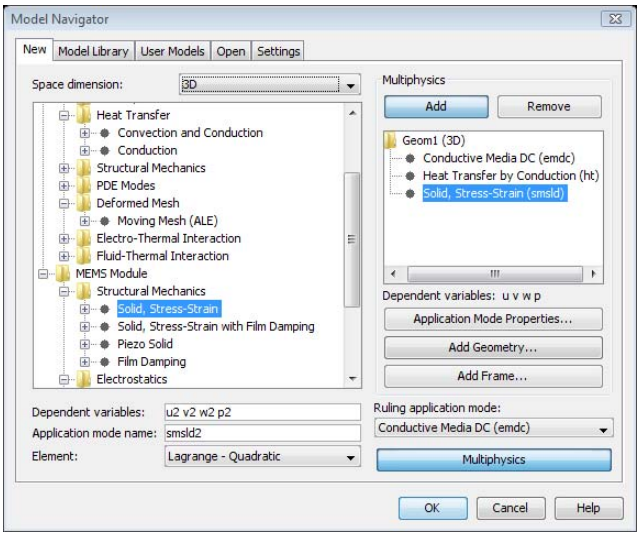


Figure 2-3: The application modes used for the microbeam model definition.

At this point, the model includes the three physics modes along with their corresponding equations and user interfaces with which you describe the subdomain settings and boundary conditions. To review the dependent variables' application-mode specific properties, go to the **Multiphysics** area, select an application mode, and then click the **Application Mode Properties** button. If you make a mistake during the setup process, simply remove an unwanted application mode or add a new one.

8 When the setup is complete, close the **Model Navigator** by clicking **OK**.

OPTIONS AND SETTINGS

It is possible to create and solve a model taking several approaches. In general, though, you probably first sketch the geometry, then supply material property values in subdomain and boundary settings. The documentation follows the order of the items as they appear on the Main menu. Thus this discussion leaves the **File** menu, skips over the **Edit** menu and picks up with selections available through the **Options** menu.

You first might want to set the axis limits to make the geometry more easy to examine. Suitable values in for this example are: **x min** `0`; **x max** `25e-6`; **y min** `0`; **y max** `5e-6`; **z min** `0`; **z max** `5e-6`. To set them, select **Options>Axes/Grid Settings**, clear the **Auto** check box, and enter those values. Note that for cases where a geometry already exists, manual scaling is not necessary—after drawing or modifying an object, simply click the **Zoom Extents** button on the Main toolbar to scale the view automatically.

Now start setting constants necessary for the model.

Constants

1 From the **Options** menu choose **Constants**.

2 Enter a constant with the **Name** `Voltage` and for its **Expression** enter `0.2[V]`. It is also advisable, although not required, to enter a description for the constant. In this case you can enter `Applied voltage` in the **Description** edit field.

3 Click **OK**.

You can later conduct experiments by changing the values of this constant. No further options or settings are now necessary, so start creating the geometry by using commands and functions on the **Draw** menu and the Draw toolbar.

GEOMETRY MODELING

In this model you create the geometry by drawing objects in 2D work planes and then extrude and combine them in 3D geometry. As discussed earlier in this chapter, in 2D you can create the geometry either by drawing objects with the mouse or by entering coordinates in dialog boxes. The graphical way is faster, whereas the direct-entry method is more precise. If you prefer the drawing method, you can activate grid points to snap the mouse on specific coordinates; to do so, click **SNAP** on the Status bar at the bottom of the screen. To change the density of the grid for snapping, go to **Options>Axes/Grid Settings**, then to the **Grid** page, clear the **Auto** button if it is activated, and make appropriate entries in the edit fields.

If you prefer entering values for geometry coordinates, note that Shift-clicking many of the buttons in the Draw toolbar immediately brings up the corresponding dialog box. This shortcut makes a quick way of specifying rectangles, circles, and other objects. Throughout the discussions in this manual you can select the method you prefer at any given time. The following discussion gives the drawing instructions.

You can also skip this geometry modeling chapter and import a ready-made geometry directly from a CAD file. To do so select **File>Import>CAD Data From File** and browse for the file `micro_beam3d.mphbin`. You can find this file in the MEMS Module's

Model Library path (*<COMSOL Installation path>*/models/MEMS_Module/Actuator_Models/microresistor_beam.mphbin).

Work-Plane Settings

1 From the **Draw** menu choose **Work-Plane Settings**.

2 Verify the default values on the **Quick** page: **Plane x-y**; **z** = **0**. Then click **OK**. This step adds a new work plane, **Geom2**, to the user interface.

Axis and Grid Settings

1 From the **Options** menu choose **Axes/Grid Settings**.

2 Clear the **Axis equal** check box.

3 Enter the following settings for the axes: **x min** = -1e-6, **x max** = 24e-6, **y min** = -1e-6, **ymax** = 5e-6.

4 Click the **Grid** tab.

5 Clear the **Auto** check box.

6 Enter the following settings for the grid: **x spacing** = 1e-6, **y spacing** = 0.5e-6.

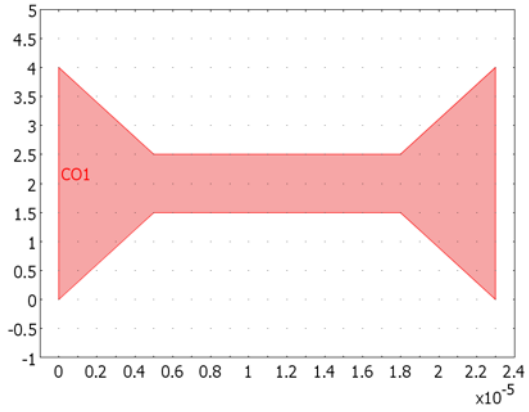
7 Click **OK**.

Drawing 2D Geometry

1 On the Draw toolbar click the **Line** button.

2 Construct the object shown in the nearby figure by left-clicking sequentially on the following x, y coordinate pairs. You can see the coordinates in the lower left corner on the screen. (0, 0), (0.5e-5, 1.5e-6), (1.8e-5, 1.5e-6), (2.3e-5, 0), (2.3e-5, 4e-6), (1.8e-5, 2.5e-6), (0.5e-5, 2.5e-6), (0, 4e-6).

3 Right-click to close the created object.



The first geometry to draw using the Line tool.

Extruding 2D Geometry

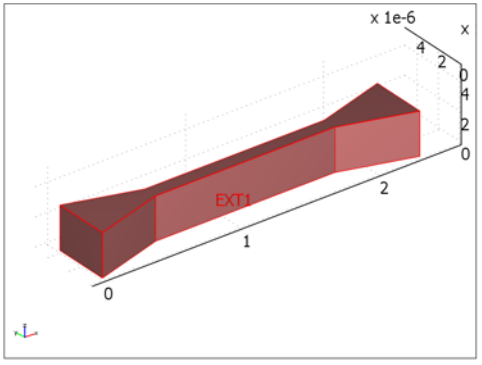
1 From the **Draw** menu choose **Extrude**.

2 In the **Distance** edit field enter `3e-6`.

3 See that **CO1** is selected in the **Objects to extrude** list; Then click **OK**.

4 With the **Geom1** geometry active, click the **Zoom Extents** button on the Main toolbar.

5 On the Camera toolbar (on the left side, the vertical toolbar closest to the drawing area), click the **Headlight** button.



The first geometry after extrusion.

Settings for the Second Work Plane

1 From the **Draw** menu choose **Work-Plane Settings**.

2 Click the **Add** button to the right of the **Workplane (2D geometry)** list.

3 Click the **Face Parallel** tab, and in the **Face Selection** list select 6.

4 In the **Offset from face** edit field enter the value `-3e-6`.

5 Click **OK**.

Axes and Grid Settings

1 With the **Geom3** geometry active, click the **Projection of All 3D geometries** button on the Visualization/Selection toolbar.

2 Click the **Zoom Extents** button on the Main toolbar.

3 From the **Options** menu choose **Axes/Grid Settings**.

4 Click the **Grid** tab, then clear the **Auto** check box.

5 Enter the following settings for the grid:

- **x spacing** = `1e-6`
- **Extra x** = `0.2e-6 12.8e-6` (make sure to separate these two entries with a space)
- **y spacing** = `1e-6`.

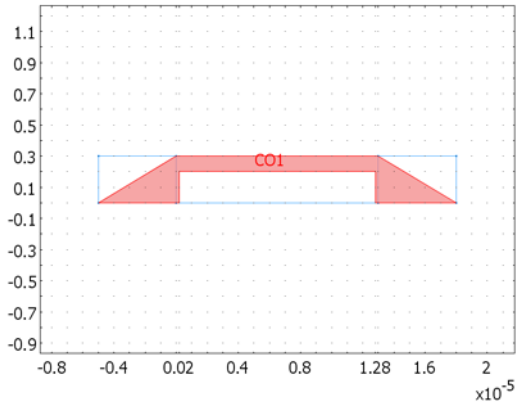
6 Click **OK**.

Drawing the Geometry

1 On the Draw toolbar click the **Line** button.

2 Construct the object shown in the nearby figure by left-clicking sequentially on the following x, y coordinate pairs: (–0.5e-5, 0), (0.02e-5, 0), (0.02e-5, 0.2e-5), (1.28e-5, 0.2e-5), (1.28e-5, 0), (1.8e-5, 0), (1.3e-5, 0.3e-5), (0, 0.3e-5).

3 Right-click to close the created geometry.



The second geometry to draw using the Line tool.

4 On the Draw toolbar click the **Fillet/Chamfer** button.

5 From the **Vertex selection** list select the two inner corners (4 and 6).

6 In the **Fillet radius** edit field enter `0.3e-6`.

7 Click **OK**.

Extruding into 3D

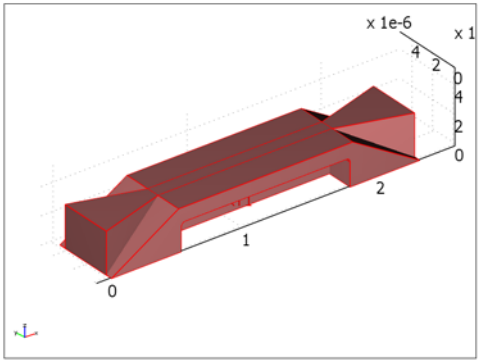
1 From the **Draw** menu choose **Extrude**.

2 Enter the **Distance** `5e-6`.

3 See that **CO2** is selected in the **Objects to extrude** list; Then click **OK**.

Combining 3D Objects

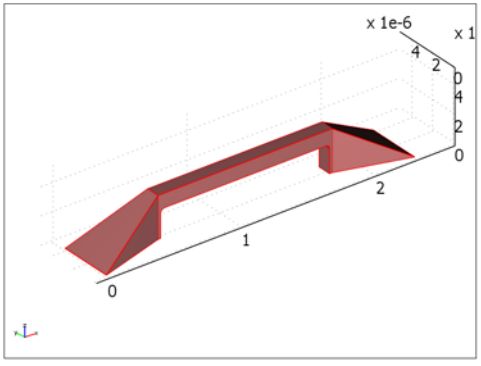
1 From the **Edit** menu choose **Select All**.



Both extruded geometries selected.

2 On the Draw toolbar click the **Intersection** button.

3 On the Draw toolbar click the **Delete interior boundaries** button.



The final geometry.

SOLVING FOR THE CONDUCTIVE MEDIA DC APPLICATION MODE

With the geometry complete, proceed to the **Physics** menu to configure the physical subdomain and add boundary settings. When you select an application mode in the **Multiphysics** menu on the Main toolbar, the corresponding subdomain and boundary settings dialog boxes in the **Physics** menu also change. In this way you shift among

various settings dialog boxes and enter values for all parameters as necessary. Start the configuration process with the electrical part.

Subdomain Settings

1. From the **Multiphysics** menu choose **Conductive Media DC (emdc)**.
2. From the **Physics** menu choose **Subdomain Settings**.
3. In the **Subdomain selection** list select Subdomain 1.
4. Click the **Load** button to open the materials library.
5. Locate the **Materials** list and select **Cu** from the library named **Metals** folder in the **MEMS Material Properties** library.
6. Click **OK** to close the **Materials/Coefficients Library** dialog box. Then click **OK** to close the **Subdomain Settings** dialog box.

Boundary Conditions

1. From the **Physics** menu choose **Boundary Settings**.
2. Press Ctrl+A to select all boundaries.
3. From the **Boundary condition** list select **Electric insulation**.
4. From the **Boundary selection** list select 1.
5. From the **Boundary condition** list select **Ground**.
6. From the **Boundary selection** list select 13.
7. From the **Boundary condition** list select **Electric potential**.
8. In the $\mathbf{V_0}$ edit field for the electric potential enter `Voltage`.
9. Click **OK**.

Mesh Generation

For most models, mesh generation can proceed through the straightforward use of default settings. However, many MEMS-related models have flat structures that require special mesh techniques where you create a quadrilateral mesh instead of working with the standard mesh generator for triangular unstructured meshes. In this model you tune the mesh by creating a finer unstructured mesh than by default.

1. From **Mesh** menu choose **Free Mesh Parameters**.
2. On the **Global** page select **Finer** from the **Predefined mesh sizes** list.
3. Click **Remesh** and then click **OK**.

The created mesh consists of approximately 5400 elements.

Computing the Solution and Solver Settings

This example employs the static solver to compute the solution. At first this model involves only 1-way couplings between application modes (from Conductive Media DC to Heat Transfer by Conduction and from Heat Transfer by Conduction to Solid, Stress-Strain). Therefore you can define and solve the application modes sequentially. Between each step COMSOL Multiphysics stores the solution and uses it for the following step. Solving application modes sequentially simplifies the computation of the solution and often reduces the total solution time.

In the second step the electrical conductivity is temperature dependent. Thus there is a 2-way coupling and you need to solve corresponding application modes simultaneously.

You initialized this model by first adding the Conductive Media DC application mode. Thus it is the *ruling application mode* and it defines the default solver settings: The *conjugate gradients* linear system solver with *algebraic multigrid* preconditioner. This combination works well for single physics and single degree of freedom problems, such as defined by the Conductive Media DC and the Heat Transfer by Conduction application modes. But the algebraic multigrid preconditioner fails to operate properly for the Solid, Stress-Strain application mode, which has many dependent variables. Furthermore, once you add couplings between the application mode, the system matrix becomes unsymmetric. With increasing unsymmetry, the operation of the conjugate gradients solver becomes gradually poorer.

There are many ways to define solver settings for this kind of problems. If you solve the application modes sequentially, you can tune the solver settings for each application mode. But in general, it is good to choose solver parameters that operate properly for each application mode and their multiphysics combinations. You can also use the segregated solver. The segregated solver continuously switches between the application modes and incrementally solves one of them at a time to reach the globally converged solution. Thus you can define the solver settings for each of the application independently of the couplings between the application modes.

This example uses the second approach and defines one set of solver settings so that the solver operates well for each step in this model: The iterative FGMRES linear system solver with the geometric multigrid (GMG) preconditioner is a robust and memory-efficient combination that works well for many multiphysics problems.

Follow the steps below to define the solver settings:

1 From the **Solve** menu choose **Solver Parameters**.

2 On the **General** page go to the **Linear system solver** list and select **FGMRES**.

3 From the **Preconditioner** list select **Geometric multigrid**.

4 Click **OK**.

Computing the Solution

Follow the steps below to compute the solution for the Conductive Media DC application mode:

1 From the **Solve** menu choose the **Solver Manager**.

2 Click the **Solve For** tab.

3 From the **Solve for variables** list select only **Conductive Media DC (emdc)**.

4 Click the **Solve** button at the bottom of the dialog box.

5 Click **OK**.

Postprocessing and Visualization

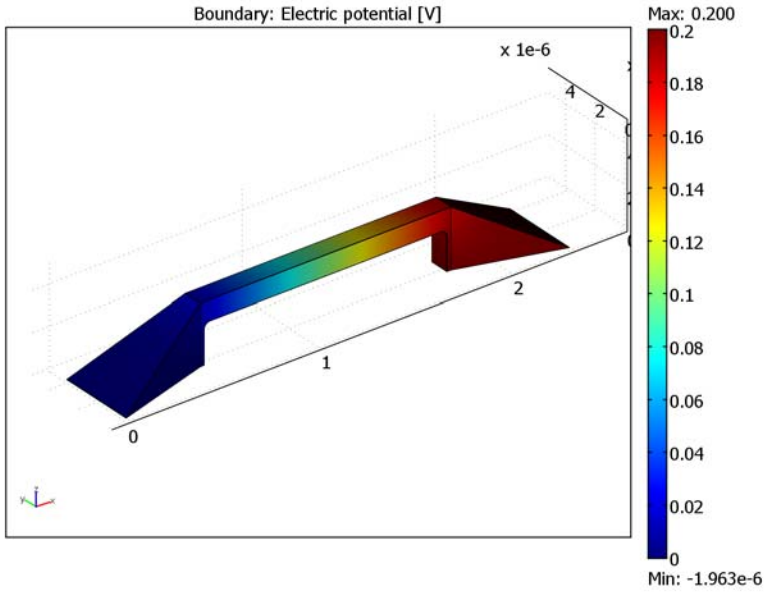
1 From the **Postprocessing** menu choose **Plot Parameters**.

2 Go to the **General** page and then the **Plot type** area; clear the **Slice** check box and then select the **Boundary** check box.

3 Click the **Boundary** tab.

4 From the **Predefined quantities** list select **Conductive Media DC (emdc)>Electric potential**.

5 Click **OK**.



Solution for the Conductive Media DC application mode.

SOLVING FOR THE HEAT TRANSFER BY CONDUCTION APPLICATION MODE

Subdomain Settings

1. From the **Multiphysics** menu choose **Heat Transfer by Conduction (ht)**.
2. From the **Physics** menu choose **Subdomain Settings**.
3. From the **Subdomain selection** list select Subdomain 1.
4. From the **Library Material** list select **Cu**.
5. In the **Q** edit field for the heat source enter `Q_emdc`. This is the name of the predefined variable for the resistive heating from the Conductive Media DC application mode. You can see the definition of `Q_emdc` and other predefined variables on the **Variables** page of the **Subdomain Settings - Equation System** dialog box, which you open by selecting **Physics>Equation System>Subdomain Settings**.
6. Click **OK**.

Boundary Conditions

1. From the **Physics** menu choose **Boundary Settings**.

2 Select all boundaries by pressing Ctrl+A.

3 From the **Boundary condition** list select **Heat flux**.

4 In the **h** edit field for the **Heat transfer coefficient** enter 5.

5 In the $\mathbf{T_{inf}}$ edit field for **External temperature** enter 298.

6 In the **Boundary selection** list select Boundaries 1 and 13.

7 From the **Boundary condition** list select **Temperature**.

8 In the $\mathbf{T_0}$ edit field for **Temperature** enter 323.

9 Click **OK**.

Computing the Solution

1 From the **Solve** menu choose the **Solver Manager**.

2 Click the **Initial Value** tab. In the **Initial value** area click the **Current solution** radio button.

3 Click the **Solve For** tab. In the **Solve for variables** list select only **Heat Transfer by Conduction (ht)**.

4 Click **Solve**, then click **OK**.

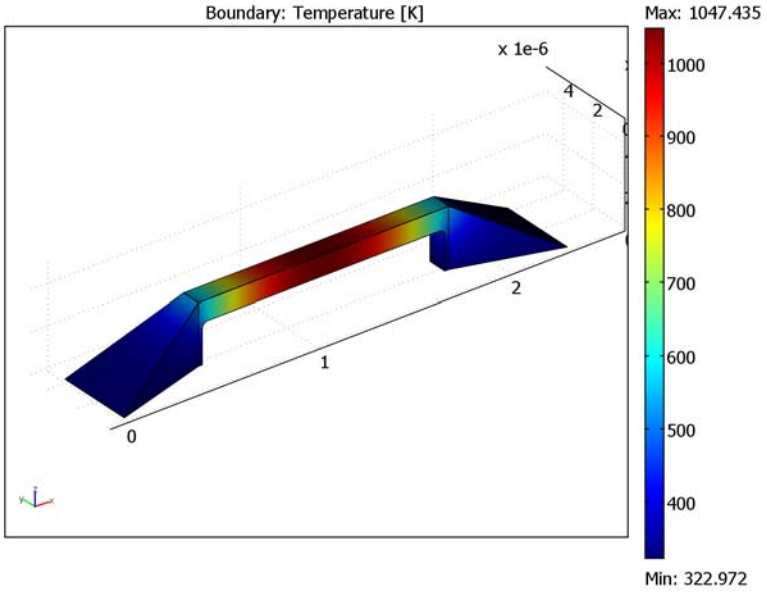
Postprocessing and Visualization

1 From the **Postprocessing** menu choose **Plot Parameters**.

2 Click the **Boundary** tab.

3 From the **Predefined quantities** list select **Heat Transfer by Conduction (ht)>Temperature**.

4 Click **OK**.



Solution for the Heat Transfer by Conduction application mode.

Based on the color bar, you can determine that the maximum temperature is 1047 K.

SOLVING FOR THE FOR SOLID, STRESS-STRAIN APPLICATION MODE

Subdomain Settings

1. From the **Multiphysics** menu choose **Solid, Stress-Strain (smsld)**.
2. From the **Physics** menu choose **Subdomain Settings**.
3. From the **Subdomain selection** list select Subdomain 1.
4. From the **Library material** list select **Cu**.
5. Click the **Load** tab.
6. Select the **Include thermal expansion** check box.
7. Find the **Temp** edit field and enter `T` as the value for the **Strain temperature**.
8. Find the **Tempref** edit field and enter `298` for the value for the **Strain ref. temperature**.
9. Click **OK**.

Boundary Conditions

1 From the **Physics** menu choose **Boundary Settings**.

2 From the **Boundary selection** list select Boundaries 1 and 13.

3 Select **Fixed** from the **Constraint condition** list.

4 Click **OK**.

Computing the Solution

5 From the **Solve** menu choose the **Solver Manager**.

6 Click the **Solve For** tab.

7 From the **Solve for variables** list select only **Solid, Stress-Strain (smsld)**.

8 Click **Solve**.

9 Click **OK**.

Postprocessing and Visualization

1 From the **Postprocessing** menu choose **Plot Parameters**.

2 Click the **General** tab.

3 Select the **Deformed shape** check box.

4 Click the **Boundary** tab.

5 From the **Predefined quantities** list select **Total displacement (smsld)**.

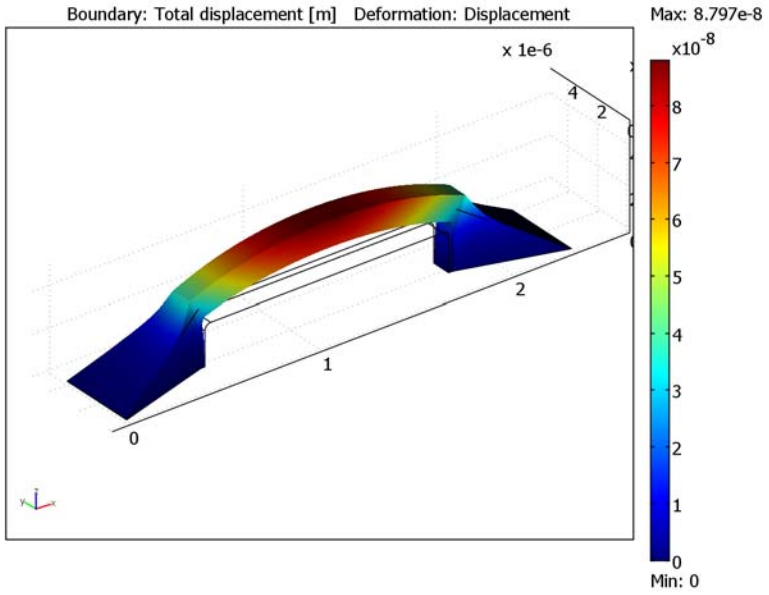
6 Click the **Deform** tab.

7 In the **Deformation data** area, select the **Subdomain Data** tab; Then select **Solid, Stress-Strain (smsld)>Displacement** from the **Predefined quantities** list.

8 In the **Deformation data** area, select the **Boundary Data** tab; Then select **Solid, Stress-Strain (smsld)>Displacement** from the **Predefined quantities** list.

9 Clear the **Scale factor: Auto** check box and enter a value of 20.

10 Click **OK**.



Solution for the Solid, Stress-Strain application mode.

Based on the color coding on the computer screen, you can determine that the maximum displacement is approximately 88 nm (8.8e-8 m).

SOLVING FOR TEMPERATURE-DEPENDENT ELECTRIC CONDUCTIVITY

Options and Settings

1. From the **Options** menu choose **Constants**.
2. Add a constant for the temperature dependency of electric conductivity; enter the **Name** `alphaT` and the **Expression** `0.0039[1/K]`.
3. Add a constant for the reference temperature; enter the **Name** `T0` and the **Expression** `293[K]`.
4. Click **OK**.

Subdomain Settings

1. From the **Multiphysics** menu choose **Conductive Media DC (emdc)**.
2. From the **Physics** menu choose **Subdomain Settings**.
3. Select Subdomain 1 from the **Subdomain selection** list.

4 From the **Conductivity relation** list, select **Linear temperature relation**.

5 Enter the following values for the parameters:

PARAMETER	VALUE
ρ_0	`1/58.1e6`
α	`alphaT`
T	`T`
T_0	`T0`

6 Click **OK**.

Computing the Solution

1 From the **Solve** menu choose the **Solver Manager**.

2 Click the **Solve For** tab. From the **Solve for variables** list select both **Heat Transfer by Conduction (ht)** and **Conductive Media DC (emdc)**.

3 Click the **Initial Value** tab. In the **Initial value** area click the **Initial value expression** radio button.

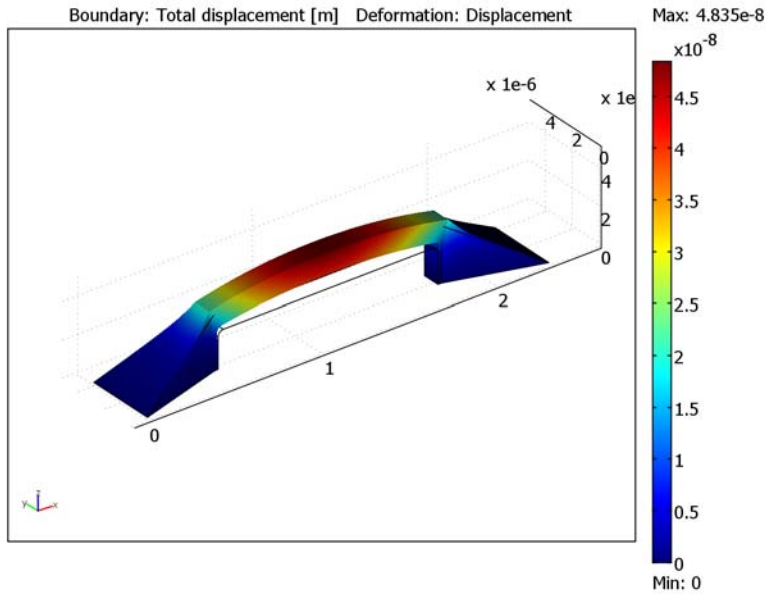
4 Click **Solve**.

5 In the **Initial value** area click the **Current solution** button.

6 Click the **Solve For** tab.

7 From the **Solve for variables** list select only **Solid, Stress-Strain (smsld)**.

8 Click the **Solve** button, then click **OK**.



Solution for temperature-dependent electric conductivity.

Based on the color coding on the screen, you can determine that the maximum displacement is now approximately 48 nm.

3

The MEMS Module Application Modes

This chapter presents an overview of the application modes in the MEMS Module. They fall into three main application areas:

- Structural mechanics—includes application modes for pure structural problems, piezoelectricity, and damping
- Electrostatics—includes application modes for modeling conducting and nonconducting materials
- Microfluidics—includes application modes for modeling fluid flow, mass transport by convection and diffusion, and electrokinetic effects

Furthermore, several predefined multiphysics couplings interface physics from these different application areas.

For each application mode, the following chapters include information about available equation formulations, material properties, boundary conditions, and postprocessing quantities. The *MEMS Module Model Library* contains examples that use combinations of these application modes to model different types of MEMS devices.

Overview of Application Modes

The application modes in the MEMS Module form a complete set of simulation tools for MEMS simulations.

The following section describes these application modes along with the physical quantities they solve for and the standard abbreviation each one uses. There are also page references to the sections that describe each application mode and their applications in detail. The physical quantities that appear in these application modes are:

- The *structural displacements* or the *velocity components* u, v, and w (uor equals u/r in the axisymmetric structural application modes)
- The *film pressure variation*, p_f
- The *electric scalar potential*, V
- The *concentration*, c
- The *level set function*, ϕ

Application Mode Guide

Table 3-1 lists the application modes in the MEMS Module. For a descriptive illustration and more details on each one, see the corresponding chapter as noted in the table's Page column.

The Name column shows the default name that appears as a label on each application mode when you use it. This name has special importance in multiphysics simulations so you can distinguish among different application modes in a model—each application mode defines its own set of variables, all of which get an underscore plus this special application mode name appended to the variable names.

The Dependent Variables column lists the dependent variables for which the application mode solves the underlying PDEs. For most 2D modes, COMSOL Multiphysics formulates the PDEs it solves for components perpendicular to the modeling plane. For axisymmetric simulations the software makes a variable transformation to avoid singularities on the rotation axis.

Finally, the Analysis Capabilities columns indicate which analysis types each application mode supports.

TABLE 3-1: MEMS MODULE APPLICATION MODES

APPLICATION MODE	PAGE	NAME	DEPENDENT VARIABLES	ANALYSIS CAPABILITIES					
				STATIC	TRANSIENT	EIGENFREQUENCY/-MODE	FREQUENCY RESPONSE	ELASTO-PLASTIC	QUASI-STATIC
STRUCTURAL MECHANICS	59								
Solid, Stress-Strain	60	smsld	u, v, w	√	√	√	√	√	√
Plane Stress	61	smps	u, v	√	√	√	√	√	√
Plane Strain	62	smpn	u, v	√	√	√	√	√	√
Axial Symmetry, Stress-Strain	63	smaxi	uor, w	√	√	√	√	√	√
Fluid-Structure Interaction[1]	136								
PIEZOELECTRICITY	157								
Piezo Solid	186	smpz3d	u, v, w, V	√	√	√	√		
Piezo Plane Stress	186	smpps	u, v, V	√	√	√	√		
Piezo Plane Strain	187	smppn	u, v, V	√	√	√	√		
Piezo Axial Symmetry	187	smpaxi	uor, w, V	√	√	√	√		
FILM DAMPING	191								
Film Damping	201	mmfd	p_f		√		√		
Solid, Stress-Strain with Film Damping[2]	208	mmfd, smsld	u, v, w, p_f		√		√		
Plane Strain with Film Damping[2]	209	mmfd, smpn	u, v, p_f		√		√		
Axial Symmetry, Stress-Strain with Film Damping[2]	210	mmfd, smaxi							
ELECTROSTATICS	211								
Conductive Media DC	212	emdc	V	√					

TABLE 3-1: MEMS MODULE APPLICATION MODES

APPLICATION MODE	PAGE	NAME	DEPENDENT VARIABLES	ANALYSIS CAPABILITIES					
				STATIC	TRANSIENT	EIGENFREQUENCY/-MODE	FREQUENCY RESPONSE	ELASTO-PLASTIC	QUASI-STATIC
Electrostatics	222	emes	V	√					
MICROFLUIDICS									
General Laminar Flow	264	mmglf	u, v, w, p	√	√				
Incompressible Navier_Stokes	264	mmglf	u, v, w, p	√	√				
Non-Isothermal Flow	264	mmglf	u, v, w, p	√	√				
Stokes Flow	264	mmglf	u, v, w, p	√	√				
Non-Isothermal Stokes Flow	264	mmglf	u, v, w, p	√	√				
Level Set Two-Phase Flow	283	mmglf	u, v, w, p, ϕ						
Convection and Diffusion	294	chcd	c						
Flow with Species Transport[3]	305	mmglf, chcd	u, v, w, p, c	√	√				
Electroosmotic Flow[4]	305	mmglf, emdc	u, v, w, p, V	√	√				
ELECTROKINETIC FLOW	319								
Electrokinetic Flow	320	chekf	c	√	√				

1. A predefined coupling that combines a structural mechanics application mode with the Incompressible Navier-Stokes and the Deformed Mesh (ALE) application modes. Names and dependent variables are those of the actual application modes.

2. A predefined coupling that combines a structural mechanics application mode with the Film Damping application mode.

3. A predefined coupling that combines a microfluidics application mode with the Convection and Diffusion application mode. Names and dependent variables are those of the actual application modes.

4. A predefined coupling that combines a microfluidics application mode with the Conductive Media DC application mode.

In addition, you can perform parametric analyses using the parametric solver. The application modes for structural mechanics make parametric analyses directly available from the Model Navigator.

To carry out various types of simulations for a given set of parameters in an application mode, you need only change the *analysis type* directly in the Model Navigator, in the **Solver Parameters** dialog box, or among the *application-mode properties*. You select application modes from the Model Navigator before starting work on a new model. You can also add them to an existing model to expand it into a multiphysics model.

When using the axisymmetric modes it is important to note that the horizontal axis represents the r direction and the vertical axis the z direction. Further, you must create the geometry in the right half-plane, that is, only for positive r.

You specify all scalar properties specific to an application mode in the **Application Scalar Variables** dialog box. Their default values are either physical constants or arbitrary values in a range common for modeling purposes.

To enter application-specific domain properties (material properties, sources, sinks, and so on) use the **Subdomain Settings** dialog box. It is possible to define subdomain parameters for problems that have multiple regions, each with different material properties. Some domain parameters might be either a scalar or a matrix depending on whether the material is isotropic or anisotropic.

The **Boundary Settings** dialog box also adapts to the current application mode. Use it to select application-specific boundary conditions. Certain boundary types require you to specify one or several properties through edit fields in the dialog box, while others generate the boundary conditions without any user-specified edit fields.

Similarly, use the **Edge Settings** and **Point Settings** dialog boxes to specify application-specific conditions on edges and points.

Finally, use the **Plot Parameters**, **Cross-Section Plot Parameters**, **Domain Plot Parameters** and other dialog boxes for postprocessing to visualize the relevant physical variables for all application modes in a model.

The following chapters contains all the details necessary to get full insight into the various application modes, including the physical assumptions and mathematical considerations upon which we base them and the functionality they offer. Each section describing a particular mode consists of the following sections:

The *PDE Formulation* section contains the equations the application mode solves.

The *Application Mode Properties* section lists those specific to the application mode. With these properties you can, for example, select the analysis type.

The *Application Scalar Variables* section lists parameters specific to the application mode. Their default values are either physical constants or arbitrary values in a range common for modeling, for example, a frequency of 50 Hz for quasi-static modes.

The *Boundary and Interface Conditions* section lists the available boundary conditions and explains their physical interpretation.

The *Line Sources* and *Point Sources* sections list the available settings on edges and points.

In the *Application Mode Variables* section you find all the variables available when formulating equations and for postprocessing (when you define a function of these variables for plots and graphs). It is also possible to use these variables in the expressions for the physical properties in the equations.

We have organized the tables that detail the application-mode variables as follows:

- The Name column gives the names of variables you can use in equations or for postprocessing. The indices `i` and `j` (using an italic font) in the variable names can mean any of the spatial coordinates. For example, `Ei` means either `Ex`, `Ey`, or `Ez` in 3D when the spatial coordinates are x, y, and z. In 2D axisymmetry `Ei` would stand for either `Er` or `Ez`. You then construct the variable names of vector and tensor components using the names of the spatial coordinates. For example, if `x1`, `y1`, and `z1` are the spatial coordinates, the variables for the vector components of the electric field are `Ex1`, `Ey1`, and `Ez1`.

  In a COMSOL Multiphysics model the name of each application-mode variable gets an underscore plus the application mode's name appended as a suffix. For example, the default name of the Electrostatics application mode is `emes`, so the variable for the x component of the electric field is `Ex_emes`.
- The Type column indicates whether the variable is defined on subdomains (S), boundaries (B), edges (E), or points (P), and it also indicates the top level where a variable is defined. Many variables that are available on subdomains are also available on boundaries, edges, and points, but then take the average value of the values in the subdomains around the boundary, edge, or point. In other words, to get a value on a boundary, you take the average of the value in the left subdomain and the value in the right subdomain.
- The Analysis column specifies for which type of analysis a variable is defined. The available analysis types might be, for example, static, transient, harmonic, and

eigenfrequency. The available analysis types are application-mode dependent; some variables are defined differently depending on the analysis type or are available only for some analysis types.

- The Description column gives a textual description of the variables.
- The Expression column gives the expression of the variables in terms of other physical quantities. In these expressions, the subscripts i and j (in italics) of vector and tensor components stand for one of the spatial coordinates. For example, E_i is either E_x, E_y, or E_z in a 3D model. Two equal subscripts that appear in an expression imply a summation. For example $\sigma_{ij}E_j = \sigma_{ix}E_x + \sigma_{iy}E_y + \sigma_{iz}E_z$.

Note: Most of the application mode variables appear in the *MEMS Module Reference Guide*.

MATERIAL LIBRARIES

All application modes in the MEMS Module support the COMSOL Multiphysics material libraries. You can store a number of material properties and use them with the MEMS application modes:

- Elastic properties
- Electric properties
- Fluid properties
- Piezoelectric properties
- Thermal properties

In addition, the MEMS Module include specific material libraries for MEMS materials, piezoelectric materials, and fluids and gases. See "The Materials/Coefficients Library" on page 309 for more information.

4

Structural Mechanics Application Modes

The Structural Mechanics application modes in the MEMS Module are known as continuum application modes. In this context “continuum” means that no simplifications are available and that you solve for the displacements without involving rotations.

The following application modes are available in the MEMS Module for structural mechanics applications:

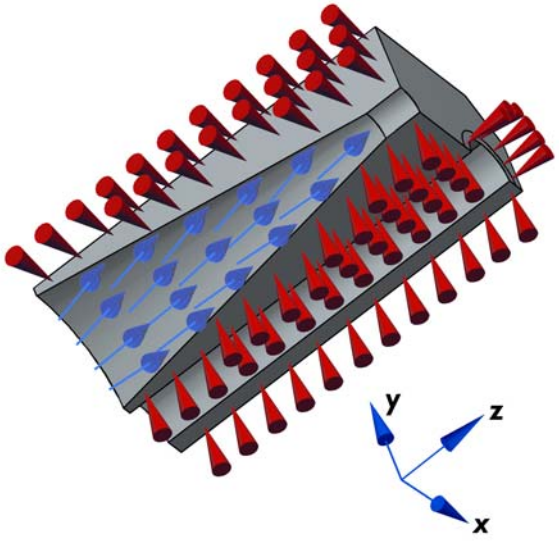
- Solid, Stress-Strain
- Plane Stress
- Plane Strain
- Axial Symmetry, Stress-Strain

In addition, this section describes predefined multiphysics couplings for modeling fluid-structure Interaction (FSI) and thermal-structure interaction.

The following chapters first review the theory of the continuum application modes, then they describe the details of each one on an individual basis. The final chapters explain several features of the MEMS Module that you should find useful when modeling structural-mechanics problems.

Solid, Stress-Strain

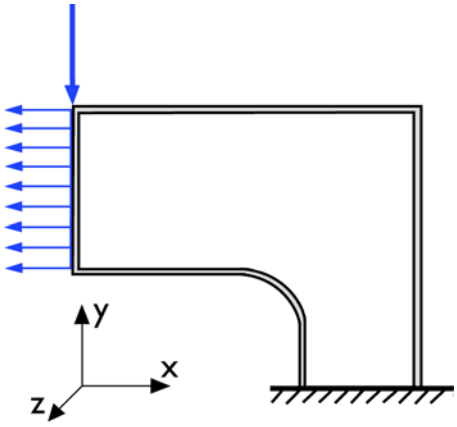
The degrees of freedom (dependent variables) in this application mode are the global displacements u, v, and w in the global x, y, and z directions, respectively, and the pressure p (only used if mixed formulation is selected).



Loads and constraints applied to a 3D solid using the Solid, Stress-Strain application mode.

Plane Stress

Use the Plane Stress application mode to analyze thin in-plane loaded plates. It solves for the global displacements (u, v) in the x and y directions, the pressure p (only used for mixed formulation), and the displacement derivative in the perpendicular direction (only used for hyperelastic material). For a state of plane stress, this mode assumes the σ_z, τ_{yz}, and τ_{xz} components of the stress tensor are zero.

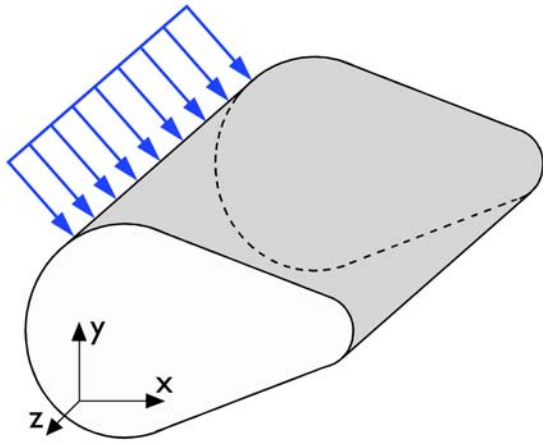


The Plane Stress application mode models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

This application mode allows loads in the x and y directions, and it assumes that these are constant throughout the material's thickness, which however 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 application mode solves for the global displacements (u, v) in the x and y directions and the pressure p (only if mixed formulation is used). The assumption that defines a state of plane strain is that the ε_z, ε_{yz}, and ε_{xz} components of the strain tensor are zero.



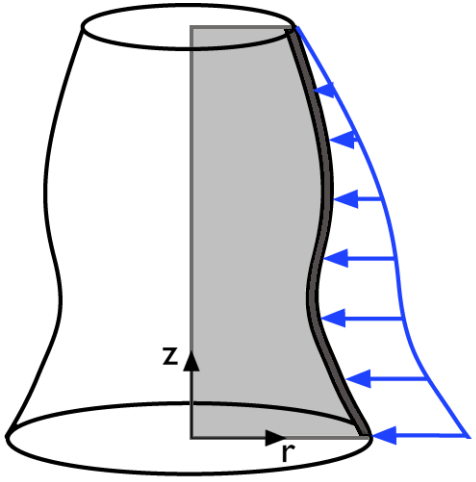
A geometry suitable for plane strain analysis.

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.

Axial Symmetry, Stress-Strain

The Axial Symmetry, Stress-Strain application mode uses cylindrical coordinates r, φ (`phi`), and z. It solves equations for the global displacement (uor, w) in the r and z directions and the pressure p (only used for mixed formulation). The dependent variable, $uor = u/r$, is introduced to avoid division by r, which causes problems on the axis where $r = 0$. This application mode assumes that the displacement v in the φ direction together with the $\tau_{r\varphi}$, $\tau_{\varphi z}$, $\gamma_{r\varphi}$, and $\gamma_{\varphi z}$ components of the stresses and strains are zero. Loads are independent of φ, and this application mode allows loads only in the r and z directions.

You can view the domain where the application mode solves the equations as the intersection between the original axially symmetric 3D solid and the half plane $\varphi = 0$, $r \geq 0$. Therefore you draw the geometry only in the half plane $r \geq 0$ and recover the original 3D solid by rotating the 2D geometry about the z-axis.



Rotating a 2D geometry to recover a 3D solid.

Note: $r = 0$ is the symmetry axis. In the Axisymmetry, Stress-Strain application mode $x \rightarrow r$ and $y \rightarrow z$.

Theory Background

Strain-Displacement Relationship

The strain consists of thermal (ε_{th}), elastic (ε_{el}), and initial (ε_0) contributions so that

$$\varepsilon = \varepsilon_{el} + \varepsilon_{th} + \varepsilon_0$$

The strain conditions at a point are completely defined by the deformation components—u, v, and w in 3D—and their derivatives. The precise relation between strain and deformation depends on the relative magnitude of the displacement.

SMALL DISPLACEMENTS

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

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_{xy} &= \frac{\gamma_{xy}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \varepsilon_y &= \frac{\partial v}{\partial y} & \varepsilon_{yz} &= \frac{\gamma_{yz}}{2} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \varepsilon_z &= \frac{\partial w}{\partial z} & \varepsilon_{xz} &= \frac{\gamma_{xz}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right). \end{aligned} \tag{4-1}$$

To express the shear strain, use either the tensor form, ε_{xy}, ε_{yz}, ε_{xz}, or the engineering form, γ_{xy}, γ_{yz}, γ_{xz}.

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

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

LARGE DISPLACEMENTS

For large-displacement analysis, the deformation is not small and you calculate the strains without this restriction. The resulting strains are known as Green or *Green-Lagrange strains*, and large displacement is sometimes referred to as *geometric nonlinearity* or *nonlinear geometry*.

Green strains are defined with reference to an undeformed geometry. Hence, they represent a Lagrangian description.

In a small-strain, large rotational analysis, the Green strain corresponds to the engineering strain in directions that follow the deformed body. The Green strain is a natural choice when formulating a problem in the undeformed state.

The Green strain components, ε_{ij}, are

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i} \cdot \frac{\partial u_k}{\partial x_j}\right) \tag{4-2}$$

ANALYSIS OF DEFORMATION AND THE DEFORMATION GRADIENT

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

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

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

$$\mathbf{x} = \mathbf{X} + \mathbf{u}$$

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

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

is used.

The deformation gradient contains the complete information about the local straining and rotation of the material. It is a positive definite matrix, as long as material cannot be annihilated. The ratio between current and original volume (or mass density) is

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

As a consequence, a deformation state where $J = 1$ is said to be incompressible. From the deformation gradient, it is possible to define the right Cauchy-Green tensor as

$$C = F^T F$$

The most commonly used definition of strain is the *engineering strain* ε; see Equation 4-1.

As can be shown by simple insertion, a finite rigid body rotation will cause nonzero values of the engineering strain. This is not in correspondence with the intuitive concept of strain, and it is certainly not useful in a constitutive law. There are several alternative strain definitions in use that do have the desired properties. The Green strain, ε, is defined as

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

Using the displacements, the Green strain can be also written as shown in Equation 4-2.

The deformation gradient and its inverse are available as variables and can be used, for instance, to model follower loads; see the Hyperelastic Seal model on page 465 in the *Structural Mechanics Model Library* for an example.

Stress-Strain Relationship

The symmetric stress tensor σ describes stress in a material:

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \qquad \tau_{xy} = \tau_{yx} \qquad \tau_{xz} = \tau_{zx} \qquad \tau_{yz} = \tau_{zy}$$

This tensor consists of three normal stresses (σ_x, σ_y, σ_z) and six (or, if symmetry is used, three) shear stresses (τ_{xy}, τ_{yz}, τ_{xz}).

For large deformations and hyperelastic material models there are more than one stress measure:

- Cauchy stress σ (the components are denoted `sx`, … in COMSOL Multiphysics) defined as force/deformed area in fixed directions not following the body. Symmetric tensor.

- First Piola-Kirchhoff stress P (the components are denoted `Px`, … in COMSOL Multiphysics). This is an unsymmetric tensor used only for hyperelastic material models.
- Second Piola-Kirchhoff stress S (the components are denoted `Sx`, … in COMSOL Multiphysics). This is a symmetric tensor, for small strains same as Cauchy stress tensor but in directions following the body.

The stresses relate to each other as

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

LINEAR ELASTIC MATERIAL

The stress-strain relationship—or the *constitutive equation*—for linear conditions including initial stress and strain and thermal effects reads:

$$\sigma = D\varepsilon_{\mathrm{el}} + \sigma_0 = D(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) + \sigma_0$$

where D is the 6-by-6 elasticity matrix and the stress and the strain are both given in column vector form:

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}$$

Note: In the following descriptions σ and ε denote either the stress and strain vectors or the corresponding tensors depending on the circumstances.

The elasticity matrix D—or the more basic flexibility (or compliance) matrix, the inverse of D—is defined differently for isotropic, orthotropic, and anisotropic materials. For an isotropic material, the flexibility matrix looks like

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

where E is the modulus of elasticity or *Young's modulus*, and ν is *Poisson's ratio*, which defines the contraction in the perpendicular direction. Inverting D^{-1} results in the following elasticity matrix:

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

For an orthotropic material, the D^{-1} matrix takes the form

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

where you supply the values of E_x, E_y, E_z, ν_{xy}, ν_{yz}, ν_{xz}, G_{xy}, G_{yz}, and G_{xz} in designated edit fields in the user interface. The software deduces the remaining components—ν_{yx}, ν_{zx}, and ν_{zy}—by using the fact that the matrices D and D^{-1} are symmetric.

Note: The definitions of the components ν_{ij} can vary depending on the application field. When specifying the material properties, make sure you use the definitions just given. If necessary, transform your material data so that it conforms with the above conventions before entering it in the user interface.

Inverting D^{-1} using only the E_x, E_y, E_z, ν_{xy},ν_{yz}, ν_{xz}, G_{xy}, G_{yz}, and G_{xz} coefficients results in the symmetric D matrix

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

where the components are as follows:

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

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

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

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

where

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

For an anisotropic material, you provide the symmetric D matrix explicitly.

Mixed Formulation

Mixed formulations are used for nearly incompressible materials. The solution is to add the negative mean stress as a new dependent variable, p (pressure).

$$p = -\left(\frac{\sigma_x + \sigma_y + \sigma_z}{3}\right)$$

The stress-strain relation for linear elastic material for 3D is

$$\sigma = D\varepsilon_{\mathrm{el}} + \sigma_0 = D(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) + \sigma_0$$

The stress σ is separated in a deviatoric part, σ_{d}, and a mean part, $-p$:

$$\sigma = \sigma_{\mathrm{d}} - mp$$

where

$$\sigma_{\mathrm{d}} = D_{\mathrm{d}}(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) + \sigma_{0\mathrm{d}}$$
$$\sigma_0 = \sigma_{0\mathrm{d}} - p_0$$
$$p_0 = -\left(\frac{\sigma_{0x} + \sigma_{0y} + \sigma_{0z}}{3}\right)$$

and m is a six-dimensional column vector. Inserting the stress-strain relation for isotropic materials results in the following expressions for the deviatoric stress and the vector m:

$$\sigma_{\mathrm{d}} = D_{\mathrm{d}}\left[\begin{bmatrix}\varepsilon_x\\ \varepsilon_y\\ \varepsilon_z\\ \varepsilon_{xy}\\ \varepsilon_{yz}\\ \varepsilon_{xz}\end{bmatrix} - \begin{bmatrix}\varepsilon_{x0}\\ \varepsilon_{y0}\\ \varepsilon_{z0}\\ \varepsilon_{xy0}\\ \varepsilon_{yz0}\\ \varepsilon_{xz0}\end{bmatrix} - \alpha(T - T_{\mathrm{ref}})m\right] + (\sigma_0 + p_0)$$

where

$$D_{\mathrm{d}} = G\begin{bmatrix} \frac{4}{3} & -\frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & \frac{4}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & -\frac{2}{3} & \frac{4}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$m = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The equation for the pressure becomes

$$\frac{p}{K} + m^T(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) - \frac{p_0}{K} = 0$$

$$K = \frac{E}{3(1-2\nu)}$$

where G is the shear modulus and K is the bulk modulus.

For orthotropic and anisotropic materials some scaling is performed to get a system of equations that produces symmetric matrices. The equations for the stress and the pressure become

$$\sigma = \sigma_{\mathrm{d}} - np$$

$$\sigma_{\mathrm{d}} = D_{\mathrm{d}}(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) + \sigma_{0\mathrm{d}}$$

$$\sigma_0 = \sigma_{0\mathrm{d}} - p_0$$

$$p_0 = -\left(\frac{\sigma_{0x} + \sigma_{0y} + \sigma_{0z}}{3}\right)$$

$$\frac{9p}{D_{\mathrm{sum}}} + m^T(\varepsilon - \varepsilon_{\mathrm{th}} - \varepsilon_0) - \frac{9p_0}{D_{\mathrm{sum}}} = 0$$

where, n, D_{d}, and m are defined as

$$m_i = \frac{3d_i}{D_{\text{sum}}}$$

$$n_i = m_i \qquad i = 1, \ldots, 3$$
$$n_i = 0.5m_i \qquad i = 4, \ldots, 6$$

$$D_{\text{sum}} = \sum_{\substack{i = 1, \ldots, 3 \\ j = 1, \ldots, 6}} D_{ij}$$

$$d_i = D_{1i} + D_{2i} + D_{3i}$$

$$D_{\text{d}ij} = D_{ij} - \frac{d_i d_j}{D_{\text{sum}}} \qquad i = 1, \ldots, 3 \qquad j = 1, \ldots, 6$$

$$D_{\text{d}ij} = D_{ij} - 0.5\frac{d_i d_j}{D_{\text{sum}}} \qquad i = 4, \ldots, 6 \qquad j = 1, \ldots, 6$$

This produces symmetric matrices.

Note: The mixed formulation creates indefinite matrices, which affects the selection of iterative solvers. See "Selecting Iterative Solvers" on page 127 for more information.

The mixed formulation are useful not only for linear elastic material but also for elasto-plastic and hyperelastic materials.

In the mixed formulation the pressure is interpolated using a polynomial of one order less than the one used for the displacement variables.

If loss factor damping is used with frequency response analysis, the loss information appears also in the pressure equation. The equation with loss factor damping for isotropic material is

$$\frac{p}{(1 + j\eta)K} + m^T\varepsilon - \frac{m^T(\varepsilon_{\text{th}} + \varepsilon_0)}{(1 + j\eta)} - \frac{p_0}{(1 + j\eta)K} = 0$$
$$K = \frac{E}{3(1 - 2\nu)}$$

and the equation for orthotropic and anisotropic materials is

$$\frac{9p}{(1+j\eta)D_{\text{sum}}} + m^T\varepsilon - \frac{m^T(\varepsilon_{\text{th}}+\varepsilon_0)}{(1+j\eta)} - \frac{9p_0}{(1+j\eta)D_{\text{sum}}} = 0$$

where η is the loss factor.

ELASTO-PLASTIC MATERIALS

In an elasto-plastic material the stress-strain relationship is

$$\sigma = D\varepsilon_{\text{el}} + \sigma_0 = D(\varepsilon - \varepsilon_{\text{p}} - \varepsilon_{\text{th}} - \varepsilon_0) + \sigma_0$$

where ε_{p} is the *plastic strain* vector.

The variable ε_{p} and a vector κ of *state parameters* describe the state of a plastic deformation. To describe the evolution of these variables, use the rate equations

$$\dot{\varepsilon}_{\text{p}} = \lambda H(\varepsilon_{\text{p}}, \kappa, v), \qquad \dot{\kappa} = \lambda G(\varepsilon_{\text{p}}, \kappa, v)$$

where v is a vector whose variables form the solution vector (with parameters such as displacements and temperature) and λ is the *plastic multiplier*. The dot stands for differentiation with respect to pseudo-time or time. The plastic multiplier is determined by the *complementarity conditions*

$$F(\varepsilon_{\text{p}}, \kappa, v) \le 0, \qquad \lambda \ge 0, \qquad F(\varepsilon_{\text{p}}, \kappa, v)\lambda = 0$$

where F is the *yield function*. The functions F, G, and H often take a simpler form when expressed in terms of the *generalized stress*, Σ,

$$\Sigma = \begin{bmatrix} \sigma \\ \chi \end{bmatrix} = \begin{bmatrix} f_1(\varepsilon - \varepsilon_{\text{p}}, v) \\ f_2(\varepsilon_{\text{p}}, \kappa, v) \end{bmatrix}$$

where σ is the vector of stress components, and χ is the vector of *conjugate forces*. The function f_1 is often a linear function (matrix D). For *associated plasticity*, which is the rule,

$$H(\sigma, \chi) = \frac{\partial}{\partial\sigma}F(\sigma, \chi), \qquad G(\sigma, \chi) = -\frac{\partial}{\partial\chi}F(\sigma, \chi)$$

For *non-associated plasticity*, which is very uncommon,

$$H(\sigma, \chi) = \frac{\partial}{\partial\sigma}Q(\sigma, \chi), \qquad G(\sigma, \chi) = -\frac{\partial}{\partial\chi}Q(\sigma, \chi)$$

where Q is a *plastic potential*.

Now consider some special cases where the *effective stress function*, ϕ, is often the *von Mises stress*.

Perfect (or Ideal) Plasticity

$$\kappa = \text{empty}, \qquad F = \phi(\sigma) - Y_0 \qquad H = \frac{\partial F}{\partial \sigma}$$

where Y_0 is the *yield stress*.

Isotropic Hardening

$$\kappa = \varepsilon_{pe}, \qquad \chi = Y = f_2(\varepsilon_{pe}), \qquad F(\sigma, Y) = \phi(\sigma) - Y \qquad G = -\frac{\partial F}{\partial \sigma} = 1$$

where ε_{pe} is the *effective plastic strain*, and Y is the yield stress. The function f_2 is often nonlinear.

Kinematic Hardening

$$\kappa = \varepsilon_p, \qquad \chi = \sigma_{shift} = f_3(\varepsilon_p), \qquad F = \phi(\sigma - \sigma_{shift}) - Y_0, \qquad G = -\frac{\partial F}{\partial \sigma_{shift}}$$

where f_3 often is a linear function.

In cases with kinematic hardening, the plastic strain is a subset of the state parameters. Then you only need the rate equation

$$\dot{\kappa} = \lambda G(\kappa, v)$$

and the complementarity conditions

$$F(\kappa, v) \le 0, \qquad \lambda \ge 0, \qquad F(\kappa, v)\lambda = 0$$

You can thus write the generalized stress, Σ, as

$$\Sigma = \begin{bmatrix} \sigma \\ \chi \end{bmatrix} = f(\kappa, v)$$

These formulas also hold for cases without kinematic hardening if you enlarge the vector of state parameters to include the plastic strain. From now on this discussion assumes this definition of κ, leading to the following special cases:

Perfect (or Ideal) Plasticity

$$\kappa = \varepsilon_p, \qquad F = \phi(\sigma) - Y_0, \qquad G = \frac{\partial F}{\partial \sigma}$$

Isotropic Hardening

$$\kappa = \begin{bmatrix} \varepsilon_p \\ \varepsilon_{pe} \end{bmatrix}, \qquad \chi = Y = f_2(\varepsilon_{pe}), \qquad F = \phi(\sigma) - Y, \qquad G = \begin{bmatrix} \dfrac{\partial F}{\partial \sigma} \\ -\dfrac{\partial F}{\partial Y} \end{bmatrix}$$

where ε_{pe} is the effective plastic strain, and Y is the yield stress. The function f_2 is often nonlinear.

Kinematic Hardening

$$\kappa = \varepsilon_p, \qquad \chi = \sigma_{shift} = f_3(\varepsilon_p), \qquad F = \phi(\sigma - \sigma_{shift}) - Y_0, \qquad G = \frac{\partial F}{\partial \sigma}$$

where f_3 often is a linear function.

Postprocessing

The effective plastic strain is calculated from the equation

$$\varepsilon_{pe} = \int_0^t \dot{\varepsilon}_{pe} dt$$

$$\dot{\varepsilon}_{pe} = \frac{\sqrt{2}}{3}\sqrt{(\dot{\varepsilon}_{px} - \dot{\varepsilon}_{py})^2 + (\dot{\varepsilon}_{py} - \dot{\varepsilon}_{pz})^2 + (\dot{\varepsilon}_{px} - \dot{\varepsilon}_{pz})^2 + 6\dot{\varepsilon}_{pxy}^2 + 6\dot{\varepsilon}_{pyz}^2 + 6\dot{\varepsilon}_{pxz}^2}$$

The plastic strain can be calculated in the node points like any other variable but this may cause problems because evaluating the plastic strain involves solving an equation system.

For postprocessing purposes, Gauss-point variables are an alternative to the stresses, plastic strains, and effective plastic strain discussed above. Gauss-point variables are normally better because they are the values that were calculated during the solution process. The Gauss-point variables have the suffix `Gp` appended to their names, for example, `sxGp` instead of `sx`.

The elasto-plastic material model requires that you use a solver that can follow the load history, that is, you need to use the nonlinear parametric or transient solver. You cannot use the nonlinear static solver together with an elasto-plastic material model.

HYPERELASTIC MATERIALS

A *hyperelastic material* is defined from its strain energy function, W_s, which is a function of the strain state. The stress in such a material is computed from the strain energy function W_s, In the following, assume that the First Piola-Kirchhoff stresses P and the displacement gradient $\nabla \mathbf{u}$ are used, so that

$$P = \frac{\partial W_s}{\partial \nabla \mathbf{u}} \tag{4-3}$$

For an isotropic material, W_s can only be a function of the strain invariants. In a total Lagrangian formulation it is convenient to use the right Cauchy-Green tensor $C = F^{\mathrm{T}}F$ for the representation of the strain. The invariants are:

$$I_1 = \mathrm{trace}(C) = C_{11} + C_{22} + C_{33}$$

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

$$I_3 = \det(C) = J^2$$

where $J \equiv \det(F)$. Due to the incompressibility, it is often a good idea to work with modified invariants, where the two first invariants have no dependency on the volume change. Such invariants can be defined as

$$\bar{I}_1 = I_1 J^{-\frac{2}{3}}$$

$$\bar{I}_2 = I_2 J^{-\frac{4}{3}}$$

COMSOL Multiphysics calculates the first Piola-Kirchhoff stress P by symbolic differentiation of the strain energy expression.

The hyperelastic material models directly supported are:

Neo-Hookean

$$W_s = \frac{1}{2}\mu(\bar{I}_1 - 3) + \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2 \tag{4-4}$$

Mooney-Rivlin

$$W_s = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + \frac{1}{2}\kappa(J_{\mathrm{el}} - 1)^2 \tag{4-5}$$

The quantity J_{el} is defined in Equation 4-9 below.

Instead of the normal approach—using the conjugate pair formed by the second Piola-Kirchhoff stress and the Green-Lagrange strain—use the first Piola-Kirchhoff stress, P, and its conjugate strain, the displacement gradient, $\nabla\mathbf{u}$. This makes it possible to utilize the capability of COMSOL Multiphysics to automatically differentiate an expression, thus making it easy to modify only the strain energy function. The first Piola-Kirchhoff stresses are calculated as

$$P = \frac{\partial W_s}{\partial \nabla \mathbf{u}}$$

The variation of the energy can then be expressed as

$$\sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} \right)_{\mathrm{test}} P_{ij}$$

Materials that are nearly incompressible cannot be solved using only displacement variables. The remedy is to introduce the pressure, p, as a dependent variable. The hyperelastic material model supports both the normal displacement-based formulation and the so-called mixed formulation that includes the pressure. The pressure is related to the volume change through the relation

$$p = -\kappa(J_{\mathrm{el}} - 1) \tag{4-6}$$

where κ is the bulk modulus.

The energy equations where the pressure is a dependent variable are:

Neo-Hookean

$$W_s = \frac{1}{2}\mu(\bar{I}_1 - 3) - p(J_{\mathrm{el}} - 1) - \frac{p^2}{2\kappa} \tag{4-7}$$

Mooney-Rivlin

$$W_s = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) - p(J_{\mathrm{el}} - 1) - \frac{p^2}{2\kappa} \tag{4-8}$$

It can be shown that these equations results in the same energy and gives the correct contributions to the displacement and pressure equations as Equation 4-4 and Equation 4-5.

The second Piola-Kirchhoff stress, S, and the Cauchy stress, σ, can then be calculated from the first Piola-Kirchhoff stress:

$$S = F^{-1}P$$

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

Thermal Expansion

If thermal expansion is present, a stress-free volume change occurs. In this case, J_{el} in the constitutive relations above must be regarded as the elastic part of the total volume change, that is

$$J_{\mathrm{el}} = \frac{J}{J_{\mathrm{th}}} = \frac{J}{(1+\varepsilon_{\mathrm{th}})^3} \tag{4-9}$$

Thermal Strain

Thermal strain depends on the present temperature, T, the stress-free reference temperature, T_{ref}, and the thermal expansion vector, α_{vec}

$$\varepsilon_{\mathrm{th}} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}_{\mathrm{th}} = \alpha_{\mathrm{vec}}(T - T_{\mathrm{ref}})$$

Depending on the material model, you set up α_{vec} up differently: For an isotropic material

$$\alpha_{vec} = \begin{bmatrix} \alpha \\ \alpha \\ \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

for an orthotropic material

$$\alpha_{vec} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and for an anisotropic material you enter the full thermal expansion vector as input:

$$\alpha_{vec} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 2\alpha_{xy} \\ 2\alpha_{yz} \\ 2\alpha_{xz} \end{bmatrix}$$

ENTROPY

For a linear thermoelastic solid, the entropy per unit volume is:

$$S = \rho C_P \log(T/T_0) + S_{elast}$$

where T_0 is the reference temperature, the volumetric heat capacity ρC_P is assumed independent of the temperature, and

$$S_{elast} = \alpha_{vec} \cdot \sigma \tag{4-10}$$

where σ is the stress vector, and α_{vec} is the thermal expansion vector. For an isotropic material, Equation 4-10 simplifies into

$$S_{\text{elast}} = \alpha(\sigma_x + \sigma_y + \sigma_z) \tag{4-11}$$

The entropy is a function of state and thus independent of the strain rate. The stress vector σ in the definitions Equation 4-10 and Equation 4-11 corresponds to no damping when used in a frequency response analysis, because the damping represents the rate-dependent (viscoelastic) effects in the material.

If you model the damping in the structural analysis via the loss factor, use the following definition for the elastic part of the entropy:

$$S_{\text{elast}} = \alpha_{\text{vec}} \cdot (\sigma - j\eta D\varepsilon)$$

where η is the loss factor, and j is the imaginary unit. For more informaton, see "Loss Factor Damping" on page 146.

Initial Stress and Strain

Initial stress refers to the stress before the system applies any loads, displacements, or initial strains, written as

$$\sigma_0 = \begin{bmatrix} \sigma_{x0} \\ \sigma_{y0} \\ \sigma_{z0} \\ \tau_{xy0} \\ \tau_{yz0} \\ \tau_{xz0} \end{bmatrix}$$

The initial strain is the one before the system has applied any loads, displacements, or initial stresses

$$\varepsilon_0 = \begin{bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \varepsilon_{z0} \\ 2\varepsilon_{xy0} \\ 2\varepsilon_{yz0} \\ 2\varepsilon_{xz0} \end{bmatrix}$$

Follower Loads

Follower loads are loads that change direction as the body deforms. The most common type of follower load is a pressure acting on a surface. In this case the force changes size due to the stretching of the surface and direction due to the change in normal direction. The following section only considers this type of follower loads.

THEORY

The continuum application modes are formulated in the reference frame (the default). All forces must be specified as force/undeformed area in a fixed coordinate system (not in a system following the body). This makes it difficult to model a pressure acting on a surface because the force changes direction due to the deformation. There is also an area effect due to the stretching of the surface. The relation between the deformed area *da* and the undeformed area *dA* is needed. To handle this, the software uses a deformed frame that computes both the deformed area and the deformed normal direction (**n**). The force is calculated as

$$\mathbf{F}dA = -P\mathbf{n}da = -P\mathbf{n}\frac{da}{dA}dA$$

where P are the follower pressure.

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

$$\mathbf{F}dA = -P\mathbf{n}\frac{da}{dA}dA\left(1+\frac{\partial w}{\partial z}\right)$$

Axial Symmetry

The extra r in the circumferential integration of the force expressions is transformed to $r + uaxi$ to account for the deformation.

Implementation

The COMSOL Multiphysics implementation of these equations in the application modes for structural analysis is based on the principle of virtual work expressed in global or local stress and strain components. The principle of virtual work states that the sum of virtual work from internal strains is equal to work from external loads.

The total stored energy, W, for a linear material from external and internal strains and loads equals:

$$W = \int_V \left(\frac{1}{2}(-\varepsilon_x\sigma_x-\varepsilon_y\sigma_y-\varepsilon_z\sigma_z-2\varepsilon_{xy}\tau_{xy}-2\varepsilon_{yz}\tau_{yz}-2\varepsilon_{xz}\tau_{xz}) + \mathbf{u}^t\mathbf{F}_V\right)dv$$
$$+ \int_S \mathbf{u}^t\mathbf{F}_S ds + \int_L \mathbf{u}^t\mathbf{F}_L dl + \sum_p \mathbf{U}^t\mathbf{F}_P .$$

The principle of virtual work states that

$$\delta W = 0 ,$$

and in order to derive the expression for the variation of W you differentiate symbolically, reaching the expression

$$\delta W = \int_V (-\varepsilon_{x\text{test}}\sigma_x-\varepsilon_{y\text{test}}\sigma_y-\varepsilon_{z\text{test}}\sigma_z$$
$$-2\varepsilon_{xy\text{test}}\tau_{xy}-2\varepsilon_{yz\text{test}}\tau_{yz}-2\varepsilon_{xz\text{test}}\tau_{xz} + \mathbf{u}^t_{\text{test}}\mathbf{F}_V)dv$$
$$+ \int_S \mathbf{u}^t_{\text{test}}\mathbf{F}_S ds + \int_L \mathbf{u}^t_{\text{test}}\mathbf{F}_L dl + \mathbf{U}^t_{\text{test}}\mathbf{F}_P .$$

The principle of virtual work for the axial symmetry case reads

$$dW = \int_A r(-\varepsilon_{r\text{test}}\sigma_r-\varepsilon_{\varphi\text{test}}\sigma_\varphi-\varepsilon_{z\text{test}}\sigma_z-2\varepsilon_{rz\text{test}}\tau_{rz}$$
$$+ r\cdot \text{uor}_{\text{test}}F_r + w_{\text{test}}F_z)dA +$$
$$\int_S r(r\cdot \text{uor}_{\text{test}}F_r + w_{\text{test}}F_z)ds + (r\cdot \text{uor}_{\text{test}}F_r + w_{\text{test}}F_z\ )/2\pi$$

To avoid division by r, the true radial displacement, u is replaced in the above equation by a new dependent variable

$$\text{uor} = \frac{u}{r} .$$

If you define the material in a local user-defined coordinate system, the variational equation in COMSOL Multiphysics is expressed in local instead of global stresses and strains.

To create the strain tensor in local coordinates, transform the global strain tensor

$$\varepsilon_l = T^T \varepsilon_g T$$

where T is the local-to-global coordinate-system transformation matrix.

Then calculate the local stress tensor from the local strain, and the global stress tensor by transforming the local stress tensor

$$\sigma_g = T \sigma_l T^T$$

SETTING UP EQUATIONS FOR DIFFERENT ANALYSES

All continuum application modes support static, eigenfrequency, transient, frequency-response, parametric, and quasi-static transient analyses. Each type might solve a different equation or employ a different solver. You control this choice with the **Analysis type** property that appear in the **Application Mode Properties** dialog box for the corresponding application mode.

Static, Parametric, and Quasi-Static Transient Analysis

These analyses all use the same equation, the difference being what solver that is used. In the following, static analysis is used as short for all the above analyses because they use the same equations.

COMSOL Multiphysics' implementation is based on the stress and strain variables. The normal and shear strain variables depend on the displacement derivatives (described in general 3D terms in the section "Theory Background" on page 64); the normal and shear stress variables depend on the strains (described in general 3D terms in the section"Stress-Strain Relationship" on page 66).

Using the shear and stress variables, you can express the principle of virtual work as

$$\begin{aligned} dW = \int_V (&-\varepsilon_{x\text{test}}\sigma_x - \varepsilon_{y\text{test}}\sigma_y - \varepsilon_{z\text{test}}\sigma_z \\ &-2\varepsilon_{xy\text{test}}\tau_{xy} - 2\varepsilon_{yz\text{test}}\tau_{yz} - 2\varepsilon_{xz\text{test}}\tau_{xz} + \mathbf{u}^t_{\text{test}}\mathbf{F}_V)dv \\ &+ \int_S \mathbf{u}^t_{\text{test}}\mathbf{F}_S ds + \int_L \mathbf{u}^t_{\text{test}}\mathbf{F}_L dl + \sum_p U^t_{\text{test}}\mathbf{F}_P \,. \end{aligned}$$

If you describe the material in a local coordinate system, dW is expressed in local stresses and strains.

Transient Analysis

For transient problems consider Newton's second law

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} - \nabla \cdot c\nabla\mathbf{u} = \mathbf{F} \ .$$

It defines the equation of motion with *no damping*.

To model viscous damping, COMSOL Multiphysics uses *Rayleigh damping*, where you specify two damping coefficients. As an example, consider a system with a single degree of freedom. The equation of motion for such a system with viscous damping is

$$m\frac{d^2u}{dt^2} + \xi\frac{du}{dt} + ku = f(t) \ .$$

In the Rayleigh damping model, you express the damping parameter ξ in terms of the mass m and the stiffness k as

$$\xi = \alpha_{dM}m + \beta_{dK}k$$

The Rayleigh damping proportional to mass and stiffness is added to the static weak term.

Frequency Response Analysis

You specify harmonic loads using two components:

- The amplitude value, F_x
- The phase, $F_{x\mathrm{Ph}}$

To derive the equations for the steady-state response from harmonic excitation loads

$$F_{x\mathrm{freq}} = F_x(f) \cdot \cos\left(\omega t + F_{x\mathrm{Ph}}(f)\frac{\pi}{180}\right)$$

$$\mathbf{F}_{\mathrm{freq}} = \begin{bmatrix} F_{x\mathrm{freq}} \\ F_{y\mathrm{freq}} \\ F_{z\mathrm{freq}} \end{bmatrix},$$

assume a harmonic response with the same angular frequency as the excitation load

$$u = u_{\text{amp}} \cos(\omega t + \phi_u)$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

You can also describe this relationship using complex notation

$$u = \text{Re}(u_{\text{amp}} e^{j\phi_u} e^{j\omega t}) = \text{Re}(\tilde{u} e^{j\omega t}) \text{ where } \tilde{u} = u_{\text{amp}} e^{j\phi_u}$$

$$\mathbf{u} = \text{Re}(\tilde{\mathbf{u}} e^{j\omega t})$$

$$F_{x\text{freq}} = \text{Re}\left(F_x(\omega) e^{jF_{xPh}(f)\frac{\pi}{180}} e^{j\omega t} \right) = \text{Re}(\tilde{F}_x e^{j\omega t})$$

where

$$\tilde{F}_x = F_x(f) e^{jF_{xPh}(f)\frac{\pi}{180}}$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{F}_x \\ \tilde{F}_y \\ \tilde{F}_z \end{bmatrix}$$

Eigenfrequency Analysis

The eigenfrequency equations are derived by assuming a harmonic displacement field, similar as for the frequency response formulation. The difference is that this analysis 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

$$f = \left| \frac{\text{Im}(j\omega)}{2\pi} \right|$$

In the eigenfrequency analysis no damping is added to the equations.

Damped Eigenfrequency Analysis

This analysis type is similar to the eigenfrequency analysis except that it adds viscous damping terms to the equation. The analysis type supports Rayleigh damping. In addition to the eigenfrequency you can also look at the quality factor, Q, and decay factor, δ, for the model:

$$Q = \frac{\mathrm{Im}(\lambda)}{2\mathrm{Re}(\lambda)}$$

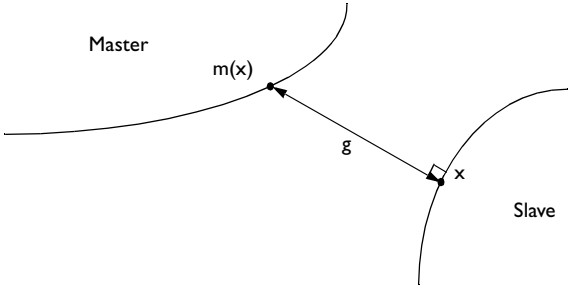
$$\delta = \mathrm{Re}(\lambda)$$

Contact Modeling

COMSOL Multiphysics solves contact problems using an *augmented Lagrangian* method. This means that the software solves the system in a segregated way. Augmentation components are introduced for the contact pressure T_n and the components T_{ti} of the friction traction vector T_t. An additional iteration level is added where the usual displacement variables are solved separately from the contact pressure and traction variables. The algorithm repeats this procedure until it fulfills a convergence criterion.

In the following equations F is the deformation gradient matrix. When looking at expressions evaluated on the slave boundaries, the expression $\mathrm{map}(E)$ denotes the value of the expression E evaluated at a corresponding master point, and g is the gap distance between the slave and master boundary.

Both the contact map operator $\mathrm{map}(E)$ and the gap distance variable are defined by the contact element `elcontact` (see the documentation of `elcontact` on page 55 of the *COMSOL Multiphysics Reference Guide* for details). For each slave point where the operator or gap is evaluated, a corresponding master point is sought by searching in the direction normal to the slave boundary.



Note that before the boundaries come in contact, the master point found is not necessarily the point on the master boundary closest to the slave point. However, as the boundaries approach one another, the master point converges to the closest point as the gap distance goes to zero.

Using the special gap distance variable, the penalized contact pressure T_{np} is defined on the slave boundary as

$$T_{np} = \begin{cases} T_n - p_n g & \text{if } g \le 0 \\ T_n e^{-\frac{p_n g}{T_n}} & \text{otherwise} \end{cases} \tag{4-12}$$

where g is the gap distance between the *slave* and *master* boundary, and p_n is the user defined normal penalty factor.

The *penalized friction traction* T_{tp} is defined on the slave boundary as:

$$T_{\mathrm{tp}} = \min\left(\frac{T_{\mathrm{tcrit}}}{|T_{\mathrm{ttrial}}|}, 1\right) T_{\mathrm{ttrial}} \tag{4-13}$$

where T_{ttrial} is defined as

$$T_{\mathrm{ttrial}} = T_t - p_t \mathrm{map}(F)(x^m - x^m{}_{\mathrm{old}}) \tag{4-14}$$

and

$$x^m = \mathrm{map}(x) \tag{4-15}$$

where x are the space coordinates.

In Equation 4-14 p_t is the user-defined friction traction penalty factor, and $x^m{}_{\mathrm{old}}$ is the value of x^m in the last time step, and

$$\mathrm{map}(F)(x^m - x^m{}_{\mathrm{old}}) \tag{4-16}$$

is the vector of slip since the last time step (approximated using a backward Euler step).

T_{tcrit} is defined as

$$T_{\mathrm{tcrit}} = \min(\mu T_{np} + \mathrm{cohe}, T_{\mathrm{tmax}}) \tag{4-17}$$

In Equation 4-17 μ is the friction coefficient, cohe is the user-defined cohesion sliding resistance, and T_{tmax} 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 weak equation on the slave boundary:

$$\int_{\text{slave}} (T_{np}\delta g + T_{tp} \cdot m(F)\delta x^m)dA + \int_{\text{slave}} (w_{\text{cn}}\delta T_n + w_{\text{ct}} \cdot \delta T_t)dA \qquad (4\text{-}18)$$

where w_{cn} and w_{ct} are contact help variables defined as:

$$w_{\text{cn}} = T_{np,i} - T_{n,i+1} \qquad (4\text{-}19)$$

$$w_{\text{ct}} = (\text{friction}\ \ (T_{tp} - (n \cdot T_{tp})n))_i - T_{t,i+1} \qquad (4\text{-}20)$$

where i is the augmented solver iteration number and friction is a Boolean variable stating if the parts are in contact.

FRICTION

The friction model is either no friction or Coulomb friction.

The friction coefficient μ is defined as

$$\begin{cases} \mu_{\text{d}} + (\mu_{\text{s}} - \mu_{\text{d}})e^{-\text{dcfric}|v_s|} & \text{if dynamic friction} \\ \mu_{\text{s}} & \text{otherwise} \end{cases} \qquad (4\text{-}21)$$

where μ_s is the static coefficient of friction and μ_{d} is the *dynamic friction coefficient*. v_s is the slip velocity, and dcfric is a decay coefficient.

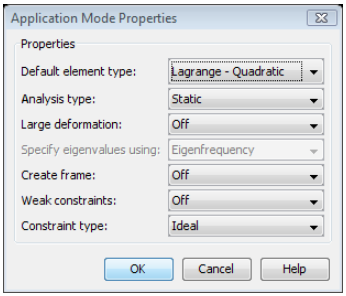
Application Mode Description

This section describes how to define a model using the continuum application modes. It reviews the following subsections:

- Application mode properties
- Scalar variables
- Material
- Constraints
- Loads
- Thermal coupling
- Damping
- Initial stresses and strains
- Perfectly matched layers (PMLs)
- Contact

Properties

To set or examine application mode properties, go to the **Physics>Properties** menu to open the **Application Mode Properties** dialog box. Here you control various global settings for the model:



Application Mode Properties dialog box for the structural mechanics application modes.

- **Default element type**: The selected finite element type that makes up the discretized finite element model is the default on all new subdomains, and the choice does not affect subdomains already created. Available elements are:
  - **Lagrange - Linear**
  - **Lagrange - Quadratic**
  - **Lagrange - Cubic**
  - **Lagrange - Quartic**
  - **Lagrange - Quintic**
  - **Lagrange - U_2P_1**
  - **Lagrange - U_3P_2**
  - **Lagrange - U_4P_3**
  - **Lagrange - U_5P_4**
- **Analysis type**: This drop-down list shows the various analyses you can perform; the default is **Static**. Your choice affects both the equations and which solver COMSOL Multiphysics uses when the **Auto select solver** option in the **Solver Parameters** dialog box is active.

ANALYSIS TYPE	COMSOL MULTIPHYSICS SOLVER
Static	Stationary
Static elasto-plastic material	Parametric

ANALYSIS TYPE	COMSOL MULTIPHYSICS SOLVER
Eigenfrequency	Eigenvalue
Damped Eigenfrequency	Eigenvalue
Time dependent	Time dependent
Frequency response	Parametric
Parametric	Parametric
Quasi-static transient	Time dependent
Linear buckling	Eigenvalue

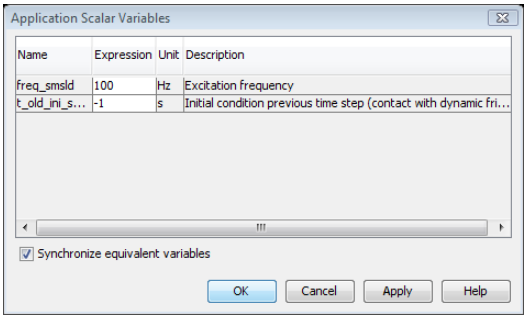
- **Large deformation**: This list controls whether or not the model should include large deformations, which only the Principle of virtual work supports (see next item). The default is **Off**.
- **Specify eigenvalues using**: This list controls how to work with eigenmode analyses. Here you should specify **Eigenvalue** or **Eigenfrequency**; this property is enabled only for eigenfrequency and linear-buckling analyses.
- **Create frame**: This list controls whether or not to create a deformed frame. The default is **Off**. A deformed frame is needed in contact modeling and to define follower forces but can also be used in a multiphysics context to define some other physics on.
- **Eigenfrequency**; this property is enabled only for eigenfrequency, damped eigenfrequency and linear-buckling analyses.
- **Weak constraints**: Controls whether or not weak constraints are active. Use weak constraints for accurate reaction-force computation. When weak constraints are enabled, all constraints are weak by default, but it is possible to change this setting for individual domains.
- **Constraint type**: Constraints can be ideal or non-ideal (see "Ideal vs. Non-Ideal Constraints" on page 301 in the *COMSOL Multiphysics Modeling Guide*).

Scalar Variables

There are three different scalar variables:

- Excitation frequency, `freq`, which is applicable only for frequency response analysis.

- Initial condition for the time in the previous time step, `t_old_ini`, which is applicable only for contact modeling using dynamic friction.
- Complex angular frequency, `jomega`, which is applicable only for eigenfrequency analysis. You normally do not need to edit the complex angular frequency.



The Application Scalar Variables dialog box in a frequency response analysis.

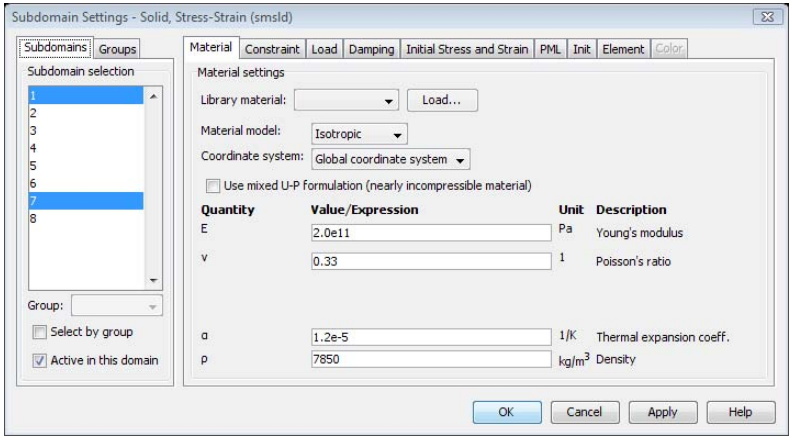
When you select a frequency response analysis, the parametric solver becomes the default solver, which makes it easy to perform a frequency sweep over several excitation frequencies in one analysis. In this case, choose **Solve>Solver Parameters**, and in the dialog box that appears go to the **General** tab. In the **Parameter** area, enter `freq_smsld` in the **Parameter name** edit field. Values that you enter in the **Parameter values** edit field override the excitation frequency you might have entered in the **Application Scalar Variables** dialog box.

To access the excitation frequency f use the variable `freq` and to access the angular excitation frequency ω use `omega`.

Material Properties

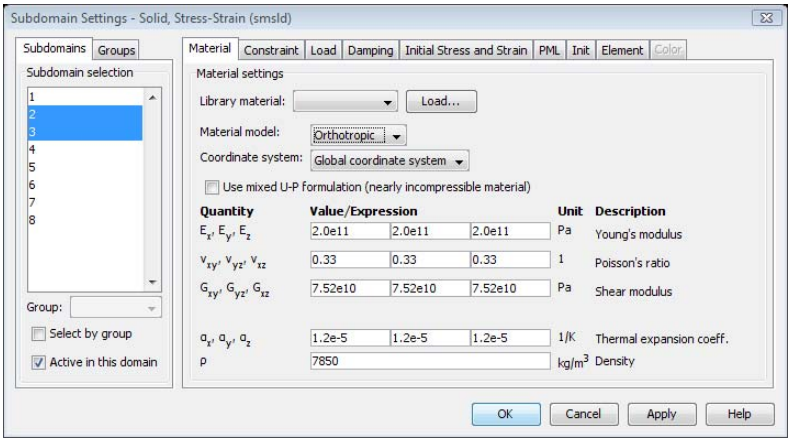
You define material properties on the **Material** page in the **Subdomain Settings** dialog box. This page has two lists: **Material model** and **Coordinate system**. Now consider the options available for each of these lists:

- **Material model**: When you select the type of material, a set of appropriate material properties appear in the dialog box.
  - **Isotropic material**: This material has the same properties in all directions.



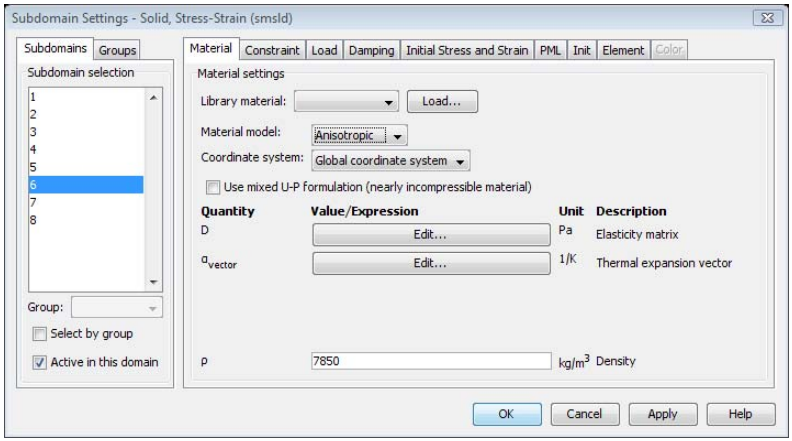
Material properties for an isotropic material.

  - **Orthotropic material**: This material has different material properties in different directions, and its stiffness depends on the properties E_i, ν_{ij}, and G_{ij} (see page 66 for details). In addition, thermal expansion depends on the parameter α_i (see page 78 for details).

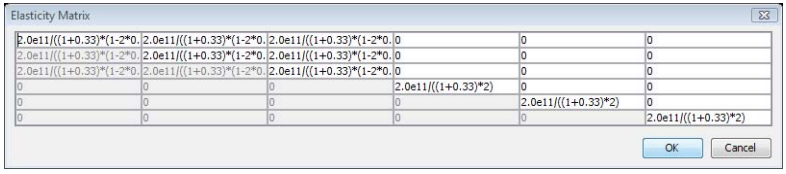


Material properties for an orthotropic material.

- **Anisotropic material**: This material has different material properties in different directions, and the stiffness comes from the symmetric *elasticity matrix*, $\boldsymbol{D}$ (see page 66 for details). Thermal expansion depends on the *thermal expansion vector*, α_{vec} (see page 78 for details).

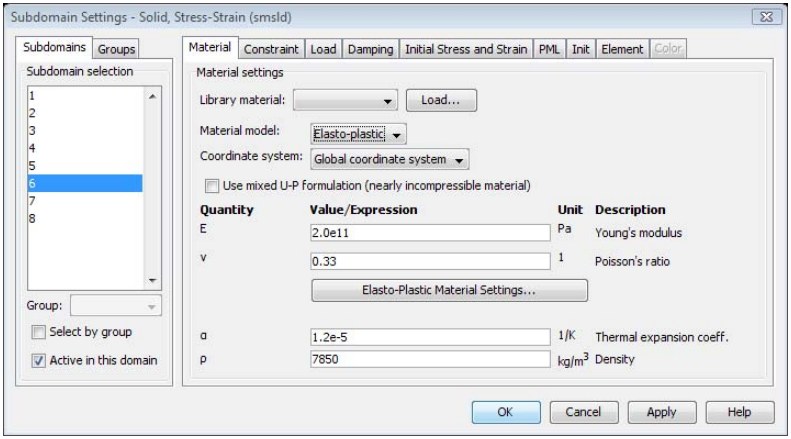


Material properties for an anisotropic material.

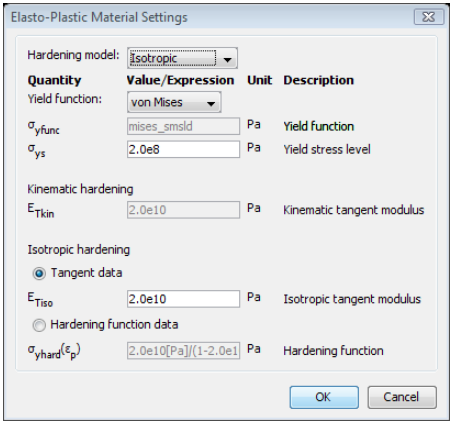


The Elasticity matrix dialog box for entering the components of the D matrix for an anisotropic material.

- **Elasto-plastic material**: A nonlinear material with possible hardening (see page 73 for details).

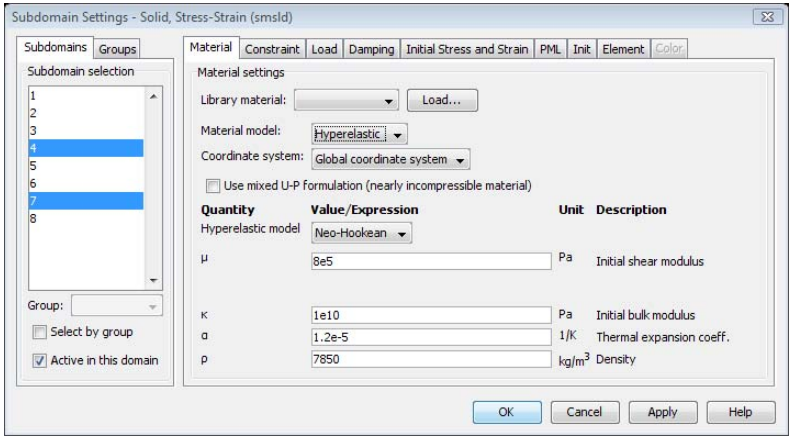


Material properties for an elasto-plastic material.



The Elasto-Plastic Material Settings dialog box for specification of elasto-plastic material data.

- **Hyperelastic material**: A hyperelastic material based on a strain energy density function, often used to model rubberlike materials (see page 76 for details).



Material properties for a hyperelastic material.

- **Coordinate system**: In this second list on the **Material** page you select the coordinate system on which the material properties are defined. Use it for orthotropic and anisotropic materials that are defined in another coordinate system other than the global system or if you need stresses and strains in a local coordinate system for postprocessing. The **Coordinate system** list is disabled if no user-defined coordinate systems are available. Note, to find the **Coordinate System Settings** dialog box go to the **Options** menu.
- **Use mixed U-P formulation (nearly incompressible material)**: Controls whether to use a mixed formulation adding the negative mean pressure as a dependent variable to solve for. This can also be controlled from the **Predefined element** list on the **Element** page. Nearly incompressible materials means a Poisson's ratio close to 0.5. See page 70 for details.

TABLE 4-1: MATERIAL PROPERTIES FOR VARIOUS MATERIAL MODELS

PARAMETER	VARIABLE	DESCRIPTION	MATERIAL MODEL
E	E	Young's modulus	Isotropic/ elasto-plastic
ν	nu	Poisson's ratio	Isotropic/ elasto-plastic
ρ	rho	Density	All
α	alpha	Thermal-expansion coefficient	Isotropic
th	`thickness`	The thickness of the geometry	All

TABLE 4-1: MATERIAL PROPERTIES FOR VARIOUS MATERIAL MODELS

PARAMETER	VARIABLE	DESCRIPTION	MATERIAL MODEL
E_i	E`i`	Young's modulus in the x_i direction	Orthotropic
ν_{ij}	nu`ij`	Poisson's ratio for the $x_i x_j$-plane	Orthotropic
G_{ij}	G`ij`	Shear modulus for the $x_i x_j$-plane	Orthotropic
α_i	alpha`i`	Thermal-expansion coefficient in the x_i direction	Orthotropic
D		Elasticity matrix for the anisotropic case	Anisotropic
α_{vec}		Thermal-expansion vector for the anisotropic case	Anisotropic
σ_{ys}	Sys	Yield stress level	Elasto-plastic
σ_{yfunc}	Syfunc	Yield function	Elasto-plastic
σ_{yhard}	Syhard	Hardening function for isotropic hardening	Elasto-plastic
E_{Tiso}	ETiso	Isotropic-tangent modulus	Elasto-plastic
E_{Tkin}	ETkin	Kinematic-tangent modulus	Elasto-plastic
C_{10}	C10	Mooney-Rivlin material parameter	Hyperelastic
C_{01}	C01	Mooney-Rivlin material parameter	Hyperelastic
μ	mu	Initial shear modulus	Hyperelastic
κ	kappa	Initial bulk modulus	Hyperelastic

The index i in the parameters E_i and α_i in Table 4-1 refers to the space coordinates x_i and represents the following names for the different application modes:

APPLICATION MODE	x_1	x_2	x_3
Plane Stress and Plane Strain	x	y	z
Solid	x	y	z
Axisymmetry Stress-strain	r	φ	z

Example: E_i for axisymmetry stress-strain means E_r, E_φ, and E_z.

The parameter ν_{ij} in Table 4-1 refers to the space coordinates $x_i x_j$ and is defined for the following combinations of i and j for the different application modes:

APPLICATION MODE	x_1x_2	x_2x_3	x_1x_3
Plane Stress and Plane Strain	xy	yz	xz
Solid	xy	yz	xz
Axisymmetry Stress-strain	$r\varphi$	φz	rz

and the parameter G_{ij} is defined for these combinations:

APPLICATION MODE	x_1x_2	x_2x_3	x_1x_3
Plane Stress and Plane Strain	xy		
Solid	xy	yz	xz
Axisymmetry Stress-strain			rz

Note: You can change the default names for the space coordinates in the same way as you can the names of the dependent variables.

Now examine the various material properties in Table 4-1.

Young's modulus It defines a material's modulus of elasticity, E. For an isotropic material it is the spring stiffness in Hooke's law, which in 1D form is

$$\sigma = E\varepsilon$$

where σ is the stress and ε is the strain. An orthotropic material uses one value of Young's modulus for each direction, E_i as defined on page 66.

Poisson's ratio Denoted by ν, it defines the normal strain in the perpendicular direction, generated from a normal strain in the other direction and follows the equation

$$\varepsilon_{\perp} = -\upsilon\varepsilon_{||}\,.$$

An orthotropic material defines three values of ν_{ij}.

Note: ν_{ij} is defined differently depending on the application field, so review page 66 for the definition within COMSOL Multiphysics. It is easy to transform among definitions, but you must check which one your material uses.

Shear Modulus Denoted by G_{ij}, it defines the relationship between engineering shear strain and shear stress. It is applicable only to an orthotropic material and follows the equation

$$\varepsilon_{ij} = \frac{\tau_{ij}}{G_{ij}}.$$

Density This entry specifies ρ, the material's density.

Thermal expansion coefficient It defines how much a material expands due to an increase in temperature following the equation

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

where ε_{th} is the thermal strain, and α is the thermal expansion coefficient. With it you model thermal strain for an isotropic material. For an orthotropic material, three values of α_i are defined for the three perpendicular directions.

Thickness (th) This property defines the out of plane thickness of the geometry for the Plane Stress and Plane Strain application modes.

Elasticity matrix It defines the elasticity matrix, D, for anisotropic materials (see page 67 for details). For the Plane Stress and Plane Strain application modes D is defined as a 4-by-4 matrix, since the out of plane shear stress and shear strain components are zero.

Thermal expansion vector It defines the thermal expansion vector, α_{vec}, for anisotropic materials (see page 78 for details).

Yield stress level (σ_{ys}) This parameter gives the stress level where plastic deformation starts. In the theory section this parameter is named Y_0.

Yield function (σ_{func}) This function detects if plasticity has occurred. In the theory section this parameter is named ϕ.

Isotropic tangent modulus This parameter is the tangent modulus used for isotropic hardening. This parameter together with σ_{ys} defines the f_2 function from the theory section as

$$f_2(\varepsilon_{\text{pe}}) = \sigma_{\text{ys}} + \frac{E_{T\text{iso}}}{1 - \dfrac{E_{T\text{iso}}}{E}}\varepsilon_{\text{pe}}$$

Kinematic tangent modulus This parameter is the tangent modulus used for kinematic hardening. This parameter is used to calculate the σ_{shift} parameter from the theory section as

$$\sigma_{\text{shift}} = \frac{E_{T\text{kin}}}{1 - \frac{E_{T\text{kin}}}{E}} \cdot \frac{2}{3} \cdot \varepsilon_p$$

Hardening function (σ_{yhard}) This hardening function applies to isotropic hardening. This parameter together with σ_{ys} defines the f_2 function from the theory section as

$$f_2(\varepsilon_{\text{pe}}) = \sigma_{\text{ys}} + \sigma_{\text{yhard}}(\varepsilon_{\text{pe}})$$

This definition implies that you have to subtract the yields stress level (σ_{ys}) when defining your hardening function.

Mooney-Rivlin material parameters Hyperelastic material model parameters.

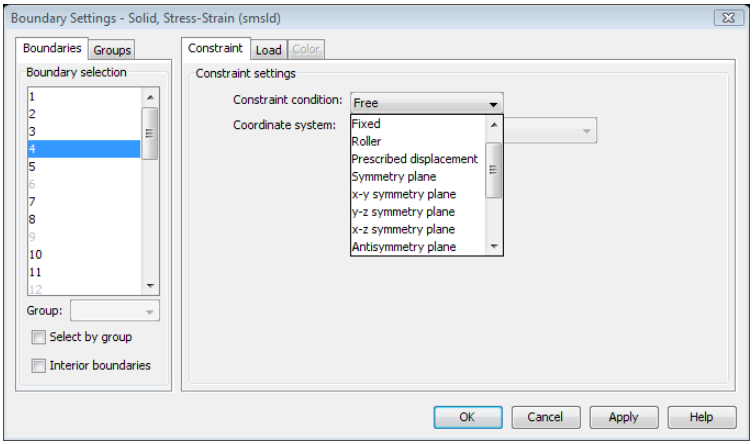
Initial shear modulus Used for Neo-Hookean hyperelastic material model.

Initial bulk modulus Used for Neo-Hookean and Mooney-Rivlin hyperelastic material models.

Constraints

A constraint specifies the displacement of certain parts of a structure. You can define constraints on all domain levels: points, edges, faces/boundaries, and subdomains in 3D; points, boundaries, and subdomains in 2D. To control a constraint, go to the **Constraint** page that appears in the following dialog boxes that you find in the **Physics** menu: **Physics>Subdomain Settings**, **Physics>Boundary Settings**, **Physics>Edge Settings**, and **Physics>Point Settings**. The following figure shows the **Boundary Settings** dialog

box for the Solid, Stress-Strain application mode, but the **Constraints settings** area has the same appearance in all structural mechanics continuum application modes



An example of a typical Constraint page, taken here from the Solid, Stress-Strain application mode Boundary Settings dialog box.

Within the dialog box the **Constraint condition** list lets you control what type of constraint you want to define. You can choose from the following options:

CONSTRAINT CONDITION	BOUNDARY	SUBDOMAIN	USE WHEN
Free	√	√	The domain has no constraint
Fixed	√	√	The displacement in the domain is fixed in all directions
Roller	√		The normal displacement is constrained
Prescribed displacement	√	√	The displacement in any direction need to be prescribed
Symmetry plane	√		The boundary is a symmetry plane
x-y symmetry plane	√		The selected coordinate system's x-y plane is a symmetry plane
y-z symmetry plane	√		The selected coordinate system's y-z plane is a symmetry plane
x-z symmetry plane	√		The selected coordinate system's x-z plane is a symmetry plane
Antisymmetry plane	√		The boundary is an antisymmetry plane
x-y antisymmetry plane	√		The selected coordinate system's x-y plane is an antisymmetry plane

CONSTRAINT CONDITION	BOUNDARY	SUBDOMAIN	USE WHEN
y-z antisymmetry plane	√		The selected coordinate system's y-z plane is an antisymmetry plane
x-z antisymmetry plane	√		The selected coordinate system's x-z plane is an antisymmetry plane
Prescribed velocity	√	√	The velocity in any direction need to be prescribed, only available for frequency response analysis
Prescribed acceleration	√	√	The acceleration in any direction need to be prescribed, only available for frequency response analysis

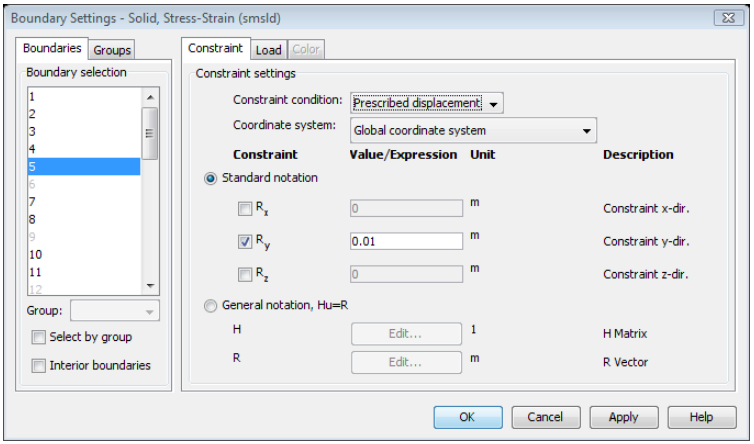
The symmetry or antisymmetry condition has the following interpretation.

CONDITION	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
x-y symmetry plane			√
y-z symmetry plane	√		
x-z symmetry plane		√	
x-y antisymmetry plane	√	√	
y-z antisymmetry plane		√	√
x-z antisymmetry plane	√		√

The **Coordinate system** list lets you control in which coordinate system you want the constraint defined. Available options are:

- Global coordinate system
- Tangent and normal coordinate system (available only on boundaries)
- User-defined coordinate systems if any local coordinate systems are defined.

When you select **Prescribed displacement** a number of new options appears in the dialog box, and the **Constraint** page takes on this appearance:



The Constraint page showing the Prescribed displacement options.

You can prescribe a constraint in two ways:

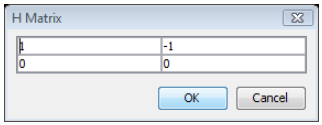
- In standard notation (select this option by clicking the **Standard notation** button), you constrain each displacement direction independently. The check boxes adjacent to the $\mathbf{R_x}$, $\mathbf{R_y}$, and $\mathbf{R_z}$ edit fields activate the constraint, whereupon you enter the value/expression of the displacement (the default value is `0`).
- In general notation (select this option by clicking the **General notation, Hu=R** button) lets you specify constraints as any linear combination of displacements components. For instance, in the 2D case, use the relationship

$$H \begin{bmatrix} u \\ v \end{bmatrix} = R \, .$$

  Enter values for the H matrix and R vector in corresponding dialog boxes by clicking the corresponding **Edit** buttons. For example, to achieve the condition $u = v$, use the settings

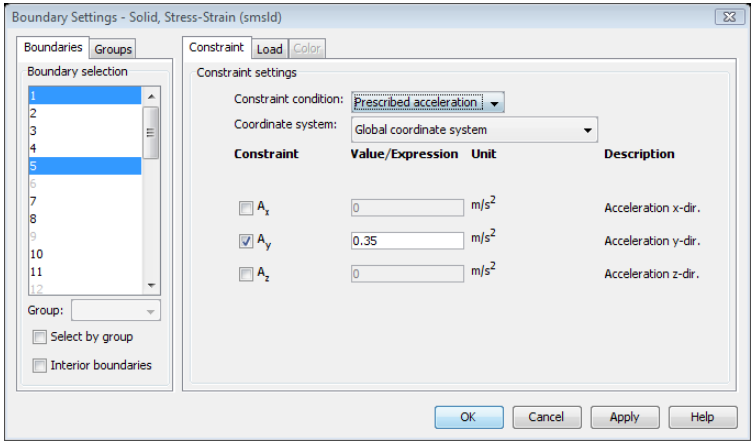
$$H = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which force the domain to move only diagonally in the *x-y* plane.



The H Matrix dialog box for the example in the text.

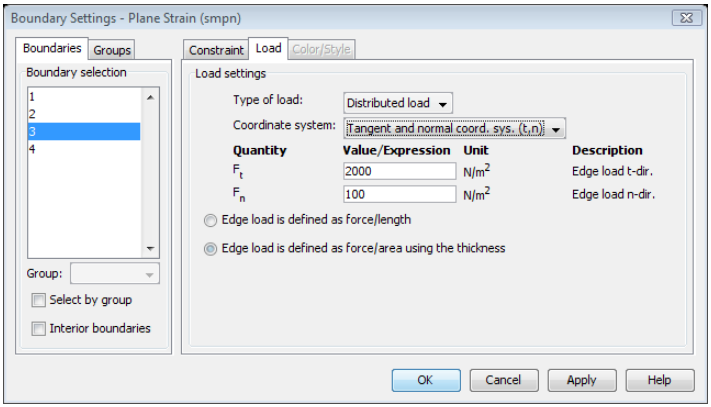
In a frequency response analysis you have the possibility to specify not only a harmonic displacement but also a harmonic velocity or acceleration. You specify a prescribed velocity or acceleration in the same way as **Prescribed displacement** using **Standard notation** by first selecting **Prescribed velocity** or **Prescribed acceleration** in the **Constraint condition** list.



Constraint page showing the Prescribed acceleration settings.

Load

"Load" is a general term for forces applied to a structure. You can specify loads on all domain types using the **Load** page that appears in the following dialog boxes that you find on the **Physics** menu: **Subdomain Settings**, **Boundary Settings**, **Edge Settings**, and **Point Settings**.



The Boundary Settings dialog box for the Plane Strain application mode shown here is representative of load pages for all domain levels in all structural mechanics application modes.

The loads on all levels except the point level are given as *distributed loads* using a force density such as; force/length, force/area, or force/volume.

For boundaries you have the option to specify between different types of loads using the **Type of load** list. You select between distributed load and *follower load* (distributed load is the default setting).

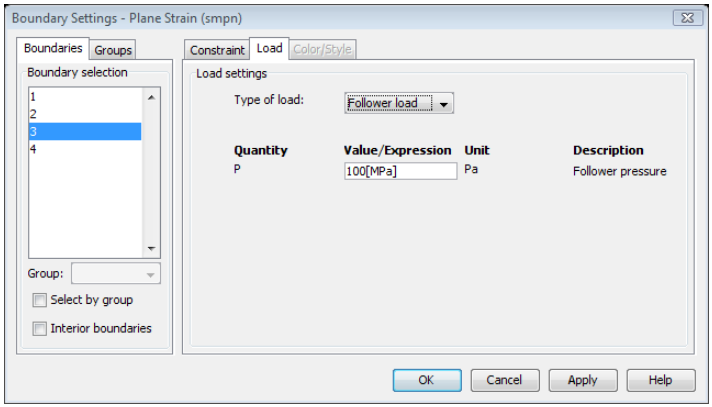
FOLLOWER LOADS

Follower loads are loads that depend on the deformation. The most common case is a pressure directed along the negative normal direction to a surface that deforms. In the following, follower loads imply such a load. Follower loads are only available on boundaries.

All loads must be applied in the undeformed reference frame; the software then computes the follower load using a frame that deforms with the structure. Both the direction and the size of the load change as the structure deforms. The **Create frame** application mode property is automatically set to **On** as soon as you specify a follower force.

Follower loads are only meaningful in a large deformation analysis. The **Large deformation** application mode property is automatically set to **On** as soon as you specify a follower force.

You select **Follower load** from the **Type of load** list on the **Load** page. You specify the pressure in the **P** edit field.



The Boundary Settings dialog box for the Plane Strain application mode showing the follower load setting.

DISTRIBUTED LOADS

Distributed load is the default setting on boundaries. On all other levels a distributed load is the only way to specify a load. For boundaries you select between distributed loads and follower loads using the **Type of load** list.

For plane stress and plane strain, two option buttons allow you to choose how to specify the load using the thickness. The following table shows how to define the loads on different domains in different application modes; the entries give the SI unit in parenthesis.

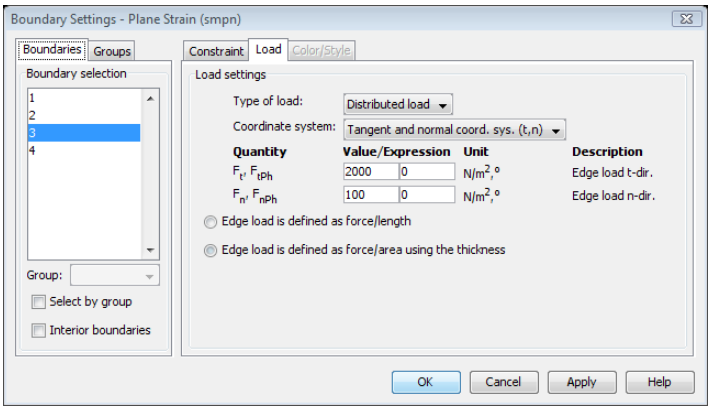
APPLICATION MODE	POINT	EDGE	BOUNDARY	SUBDOMAIN
Plane Stress, Plane Strain	force (N)		force/area (N/m^2) or force/length (N/m)	force/volume (N/m^3) or force/area (N/m^2)
Axisymmetry, Stress-Strain	total force along the circumferential (N)		force/area (N/m^2)	force/volume (N/m^3)
Solid, Stress-Strain	force (N)	force/length (N/m)	force/area (N/m^2)	force/volume (N/m^3)

Within the dialog box, the **Coordinate system** list lets you control in which coordinate system you want to define the load. Available options are:

- Global coordinate system
- Tangent and normal coordinate system (available only on boundaries)

- User-defined coordinate systems, if any local coordinate systems are defined.

For a frequency response analysis you have additional input data. To control the analysis type, use the **Application Mode Properties** dialog box. When frequency response is the analysis type, the **Load** page takes on this appearance:



The Load page that appears for frequency response analysis.

For frequency response analysis, the application mode splits the harmonic load into two parameters:

- The amplitude, F
- The phase (F_{Ph})

Together they define a harmonic load whose amplitude and phase shift can vary with the excitation frequency, f

$$F_{\mathrm{freq}} = F(f) \cdot \cos(2\pi f + F_{\mathrm{Ph}}(f)) .$$

For subdomains, you have additional options to control if and how the analysis should include thermal strains (explained in the following section).

Thermal Coupling

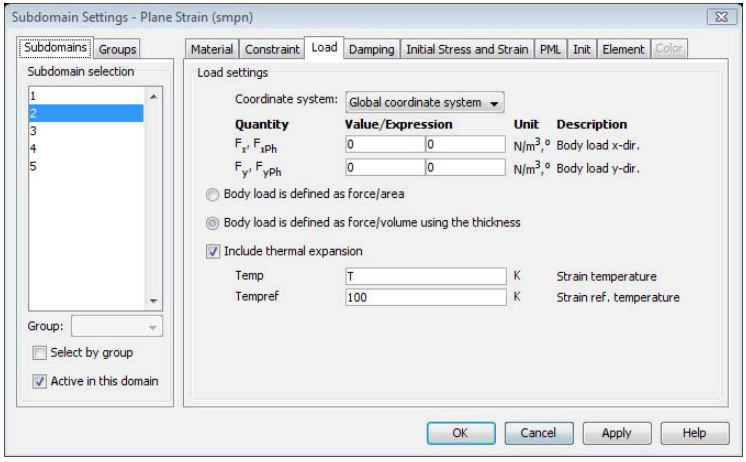
Material expands with temperature, causing thermal strains to develop in the material. The thermal strains, taken together with the initial strains and elastic strains from structural loads, form the total strain

$$\varepsilon = \varepsilon_{\mathrm{el}} + \varepsilon_{\mathrm{th}} + \varepsilon_0$$

where

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

Thermal coupling means that the analysis includes thermal expansion. Details on thermal coupling appear on page 78. You specify thermal effects on the **Load** page in the dialog box that appears when you choose **Physics>Subdomain Settings**.



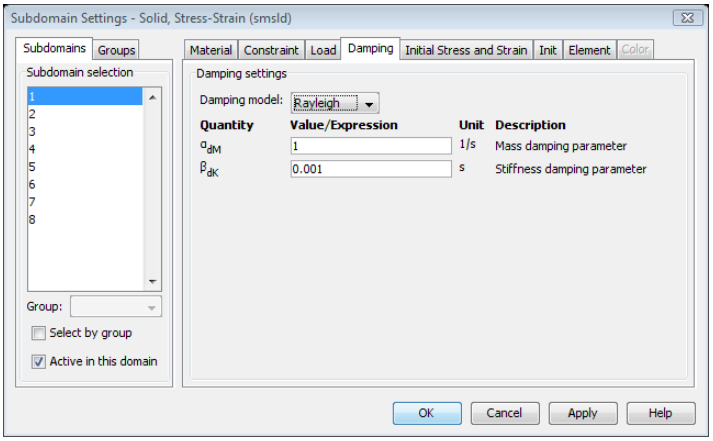
You specify thermal effects on the Load page.

The **Include thermal expansion** check box instructs the model to add thermal effects. Specify the strain temperature, T, and reference temperature, T_{ref}, in the **Temp** and **Tempref** edit fields, but you specify the thermal expansion coefficient on the **Material** page (see page 93). T and T_{ref} can be any expression and are typically another dependent variable for temperature solved for in a COMSOL Multiphysics heat transfer application mode. Any type of analysis can use this temperature coupling.

Note: Special approach is required if the structural analysis is performed in the frequency domain. This includes the following analysis types: **Frequency response**, **Eigenfrequency**, and **Damped eigenfrequency**. The coupled displacement-temperature field presents thermoelastic oscillations of small amplitude, which are initialized to zero. You need to set the strain reference temperature **Tempref** to zero and use a special form of the heat balance equation.

Damping

In transient and frequency response analyses you have the possibility to model undamped or damped problems. You can specify damping on the subdomain level using the **Damping** page that appears in the **Subdomain Settings** dialog box. From the **Damping models** list you can select **No damping**, **Rayleigh**, or **Loss factor**, and the contents of the dialog box changes for each of these damping models.



Damping page when Rayleigh damping is selected.

Note: Loss factor damping is valid only for frequency response analysis. If you choose transient analysis and loss factor damping, COMSOL Multiphysics solves the model with no damping.

Table 4-2 and the subsequent text describe the parameters that define damping:

TABLE 4-2: PARAMETERS FOR DAMPING MODELS

PARAMETER	VARIABLE	DESCRIPTION	DAMPING MODEL
α_{dM}	alphadM	Mass-damping parameter	Rayleigh
β_{dK}	betadK	Stiffness-damping parameter	Rayleigh
η	eta	Loss factor	Loss factor

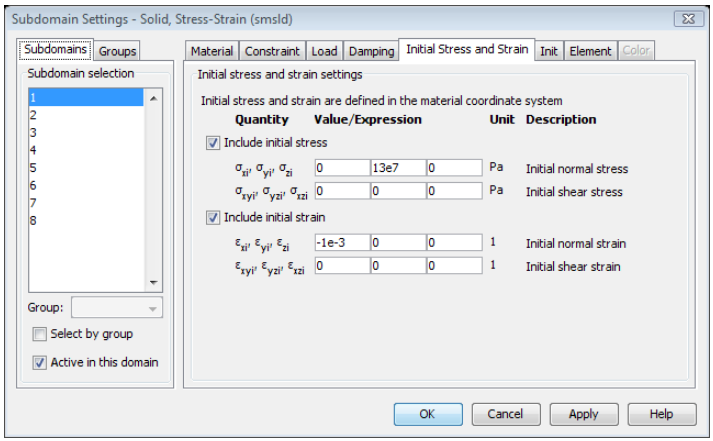
Mass damping parameter Defines the Rayleigh damping model's mass damping, α_{dM}.

Stiffness damping parameter Defines the Rayleigh damping model's stiffness damping, β_{dK}.

Loss factor Defines the loss factor η for the loss factor damping model.

Initial Stress and Strain

An analysis can include initial stress and strain, which is the stress/strain state in the structure before the model applies any constraint or load. Initial strain can, for instance, describe moisture-induced swelling, and initial stress can describe stresses from heating. In fact, you can think of initial stress and strain as different ways to express the same thing. To specify them, go to the **Initial Stress and Strain** page in the **Physics>Subdomain Settings** dialog box.



Dialog box for setting up initial stress and strain.

You control the option to include initial stress and strain independently using like-named two check boxes.

In the following table, the index l for parameter σ_{0l}/`sli` and ε_{0l}/`eli` refers to the space coordinates x_l.

PARAMETER	VARIABLE	DESCRIPTION
σ_{0l}	sli	Initial normal stress
τ_{0lk}	slki	Initial shear stress

PARAMETER	VARIABLE	DESCRIPTION
ε_{0l}	eli	Initial normal strain
ε_{0lk}	elki	Initial shear strain

The index l runs over the following coordinate names for the different application modes.

APPLICATION MODE	x_1	x_2	x_3
Plane Stress and Plane Strain	x	y	z
Solid, Stress-Strain	x	y	z
Axial Symmetry, Stress-Strain	r	φ	z

Example: σ_{0l} for axial symmetry stress-strain means σ_{0r}, $\sigma_{0\varphi}$, and σ_{0z}.

The parameters σ_{0lk}/`slki` and ε_{0lk}/`elki` in the first table refer to the space coordinates $x_l x_k$ and are defined for the following combinations of l and k for the different application modes:

APPLICATION MODE	x_1x_2	x_2x_3	x_1x_3
Plane Stress and Plane Strain	xy		
Solid, Stress-Strain	xy	yz	xz
Axial Symmetry, Stress-Strain	rz		

Perfectly Matched Layers (PMLs)

In frequency response analysis or elastic waves, you can use perfectly matched layers to simulate absorbing boundaries. A PML is strictly speaking not a boundary condition but an additional domain that absorbs the incident radiation without producing reflections. It provides good performance for a wide range of incidence angles and is not particularly sensitive to the shape of the wave fronts. The PML formulation introduces a complex-valued coordinate transformation under the additional requirement that the wave impedance should not be affected. The following sections describe how to create Cartesian, cylindrical, and spherical PMLs for elastic waves.

For an account of elastic waves in solids, see Chapters 4 and 5 of Ref. 1. For background information about PMLs in elastodynamics, see Ref. 2.

PML IMPLEMENTATION

For a PML that absorbs waves in the coordinate direction ξ, the implementation uses the following coordinate transformation inside the PML:

$$\xi' = \text{sign}(\xi - \xi_0)\left|\xi - \xi_0\right|^n \frac{L}{\delta\xi^n}(1 - i) \tag{4-22}$$

The scaled PML width, L; the coordinate of the inner PML boundary, ξ_0; and the (actual) width of the PML, $\delta\xi$, are input parameters for each orthogonal absorbing coordinate direction.

The scaling exponent, n is an input parameter for each PML subdomain. The default value of n is 1, giving a linear scaling that works well in most cases, and the useful range is roughly between 1 and 2; increasing the exponent allows you to use fewer mesh elements to resolve wavelengths much smaller than the scaled PML width.

Usually, set L equal to one wavelength. What the wavelength is, depends on the type of elastic wave you are considering. For example, for longitudinal (acoustic) waves, the wavelength is given by (Ref. 1)

$$\lambda = \frac{1}{f}\sqrt{\frac{(1-\nu)}{(1+\nu)(1-2\nu)}\frac{E}{\rho}}$$

where f is the frequency, E is Young's modulus, ν is Poisson's ratio and ρ is the density. If your analysis includes several wave types of different wavelengths, set L to the longest one. For this case, you can also try to set the scaling exponent, n, equal to 2.

The parameters ξ_0 and $\delta\xi$ get default settings that the software deduces from the drawn geometry and stores in so-called guess variables. You can inspect the values of the guess variables on the **Variables** page of the **Subdomain Settings - Equation System** dialog box or at the corresponding node of the **Model Tree**.

The default settings defined by the guess variables work nicely in most cases, but they might fail for PML subdomains of nonstandard shape. Examples of geometries that work nicely are shown in the following figures for each of the available PML types:

- **Cartesian**—PMLs absorbing in Cartesian coordinate directions.
- **Cylindrical**—PMLs absorbing in cylindrical coordinate directions from a specified axis. For axisymmetric geometries the cylinder axis is the z-axis.
- **Spherical**—PMLs absorbing in the radial direction from a specified center point.

  For each of the above PML types, you can choose the coordinate directions in which the PML absorbs waves, that is, for which directions a coordinate transformation of

the type Equation 4-22 applies. To allow complete flexibility in defining a PML there is, in addition, a fourth option:

- **User defined**—General PMLs or domain scaling with user-defined coordinate transformations.

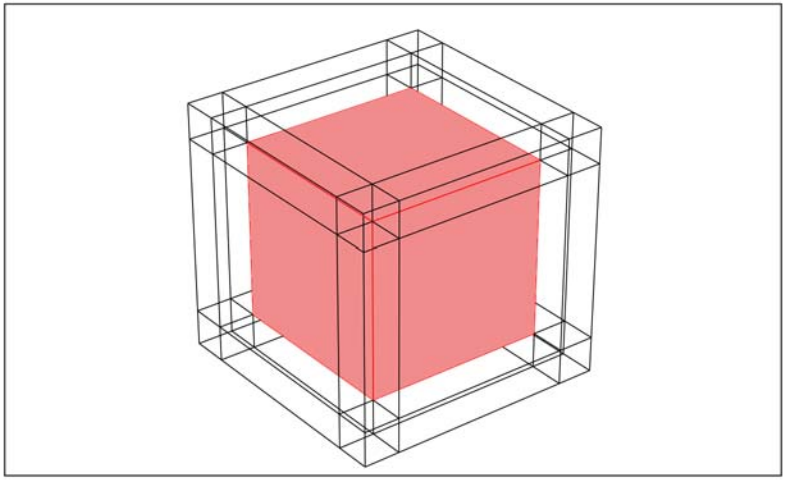


Figure 4-1: A cube surrounded by typical PML regions of the type "Cartesian."

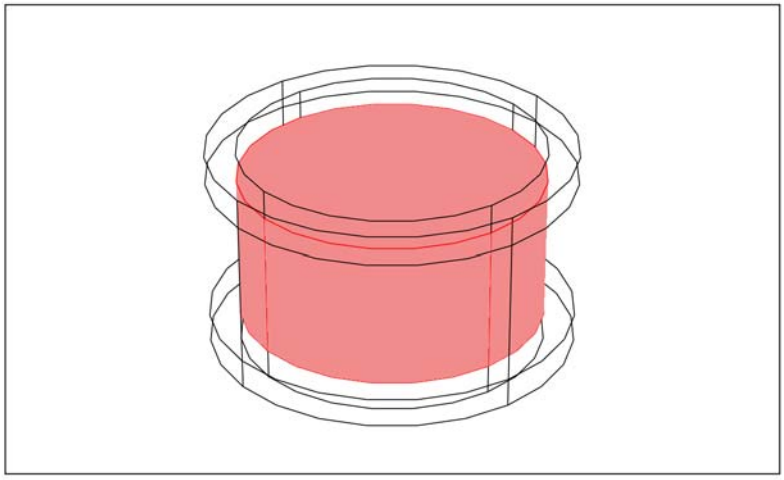


Figure 4-2: A cylinder surrounded by typical cylindrical PML regions.

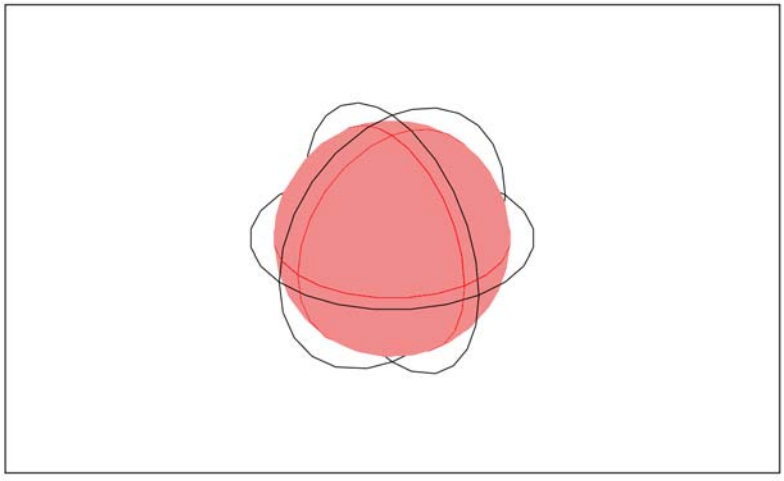


Figure 4-3: A sphere surrounded by a typical spherical PML region.

SETTING UP A PML

On the **PML** page in the **Subdomain Settings** dialog box, you can select different types of PMLs depending on what kind of wave you have:

PML TYPE	APPLICATION MODE	DESCRIPTION
None	all	No PML
Cartesian	Solid, Stress-Strain; Plane Stress; and Plane Strain	Absorbs waves in the specified Cartesian coordinate directions
Cylindrical	all	Absorbs cylindrical waves
Spherical	Solid, Stress-Strain; Axial Symmetry, Stress-Strain	Absorbs spherical waves
User defined	all	Define your own scaled space variables

The PML type **None** is default. To add a PML, select one of the other types.

Cartesian PMLs

When choosing the Cartesian PML type you can use a user-defined coordinate system to define the directions. If you want a curved coordinate system you must use the cylindrical or spherical PML type.

Select the check box for the directions in which you want the waves to be absorbed. For each of these directions, enter the scaled PML width, L in the associated edit field. Make sure all material properties are the same in the PML as in the adjacent subdomain.

Cylindrical PMLs

In 2D, a cylindrical PML always absorbs waves in the radial direction. In the other dimensions, you can decide how the PML absorbs the wave: in the radial direction, the z direction, or both.

Select the directions in which you want the PML to absorb the waves and enter the scaled PML widths in those directions. To define a cylindrical PML you also need to enter the center point of the cylindrical coordinates and, in 3D, the cylinder axis direction.

Spherical PMLs

A spherical PML always absorbs waves in the radial direction. Enter the scaled PML width, L. Define the spherical coordinates by entering the center point.

User-Defined PMLs

When using a PML, the algorithm scales the equation in this domain so that instead of the coordinates used in the rest of the model, the coordinates PMLx, PMLy, and PMLz appear in the equation. If you want to scale the equation in some other way than the automatic PML options provide, use a user-defined PML. In this case you enter your own **User-defined PML coordinates**.

REFERENCES

1. L.M. Brekhovskikh and V. Goncharov, *Mechanics of Continua and Wave Dynamics*, 2nd ed., Springer-Verlag, 1994.

2. W.C. Chew and Q.H. Liu, "Perfectly Matched Layers for Elastodynamics: A New Absorbing Boundary Condition," *J. Comp. Acoustics*, vol. 4, pp. 341–359, 1996.

Contact Modeling

You can model contact between two boundaries. The boundaries need to be connected to a subdomain active in the same application mode. To be able to model contact you need the following:

- An application mode modeling the deformation that supports contact modeling: The Plane Strain; Plane Stress; Axial Symmetry, Stress-Strain; or the Solid Stress-Strain application mode.
- A deformed frame controlled by the application mode. This is done by setting the application mode property **Create frame** to **On**. The program does this automatically when you add a contact pair.
- Use of assembly mode, if the parts are in contact initially. In this case select **Use Assembly** from the **Draw** menu. Read more about assemblies in the section "Using Assemblies" on page 351 in the *COMSOL Multiphysics Modeling Guide*.
- A contact pair. A contact pair consists of a number of slave and master boundaries. The slave is constrained not to penetrate the master boundary. You can create contact pairs from the **Contact Pairs** dialog box, which you open from the **Physics** menu. A description of how to create contact pairs appears later in this section. Additionally, if some parts of the boundaries are in initial contact, you can use the **Create Pairs** dialog box to automatically detect and define contact pairs. For more information about the **Create Pairs** dialog box, see the section "Creating Pairs" on page 353 in the *COMSOL Multiphysics Modeling Guide*.
- Contact parameters, specified to suit your model. You can inspect and change the contact parameters in the **Boundary Settings** dialog box by selecting the appropriate

contact pair on the **Pair** page. There are three different pages, described further on in this section, to do this on:

- **Contact**
- **Contact, Initial**
- **Contact, Advanced**

- Solver parameters, specified to suit your model. You can set the parameters for the augmented and nonlinear solvers in the **Solver Settings** dialog box, which is described in the section "Nonlinear Solver Settings" on page 368 in the *COMSOL Multiphysics User's Guide*. You can find recommendations for solver settings specifically for contact models on page 129 of this book.

Note: The current version supports contact in the continuum application modes: Plane Stress; Plane Strain; Axial Symmetry, Stress-Strain; and Solid, Stress-Strain.

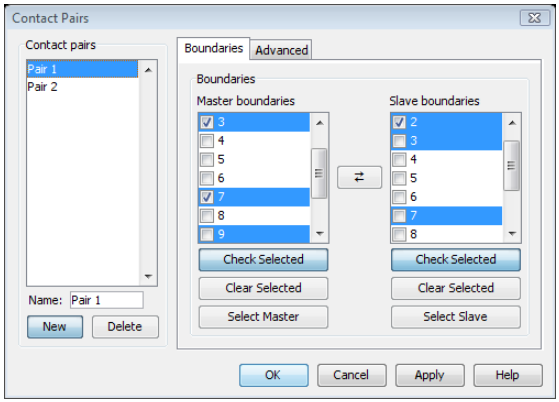
Note: You cannot use contact pairs together with the transient solver. Use the parametric solver with the time t as the parameter to solve contact problems with friction.

This section provides information about how to set up contact pairs and specify contact parameters. You can read about the theory behind the implementation of the contact modeling method on page 86 and about tips for creating a contact model on page 118.

For general information about modeling with pairs, see the section "Specifying Physics Settings on Pairs" on page 361 in the *COMSOL Multiphysics Modeling Guide*.

THE CONTACT PAIRS DIALOG BOX

To define contact pairs, choose **Physics>Contact Pairs**. This opens the **Contact Pairs** dialog box.



Boundaries Page

Each pair has a name. The application modes use this name to refer to the pair. The name must be unique.

The two boundary lists show the master and slave domains of the pair selected in the list to the left. The check boxes beside the domain numbers indicate which domains belong to the master and the slave, respectively.

Clicking the **Check Selected** buttons below the lists selects the check boxes of the boundaries highlighted in the list. This is equivalent to selecting the individual check boxes and is a quick way to select multiple check boxes. Clicking the **Clear Selected** button similarly clears the check boxes of the selected domains.

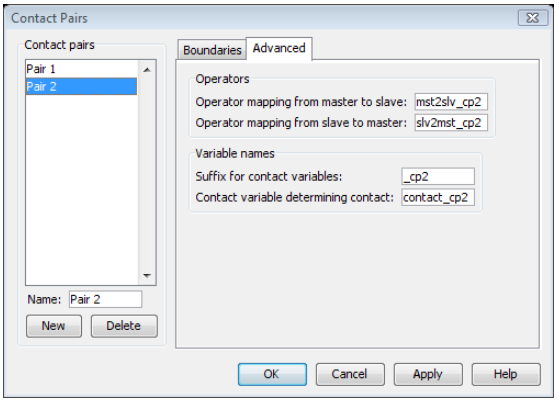
Use the buttons **Select Master** and **Select Slave** to select the master and slave domains in the main window and in the selection lists.

Clicking the arrow button between the selection lists interchanges the master and slave domains.

For best results you should consider the guidelines on page 118 when selecting your master and slave boundaries.

Advanced Page

On the **Advanced** page in the **Contact Pairs** dialog box you can define the names of the contact pair's coupling operators.



When creating a contact pair COMSOL Multiphysics automatically defines the operators and gives them a name. The names have to be unique within the whole model. The application modes use the operators to set up the contact condition preventing the slave from penetrating the master.

A *map operator* evaluates its argument on one side of the pair and makes the result available on the other side. In the previous figure you can see two operators: `mst2slv_cp2`, mapping from the master of the pair to the slave, and `slv2mst_cp2`, mapping in the other direction. For example, if `u` is a variable on the master side you can use the expression `mst2dst_cp2(u)` on the slave side.

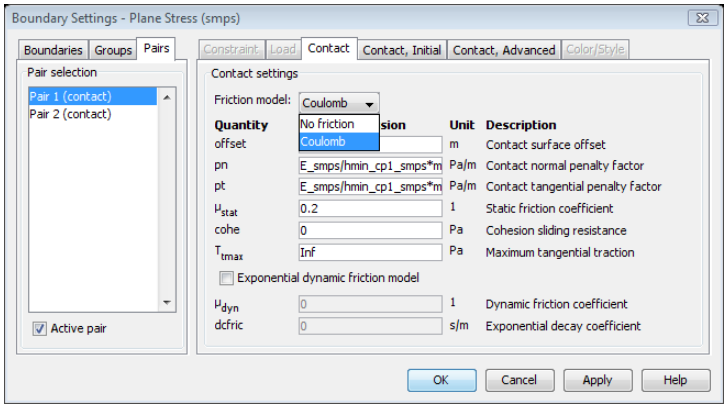
The application modes create a number of variables on the boundaries of the contact pair. To make the variable names unique the software adds a suffix to them. You can edit the suffix name in the **Suffix for contact variables** edit field. A special variable is the contact variable stating if the boundaries are in contact or not, which you can use in logical expressions. The expression `if(contact_cp1,1,2)` on the slave side evaluates to 1 for points where the boundaries are in contact and to 2 in the other points. You can edit the name of the contact variable in the **Contact variable determining contact** edit field.

BOUNDARY SETTINGS DIALOG BOX

The **Boundary Settings** dialog box, which you can open from the **Physics** menu, has three pages dedicated for contact settings: the **Contact**; **Contact, Initial**; and **Contact, Advanced** pages. In the following you can find a description of these.

Contact Page

You specify the most important settings for your contact problem on the **Contact** page.



The Contact page for the Plane Stress application mode.

The **Active pair** check box lets you select if you want to use the contact pair in this application mode or not. If you want to model friction between the contact pairs, select **Coulomb** from the **Friction model** list.

The following table specifies the contact pair parameters on the **Contact** page:

PARAMETER	DESCRIPTION	SI UNIT	NO FRICTION	COULOMB
offset	An optional offset specifying at what distance from the geometrical boundary contact appears, positive in the normal direction	m	√	√
pn	The normal penalty factor	Pa/m	√	√
pt	The tangential penalty factor	Pa/m		√
μ_{stat}	Static friction coefficient	-		√
cohe	Cohesion sliding resistance, the friction force at zero contact pressure	Pa		√
T_{tmax}	The maximum tangential traction	Pa		√

PARAMETER	DESCRIPTION	SI UNIT	NO FRICTION	COULOMB
μ_{dyn}	Dynamic friction coefficient, only used with the dynamic friction option	-		√
dcfric	Decay coefficient, only used with the dynamic friction option	s/m		√

The convergence is sensitive to the value of the penalty factors. Their value should be of the same order as the stiffness of the boundary divided by a typical length scale, that is, the mesh size. The default value for both the normal and tangential penalty factors is set according to

$$\mathrm{p} = \frac{E}{h_{\min}} \cdot \min(10^{-3} \cdot 5^{\mathrm{auglagiter}}, 1) \tag{4-23}$$

The Young's modulus is denoted E and the smallest mesh size on the slave boundary, $h_{\min}$, is included in order to get a typical length scale. The `auglagiter` variable is the iteration number in the augmented Lagrange solver. It is used to make the penalty parameter soft at the beginning (to help the solver get started) and to gradually make it stiffer (to speed up convergence).

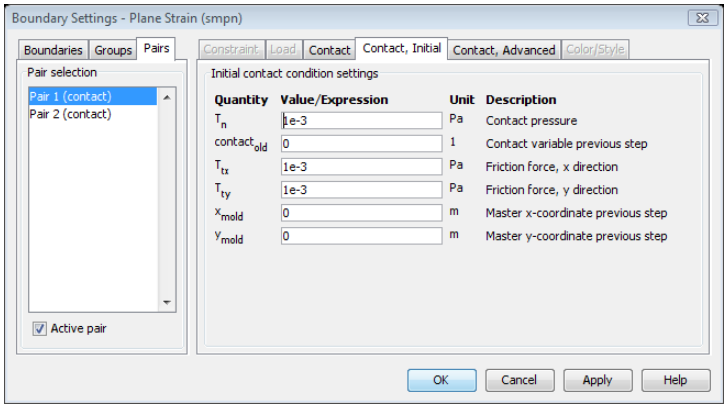
The default values, using Young's modulus only work for linear isotropic materials, for which the Young's modulus is defined. For other types of materials you need to substitute E with a suitable value or define it as a constant or expression variable. Read more about selecting the penalty factor on page 119.

If you select the **Exponential dynamic friction model** check box you get a friction coefficient that varies between the static and dynamic friction coefficient depending on the slip velocity and the dcfric decay coefficient in the following way.

$$\mu_{\mathrm{dyn}} + (\mu_{\mathrm{stat}} - \mu_{\mathrm{dyn}}) e^{-\mathrm{dcfric}|v_s|} \tag{4-24}$$

Contact, Initial Page

You specify the initial conditions for your contact problem on the **Contact, Initial** page.



The Contact, Initial page for the Plane Strain application mode.

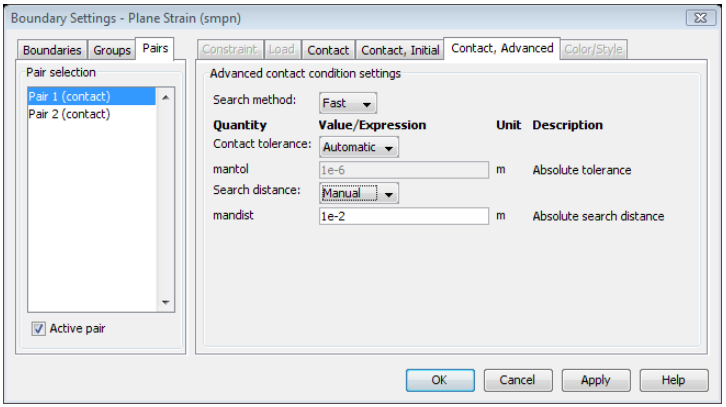
The following table specifies the contact pair parameters on the **Contact, Initial** page:

PARAMETER	DESCRIPTION	SI UNIT	NO FRICTION	COULOMB
T_n	The initial value for the contact pressure.	Pa	√	√
$contact_{old}$	The initial value for the `contact` variable in the previous step.			√
T_{txi}	The initial value for the friction force components.	Pa		√
xi_{mold}	The initial value for the coordinates of the master point in the previous step.	m		√

Turn to page 120 to read about how the initial contact pressure can influence your contact model.

Contact, Advanced Page

You have the option to specify more advanced contact pair settings on the **Contact, Advanced** page.



The Contact, Advanced page for the Plane Strain application mode.

You specify what search method to use in the **Search method** list. The default option is **Fast**. Under some rare circumstances this method can fail to detect contact and find the corresponding master points. Then select the more robust but slower option **Direct** instead.

You have two options to calculate the contact tolerance: **Automatic** or **Manual**. That is, at what distance between the two bodies they are regarded as being in contact (used for friction and multiphysics contact). You select this from the **Contact tolerance** list. **Automatic** means that the software calculates the tolerance from the size of the bounding box of the total geometry. **Manual** means that you specify the value yourself in the **mantol** edit field.

In a similar way as for the contact tolerance you have the option to specify the **Search distance**. The search distance sets the radius from any slave point within which the program looks for possible contact between the slave and master boundary. A shorter distance speeds up the search algorithm because the vast majority of boundary elements can quickly be excluded from the search process. But a too small value might result in missed contact detection.

The **Automatic** setting means that the program calculates the search distance from the size of the bounding box of the total geometry. If the total size of the geometry is not representative for the size of the contact areas, you can use the **Manual** setting and

specify the value yourself in the **mandist** edit field. A suitable search radius is usually on the order of the largest mesh elements involved in the contact process.

Fluid-Structure Interaction

The Fluid-Structure Interaction (FSI) predefined multiphysics coupling combines fluid flow with structural mechanics by using a Moving Mesh (ALE) application mode to capture the movement of the fluid domain. The structural mechanics application mode uses the large-deformation option, and the fluid flow application mode enables weak constraints that provide the fluid loads on the structure. The fluid flow application mode is defined on an ALE frame, whereas the structural mechanics application mode for the solid is defined on a reference frame. The FSI couplings appear on the boundaries between the fluid and the solid, and there are also predefined settings for the subdomain properties. These settings are grouped into easily identifiable groups that you assign to the relevant subdomains and boundaries.

Theory Background

The fluid flow is described by the Navier-Stokes equations (Equation 6-1 in the *COMSOL Multiphysics Modeling Guide*), which provide a solution for the velocity field $\mathbf{u}$. The total force exerted on the solid boundary by the fluid is the negative of the reaction force on the fluid,

$$\mathbf{f} = -\mathbf{n} \cdot (p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)) \tag{4-25}$$

where p denotes pressure, η 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) coordinate system while the structural mechanics application modes are defined in the reference (undeformed) coordinate system, a transformation of the force is necessary. This is done according to

$$\mathbf{F} = \mathbf{f} \cdot \frac{dv}{dV} \tag{4-26}$$

where dv and dV are the mesh element scale factors for the spatial frame and the reference frame, respectively.

The FSI predefined multiphysics coupling computes the reaction force on the fluid by turning on the weak constraints option for the fluid application mode, which adds Lagrange multipliers as additional dependent variables. Due to the derivatives present in the boundary condition for the velocity field, non-ideal weak constraints are used.

You can read about weak constraints in the section "Using Weak Constraints" on page 300 of the *COMSOL Multiphysics Modeling Guide*.

Application Mode Description

A combination of the following three application modes make up the FSI predefined multiphysics coupling:

- A continuum application mode:
  - Plane Strain in 2D
  - Axial Symmetry, Stress-Strain in 2D axisymmetry
  - Solid, Stress-Strain in 3D
- Moving Mesh (ALE) from COMSOL Multiphysics
- The General Laminar Flow application mode, with non-ideal constraints and the settings for incompressible Navier-Stokes (the application mode's name with this setting is Incompressible Navier-Stokes)

This section describes settings specific to the FSI predefined coupling.

PROPERTIES

The FSI predefined multiphysics coupling change some of the application mode properties from their default settings according to the following table:

APPLICATION MODE	PROPERTY	SETTING
Continuum application modes	**Large deformation**	**On**
Moving Mesh (ALE)	**Smoothing method**	**Winslow**
	Weak constraint	**Off**
Incompressible Navier-Stokes	**Constraint type**	**Non-ideal**

ANALYSIS TYPE

There are two available analysis types in the Model Navigator, a static analysis that uses the stationary solver and a transient analysis that uses the time-dependent solver.

SUBDOMAIN SETTINGS

From within the **Subdomain Settings** dialog box for each application mode, you can assign a group of settings to each subdomain by selecting it from the **Group** list. The following groups are available:

- **Fluid domain**. This group contains subdomain settings for the fluid domain.
  - In the structural mechanics application mode, this group makes this application mode inactive.
  - In the Incompressible Navier-Stokes application mode, this group uses the default properties for the fluid. Change these properties to match the fluid in your model.
  - In the Moving Mesh (ALE) application mode, this group defines free mesh displacement
- **Solid domain**. This group contains subdomain settings for the solid domain.
  - In the structural mechanics application mode, this group uses the default properties for the solid. Change these properties to match the solid in your model.
  - In the fluid flow application mode, this group makes this application mode inactive.
  - In the Moving Mesh (ALE) application mode, this group defines physics-induced mesh displacement using the displacements from the structural mechanics application mode. Note that in 2D axisymmetry, these displacements are defined as prescribed mesh displacements because the dependent variables in the Axial Symmetry, Stress-Strain application mode differ from the actual displacements, which instead are available as variables. The following table shows the applied settings for the different space dimensions.

SPACE DIMENSION	SELECTION	EDIT FIELD	EXPRESSION
2D	**Physics induced displacement**	**dx**	u
		dy	v
3D	**Physics induced displacement**	**dx**	u
		dy	v
		dz	w
2D axial symmetry	**Prescribed displacement**	**dr**	uaxi_smaxi
		dz	w

BOUNDARY SETTINGS

You can apply predefined boundary settings by selecting a group from the **Group** list in the **Boundary Settings** dialog box for the application modes. Each of the following groups is available only in one of the application modes:

- **Fluid load**. This group, found in the structural mechanics application mode, defines the fluid load on the structure using the variable for the total force per area times a factor for the area effect, for example, `T_x_mmglf*dvol_ale/dvol`. The expression includes a factor for the area effect because the total force variable comes from the deformed mesh, whereas the forces in the structural mechanics application mode must be based on the undeformed area. This factor is the mesh element scale factor for the ALE frame divided by the mesh element scale factor for the reference frame. Also, for axisymmetric models, an additional factor `(R+uaxi)/R` takes the radial displacement into account.
- **Structural velocity**. This group, found in the fluid flow application mode, is only applicable for transient analysis, where the time derivatives of the structural displacements define the fluid's velocity. **Moving leaking wall** is set as **Boundary condition** with components according to the following table.

SPACE DIMENSION	EDIT FIELD	EXPRESSION
2D	u_w	`ut`
	v_w	`vt`
3D	u_w	`ut`
	v_w	`vt`
	w_w	`wt`
2D axial symmetry	u_w	`uaxi_t_smaxi`
	v_w	`wt`

- **Structural displacement**. Use this setting in the Moving Mesh (ALE) application mode at the boundaries of the solid domain. The settings define the mesh displacements as the structural displacements, according to the table below.

SPACE DIMENSION	EDIT FIELD	EXPRESSION
2D	dx	u
	dy	v
3D	dx	u
	dy	v
	dz	w
2D axial symmetry	dr	uaxi_smaxi
	dz	w

- **Fixed**. This group, found in the Moving Mesh (ALE) application mode, defines the mesh displacements to be zero. Use this setting at the exterior boundaries of the fluid domain.

In addition to the above predefined settings, you typically define standard boundary conditions such an inflow velocities, slip, and no-slip conditions in the fluid flow application mode and one or several fixed boundaries in the structural mechanics application mode.

Example Model

"ALE Fluid-Structure Interaction" on page 294 in the *MEMS Module Model Library* shows a transient 2D FSI simulation.

Thermal-Structure Interaction

The Thermal-Structure Interaction predefined multiphysics coupling combines a continuum application mode from the MEMS Module with a heat transfer application mode from the Heat Transfer Module or COMSOL Multiphysics. The coupling appears on the subdomain level, where the temperature from the heat transfer application mode acts as a thermal load for the structural mechanics application mode.

Theory Background

Read about constitutive equations including thermal expansion in the section dealing with the theory background for the continuum application modes, on page 164 of this manual.

Application Mode Description

A combination of the following two application modes make up the Thermal-Structure Interaction predefined coupling:

- A continuum application mode from the MEMS Module:
  - Plane Strain or Plane Stress in 2D
  - Axial Symmetry, Stress-Strain in 2D axisymmetry
  - Solid, Stress-Strain in 3D
- The General Heat Transfer application mode from the Heat Transfer Module, if your license includes that module, or the Heat Transfer, Conduction application mode from COMSOL Multiphysics

This section describes settings specific to the Thermal-Structure Interaction predefined multiphysics coupling.

ANALYSIS TYPE

There are three available analysis types in the Model Navigator: a static analysis type, which uses the stationary solver, and the transient and quasi-static analysis types, which use the time-dependent solver.

SUBDOMAIN SETTINGS

Both application modes of this predefined multiphysics coupling are active on all subdomains of the model. Thermal expansion is enabled for all subdomains in the

structural mechanics application mode. In the graphical user interface, you can find this on the **Load** page of the **Subdomain Settings** dialog box for the structural mechanics application mode, where the predefined coupling automatically selects the **Include thermal expansion** check box. On the same page, the expression in the **Temp** edit field is the dependent variable for temperature from the heat transfer application mode, typically T.

BOUNDARY SETTINGS

The Thermal-Structure Interaction predefined coupling does not define any coupled constraints or loads on the boundaries. You can set those individually for the structural and thermal analyses.

Example Model

See the model "Thermal Expansion in a MEMS Device Using the Material Library" on page 227 of the *MEMS Module Model Library* for an example model that uses the Thermal-Structure Interaction predefined multiphysics coupling.

Contact Modeling

In the MEMS Module you can create models involving contact, with or without friction, between parts. Contact is implemented based on the *augmented Lagrangian* method, which is described on page 186. When modeling contact between structural parts you need to set up *contact pairs*, which define where the parts may come into contact. A contact pair consists of two sets of boundaries, which are the *master domains* and the *slave domains*. The 2D and 3D structural continuum application modes use the pairs to set up equations that prevent the *slave boundaries* to penetrate the *master boundaries*. The present section provides some advice regarding important aspects of creating contact models. You can find tips regarding solver settings for contact models in the section "Solver Settings for Contact Modeling" on page 155.

On page 217 you can read about how to specify contact pairs and define the physics for these in the graphical user interface.

When creating contact models it can often be to advantage to set up a prototype in 2D before attempting a 3D model. Similarly it is often good to start using linear elements to ease convergence toward a solution. When you have got this working, you can switch to quadratic elements if you want to.

Constraints

Make sure that the bodies are sufficiently constrained, also in the initial position. If the bodies are not in contact in the initial configuration, and there are no constraints on the bodies, you have an underconstrained state. This causes the solver to fail. One way to fix this problem is to set initial values for the displacement variables so that you have a small penetration in the initial configuration. Another way is to use a displacement-controlled model rather than a force-controlled one.

Contact Pairs

For efficiency, only include those boundaries that may actually come in contact in the slave. For the master, it is often a bit more efficient to make it so large that every slave point "has" a corresponding master point. Note that the corresponding master point is obtained by following the normal to the slave until it reaches the master.

To decide which boundaries should be assigned as master and slave in a contact pair consider the following guidelines:

- Make sure that the master boundary stiffness in the normal direction is higher than the slave boundary stiffness. This is especially important if the difference in stiffness is quite large, for example, over ten times larger. Keep also in mind that for elasto-plastic or hyperelastic materials there can be a significant change in stiffness during the solution process, and choose the master and slave boundaries accordingly. For such materials you might have to also adjust the penalty factor as the solution progresses.
- When the contacting parts have approximately the same stiffness, you can instead consider the geometry of the boundaries. The master should be concave and the slave convex rather than the opposite.

Once you have chosen the master and slave boundaries you should mesh the slave finer than the master. Do not make the slave mesh just barely finer than the master because this often causes unphysical oscillations in the contact pressure. Make the slave at least two times finer than the master.

Boundary Settings for Contact Pairs

PENALTY FACTORS

Note that 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 parameters. If the iteration process fails in some of the first augmented iterations, lower your penalty parameters. If the model seems to converge but very slowly, consider increasing the maximum value of your penalty parameters.

Increasing the penalty factor can lead to an ill-conditioned Jacobian matrix and convergence problems in the Newton iterations. You can often see this by noting that the damping factor becomes less than 1 for many Newton iterations. If this occurs, decrease the penalty factors.

The default values for the penalty factors, using Young's modulus, only work for linear isotropic materials, for which the Young's modulus is defined. For other types of materials you need to substitute $\boldsymbol{E}$ with a suitable value or define it as a constant or expression variable. For elasto-plastic materials you may find that the default value works fine until there is a significant decrease in stiffness due to plastic deformation. This can give rise to convergence problems for the nonlinear solver, since the penalty

factor becomes too large. To aid convergence you can specify an expression for the stiffness that depends for example on the solver parameter.

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.

Multiphysics Contact

Multiphysics contact problems are often very ill-conditioned, which leads to convergence problems for the nonlinear solver. For example, take heat transfer through the contact area, where initially only one point is in contact. The solution for the temperature is extremely sensitive to the size of the contact area (that is, the problem to determine the temperature is ill-conditioned). Therefore it is important to resolve the size of the contact area accurately, that is, to use a very fine mesh in the contact area. If the contact area is larger, you do not need as fine mesh 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.

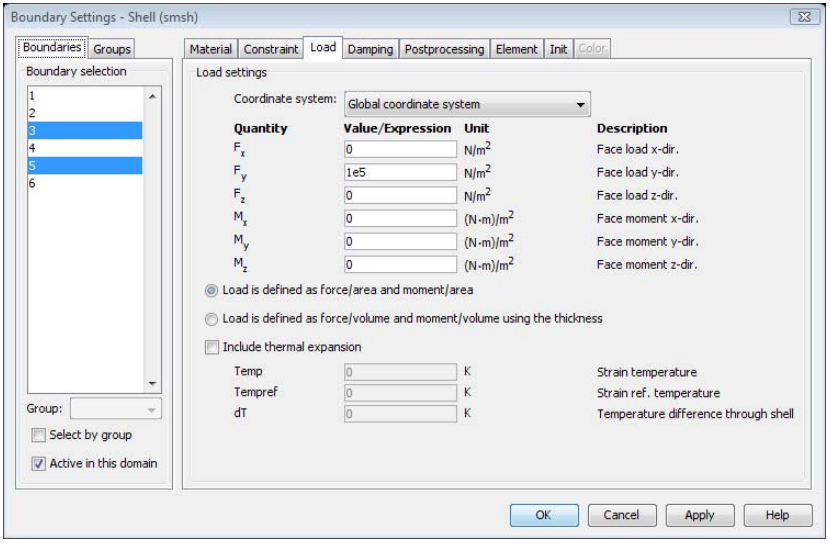
Coordinate Systems

It can be very useful to use different coordinate systems in situations such as when specifying loads, constraints, and anisotropic materials, and when postprocessing the results. The MEMS Module provides a number of different coordinate systems:

- A global coordinate system, where the geometry is created in 3D (x, y, z).
- A local geometrical coordinate system, on 2D boundaries (t, n) and on 3D faces (t_1, t_2, n).
- Application-mode specific coordinate systems: a shell coordinate system and a 3D Euler beam coordinate system.
- A user-defined coordinate system.

To specify the coordinate system, select it from the **Coordinate system** list on the **Constraint**, **Load**, and **Material** pages.

The following figure shows the **Load** page in the **Boundary settings** dialog box for the Shell application mode.



Global Coordinate System

You can use the global coordinate system in all application modes to specify loads and constraints on all domain levels: points, edges, faces, and subdomains. It is the default setting for loads and constraints in all application modes on all domain levels except boundary constraints for the Mindlin plate application mode. The default name of the space coordinates are the following for the different geometries:

GEOMETRY	DEFAULT NAME OF SPACE COORDINATES
2D	$x\ y\ z$
3D	$x\ y\ z$
Axial symmetry 2D	$r\ \varphi\ z$

The name of the space coordinates can be changed when creating a geometry from the Model Navigator, see "Creating Cartesian and Cylindrical Coordinate Systems" on page 27 in *COMSOL Multiphysics User's Guide* for details.

Local Geometrical Coordinate System

Boundaries in 2D and 3D have geometric variables describing the parametrization of the geometry defined on them. These variables define directions that define a local coordinate system that can be used when specifying loads and constraints.

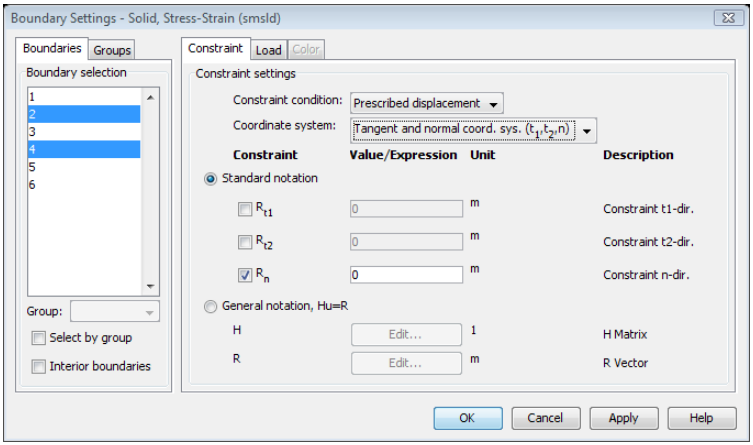
In 2D the local geometrical coordinate system is (t, n), representing the tangential and normal direction of the boundary. For interior boundaries and free edges this coordinate system is right-oriented. For exterior boundaries the normal is always directed out from the domain.

In 3D the local geometrical coordinate system is (t_1, t_2, n) representing two tangential directions and one normal direction. t_1 and t_2 depend on the parametrization of the geometry. For interior boundaries and free faces this coordinate system is right-oriented but not always orthogonal. For exterior boundaries the normal is always directed out from the domain. Common applications for this coordinate system include specifying pressure or normal displacement on a surface.

Note: t_1 and t_2 depend on how the geometry was created and are usually perpendicular to each other.

Read more about this topic in "Geometric Variables" on page 165 in the *COMSOL Multiphysics User's Guide*.

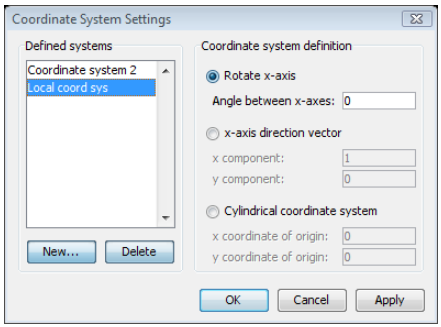
The **Constraint** page in the **Boundary settings** dialog box for the Solid, Stress-Strain application mode shows how local coordinate systems work.



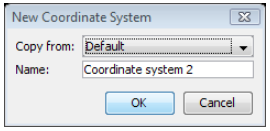
User-Defined Coordinate System

User-defined coordinate systems can be used on all domain levels in all application modes. For the continuum application modes, they can define orthotropic and anisotropic material properties in a coordinate system other than the global Cartesian system.

Create a user-defined coordinate system by going to the **Options** menu and opening the **Coordinate systems settings** dialog box. Depending on the active geometry, the software creates a 2D or 3D coordinate system.



The **New** button opens the **New coordinate system** dialog box.

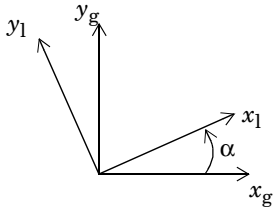


In the **Copy from** list you select from which existing coordinate system you want to copy the coordinate-system settings.

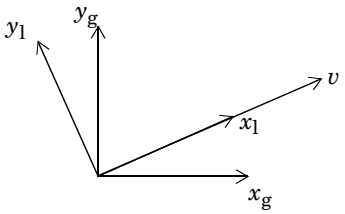
In the **Name** edit field you enter the name of the coordinate system, and it is the name that appears in all coordinate-system lists.

The software creates a coordinate system in one of three ways, which you control with option buttons:

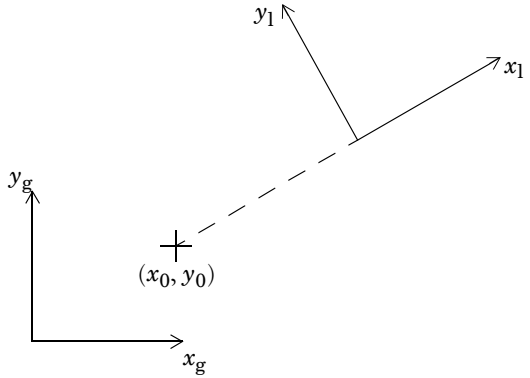
- **Rotate x-axis** The local x_1-axis direction is specified using an angle (α) between the global and local x-axes.



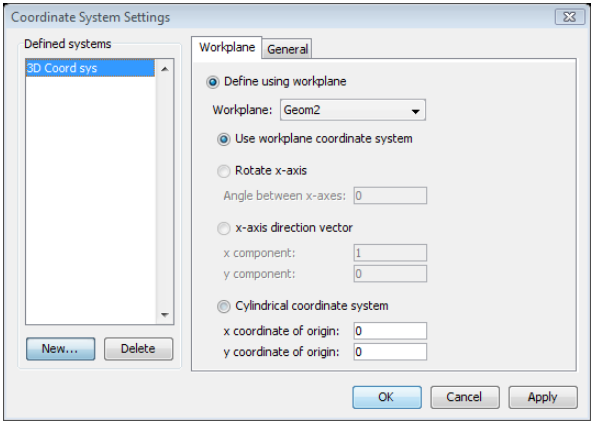
- **x-axis direction vector**: The local x_1-axis direction is specified using a direction vector v.



- **Cylindrical coordinate system** A local cylindrical coordinate system (x_1, y_1) with origin at (x_0, y_0) is specified.



3D GEOMETRY



The **New** button works in the same way as for the 2D geometry case.

The software defines the coordinate system in two ways, which you control with the **Define using work plane** and **Define using global coordinates** option buttons.

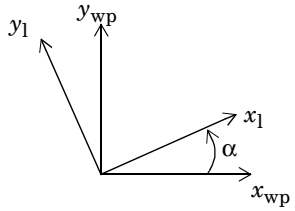
Define Using Work Plane

Define using work plane is enabled when a least one work plane/2D geometry exists.

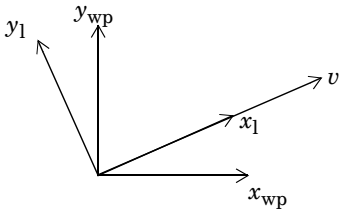
Which work plane the local coordinate system is based upon is controlled from the **Work plane** list.

Four options are available, which you control with option buttons:

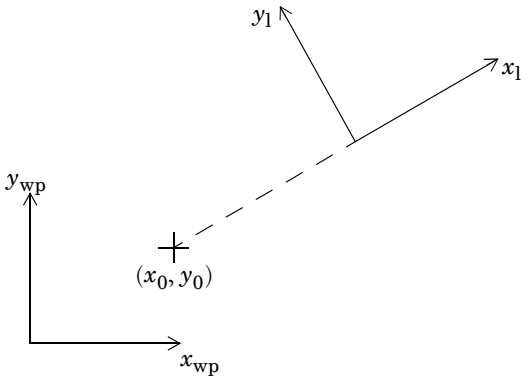
- **Use work plane coordinate system** The local coordinate system is the same as the work plane. You control the definition of the work plane by going to the **Draw** menu and opening the **Work-Plane Settings** dialog box. Get details about the creation of work planes in "Creating and Using 2D Work Planes" on page 59 in the *COMSOL Multiphysics User's Guide*.
- **Rotate x-axis** The local x_l-axis direction is specified using an angle (α) between the work planes x_{wp}-axis and the local x_l-axis.



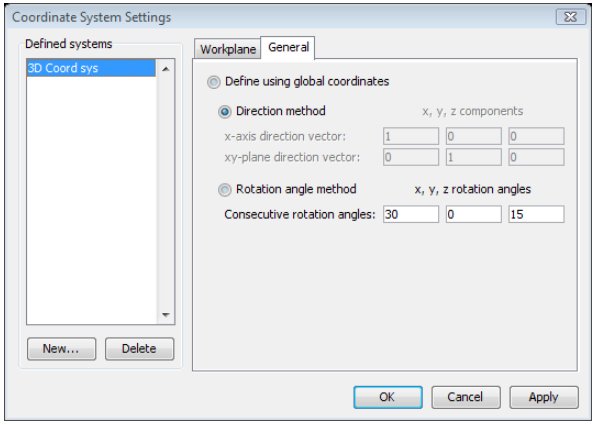
- **x-axis direction vector** The local x_l-axis direction is specified using a direction vector v.



- **Cylindrical coordinate system** A local cylindrical coordinate system (x_l, y_l) with origin at (x_0, y_0) in the work plane coordinates is specified.

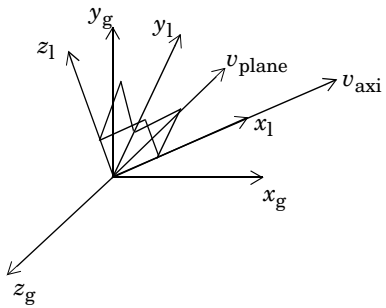


Define Using Global Coordinates

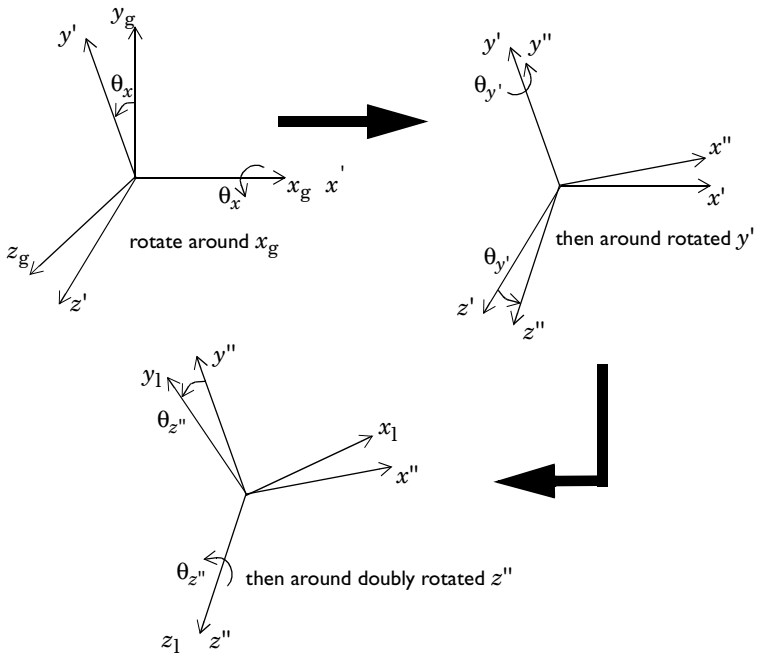


There are two different options available, which you control with option buttons:

- **Direction method** The local x_l-axis direction is specified using a direction vector v_{axi}. The local $x_l y_l$-plane is specified using a direction vector v_{plane} which is a vector lying in the local $x_l y_l$-plane.



- **Rotation angle method** The local coordinate system (x_1, y_1, z_1) is specified using three consecutive rotation angles θ_x, $\theta_{y'}$, $\theta_{z''}$.



Damping

Damping is important in time-dependent and frequency response analysis. This section describes how to model it in the MEMS Module using different damping models.

Rayleigh Damping

A common model for viscous damping is *Rayleigh damping*, where the damping is assumed to be proportional to a linear combination of the stiffness and mass. To illustrate this, consider a system with a single degree of freedom. The following equation of motion describes the dynamics of such a system with viscous damping:

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

In the Rayleigh damping model, the damping parameter c is expressed in terms of the mass m and the stiffness k as

$$c = \alpha_{dM}m + \beta_{dK}k$$

where α_{dM} and β_{dK} are the mass and stiffness damping parameters, respectively.

The problem with the Rayleigh damping model is getting good values for the damping parameters. A much more physical damping measure is the damping ratio—the ratio between actual and critical damping, often expressed as a damping factor in percentage of the critical damping. You can find commonly used values of damping factors in the literature.

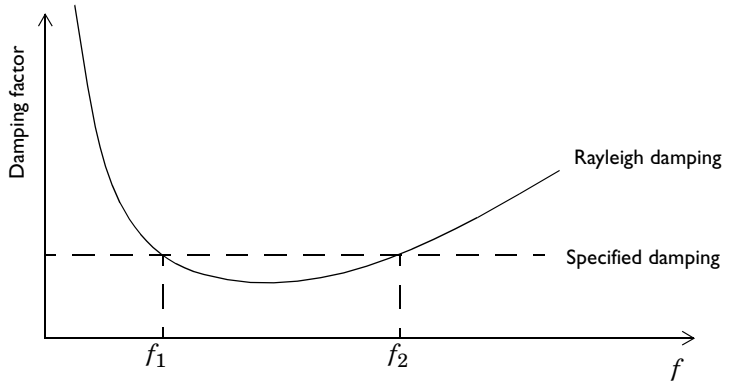
It is possible to transform damping factors to Rayleigh damping parameters. For a specified damping factor ξ at a frequency f

$$\xi = \frac{1}{2}\left(\frac{\alpha_{dM}}{2\pi f} + \beta_{dK}2\pi f\right)\,.$$

Using this relationship at two frequencies f_1 and f_2 with different damping factors ξ_1 and ξ_2 results in an equation system

$$\begin{bmatrix} \frac{1}{4\pi f_1} & \pi f_1 \\ \frac{1}{4\pi f_2} & \pi f_2 \end{bmatrix} \begin{bmatrix} \alpha_{dM} \\ \beta_{dK} \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}.$$

Using the same damping factors ξ_1 and ξ_2 does not result in the same damping factors in the interval. It can be shown that the damping parameters have the same damping at the two frequencies and less damping in between, as you can see in the following figure.



Note: All Structural Mechanics application modes in the MEMS Module use nonzero default values for α_{dM} and β_{dK}. You must change these default values to meet the specific modeling situation.

Loss Factor Damping

Loss factor damping (sometimes referred to as material or structural damping) takes place when viscoelastic materials are modeled in the frequency domain. The complex modulus $G^*(\omega)$ is the frequency-domain representation of the stress relaxation function of viscoelastic material. It is defined as

$$G^* = G' + jG'' = (1 + j\eta)G'$$

where G' is the storage modulus, G'' is the loss modulus, and their ratio $\eta = G''/G'$ 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 can all be frequency dependent.

In COMSOL Multiphysics, the loss information appears as a multiplier of the total strain in the stress-strain relationship:

$$\sigma = D((1 + j\eta)\varepsilon - \varepsilon_{\text{th}} - \varepsilon_0) + \sigma_0 .$$

For hyperelastic material, the loss information appears as a multiplier in the first Piola-Kirchhoff stress, P:

$$P = (1 + j\eta)\frac{\partial W_{\text{hyp}}}{\partial \nabla \mathbf{u}}$$

The loss factor damping is available for frequency response analysis in all application modes, but it is not defined for elasto-plastic materials.

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 η as the damping parameter, but its implementation is different from the actual loss factor damping.

The piezoelectric application modes have built-in support for this type of damping. For the other application modes, you can model it using the stiffness damping parameter β_{dK}, defined as the loss factor, η, divided by the excitation frequency:

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

You must also set the mass damping factor, α_{dM}, to zero.

Explicit Damping

Another way to model damping is to specify it explicitly as a viscous force. In a transient analysis you do so by specifying a force that depends on the velocities with opposite signs:

$$\mathbf{F} = -c \cdot \mathbf{v}$$

where $\mathbf{v}$ is the velocity vector

$$\mathbf{v} = \begin{bmatrix} u_t \\ v_t \\ w_t \end{bmatrix}$$

and u_t is the velocity component in the x direction, typically named `ut`.

By specifying damping locally, it is possible to define explicit damping on all domain levels.

In a frequency response analysis you define explicit damping in a similar way, but the name of the velocity variable changes and includes the application mode name, for example, `u_t_smsld` for the Solid, Stress-Strain application mode.

No Damping

To create an undamped model, you can also select to use **No damping** from the **Damping model** list.

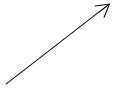
Symbols for Loads and Constraints

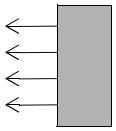
To make it easier to specify a model, you can choose to display load and constraint symbols on a geometry. This is done automatically, but in some situations it might take too long, so the software provides an option to deactivate the automatic update of symbols. This option appears on the **Visualization** page in the **Preferences** dialog box. To read more about that dialog box see the section "Saving Preferences for Labels, Rendering, and Highlighting" on page 119 in the *COMSOL Multiphysics User's Guide*. In the **Preferences** dialog box you also have the option to select whether the symbols from the current domain type or all domain types should be plotted. A manual update of symbols is possible from the **Options** menu by selecting **Update Symbols** or by clicking the **Update Symbols** button on the Visualization/Selection toolbar. Scaling the size of the symbols is possible in the **Visualization/Selection** dialog box; see "Scaling of Load and Constraint Symbols" on page 119 in the *COMSOL Multiphysics User's Guide*.

Load Symbols

You can plot load symbols on points, boundaries, edges, and subdomains. The loads are normalized with respect to the maximum value within a domain type.

The following table lists all load symbols together with the application modes where they appear.

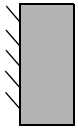
LOAD SYMBOL	DESCRIPTION	APPLICATION MODES
	Force at a point or in a subdomain.	Plane Stress, Piezo Plane Stress, Plane Strain, Piezo Plane Strain, Axial Symmetry, Stress-Strain, Piezo Axial Symmetry

LOAD SYMBOL	DESCRIPTION	APPLICATION MODES
	Boundary force.	Plane Stress, Piezo Plane Stress, Plane Strain, Piezo Plane Strain, Axial Symmetry, Stress-Strain, Piezo Axial Symmetry
	Force in the direction indicated by the direction of the arrow.	Solid, Stress-Strain, Piezo Solid

Constraint Symbols

Constraint symbols can be plotted on points, boundaries, edges, and subdomains. The following table lists all constraint symbols together with the application modes where they appear.

CONSTRAINTS SYMBOL	DESCRIPTION	APPLICATION MODE
	Displacement constrained in the direction indicated by the roller.	Plane Stress, Piezo Plane Stress, Plane Strain, Piezo Plane Strain, Axial Symmetry, Stress-Strain, Piezo Axial Symmetry
	All degrees of freedom constrained.	Plane Stress, Piezo Plane Stress, Plane Strain, Piezo Plane Strain, Axial Symmetry, Stress-Strain, Piezo Axial Symmetry

CONSTRAINTS SYMBOL	DESCRIPTION	APPLICATION MODE
	Clamped edge, all degrees of freedom constrained.	Plane Stress, Piezo Plane Stress, Plane Strain, Piezo Plane Strain, Axial Symmetry, Stress-Strain, Piezo Axial Symmetry
	Displacements constrained in the directions indicated by the arrows.	Solid, Stress-Strain, Piezo Solid

Solver Settings

A large number of possible solver settings are available in COMSOL Multiphysics. To make it easier for you to select a solver and its associated solver parameters, the various application modes use different default settings depending on the analysis type. In some situations you must change the default settings. This section helps you select a solver and its solver settings to solve structural mechanics and multiphysics problems. Further details about all solver settings appear in the chapter "Solving the Model" on page 359 in the *COMSOL Multiphysics User's Guide*.

Symmetric Matrices

The **Matrix symmetry** list appears on the **General** page in the **Solver Parameters** dialog box. Here you specify if the assembled matrices (stiffness matrix, mass matrix) resulting from your equations are symmetric or not.

Normally the matrices from a single-physics structural mechanics problem are symmetric, but there are exceptions:

- Multiphysics models solving for several physics simultaneously, for example, heat transfer and structural mechanics. Solving for several structural mechanics application modes, such as shells combined with beams, does not create unsymmetric matrices.
- Elasto-plastic analysis.

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. The default solver in 2D, UMFPACK, does not support symmetry—but it is faster than SPOOLES, the default solver in 3D. SPOOLES uses less memory, but memory is usually not a major issue in 2D.

Note: Selecting the **Symmetric** option for a model with unsymmetric matrices produces incorrect results.

Complex matrices can be unsymmetric, symmetric, or Hermitian. Hermitian matrices do not appear in structural mechanics problems.

Note: Selecting the **Hermitian** option for a model with complex-valued symmetric matrices produces incorrect results.

Selecting Iterative Solvers

The **Linear system solver** list appears on the **General** page in the **Solver Parameters** dialog box. The default solver is **Direct (SPOOLES)** in 3D and **Direct (UMFPACK)** in 2D. For large 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 have more options than the direct solvers. The following table makes suggestions on which iterative solver and preconditioner to use for different analyses for large problems.

ANALYSIS	LINEAR SYSTEM SOLVER	PRECONDITIONER
Static analysis, single physics	Conjugate gradients	Geometric multigrid
Quasi-static transient analysis, single physics	Conjugate gradients	Geometric multigrid
Parametric analysis, single physics	Conjugate gradients	Geometric multigrid
Eigenfrequency analysis, single physics	Conjugate gradients	Geometric multigrid
Static analysis, multiphysics	GMRES	Geometric multigrid
Eigenfrequency analysis, multiphysics	GMRES	Geometric multigrid
Frequency response analysis	GMRES	Geometric multigrid
Elasto-plastic analysis	GMRES	Geometric multigrid
Time-dependent analysis	Conjugate gradients	Geometric multigrid

Specifying a positive shift greater than the lowest eigenfrequency results in indefinite matrices. The conjugate gradients iterative solver does not work for indefinite matrices. Get more details about solver settings in Chapter 6, “Selecting a Solver,” in the *COMSOL Multiphysics User’s Guide*.

Note: Check the mesh quality when using the geometric multigrid (GMG) preconditioner. It does not work well when using the option to scale the geometry before meshing (on the **Advanced** tab in the **Free Mesh Parameters** dialog box). When using extruded meshes, you might need to create the mesh cases manually.

The conjugate gradients solver does not work together with a mixed formulation because it results in an indefinite stiffness matrix. For this type of problems the following solver combinations work:

LINEAR SYSTEM SOLVER	PRECONDITIONER	SMOOTHER
GMRES	GMG	Vanka
GMG	-	Vanka
GMRES	Incomplete LU	-

When using the Vanka smoother for a mixed-formulation problem, specify the pressure as the Vanka variable. Get more information about using the Vanka smoother in the section "The Vanka Algorithm" on page 531 in the *COMSOL Multiphysics Reference Guide*.

Specifying the Absolute Tolerance

The absolute-tolerance parameters used for time-dependent problems are very problem specific. As a rule of thumb, set the absolute tolerance to be at least one order of magnitude smaller than the typical displacement.

The default value is 0.001 for all solution components. When solving mixed problems with both displacements and pressure, this default results in very small tolerance conditions for the pressure. One way to help the solver is to specify individual tolerance values for all solution components. This speeds up the solution and usually does not affect the accuracy. For example, when solving a model using the 3D Solid, Stress-Strain application mode for a mixed problem with a typical displacement amplitude of 10^{-5} and an internal pressure amplitude of 10^{5}, specify `u 1e-7 v 1e-7 w 1e-7 p 1e3` in the **Absolute tolerance** edit field (that is, use space-separated pairs of variable names and the absolute tolerance for that variable).

Solver Settings for Contact Modeling

You solve contact problems using the augmented Lagrangian method. The augmented solution components are specified on the **Stationary** page in the **Solver Parameters** dialog box. The augmented solution components are the contact pressure and the friction traction components. By default the solver finds these components automatically.

If the model includes friction, some solution components from the previous solution step are needed. You specify these variables on the **Parametric** page in the **Solver Parameters** dialog box. The components are the master coordinates, the contact variable, and, if dynamic friction is modeled, the time. By default the program finds these components automatically.

MANUAL SCALING

You need to use manual scaling if the parts are not in contact initially (initial value of contact pressure is zero) or if the model includes friction. Select **Manual** from the **Type of scaling** list on the **Advanced** page in the **Solver Parameters** dialog box. In the **Manual scaling** edit field, enter the name of all the solution components together with their approximate order of magnitude. For example, solving a plane stress problem with one contact pair including friction, where the displacements in both directions are around 10^{-3}, the contact pressure is around 1000, and the friction traction components are around 100. Then enter `u 1e-3 v 1e-3 Tn_cp1_smps 1000 Ttx_cp1_smps 100 Tty_cp1_smps 100` (using space-separated pairs of variable names and scaling factors) in the **Manual scaling** edit field.

To get the list of degrees of freedom in the model, go to the **Solver Manager** dialog box and look at the **Solve For** page. For each degree of freedom, use a positive value that is of the order of the typical value of that variable. You need not specify scaling factors for the friction history variables containing `_old`, for instance `contact_cp1_old_smps`, `xm_old_cp1_smps`, `ym_old_cp1_smps`.

You can read more about how to prevent ill-conditioned matrices by scaling of variables and equations on page 497 of the *COMSOL Multiphysics Reference Guide*.

TOLERANCES

You find tolerance settings for both the augmented Lagrangian solver and the nonlinear solver on the **Stationary** page of the **Solver Parameters** dialog box.

Specify the tolerance for the augmented Lagrangian solver (`augtol`) in the **Tolerance** edit field under the **Augmented Lagrangian solver** group label. It controls the accuracy

of the so-called augmentation components T_n, T_{tx}, T_{ty} (that is, the contact pressure and the friction tractions). The accuracy in these components is the product of the manual scaling value with augtol. For example, if the manual scale for T_n is set to 10^8, the default `augtol` = 10^{-3} gives an error $10^8 * 10^{-3} = 10^5$ or about 0.1% in T_n.

The tolerance for the nonlinear solver (`ntol`) controls the accuracy of the displacement variables (and other variables in a multiphysics model). You can change its value in the **Relative Tolerance** edit field in the **Nonlinear settings** area.

Do not use a too coarse `ntol`, especially if the body is stiff, because this causes too large errors in the determination of the contact tractions, which leads to nonconvergence in the augmented Lagrangian iterations. You can estimate `ntol` by looking at the scaling of the dependent variables and the penalty factors:

$$\text{ntol} < \frac{T_{\min} \cdot \text{augtol}}{p_{\max} \cdot u_{\max}}$$

where $T_{\min}$ denotes the minimum of the contact traction scales, $p_{\max}$ the maximum penalty factor, and $u_{\max}$ the maximum of the displacement scale factors. For example, for a material with Young's modulus of 10^{11}, a minimum mesh size of 10^{-2}, and with the manual scaling set to

```
u 1e-4 v 1e-3 Tn_cp1_smps 1e8 Ttx_cp1_smps 1e6 Tty_cp1_smps 1e6
```

using the default values for the penalty factors, the nonlinear tolerance is

$$\text{ntol} < \frac{10^6 \cdot 10^{-3}}{10^{13} \cdot 10^{-3}} = 10^{-7}$$

AUGMENTED LAGRANGIAN SOLVER

You select the augmented Lagrangian solver from the **Solver** list on the **Stationary** page of the **Solver Parameters** dialog box. This solver controls the updating of the contact tractions in each augmented Lagrange iteration. Because these degrees of freedom are rather few there is no performance issue here. The default lumped solver is used for 2D problems because this gives less undershoots in the contact tractions at the ends of the segments in contact. The lumped solver is an approximation that replaces the boundary mass matrix with a lumped diagonal matrix.

In 3D, the UMFPACK solver is used as default because lumping does not work for quadratic elements.

5

Piezoelectric Application Modes

This chapter describes the application modes for modeling piezoelectric effects in the MEMS Module.

Theory Background

The Piezoelectric Effect

The piezoelectric effect manifests itself as a transfer of electric to mechanical energy and vice-versa. It is observable in many crystalline materials, while some materials such as quartz, Rochelle salt, and lead titanate zirconate ceramics display the phenomenon strongly enough for the phenomenon 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 CONVENTIONS

The documentation and the user interface use piezoelectric conventions as far as possible. These conventions differ from those used in other structural mechanics application modes. For instance, the numbering of the shear components in the stress-strain relation differs, as the following section describes. However, the names of the stress and strain components remain the same as in the other structural mechanics application modes.

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 or strain-charge form:

STRESS-CHARGE

$$\mathbf{T} = c_E\mathbf{S} - e^T\mathbf{E}$$
$$\mathbf{D} = e\mathbf{S} + \varepsilon_S\mathbf{E}$$

STRAIN-CHARGE

$$\mathbf{S} = s_E\mathbf{T} + d^T\mathbf{E}$$
$$\mathbf{D} = d\mathbf{T} + \varepsilon_T\mathbf{E}$$

The naming convention differs in piezoelectric theory compared to structural mechanics theory, but the piezoelectric application modes use the structural mechanics nomenclature. The strain is named ε instead of $\mathbf{S}$, and the stress is named σ instead of $\mathbf{T}$. This makes the names consistent with those used in the other structural mechanics application modes.

The numbering of the strain and stress components is also different in piezo and structural mechanics theory, and it is quite important to keep track of this aspect in order to give the correct material data. In structural mechanics the following is the most common numbering convention, and it is also the one used in the other structural mechanics application modes:

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \end{bmatrix}$$

In contrast, textbooks on piezoelectric effects and the IEEE standard on piezoelectric effects use the following numbering convention:

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \\ 2\varepsilon_{xy} \end{bmatrix}$$

The piezoelectric application modes employ the immediately preceding piezo numbering convention to make it easier to work with materials data and avoid mistakes.

The constitutive relation using COMSOL Multiphysics symbols for the different constitutive forms are thus:

STRESS-CHARGE

$$\sigma = c_E \varepsilon - e^T \mathbf{E}$$
$$\mathbf{D} = e\varepsilon + \varepsilon_0 \varepsilon_{rS} \mathbf{E}$$

STRAIN-CHARGE

$$\varepsilon = s_E \sigma + d^T \mathbf{E}$$
$$\mathbf{D} = d\sigma + \varepsilon_0 \varepsilon_{rT} \mathbf{E}$$

Most material data appears in the strain-charge form, and you can easily transform into the stress-charge form. COMSOL Multiphysics allows you to use both constitutive forms; simply select one, and COMSOL Multiphysics makes any necessary transformations. The following equations transform strain-charge material data to stress-charge data

$$c_E = s_E^{-1}$$
$$e = d \; s_E^{-1}$$
$$\varepsilon_S = \varepsilon_0 \varepsilon_{rT} - d \; s_E^{-1} \; d^T$$

Material Models

In addition to modeling piezoelectric materials, the Piezoelectric application mode provides different material models for easier modeling of piezo components. This means, that in the subdomain settings of the application mode, you can define the material of each domain as:

- Piezoelectric
- Decoupled, isotropic
- Decoupled, anisotropic

The Piezoelectric material operates as described in the chapter above, whereas using the two other material models, you can model structural and electrical problems or either of them independently.

The structural part of the *Decoupled, isotropic* and *Decoupled, anisotropic* material operates as the linear elastic material with small deformations as described in "Structural Mechanics Application Modes" on page 59. However, the initial stress and

strain and thermal expansion are not supported within the Piezoelectric application mode.

For the *Decoupled, isotropic* material you define the material using the Young's modulus, E, and the Poisson ratio, ν. For the *Decoupled, anisotropic* material you define the full 6-by-6 elasticity matrix D. Note here, that you define D using the standard structural mechanics ordering. Thus the ordering of the D is different from the ordering of the piezoelectric c_E matrix.

Depending on the value of the *Electrostatics formulation* property (See "Electrical Formulations" below), the electrical part of the *Decoupled, isotropic* and *Decoupled, anisotropic* material solves either the electrostatics equation:

$$-\nabla \cdot (\varepsilon_0 \varepsilon_r \nabla V) = \rho_v$$

where ε_0 is the electrical permittivity of free space, ε_r is the relative electrical permittivity, and ρ_v is the volume charge density, or the quasi-static electric currents equation:

$$-\nabla \cdot ((\sigma_e + j\omega\varepsilon_0\varepsilon_r)\nabla V) = 0$$

where σ_e is the electrical conductivity of the material (note that σ is used also for the structural stress vector).

In frequency response analysis the conductivity appears also into the electrostatics equation:

$$-\nabla \cdot \left(\left(\frac{\sigma_e}{j\omega} + \varepsilon_0\varepsilon_r\right)\nabla V\right) = \rho_v$$

and thus you can define and use conductivity of the material independently of the Electrostatics formulation property.

For a *Decoupled, isotropic* material you define ε_r and σ_e as scalars, but for a *Decoupled, anisotropic* material you define them as 3-by-3 matrices.

Electrical Formulations

The default formulation of the equations in the Piezoelectric application modes is such that the resulting equation system with piezoelectric material is symmetric. This allows reduced memory requirements with solvers that utilize symmetry information.

The drawback of this design is that by default the Piezoelectric application modes are not electrically compatible with the Electrostatics application mode found in the AC/DC Module and the MEMS Module, nor is it compatible with the Quasi-Statics -Electric, Electric currents application modes in the AC/DC Module.

The Piezoelectric application modes support an application mode property, Electrostatics formulation, which makes them compatible with the electrostatic or quasi-static application modes so that it is possible to couple them in a model. The Electrostatics formulation property has the following choices:

- Symmetric, Electrostatics: The default implementation creates a symmetric equation system, but the application mode is not compatible with the other application modes.
- Unsymmetric, Electrostatics: This implementation creates an unsymmetric equation system which is compatible with the Electrostatics application modes.
- Unsymmetric, Electric currents: This implementation creates an unsymmetric equation system which is compatible with the Quasistatics - Electric, Electric currents application modes.

At the equation level the difference between these formulation is the following. The default formulation is that the variational electrical energy is written using a positive sign:

$$\delta We = \int (\mathbf{D} \cdot \hat{\mathbf{E}}) d\Omega$$

Here $\mathbf{D}$ is the electric displacement vector, and $\hat{\mathbf{E}}$ is the test function for the Electric field. Ω is the integration domain.

On the other hand, the formulation compatible with the Electrostatics application mode uses variational electrical energy with the negative sign:

$$\delta We = -\int (\mathbf{D} \cdot \hat{\mathbf{E}}) d\Omega$$

Finally, the electric currents formulation uses the following variational electrical energy:

$$\delta We = \int (\mathbf{J} \cdot \nabla \hat{V}) d\Omega$$

where $\mathbf{J}$ is the electric current density vector, and $\nabla \hat{V}$ is the test function for the potential gradient.

The use of the Unsymmetric, electric currents formulation sets certain limitations: you cannot model any charges, and any boundary conditions that use charges or electric displacement are written in terms of electric current. Also, this formulation only appears in the frequency response analysis.

The Piezoelectric Application Modes

This section describes the interface for defining a model using the piezoelectric application modes:

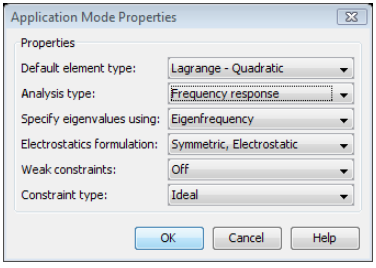
- Piezo Solid (in 3D)
- Piezo Plane Stress (in 2D)
- Piezo Plane Strain (in 2D)
- Piezo Axial Symmetry (in axisymmetric 2D)

It consists of the following sections:

- "Application Mode Properties" (the next section)
- "Scalar Variables" on page 165
- "Material Properties" on page 166
- "Electric Boundary Conditions" on page 176
- "Constraints" on page 180
- "Loads and Charges" on page 182
- "Structural Damping" on page 184

Application Mode Properties

To set or examine material properties, go to the **Physics** menu and open the **Application Mode Properties** dialog box.



Here you control various global settings for the model, which include:

- **Default element type**: A list of elements, where the selection becomes the default on all new subdomains. The default is to use second-order Lagrange elements.

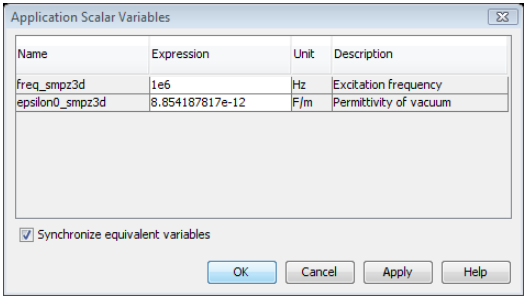
- **Analysis type**: A list of analyses to perform. It affects both the equations and which solver to use with the **Auto select solver** option in the **Solver Parameters** dialog box. The default is static analysis. You can also select transient, eigenfrequency, damped eigenfrequency, and frequency response analysis types.
- **Specify eigenvalues using**: A list controlling whether the application mode works with eigenvalues or eigenfrequencies.
- **Electrostatics formulation**: Select the electrical formulation to use:
  - **Symmetric, Electrostatic**: the default setting.
  - **Unsymmetric, Electrostatic**: for compatibility with the Electrostatics application mode.
  - **Unsymmetric, Electric currents**: for compatibility with the application modes for electric currents in the AC/DC Module (Electric Currents in 3D, In-Plane Electric Currents in 2D, and Meridional Electric Currents in 2D axial symmetry). Available for frequency response analysis.
- **Weak constraints**: Controls whether or not weak constraints are active Use weak constraints for accurate reaction-force computation. When weak constraints are enabled, all constraints are weak by default, but it is possible to change this setting for individual domains.
- **Constraint type**: Constraints can be ideal or nonideal (see "Ideal vs. Non-Ideal Constraints" on page 301 in the *COMSOL Multiphysics Modeling Guide*).

Scalar Variables

The piezoelectric application modes have the following scalar variables:.

PROPERTY	VARIABLE	DEFAULT	SI UNIT	DESCRIPTION
ε_0	`epsilon0`	`8.854187817e-12`	F/m	Permittivity of vacuum
f	`freq`	`1e6`	Hz	Excitation frequency
$j\omega$	`jomega`	`-lambda`	rad/s	Complex angular frequency

You control the scalar variables by going to the **Physics** menu and opening the **Application Scalar Variables** dialog box.



The excitation frequency (the frequency of the harmonic forces, potential, and displacement) is available only for frequency response analysis. The equations and documentation describing frequency response use the angular excitation frequency, $\omega = 2\pi f$, which is available as the variable `omega`. The complex angular frequency is available for eigenfrequency analysis and damped eigenfrequency analysis.

When you select **Frequency response** as the analysis type, the default solver is the parametric solver. This default makes it easy to perform a frequency sweep over several excitation frequencies in one analysis. In this case enter `freq` as the **Parameter name** on the **General** page in the **Solver Parameters** dialog box. The values you enter in the **Parameter values** edit field override the excitation frequency you might have entered in the **Application Scalar Variables** dialog box.

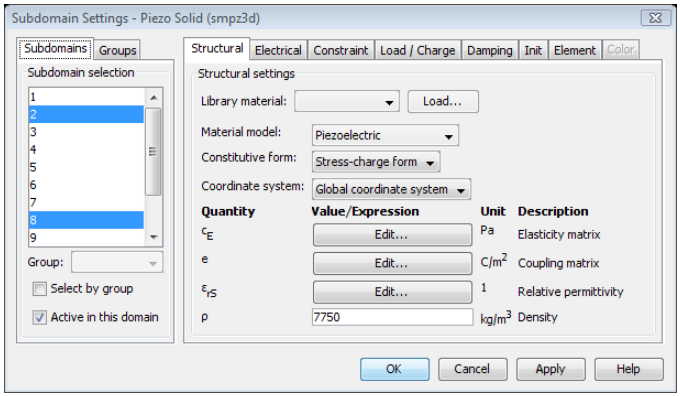
Material Properties

The **Subdomain Settings** window has two pages where you define the material properties: the **Structural** page and the **Electrical** page. On top of both pages you find the **Library material** list and the **Load** button for importing and selecting data from the material libraries and the **Material model** list for selecting the material model for each domain. These settings are shared between the pages, and if you change the **Structural** page, the settings change also on the **Electrical** page. Note that loading a material from a material library does not change the material model, so you need to change it manually in the **Material model** list to match the type of material.

Everything else you see and define on the pages depends on the material model you select. Setting for different material model are described in the following chapters.

SUBDOMAIN SETTINGS FOR PIEZOELECTRIC MATERIAL

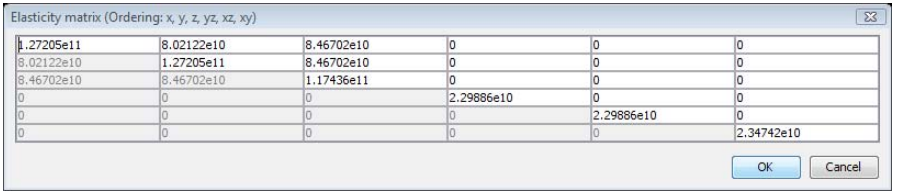
The piezoelectric material is a complete structural-electrical material, and thus you define all piezoelectric material properties on the **Structural** page.



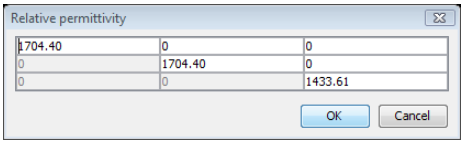
The **Structural** page has two lists in 3D, three lists in 2D, and three lists in axial symmetry:

- **Constitutive form**: Select the constitutive form from those in the following list. Depending on the selection, different material properties are shown in the dialog box.
  - **Stress-charge form**: Define the constitutive relation of the material on the stress-charge form through the c_E, e, and ε_{rS} matrices. The previous figure shows

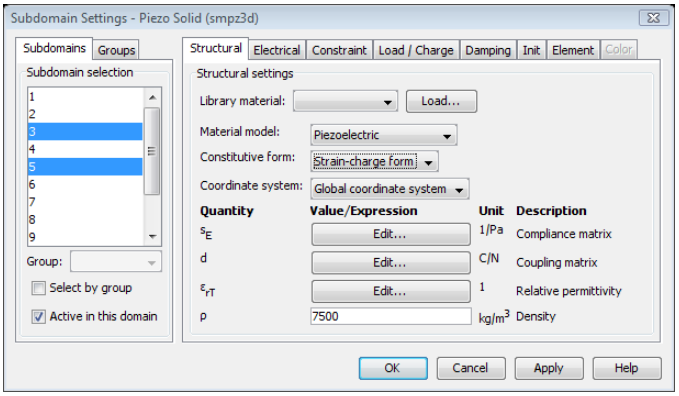
the **Material** page for stress-charge, while the following figure shows the **Elasticity matrix** dialog box for entering the c_E matrix.



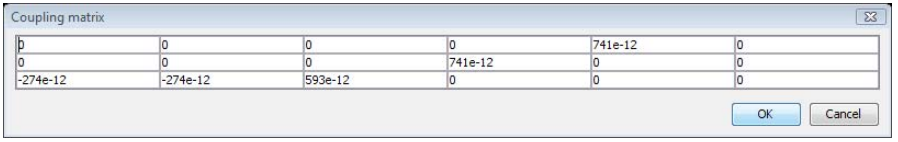
The figure below shows the **Relative permittivity** dialog box for entering the ε_{rS} matrix components.



- **Strain-charge form**: You define the constitutive relation of the material on the strain-charge form through the s_E, d, and ε_{rT} matrices (see page 160 for details). The following figure shows the **Material** page for strain-charge.



The next graphic shows the **Coupling matrix, strain-charge form** dialog box for entering the d matrix components.



- **Material orientation** (2D and axisymmetry only): Here you select how the 3D

material properties are oriented relative the 2D/axial symmetric analysis plane. There are six options: *xy*, *yz*, *zx*, *yx*, *zy*, and the default *xz*-plane. The plane represents how the 3D material is oriented relative the 2D/axial symmetric analysis plane: The first letter indicates which 3D direction coincides with the *x* direction in 2D or the *r* direction for axisymmetry; the second letter indicates which 3D direction coincides with the *y* direction in 2D or the *z* direction for axisymmetry. The material coordinates names are fixed and do not depend of the names of the space coordinates (independent variables), which have different defaults in 2D and axial symmetry.

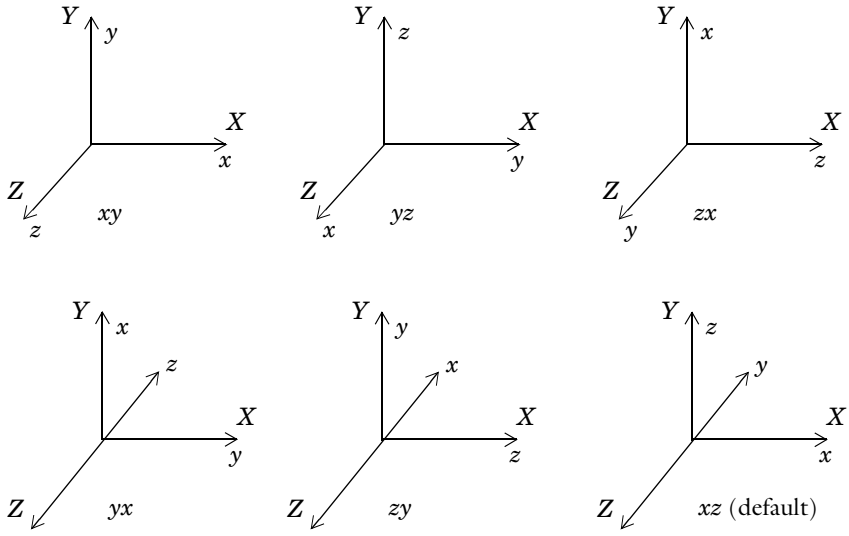


Figure 5-1:Orientation of 3D material xyz relative the 2D analysis coordinate system XYZ.

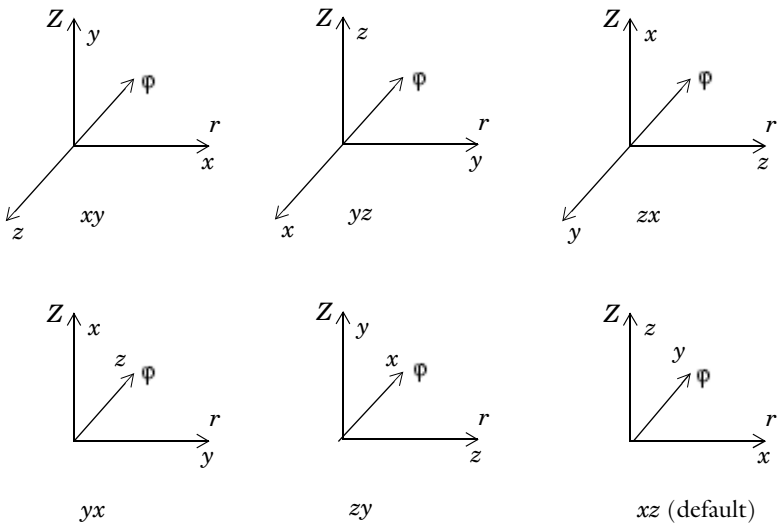


Figure 5-2: Orientation of 3D material xyz relative the axisymmetric analysis coordinate system r φ Z.

- **Coordinate system**: Select the coordinate system where the material properties are defined. This choice is useful if you want to define the material in a coordinate system other than the global system, or if you need results in a local coordinate system for postprocessing. The **Coordinate system** list contains only the global coordinate system unless you have made available a user-defined coordinate system. You find the **Coordinate System Settings** dialog box on the **Options** menu. Read more about creating a coordinate system and how to use it in "Coordinate Systems" on page 136.

The following table shows the material properties for the union of all constitutive forms and all piezoelectric application modes.

PARAMETER	VARIABLE	DESCRIPTION	CONSTITUTIVE FORM
c_E	cElk	Elasticity matrix	Stress-charge
s_E	sElk	Compliance matrix	Strain-charge
e	eik	Coupling matrix, stress-charge form	Stress-charge
d		Coupling matrix, strain-charge form	Strain-charge

PARAMETER	VARIABLE	DESCRIPTION	CONSTITUTIVE FORM
ε_{rS}		Relative permittivity matrix, stress-charge form	Stress-charge
ε_{rT}		Relative permittivity matrix, strain-charge form	Strain-charge
ρ	`rho`	Density	All
th	`thickness`	Thickness of the geometry (2D only)	All

Elasticity matrix defines the stress-strain relation matrix c_E

$$\sigma = c_E \varepsilon$$

where σ is the stress, and ε is the strain.

Coupling matrix defines the piezo coupling matrix e used in the stress-charge form of the constitutive equation

$$\sigma = c_E \varepsilon - e^T \mathbf{E}$$

where σ is the stress, ε is the strain, and $\mathbf{E}$ is the electric field.

Compliance matrix defines the strain-stress relation matrix s_E

$$\varepsilon = s_E \sigma$$

where σ is the stress, and ε is the strain.

Coupling matrix defines the piezo coupling matrix d used in the strain-charge form of the constitutive equation

$$\varepsilon = s_E \sigma + d^T \mathbf{E}$$

where σ is the stress, ε is the strain, and $\mathbf{E}$ is the electric field.

Relative permittivity the relative permittivity, ε_{rS} and ε_{rT}, appears in the constitutive relation on stress-charge and strain-charge forms, respectively.

$$\mathbf{D} = e\varepsilon + \varepsilon_0 \varepsilon_{rS} \mathbf{E}$$

$$\mathbf{D} = d\sigma + \varepsilon_0 \varepsilon_{rT} \mathbf{E}$$

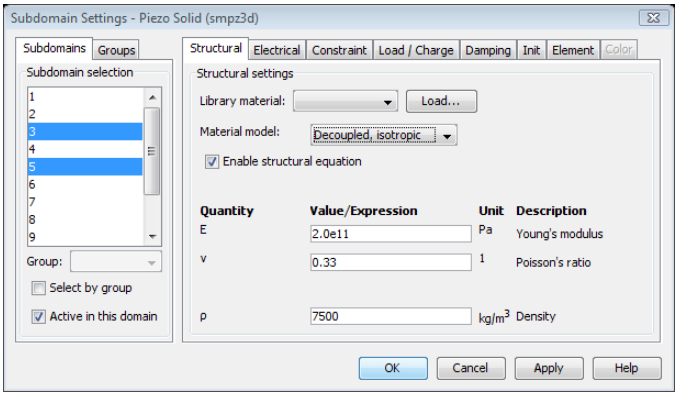
Density this material property, ρ, specifies the material's density.

Thickness this material property, thickness, specifies the material's thickness and appears in 2D only.

SUBDOMAIN SETTINGS FOR DECOUPLED, ISOTROPIC MATERIAL

With this material model you specify material properties on the **Structural** page and the **Electrical** page.

You define the structural material properties on the **Structural** page:



On the first row after the **Material Model** list you find the **Enable structural equation** check box. Use this check box to activate the structural equation or inactivate it to model only electrical problems. By default the **Enable structural equation** check box is selected. If this setting is selected you can define the following structural material properties:

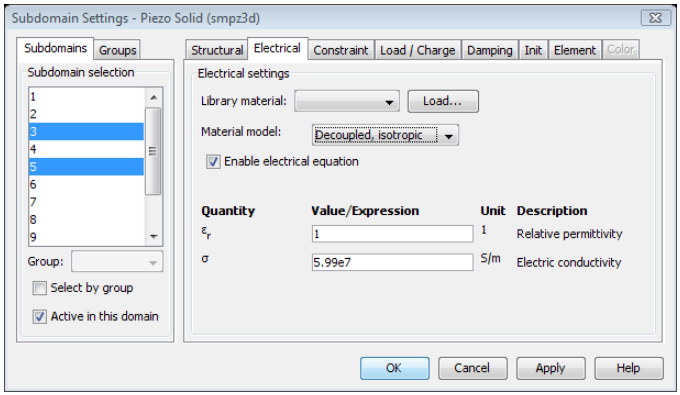
Young's modulus This material property, E, is the modulus of elasticity of the material. It is used to form the elasticity matrix D for the stress strain relationship as described in the chapter "Material Models" on page 160.

Poisson's ratio This material property, ν, defines the contraction of the structure in the perpendicular direction. It is used to form the elasticity matrix D for the stress strain relationship as described in the chapter "Material Models" on page 160.

Density this material property, ρ, specifies the material's density.

Thickness this material property, thickness, specifies the material's thickness and appears in 2D only.

You define the electrical material properties on the **Electrical** page:



On the first row after the **Material Model** list you find the **Enable electrical equation** check box. Use this check box to activate the electrical equation or inactivate it to model only structural problems. If you select it and clear the **Enable structural equation** check box, only the electrical equation is active. By default the **Enable electrical equation** check box is selected. If this setting is selected you can define the following electrical material properties:

Relative permittivity This material property, ε_r, defines the isotropic relative electrical permittivity of the material.

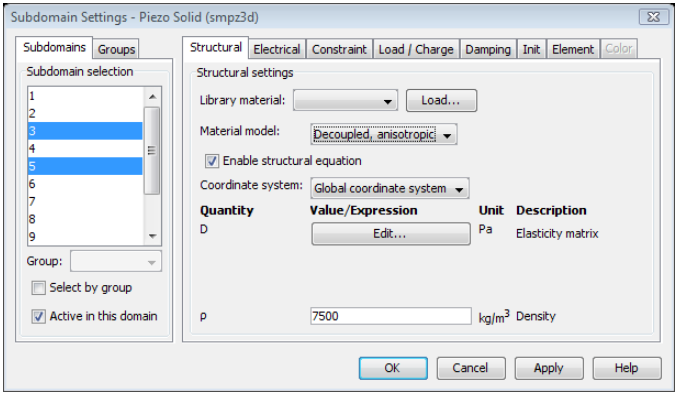
Electric conductivity This material property, σ, defines the isotropic electrical conductivity of the material. This setting only appears for frequency response analysis.

Thickness this material property, thickness, specifies the material's thickness and appears in 2D only.

SUBDOMAIN SETTINGS FOR DECOUPLED, ANISOTROPIC MATERIAL

With this material model you specify material properties on the **Structural** page and the **Electrical** page.

You define the structural material properties on the **Structural** page:

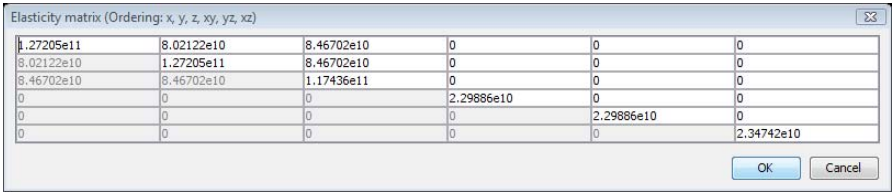


On the first row after the **Material Model** list you find the **Enable structural equation** check box. Use this check box to activate the structural equation or inactivate it to model only electrical problems. By default, **Enable structural equation** is selected. If this setting is selected you can define the following structural material properties:

Material orientation (2D and axisymmetry only): Here you select how the 3D material properties are oriented relative the 2D/axial symmetric analysis plane. There are six options: *xy*, *yz*, *zx*, *yx*, *zy*, and the default *xz*. This setting works the way same as for the piezoelectric material (See description on page 168).

Coordinate system Select the coordinate system where the material properties are defined. This setting works the way same as for the piezoelectric material (See description on page 170).

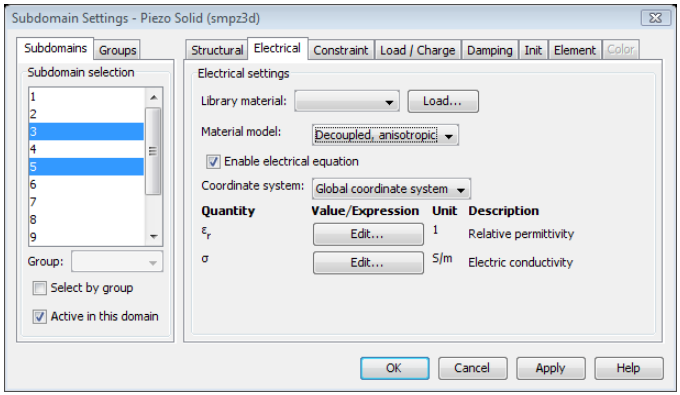
Elasticity matrix This material property, D, defines the elasticity matrix of the anisotropic material (See "Material Models" on page 160.). You define D as a symmetric 6-by-6 matrix:



Density this material property, ρ, specifies the material's density.

Thickness this material property, thickness, specifies the material's thickness and appears in 2D only.

You define the electrical material properties on the **Electrical** page:

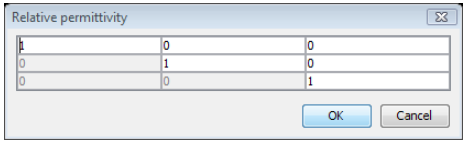


On the first row after the **Material Model** list you find the **Enable electrical equation** check box. Use this check box to activate the electrical equation or inactivate it to model only structural problems. By default **Enable electrical equation** is selected. If this setting is selected you can define the following electrical material properties:

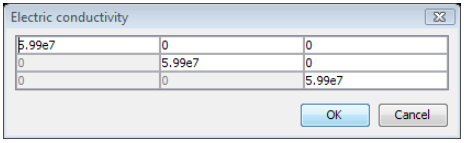
Material orientation (2D and axisymmetry only) This is the same setting as the **Material orientation** in the **Structural** page.

Coordinate system This is the same setting as the **Coordinate system** on the **Structural** page.

Relative permittivity This material property, ε_r, defines the anisotropic relative electrical permittivity of the material. You define ε_r using a symmetric 3-by-3 matrix:



Electric conductivity This material property, σ, defines the anisotropic electrical conductivity of the material. This setting only appears for frequency response analysis. You define σ using a symmetric 3-by-3 matrix:



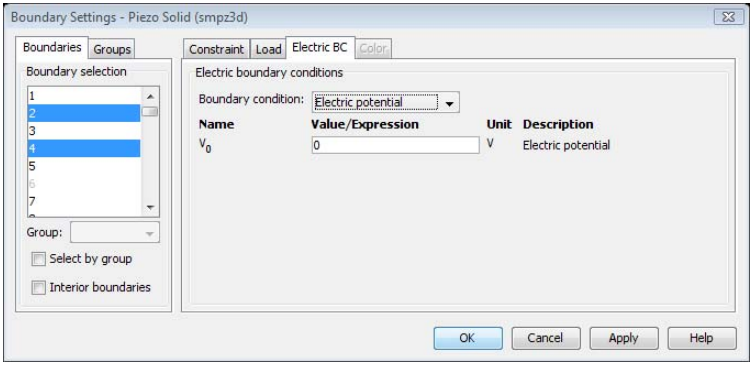
Thickness this material property, thickness, specifies the material's thickness and appears in 2D only.

PIEZOELECTRIC MATERIALS PROPERTIES LIBRARY

A library of about 25 common piezoelectric materials is available through the **Materials/ Coefficients Library** dialog box. For more information about these materials, see "Piezoelectric Material Properties Library" on page 321.

Electric Boundary Conditions

The electric boundary conditions in the piezoelectric application modes depend on the setting of the **Electrostatics formulation** property in the **Application Mode Properties** dialog box. You specify the electric boundary conditions on the **Electric BC** page in the **Boundary Settings** dialog box.



The **Electric BC** page also has a **Boundary condition** list where you select the type of electric boundary condition; the software enables different edit fields depending on the selected type.

BOUNDARY CONDITIONS FOR ELECTROSTATICS

For the Unsymmetric, Electrostatic and Symmetric, Electrostatic formulations, the boundary conditions include:

Electric Displacement

$$\mathbf{n} \cdot \mathbf{D} = \mathbf{n} \cdot \mathbf{D}_0$$

This boundary condition specifies the normal component of the electric displacement at a boundary. Enter the components of the electric displacement $\mathbf{D}_0$.

Surface Charge

$$-\mathbf{n} \cdot \mathbf{D} = \rho_s, \qquad \mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

This boundary condition specifies the surface charge density ρ_s at an exterior boundary (left equation) or at the interior boundary between two media with electric displacement $\mathbf{D}_1$ and $\mathbf{D}_2$, respectively.

Zero Charge/Symmetry

$$\mathbf{n} \cdot \mathbf{D} = 0$$

This boundary condition specifies that the normal component of the electric displacement is zero. The Zero charge/Symmetry boundary condition is also useful at symmetry boundaries where the potential is symmetric with respect to the boundary.

Electric Potential

$$V = V_0$$

This boundary condition specifies the voltage V_0 at the boundary. Because the application mode computes the electric potential, you must define its value at some boundary in the geometry to be fully determined.

Ground

$$V = 0$$

This boundary condition is a special case of the previous one specifying zero potential. The Ground boundary condition is also be useful at symmetry boundaries, where the potential is antisymmetric with respect to the boundary.

Continuity

$$\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

This boundary condition specifies that the normal component of the electric displacement is continuous across an interior boundary or across a boundary between a piezoelectric and an electrostatic domain if you use the Unsymmetric, Electrostatic formulation. Using the Symmetric, Electrostatic formulation the Continuity condition is only available for interior boundaries, where it is the default.

Floating Potential

This condition the potential on the boundary to a spatially constant value such that the total charge on the boundary equals the user defined total charge Q_0:

$$\int_{\partial\Omega} \rho_s = Q_0$$

You also define the group index, which defines how the boundaries are grouped in to a set of electrodes.

Axial Symmetry

$$E_r = 0$$
$$\frac{\partial E_z}{\partial r} = 0$$

This boundary condition is the natural Neumann boundary condition, which you use on the z-axis ($r = 0$) to maintain the symmetry conditions. The Axial Symmetry boundary condition is available only in the Piezo Axial Symmetry application mode.

BOUNDARY CONDITIONS FOR ELECTRIC CURRENTS

For the Unsymmetric, Electric currents formulations, the boundary conditions include:

Ground

$$V = 0$$

This boundary condition is a special case of the previous one specifying zero potential. The Ground boundary condition is also be useful at symmetry boundaries, where the potential is antisymmetric with respect to the boundary.

Electric Potential

$$V = V_0$$

This boundary condition specifies the voltage V_0 at the boundary. Because the application mode computes the electric potential, you must define its value at some boundary in the geometry to be fully determined.

Current Flow

$$\mathbf{n} \cdot \mathbf{J} = \mathbf{n} \cdot \mathbf{J}_0$$

This boundary condition specifies the current flow. Enter the components of the current density $\mathbf{J}_0$.

Inward Current Flow

$$-\mathbf{n} \cdot \mathbf{J} = J_n$$

This boundary condition specifies the normal current density J_n at an exterior boundary.

Electric Insulation

$$\mathbf{n} \cdot \mathbf{J} = 0$$

This boundary condition specifies that the normal component of the electric current is zero; that is, the boundary is electrically insulated.

Current Source

The current source boundary condition

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = J_n$$

is applicable to interior boundaries that represent either a source or a sink of current.

Continuity

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

This boundary condition specifies that the normal component of the electric current is continuous across the interior boundary (where it is the default setting) or across a boundary between a piezoelectric and an domain with electric currents.

Floating Potential

This condition the potential on the boundary to a spatially constant value such that the total current through the boundary equals the user defined total current I_0:

$$\int_{\partial\Omega} -\mathbf{n} \cdot \mathbf{J} = I_0$$

You also define the group index, which defines how the boundaries are grouped in to a set of electrodes.

Axial Symmetry

This boundary condition is the natural Neumann boundary condition, which you use on the z-axis ($r = 0$) to maintain the symmetry conditions. The Axial Symmetry boundary condition is available only in the Piezo Axial Symmetry application mode.

CONVERSION OF ELECTRIC BOUNDARY CONDITIONS

Some boundary conditions are applicable only for the formulations for electrostatics, whereas others apply only to the formulation for electric currents. Table 5-1 contains the boundary conditions that the software converts when changing from one formulation to the other:

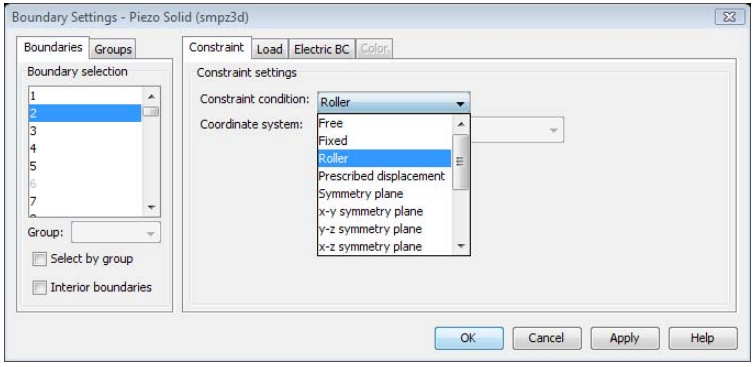
TABLE 5-1: BOUNDARY CONDITION CONVERSIONS

BOUNDARY CONDITION FOR ELECTROSTATICS	BOUNDARY CONDITION FOR ELECTRIC CURRENTS
Electric displacement	Current flow
Zero charge/Symmetry	Electric insulation
Surface charge (exterior boundaries)	Inward current flow
Surface charge (interior boundaries)	Current source

Constraints

A constraint specifies the displacement or potential of certain parts of a structure. You can define constraints for the displacements on all domain levels including points, edges, faces/boundaries, and subdomains (in 3D), and points, boundaries, and subdomains (in 2D). In addition, you can define constraints for the potential on points and edges in 3D, and for points in 2D. To control them, go to the **Constraint** page in the **Subdomain**/**Boundary**/**Edge**/**Point Settings** dialog boxes, and set constraints on boundaries from the **Electric BC** page. The following figure shows the **Boundary Settings**

dialog box for the Piezo Solid application mode, but the page has the same appearance in all piezoelectric application modes.

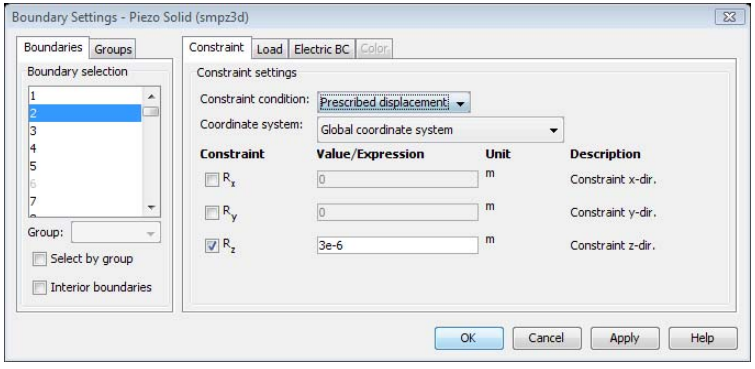


Use the **Constraint condition** list in this dialog box to select the type of constraint that you want to define. See "Constraints" on page 96 for details.

The **Coordinate system** list lets you control in which coordinate system you want the constraint defined. Available options are:

- Global coordinate system
- Tangent and normal coordinate system, available only on boundaries
- User-defined coordinate systems, if any local coordinate systems are defined. (Read more about creating a coordinate system in the section "Coordinate Systems" on page 136.)

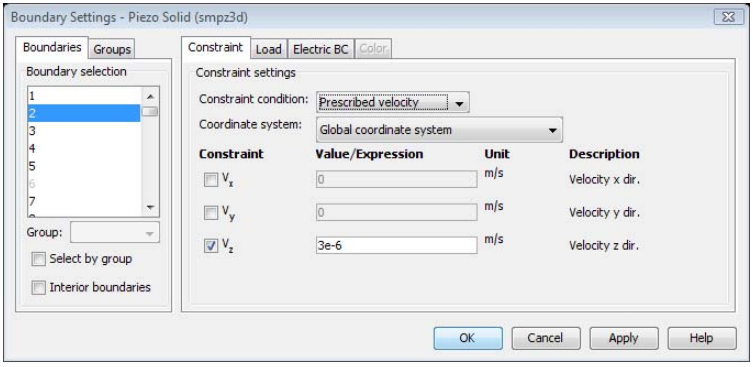
When you select **Prescribed displacement** a number of new options appears in the dialog box and the **Constraint** page takes on this appearance:



The Constraint page showing the prescribed displacement options.

The check boxes adjacent to the $\mathbf{R_x}$, $\mathbf{R_y}$, and $\mathbf{R_z}$ edit fields activate the constraint, whereupon you enter the value/expression of the displacement (the default value is `0`).

In a frequency response analysis you have the possibility to specify not only a harmonic displacement but also a harmonic velocity or acceleration. You specify the **Prescribed velocity** and **Prescribed acceleration** in the same way as **Prescribed displacement**.

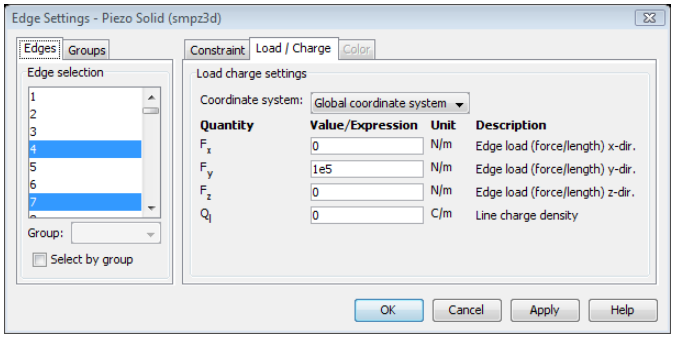


Constraint page showing the prescribed velocity settings.

Loads and Charges

Load is a general name for forces applied to a structure. You can specify loads on all domain types. To do so, click the **Load** tab in the **Boundary Settings** dialog boxes or the **Load/Charge** tab in the **Subdomain Settings**, **Edge Settings**, and **Point Settings** dialog boxes, where you can also specify a charge density. The formulation for electric currents does not include charges, so in that case, the name of the tab is **Load** also in the **Subdomain Settings**, **Edge Settings**, and **Point Settings** dialog boxes. The following

image shows the **Edge Settings** dialog box for the Piezo Solid application mode, but the tab looks similar on all domain levels in all piezoelectric application modes.



SPECIFYING LOADS

For plane stress and plane strain, option buttons allow you to specify the load in different ways using the thickness. The following table summarizes the options for defining loads on different domains in different application modes; the SI unit appears in parenthesis.

APPLICATION MODE	POINT	EDGE	BOUNDARY	SUBDOMAIN
Plane Stress, Plane Strain	force (N)		force/area (N/m^2) or force/length (N/m)	force/volume (N/m^3) or force/area (N/m^2)
Axial symmetry	total force along the circumferential (N)		force/area (N/m^2)	force/volume (N/m^3)
Solid	force (N)	force/length (N/m)	force/area (N/m^2)	force/volume (N/m^3)

With the **Coordinate system** list you control in which coordinate system the load is defined. Available options are:

- Global coordinate system
- Tangent and normal coordinate system, only available on boundaries
- User-defined coordinate systems, if there are any local coordinate systems defined. Read more about creation of coordinate system in the coordinate system section.

SPECIFYING CHARGES

You can specify a charge on the **Edge**/**Point** level when you use a formulation for electrostatics. For plane stress and plane strain, option buttons allow you to specify the charge in different ways using the thickness. The following table summarizes the

options for defining charge on different domains in different application modes; the SI units appears in parenthesis.

APPLICATION MODE	POINT	EDGE	SUBDOMAIN
Plane Stress, Plane Strain	charge (C)		charge/volume (C/m^3) or charge/area (C/m^2)
Axial symmetry	total charge along the circumferential (C)		charge density (C/m^3)
Solid	force (C)	charge/length (C/m)	charge density (C/m^3)

To specify charge density on boundaries, click the **Electric BC** tab.

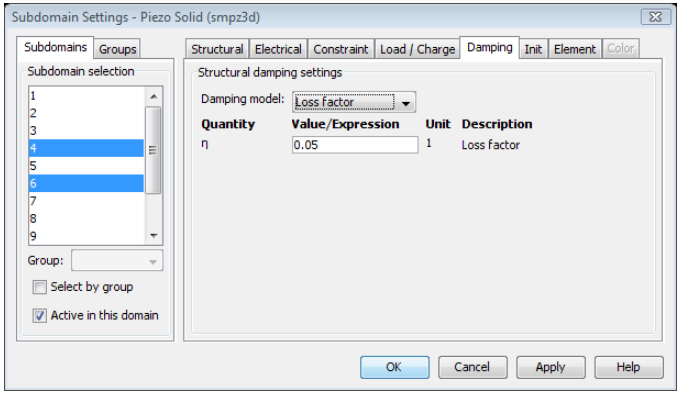
Structural Damping

For time-dependent analysis, you can specify viscous damping (structural damping) using Rayleigh damping, where the damping matrix is specified to be proportional to the mass and stiffness matrix:

$$C = \alpha_{dM} M + \beta_{dK} K$$

For frequency response analysis you can specify viscous damping using either Rayleigh damping, loss factor damping, or equivalent viscous damping.

To specify structural damping parameters, go to the **Damping** page in the **Subdomain Settings** dialog box, and choose the type of damping model from the **Damping model** list. The layout of the dialog box changes for each damping model.



The Damping page when loss factor damping is selected.

Note: Loss factor damping and equivalent viscous damping are valid only for frequency response analysis. If you choose a transient analysis and either of these damping types, COMSOL Multiphysics solves the model with no damping.

Table 5-2 and the following text describe the parameters that define damping:

TABLE 5-2: PARAMETERS FOR DAMPING MODELS

PARAMETER	VARIABLE	DESCRIPTION	DAMPING MODEL
α_{dM}	`alphadM`	Mass-damping parameter	Rayleigh
β_{dK}	`betadK`	Stiffness-damping parameter	Rayleigh
η	`eta`	Loss factor	Loss factor, Equivalent viscous

Mass damping parameter Defines the Rayleigh damping model's mass damping, α_{dM}.

Stiffness damping parameter Defines the Rayleigh damping model's stiffness damping, β_{dK}.

Loss factor Defines the loss factor η for the loss factor damping and equivalent viscous damping models.

The Piezo Solid Application Mode

Use the Piezo Solid application mode for analysis of 3D structures that exhibit piezoelectric effects.

VARIABLES AND SPACE DIMENSIONS

The degrees of freedom (dependent variables) are the global displacements u, v, and w in the global x, y, and z directions, and the electric potential, V.

PDE FORMULATION

The implementation of this application mode uses the principle of virtual work, described in general terms in the section "Implementation" on page 81.

APPLICATION MODE VARIABLES

For information about available application mode variables, see "Piezoelectric Application Modes" on page 28 in the *MEMS Module Reference Guide*.

The Piezo Plane Stress Application Mode

Use the Piezo Plane Stress application mode to analyze thin in-plane loaded plates that exhibit piezoelectric effects.

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 electric potential V.

PDE FORMULATION

The implementation of this application mode uses the principle of virtual work, which this manual describes in general terms in the section "Implementation" on page 81.

Application Mode Parameters

For details about the application mode parameters that define the loads, charges, material properties, constraints, and electric boundary conditions, see the sections earlier in this chapter.

APPLICATION MODE VARIABLES

For information about available application mode variables, see "Piezoelectric Application Modes" on page 29 in the *MEMS Module Reference Guide*.

The Piezo Plane Strain Application Mode

Use the Piezo Plane Strain application mode to compute the global displacements (u, v) in the x and y directions and the electric potential for a piezoelectric structure in a state of plane strain. The plane strain condition assumes that the ε_z, ε_{yz}, and ε_{xz} components of the strain tensor are zero.

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 electric potential V.

PDE FORMULATION

The implementation of this application mode uses the principle of virtual work, described in general terms in the section "Implementation" on page 81.*Application Mode Parameters*

For details about the application mode parameters that define the loads, charges, material properties, constraints, and electric boundary conditions, see the sections earlier in this chapter.

APPLICATION MODE VARIABLES

For information about available application mode variables, see "Piezoelectric Application Modes" on page 29 in the *MEMS Module Reference Guide*.

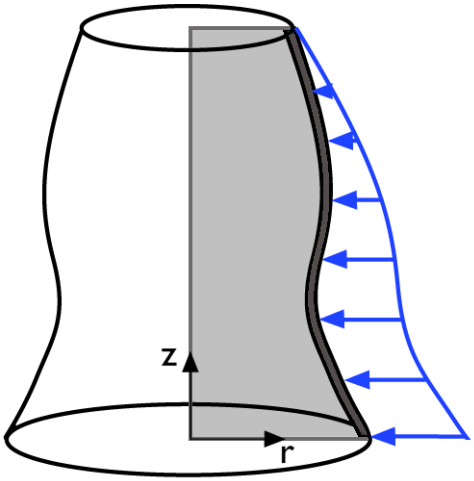
The Piezo Axial Symmetry Application Mode

Use the Piezo Axial Symmetry application mode to analyze axisymmetric models of materials showing piezoelectric effects.

This application mode uses cylindrical the coordinates r, φ (`phi`), and z. It solves the equations for the global displacement (u, w) in the r and z directions. It assumes that the displacement v in the φ direction together with the $\tau_{r\varphi}$, $\tau_{\varphi z}$, $\gamma_{r\varphi}$, and $\gamma_{\varphi z}$ components of the stresses and strains are zero. Loads are independent of φ, and it allows loads only in the r and z directions.

You can consider the domain where the software solves the equations as the intersection between the original axially symmetric 3D solid and the half plane $\varphi = 0$, $r \geq 0$. Therefore it is necessary to draw the geometry only in the half plane $r \geq 0$. The

software recovers the original 3D solid by rotating the 2D geometry about the z-axis as seen in the following figure:



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

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

To avoid division by r (which causes problems on the axis, where $r = 0$), the program automatically transforms the equations by multiplying by r. When using the principle of virtual work, you normally do not think of this multiplication as a transformation but merely as an integration around the circumference. Integrating over the volume, you must multiply the integrand by $2\pi r$. The application mode introduces and solves for a new dependent variable

$$\mathrm{uor} = \frac{u}{r}$$

instead of the true radial displacement, u.

Note: $r = 0$ is the symmetry axis. x -> r and y -> z in the Piezo Axial Symmetry, application mode.

VARIABLES AND SPACE DIMENSIONS

The degrees of freedom (dependent variables) are `uor` the radial displacement divided by r and w the global displacement in the z direction and the electric potential V.

PDE FORMULATION

The implementation of this application mode uses the principle of virtual work, described in general terms in the section "Implementation" on page 81.

Application Mode Parameters

For details about the application mode parameters that define the loads, charges, material properties, constraints, and electric boundary conditions, see the sections earlier in this chapter.

APPLICATION MODE VARIABLES

For information about available application mode variables, see "Piezoelectric Application Modes" on page 29 in the *MEMS Module Reference Guide*.

6

Film Damping Application Modes

Film damping often appears in microsystems, where a narrow gas film surrounds vibrating structures. It is an important factor in accelerometers and resonators. Film damping can be an unwanted phenomenon, but it can also be used for adjusting the transient operation of a component. When a structure's movement is mostly along its boundary normals, resulting in compression of the gas film, one usually refers to *squeezed film damping*. Tangential movements result in *slide film damping* or, looking from another perspective, the film is used for lubrication.

The Film Damping application mode allows you to model squeezed film damping, slide film damping, or their combination using the modified Reynolds equation. This application mode is a boundary mode, that is, if you draw a block in 3D, the application mode simulates the gas film outside the block. You can use this application mode to model the pressure distribution in the gas film. You can also do coupled analyses, where you model the film damping and structural problems simultaneously, with the film pressure acting as a boundary load on the structure.

For squeezed film damping, the Film Damping application mode can model structures with as well as without perforations, the former by adding perforation-specific extensions to the modified Reynolds equation.

The Film Damping application mode is available in 2D, 3D, and 2D axisymmetry. There are also three predefined multiphysics couplings—Plane Strain with Film Damping, Solid, Stress-Strain with Film Damping, and Axial Symmetry, Stress-Strain with Film Damping—facilitating the creation of coupled models.

Theory Background

Conditions for Film Damping

The drawing in Figure 6-1 shows an example system where film damping is expected to appear: a thin channel of fluid located between two moving structures. The upper structure is here referred to as *the moving structure*, whose damping is of interest, and the lower one is referred to as the *channel base*. Both structures can be in arbitrary motion but one usually distinguishes between squeezed film damping for mostly normal movements and slide film damping (or lubrication) for mostly tangential movements.

The gas film poses two kinds of forces to the moving structure. Initially both structures are surrounded by gas with a constant pressure p_a, and the gas can freely move into and out of the gap. Due to the movements (normal displacement and tangential velocity) of the structures, an additional and usually time-dependent pressure component, the film pressure p_f, appears in the gas inside the gap. Thus an effective force equal to $\mathbf{F}_n = -\mathbf{n}p_f$ affects the structure in the normal direction ($\mathbf{n}$ is the normal unit vector from the structure to the fluid). Another force that affects the moving structure is the viscous drag force of the fluid $\mathbf{F}_t$, which resists the tangential movement of the structure.

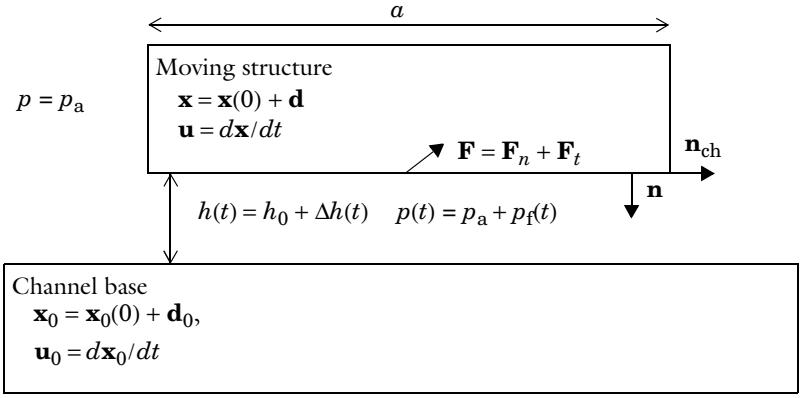


Figure 6-1: Squeezed film damping takes place when a narrow gas film is located between a fixed and a moving structure.

In MEMS devices, the aspect ratios are very large, which means that the relative dimensions of the system are such that the horizontal dimension, a, is always much

larger than the gap, or film thickness, h. It follows that the inertial effects in the fluid are negligible compared to the viscous effects (below MHz frequencies), the pressure is constant over the film thickness, and the velocity has a parabolic profile. Also, it can be assumed that the curvature of the channel is small and that the channel boundaries are almost parallel. Furthermore, owing to the small thickness of the gas film, the fluid is practically always isothermal.

Given these assumptions, solving the full gas flow problem—described by the Navier-Stokes equations in the gap—reduces to solving the Reynolds equation in the channel boundary. The classical Reynolds equation is valid for large-scale problems. In microsystems, where the continuum assumption often is not valid, you can use a so-called modified Reynolds equation that also covers the rarefied gas effects taking place at the microscale.

The Modified Reynolds Equation

The following equation shows the time-dependent formulation of the modified Reynolds equation (Ref. 1, Ref. 2, and Ref. 3). It solves the film-pressure variation, p_f, in the gap:

$$\nabla_t \cdot (h^3 Q_{ch} p \nabla_t p_f - 6\eta p h(\mathbf{u}_t + \mathbf{u}_{0,t})) = 12\eta\left(\frac{dp_f}{dt}h + p\frac{dh}{dt}\right) \qquad (6\text{-}1)$$

In this equation, $h = h_0 + \Delta h(t)$ is the air gap's height consisting of the initial gap and the deformation in the normal direction of the boundary, η denotes the fluid viscosity at normal conditions, and $p = p_a + p_f$ is the total fluid pressure consisting of the initial/ambient pressure and the variation. The term Q_{ch} is the relative flow rate function that accounts for the rarefied gas effects (See "Rarefaction and Slip Effects" on page 197.) The subscript t in ∇_t refers to the fact that this equation is solved on a boundary and that all gradients are computed using the tangential derivatives along the boundary. $\mathbf{u}_t$ and $\mathbf{u}_{0,t}$ are the tangential velocities of the moving structure and the channel base, respectively.

For modeling the frequency response of a microsystem, a linearized equation is used:

$$\nabla_t \cdot (h_0^3 Q_{ch} \nabla_t p_f - 6\eta h_0(\mathbf{u}_t + \mathbf{u}_{0,t})) = j\omega 12\eta\left(p_f\frac{h_0}{p_a} + \Delta h\right) \qquad (6\text{-}2)$$

where p_f and Δh correspond to complex pressure variation and gap deformation, respectively, that vary with angular frequency, ω. Also, $\mathbf{u}_t$ and $\mathbf{u}_{0,t}$ are the complex valued tangential velocities of the moving structure and the channel base, respectively.

Equation 6-1 and Equation 6-2 give two representations of the *full formulation* of the modified Reynolds equation. For slide film damping, it is often possible to make the simplifying assumption that the gap height h is constant in time. Then, h in Equation 6-1 can be replaced by h_0 and the time derivative dh/dt vanishes, as does the Δh term in Equation 6-2.

For squeezed film damping (Ref. 4, Ref. 5, and Ref. 6), a common assumption is that the tangential velocity components are zero. Thus, the second term under the divergence disappears from the squeezed film formulation of the modified Reynolds equation.

An important measure of the validity of the Reynolds equation to is the squeeze number σ (Ref. 1, Ref. 2):

$$\sigma = \frac{12\eta a^2 \omega}{p_a h_0^2},$$

where a is a typical horizontal dimension of the moving structure. In order that the Reynolds equation would properly describe the flow problem in the thin gas film σ must be much smaller the one: $\sigma << 1$.

Structural Loads

The boundary adjacent to the gas film experiences forces in normal and tangential directions. The normal load results directly from the pressure load p_f (assuming that the ambient pressure surrounds the body elsewhere):

$$\mathbf{F}_n = -\mathbf{n}p_f$$

where $\mathbf{n}$ is the outward pointing normal of the structure.

The tangential load results from the viscosity of the fluid and the velocity distribution over the film thickness. The validity of the Reynolds equation assumes that the fluid has parabolic velocity profile over the film thickness. Thus, rarefied gas the load expression is:

$$\mathbf{F}_t = -\eta \frac{\mathbf{u}_t - \mathbf{u}_{0,t}}{h(1 + 2\mathrm{Kn})} - \nabla_t p_f$$

and where continuum assumption holds:

$$\mathbf{F}_t = -\eta \frac{\mathbf{u}_t - \mathbf{u}_{0,t}}{h} - \frac{h}{2}\nabla_t p_\mathrm{f}$$

Here $\mathbf{u}_t$ and $\mathbf{u}_{0,t}$ are the tangential velocities of the boundary, and channel base, respectively.

Gas Outflow Conditions

When the system's aspect ratio is very large, a valid assumption is that the film-pressure variation vanishes at the end of the channel. Thus, the corresponding end condition is $p_\mathrm{f} = 0$.

If the aspect ratio is not very large, this condition is no longer valid, and the pressure variation continues outside of the gap. For these cases you can use a so-called boundary flow condition (Ref. 5),

$$\mathbf{n}_\mathrm{ch} \cdot \nabla_t p_\mathrm{f} = \frac{p_\mathrm{f}}{\Delta L} \tag{6-3}$$

where the elongation ΔL is the distance where p_f goes to zero outside the boundary assuming that the pressure drop is linear. The elongation ΔL can be a constant or it can be relative to the gap:

$$\mathbf{n}_\mathrm{ch} \cdot \nabla_t p_\mathrm{f} = \frac{p_\mathrm{f}}{\Delta L_r h} \tag{6-4}$$

Here, $\mathbf{n}_\mathrm{ch}$ is the outflow normal from the flow channel, as shown in Figure 6-1.

Table 6-1 summarizes the relative boundary elongation values for linear and torsional dampers from Ref. 5. This data especially concerns the squeezed film damping.

TABLE 6-1: RELATIVE ELONGATION FOR DIFFERENT ASPECT RATIONS

ASPECT RATIO	RELATIVE ELONGATION ($\Delta L/h$)	
a/h	LINEAR	TORSIONAL
4	0.8275	1.971
8	0.727	1.002
16	0.6805	0.791
32	0.659	0.7115

MEASURES OF RAREFACTION

There are several ways to identify the rarefaction level of the gas in the gap. The Knudsen number in the gap, **Kn**, is given by

$$\mathbf{Kn} = \frac{\lambda}{h}$$

where λ is the mean free path of the gas molecules at pressure $p_a + p_f$, and h is the local gap height. λ is defined by

$$\lambda = \frac{\lambda_0 p_{\lambda,0}}{p_a + p_f}$$

where λ_0 is the mean free path defined at constant pressure $p_{\lambda,0}$.

The scaled Knudsen number, **Ks**, is a product of the slip coefficient, σ_p, and Knudsen number, **Kn** (Ref. 5):

$$\mathbf{Ks} = \sigma_P \mathbf{Kn}\,.$$

The slip coefficient, σ_p, is defined by

$$\sigma_p = \frac{2 - \alpha_v}{\alpha_v}(1.016 - 0.1211(1 - \alpha_v)) \qquad (6\text{-}5)$$

where α_v is the tangential momentum accommodation coefficient. For rough surfaces with diffuse molecular reflection $\alpha_v = 1$, but for polished surfaces $\alpha_v < 1$. In this example assume that both boundaries of the gap have the same α_v value or that α_v represents their average effect.

Another measure of the gas rarefaction is the scaled inverse Knudsen number, D (Ref. 7):

$$D = \frac{\sqrt{\pi}}{2\mathbf{Kn}}$$

When modeling with time-dependent analysis, these parameters are always computed for the system's instantaneous state, but for frequency-response analysis these parameters are constants and correspond to the initial conditions: $\Delta h = 0, p_f = 0$.

RELATIVE FLOW RATE

The relative flow rate function, Q_{ch}, describes how the viscous flow within the narrow gap changes when the continuum assumption is not valid or if the no-slip condition for the flow is not the best assumption. The following list gives four alternative formulations of the relative flow function.

No-slip model: Q_{ch} is a constant and the continuum assumption holds (Ref. 4):

$$Q_{\mathrm{ch}} = 1 \tag{6-6}$$

Slip model: Q_{ch} is a function of Ks. This model assumes minor rarefaction so that the no slip condition in the channel boundaries is not valid (Ref. 4, Ref. 5):

$$Q_{\mathrm{ch}} = 1 + 6\mathrm{Ks} \tag{6-7}$$

Model 1: Q_{ch} is a function of Ks. For this model, the modified Reynolds equation is valid with 5% accuracy for $0 \le \mathrm{Kn} \le 880$ (Ref. 7):

$$Q_{\mathrm{ch}} = 1 + 9.638\mathrm{Ks}^{1.159} \tag{6-8}$$

Model 2: Q_{ch} is a function of D and α. For this model, the modified Reynolds equation is valid with 1% accuracy for $D \ge 0.01$ $(\mathrm{Kn} < 88.6)$ and $0.7 \le \alpha \le 1$ (Ref. 7):

$$Q_{\mathrm{ch}} = 1 + \frac{6}{\alpha^{1.34} D\sqrt{\pi}} \ln\left\langle \frac{1}{D} + 4.1 \right\rangle + \frac{\alpha}{1.15D} + \frac{7.8(1-\alpha)}{D + 0.08D^{2.83}} + \frac{3.84\alpha D^{-0.83}}{1 + 1.12D^{0.72}} \tag{6-9}$$

Modified Reynolds Equation for Perforated Structures

It is possible to extend the squeezed film formulation of the modified Reynolds equation to cases where the moving structure has perforations (Ref. 5, Ref. 6). Sometimes this is referred to as the *extended Reynolds equation*. The time-dependent form of the modified Reynolds equation with perforation effects is given by

$$\nabla_t \cdot (D_h h^3 Q_{\mathrm{ch}} p \nabla_t p_{\mathrm{f}}) = 12\eta\left(C_h \frac{dp_{\mathrm{f}}}{dt} h + p\frac{dh}{dt} + pY_h \Delta p_h\right).$$

In general, this equation uses the same definitions as Equation 6-1 without tangential velocity components, but the terms D_h, C_h, and Y_h are extensions specific to

perforated structures: *relative diffusivity*, *relative compressibility*, and *perforation admittance*, respectively. The last term in the equation represents the flow through the perforations, and it depends on the pressure difference over the perforations. A valid assumption for a system such as that of the type in Figure 6-1 is that $\Delta p_h = p_f$.

Depending on their assigned values, D_h, C_h, and Y_h can represent an average value of perforations of a perforation distribution (the homogenized method, Ref. 6), or they can be assigned separately on hole-by-hole basis (the Perforation Profile Reynolds or PPR method, Ref. 8). Setting $D_h = 1$, $C_h = 1$, and $Y_h = 0$, the equation (Equation 6-1) for nonperforated structures results.

Here also the frequency-response analysis uses the following linearized equation:

$$\nabla_t \cdot (D_h h_0^3 Q_{ch} \nabla_t p_f) = j\omega 12\eta\left(C_h p_f \frac{h_0}{p_a} + \Delta h\right) + 12\eta Y_h \Delta p_h$$

Geometry Orientations

The Film Damping application mode calculates the film-thickness variation as the difference of the normal displacements of the moving structure and the channel base $dh = -d_n - d_{0,n}$, where the '–' sign emphasizes that the film gets narrower for boundary deformation toward the fluid.

The normal deformation, on the other hand, is a product of the boundary normal ($\mathbf{n}$ and $\mathbf{n}_0$) and the boundary deformation ($\mathbf{d}$ and $\mathbf{d}_0$) of the structure and the channel base. The implementation of the Film Damping application mode assumes that the film boundaries are parallel so that the two normals are parallel but have opposite signs: $\mathbf{n} = -\mathbf{n}_0$. Thus,

$$d_n = \mathbf{n}\cdot\mathbf{d}$$

$$d_{0,n} = -\mathbf{n}\cdot\mathbf{d}$$

The orientation of the normal vector $\mathbf{n}$ has also an effect on how to view the relative locations of the structures in Figure 6-1.

If you use simple blocks, the normal $\mathbf{n}$ always points out of the structure. Thus the fixed surface is also outside of the structure. But if the geometry consists only of a single boundary, the $\mathbf{n}$ vector always points along the positive coordinate directions. Thus, for a boundary on the *xy*-plane, the fixed surface resides on top of the boundary. Also, if the geometry consists of several blocks, the $\mathbf{n}$ vector always points along the positive coordinate directions for all interior boundaries. If you want to assign film

damping on these boundaries, be careful to have the correct sign on the components of **n**.

References

1. K.S. Breuer, "Chapter 9. Lubrication in MEMS," *The MEMS Handbook*, M. Gad-el-Hak, ed., CRC Press, 2002.

2. M. Bao, "Chapter 3. Air Damping," *Analysis and Design Principles of MEMS Devices*, Elsevier, 2005.

3. A. Burgdorfer, "The Influence of the molecular mean free path on the performance of hydrodynamic gas lubricated bearings," *Transducers of the ASME*, March 1959.

4. S.D. Senturia, *Microsystem Design*, Kluwer Academic Publishers, 2003.

5. T. Veijola, A. Pursula, and P. Råback, "Extending the validity of squeezed-film damper models with elongations of surface dimensions," *J. Micromech. Microeng.*, vol. 15, pp. 1624–1636, 2005.

6. M. Bao, H. Yang, Y. Sun, and P.J. French, "Modified Reynolds equation and analytical analysis of squeeze-film air damping of perforated structures," *J. Micromech. Microeng.*, vol. 13, pp. 795–800, 2003.

7. T. Veijola, H. Kuisma, and J. Lahdenperä, "The influence of gas-surface interaction on gas film damping in a silicon accelerometer," *Sensors and Actuators*, vol. A 66, pp. 83–92, 1998.

8. T. Veijola and P. Råback, "A method for solving arbitrary MEMS perforation problems with rare gas effects," *NSTI-Nanotech*, vol. 3, 2005.

The Film Damping Application Mode

This section describes the user interfaces and other details of the Film Damping application mode in the MEMS Module, reviewing the following parameters:

- Variables and space dimensions
- Properties
- Scalar variables
- Boundary settings
- Edge and point settings
- Application mode variables

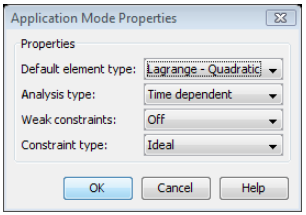
Variables and Space Dimensions

The Film Damping application mode is available in 2D, 2D axisymmetric, and 3D geometries. This application mode is a boundary mode, which means that the boundary level is the highest level where this application mode has equations and user interfaces; it does not have any user interfaces or equations at the subdomain level. Using boundary equations, this application mode models the pressure in a narrow flow channel that is assumed to reside just outside of the boundary.

This application mode has only one dependent variable, the gas film pressure, p_f.

Properties

The following figure shows the **Application Mode Properties** dialog box of the Film Damping application mode (choose **Physics>Properties**). Here you control global settings for a model:



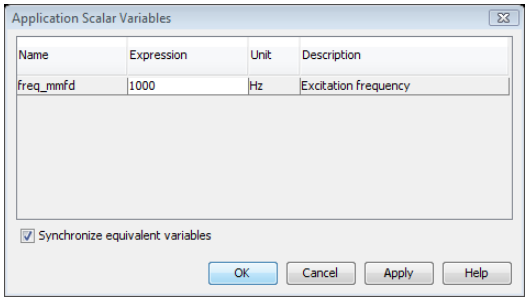
Application mode properties for the Film Damping application mode.

- **Default element type**: The selected element type is the default for all new film damping boundaries, and it does not affect any boundaries already created. Available elements are:
  - **Lagrange - Linear**
  - **Lagrange - Quadratic**
  - **Lagrange - Cubic**
  - **Lagrange - Quartic**
  - **Lagrange - Quintic**
- **Analysis type**: This list shows the analyses you can perform:
  - **Time dependent** (the default)
  - **Frequency response**

  If a model couples film damping to a structural application mode, you must use the same analysis type in both application modes—the software does not synchronize the analysis types between application modes.
- **Weak constraints**: This list controls if the model includes weak constraints or not. The default setting is **Off**.
- **Constraint type**: Choose the type of constraint in the model by selecting **Ideal** (the default) or **Non-ideal** from this list (see "Ideal vs. Non-Ideal Constraints" on page 301 in the *COMSOL Multiphysics Modeling Guide*).

Scalar Variables

The following figure shows the **Application Scalar Variables** dialog box for the Film Damping application mode (choose **Physics>Scalar Variables** to open). This dialog box is available only if you have selected a frequency-response analysis.



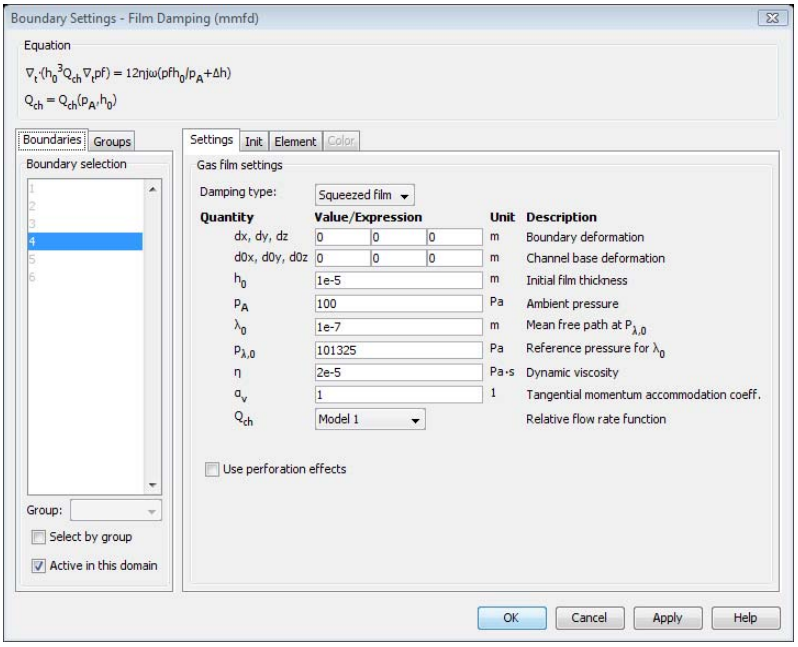
The Application Scalar Variables dialog box when analysis type is frequency response.

The Film Damping application mode has only one scalar variable: The excitation frequency in frequency-response analysis. If you create a model that couples structural and Film Damping application modes, you should write `freq_smps` to the expression field (see that the ending correspond to the structural mode you use) and clear the **Synchronize equivalent variables** check box. This way the parametric sweep of `freq_smps` uses the same frequency values for both application modes.

Boundary Settings

The **Boundary Settings** dialog box has four tabbed pages: on the **Settings** page you define the material for the fluid as well as the dimensions and deformation of the channel; on the **Init** page you define the initial value for the dependent variable; on the **Element** page you define the element to use; and on the **Color/Style** page you define colors for the boundary groups.

The following figure shows the **Settings** page in 3D:



Film damping boundary settings.

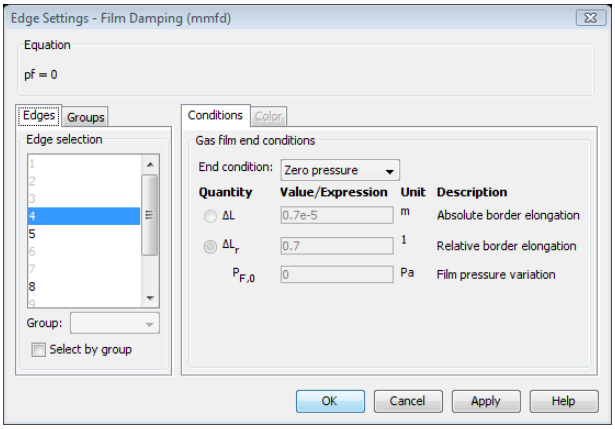
The following list shows the parameters that you can define on the **Settings** page:

- **Damping type**: define the formulation of the modified Reynolds equation. Available options are **Squeezed film**, **Slide film**, and **Full Reynolds**. Each will set the form of the Reynolds equation as described in the chapter "The Modified Reynolds Equation" on page 194.
- **Boundary deformation** (**dx, dy, dz**): define the boundary deformation. From these values the application mode calculates the normal deformation and the tangential velocity of the boundary.
- **Channel base deformation** (**d0x, d0y, d0z**): define the boundary deformation of the opposite side of the film. From these values the application mode calculates the normal deformation and the tangential velocity of the channel base boundary.
- **Initial film thickness** ($\mathbf{h_0}$): defines the initial gap height, which can have spatial variations.
- **Ambient pressure** ($\mathbf{p_A}$): defines the system's ambient pressure.

- **Mean free path at $p_{\lambda,0}$** (λ_0): defines the mean free path of the gas molecules at the reference pressure, $p_{\lambda,0}$.
- **Reference pressure for λ_0** ($p_{\lambda,0}$): defines the reference pressure for λ_0.
- **Dynamic viscosity** (η): defines the gas' dynamic viscosity.
- **Tangential momentum accommodation coeff.** (α_v): defines the tangential momentum accommodation coefficient for the gas molecule–boundary interaction.
- **Relative flow rate function** (Q_{ch}): from this list you can select the relative flow-rate function to be used. Available choices are **No slip**, **Slip**, **Model 1**, **Model 2**, and **User defined** according to the equations in the chapter "Relative flow rate" on page 198.
- **User-defined Q_{ch}** ($Q_{ch,u}$): this setting is not shown by default. It becomes visible when you select **User defined** from the **Relative flow rate function** list.
- **Use perforation effects**: activating this check box allows you to define perforation settings, as described in the chapter "Modified Reynolds Equation for Perforated Structures" on page 198. This selection is available only if you are modeling squeezed film damping.
- If you have activated the **Use perforation effects** check box you will have four more parameters to define:
  - Perforation coefficients (D_h, C_h, Y_h): define a perforated structure's relative diffusivity, relative compressibility, and perforation admittance, respectively.
  - **Perforation pressure difference** (Δp_h): defines the pressure difference over the perforation.

Edge and Point Settings

The Reynolds equation requires end conditions for the fluid flow, conditions that are defined on the edge level in 3D and on the point level in 2D. The following figure shows the **Edge Settings** dialog box for a 3D model.



The Edge Settings dialog box for the Film Damping application mode.

The following list details the settings you can define in edge and point settings dialog box:

- **End condition**: from this list you can select the edge/point condition. Available choices in all geometry types are:
  - **Zero pressure**: Defines $p_f = 0$ on the edge/point.
  - **Pressure**: Defines $p_f = p_{f,0}$ on the edge/point.
  - **Border flow**: Defines the pressure gradient according to Equation 6-3 or Equation 6-4 on page 196.
  - **Closed/Symmetry**: Defines $-\mathbf{n} \cdot \nabla p_f = 0$ condition on the edge/point.
  - **Neutral**: Removes any equations from the equation system. This condition is intended for internal edges in 3D, for pair and for periodic conditions.
  - In 2D axial symmetry you can also select **Axial symmetry** condition for points on the axial symmetry line.

  The **End condition** list is available only on edges/point where the film damping boundary (and thus the channel) ends. The list is inactive for any other edge/point between two film damping boundaries.

- **Absolute border elongation** (ΔL): if you have selected the **Border flow** condition you can define absolute elongation.
- **Relative border elongation** (ΔL_r): if you have selected the **Border flow** condition you can define relative elongation.
- **Film pressure variation** ($p_{F,0}$): if you have selected the **Pressure** condition you can define pressure variations.

Application Mode Variables

A number of variables and physical quantities are available for postprocessing and for use in equations and boundary conditions. For information of these variables see "Film Damping Application Modes" on page 66 in the *MEMS Module Reference Guide*.

Predefined Structural-Film Damping Multiphysics Couplings

Solid, Stress-Strain with Film Damping

Solid, Stress-Strain with Film Damping is a predefined multiphysics coupling that combines the Solid, Stress-Strain and the Film Damping application modes from the MEMS Module. You find this predefined coupling in the list of application modes under **MEMS Module>Structural Mechanics** in the Model Navigator when the space dimension is 3D.

There is a 2-way coupling on the boundaries of the application modes: the structural deformation enters the film damping application mode, and the solved film pressure appears as a boundary load on the structural boundaries.

When building a model, you work with predefined boundary groups: either select **Film damping** or **No damping** groups. In addition, you should assign the same group on both application modes. Initially (in the **default** group) all structural boundaries are free, and the Film Damping application mode is inactive on all boundaries. Thus you only need to activate film damping on selected boundaries to get started. A detailed description of the predefined boundary groups is available in Table 6-2 and Table 6-3.

By default, Solid, Stress-Strain is the ruling application mode of the two modes. Thus the **Analysis** list in the **Solver Parameters** dialog box shows the analyses available for the structural mode. When you change the analysis type, you also need the change the type for the Film Damping application mode: Select **Film damping (mmfd)** from the **Multiphysics** menu, then go to **Physics>Properties** and change the analysis type.

When you work with frequency-response analysis and run a frequency sweep with the parametric solver, the solver sweeps through the structural frequency `freq_smsld` by default. Consequently, you must synchronize the frequency variables of these application modes. To do this, follow these steps:

1. From the **Physics** menu, select **Scalar Variables**.
2. In the dialog that opens, clear the **Synchronize equivalent variables** check box.
3. In the **Expression** field for **freq_mmfd**, type `freq_smsld`.
4. Click **OK**.

TABLE 6-2: PREDEFINED GROUPS FOR FILM DAMPING APPLICATION MODE

DEFAULT	
Active in this domain	off
Boundary deformation dx, dy, dz	0 0 0
NO DAMPING	
Active in this domain	off
Boundary deformation dx, dy, dz	0 0 0
FILM DAMPING	
Active in this domain	on
Boundary deformation dx, dy, dz	u v w

TABLE 6-3: PREDEFINED GROUPS FOR SOLID, STRESS-STRAIN APPLICATION MODE. LOAD ARE DEFINED IN THE LOAD TAB OF THE BOUNDARY SETTINGS DIALOG.

DEFAULT	
F_x	0
F_y	0
F_z	0
NO DAMPING	
F_x	0
F_y	0
F_z	0
FILM DAMPING	
F_x	F_x_mmglf
F_y	F_y_mmglf
F_z	F_z_mmglf

Plane Strain with Film Damping

Plane Strain with Film Damping is a predefined multiphysics coupling that combines the Plane Strain and the Film Damping application modes from the MEMS Module. You can find this in the list of application modes in the **MEMS Module>Structural Mechanics** folder in the Model Navigator when the space dimension is 2D.

This is the 2D version of the Solid, Stress-Strain with Film Damping application mode. Thus, for a detailed description about the predefined groups and analysis types, see the just described section "Solid, Stress-Strain with Film Damping" on page 208. Also, when reading that material, keep in mind that the default abbreviation of the Plane Strain application mode is `_smpn`.

Axial Symmetry, Stress-Strain with Film Damping

Axial Symmetry, Stress-Strain with Film Damping is a predefined multiphysics coupling that combines the Axial Symmetry, Stress-Strain and the Film Damping application modes from the MEMS Module. You can find this predefined coupling as en entry in the list of application modes in the **MEMS Module>Structural Mechanics** folder in the Model Navigator when the space dimension is axial symmetry (2D).

This is the 2D axisymmetric version of the Solid, Stress-Strain with Film Damping application mode. Thus, for a detailed description of the predefined groups and analysis types, see "Solid, Stress-Strain with Film Damping" on page 208. Also, when reading that material, keep in mind that the default abbreviation of the Axial Symmetry, Stress-Strain application mode is `_smaxi`, the spatial dimensions are r and z, and that the radial deformation or the structure is given by `uaxi_smaxi` instead of u.

7

Electrostatics Application Modes

This chapter covers the following application modes in the MEMS Module:

- Electrostatics
- Conductive Media DC

Before detailing these two application modes, the text provides an overview of electrostatic fields.

The sections following the documentation of the application modes describe features in the MEMS Module that are useful when modeling electrostatics problems.

Electrostatic Fields

Modeling of static electric fields is carried out using the electric potential *V*. By combining the definition of the potential with Gauss' law and the equation of continuity, you can derive the classical Poisson's equation.

Under static conditions, the electric potential *V* is defined by the equivalence

$$\mathbf{E} = -\nabla V$$

Using this together with the constitutive relation $\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$ between $\mathbf{D}$ and $\mathbf{E}$, you can rewrite Gauss' law as Poisson's equation

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

This equation holds for nonconducting media and is used in the *Electrostatics* application mode. When handling conducting media, the equation of continuity is considered. In a stationary coordinate system, the point form of *Ohm's law* states that

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}^e$$

where $\mathbf{J}^e$ is an externally generated current density. The static form of the equation of continuity then gives us

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma \nabla V - \mathbf{J}^e) = 0$$

To handle current sources the equation can be generalized to

$$-\nabla \cdot (\sigma \nabla V - \mathbf{J}^e) = Q_j$$

This equation forms the basis for the *Conductive Media DC* application mode.

Conductive Media DC Application Mode

The *Conductive Media DC* application mode is available for 3D, 2D in-plane, and 2D axisymmetric models.

PDE FORMULATION

In the *Conductive Media DC* application mode the equation

$$-\nabla \cdot (\sigma \nabla V - \mathbf{J}^e) = Q_j$$

is solved.

The *in-plane Conductive Media DC* application mode assumes that your model has a symmetry where the electric potential varies only in the x and y directions and is constant in the z direction. This implies that the electric field $\mathbf{E}$ is tangential to the x-y plane. The application mode solves the following equation where d is the thickness in the z direction.

$$-\nabla \cdot d(\sigma \nabla V - \mathbf{J}^e) = dQ_j$$

The *axisymmetric Conductive Media DC* application mode is useful in the situation where the fields and the geometry are axially symmetric. In this case the electric potential is constant in the φ direction, which implies that the electric field is tangential to the rz-plane.

Writing the equation in cylindrical coordinates and multiplying it by r to avoid singularities at $r = 0$, the equation becomes

$$-\begin{bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial z} \end{bmatrix}^T \cdot \left(r\sigma \begin{bmatrix} \frac{\partial V}{\partial r} \\ \frac{\partial V}{\partial z} \end{bmatrix} - r\mathbf{J}^e \right) = rQ_j$$

Specifying the Conductivity

You can provide the conductivity using two different types of conductivity relations:

- The conductivity, either as an isotropic conductivity (a scalar number or expression) or as an anisotropic conductivity, using several components of a conductivity tensor to define an anisotropic material. See "Modeling Anisotropic Material" on page 215 in the *COMSOL Multiphysics User's Guide* for information about entering anisotropic material properties.
- A linear temperature relation for modeling a temperature-dependent conductivity (Joule heating or resistive heating). In this case the following equation describes the conductivity:

$$\sigma = \frac{1}{\rho_0(1 + \alpha(T - T_0))}$$

where ρ_0 is the resistivity at the reference temperature T_0. α is the temperature coefficient of resistivity, which describes how the resistivity varies with temperature. T is the current temperature, which can be a value that you specify or the

temperature from a heat transfer application mode (in the Joule Heating predefined multiphysics coupling, this is the default setting).

Select the type of conductivity from the **Conductivity relation** list.

BOUNDARY CONDITIONS

The relevant interface condition at interfaces between different media for this mode is

$$\mathbf{n}_2 \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

This is fulfilled by the natural boundary condition

$$\mathbf{n} \cdot [(\sigma \nabla V - \mathbf{J}^e)_1 - (\sigma \nabla V - \mathbf{J}^e)_2] = -\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

Current Flow

The current flow boundary condition

$$\mathbf{n} \cdot \mathbf{J} = \mathbf{n} \cdot \mathbf{J}_0$$

specifies the normal component of the current density flowing across the boundary.

Inward Current Flow

The inward current flow boundary condition

$$-\mathbf{n} \cdot \mathbf{J} = J_n$$

is similar to the above current flow boundary condition. In this case you specify the normal component of the current density rather than the complete vector. When the normal component J_n is positive the current flows inward through the boundary.

Distributed Resistance

You can use the distributed resistance boundary condition

$$\mathbf{n} \cdot \mathbf{J} = \frac{\sigma}{d}(V - V_{\text{ref}}), \qquad \mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = \frac{\sigma}{d}(V - V_{\text{ref}})$$

to model a thin sheet of a resistive material. The sheet has thickness d and is connected to the potential V_{ref}.

Electric Insulation

The electric insulation boundary condition

$$\mathbf{n} \cdot \mathbf{J} = 0$$

specifies that there is no current flowing across the boundary.

You can also use this boundary condition at symmetry boundaries where the potential is known to be symmetric with respect to the boundary.

Electric Potential

The electric potential boundary condition

$$V = V_0$$

specifies the voltage at the boundary. Because the potential is the dependent variable that the application mode solves for, its value has to be defined at some point or boundary in the geometry to be fully determined.

Ground

The ground boundary condition

$$V = 0$$

is a special case of the previous one specifying a zero potential. You can also use this boundary condition at symmetry boundaries, where the potential is known to be antisymmetric with respect to the boundary.

Current Source

The current source boundary condition

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = J_n$$

represents either a source or a sink of current at interior boundaries.

Continuity

The continuity boundary condition

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

specifies that the normal component of the electric current is continuous across the interior boundary.

Floating Potential

To set a floating potential with an integral constraint, use this boundary condition:

$$\int_{\partial\Omega} \mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = I_0 \quad \int_{\partial\Omega} -\mathbf{n} \cdot \mathbf{J} = I_0$$

This sets the potential to a constant value on the boundary such that the integral is fulfilled. See the section "Floating Potentials and Electric Shielding" on page 17 for an example.

Circuit Terminal

The circuit terminal is a special version of the floating potential boundary condition specialized for connection to external circuits. See the section "SPICE Circuit Import" on page 70 for more information.

Electric Shielding

The electric shielding boundary condition describes a thin layer of a dielectric medium that shields the electric field. See the section "Floating Potentials and Electric Shielding" on page 17 for an example.

Port

Use the port condition on the electrodes to calculate the conductance or resistance. This condition forces the potential or current to one or zero depending on the settings. See the section "The Port Page" on page 50 for more information about port conditions.

Contact Resistance

You can use the contact resistance boundary condition

$$(\mathbf{n} \cdot \mathbf{J})_1 = \frac{\sigma}{d}(V_1 - V_2)$$

$$(\mathbf{n} \cdot \mathbf{J})_2 = \frac{\sigma}{d}(V_2 - V_1)$$

to model a thin layer of a resistive material. The layer has the thickness d and the conductivity σ. This boundary condition is only available at the border between the parts in an assembly.

Axial Symmetry

Use the axial symmetry boundary condition on the symmetry axis $r = 0$ in axisymmetric models only. For a thorough discussion of this boundary condition, see "Axial Symmetry" on page 1 for the generalized electrostatic formulation.

Periodic Boundary Condition

The periodic boundary condition sets up a periodicity between the selected boundaries. See section "Periodic Boundary Conditions" on page 26 for more details on this boundary condition.

LINE SOURCES

In 3D line sources can be specified along the edges of the geometry.

Line Current Source

A line current source Q_{jl} can be applied to edges. This source represents electric current per unit length.

POINT SOURCES AND CONSTRAINTS

Point sources and constraints can be specified in 2D and 3D.

Point Current Source

A point current source Q_{j0} can be applied to points. This source represents an electric current flowing out of the point.

Point Constraint (Electric Potential)

The electric potential can be constrained to the value V_0 in a point using a constraint.

APPLICATION MODE VARIABLES

See the section "Conductive Media DC Application Mode" on page 5 of the *MEMS Module Reference Guide*.

Electrostatics Application Mode

You can use the *Electrostatics* application mode for 3D, 2D in-plane and 2D axisymmetric models.

Applications involving *electrostatics* include high-voltage apparatus, electronic devices, and capacitors. The statics means that the time rate of change is slow and that wavelengths are very large compared to the size of the domain of interest.

PDE FORMULATION

The *3D Electrostatics* application mode solves the equation

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

This equation assumes the constitutive relation $\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$. The corresponding equations for the constitutive relations $\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E}$ and $\mathbf{D} = \varepsilon_0 \varepsilon_r \mathbf{E} + \mathbf{D}_r$ can also be handled.

The *in-plane Electrostatics* application mode assumes a symmetry where the electric potential varies only in the x and y directions and is constant in the z direction. This

implies that the electric field $\mathbf{E}$ is tangential to the *xy*-plane. Given this symmetry, it solves the same equation as in the 3D case.

Use the *axisymmetric Electrostatics* application mode when the fields and the geometry are axially symmetric. In this case, the electric potential is constant in the φ direction, which implies that the electric field is tangential to the *rz*-plane.

Writing the equation in cylindrical coordinates and multiplying it by r to avoid singularities at $r = 0$, the equation becomes

$$-\begin{bmatrix}\frac{\partial}{\partial r}\\ \frac{\partial}{\partial z}\end{bmatrix}^T \cdot \left(r\varepsilon_0 \begin{bmatrix}\frac{\partial V}{\partial r}\\ \frac{\partial V}{\partial z}\end{bmatrix} - r\mathbf{P} \right) = r\rho$$

APPLICATION SCALAR VARIABLES

The application-specific scalar variable in this mode is given below.

PROPERTY	NAME	DEFAULT	DESCRIPTION
ε_0	epsilon0	$8.854187817 \cdot 10^{-12}$ F/m	Permittivity of vacuum

BOUNDARY CONDITIONS

The relevant interface condition at interfaces between different media for this application mode is

$$\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

In the absence of surface charges this is fulfilled by the natural boundary condition

$$\mathbf{n} \cdot [(\varepsilon_0 \nabla V - \mathbf{P})_1 - (\varepsilon_0 \nabla V - \mathbf{P})_2] = -\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

Electric Displacement

Use the electric displacement boundary condition

$$\mathbf{n} \cdot \mathbf{D} = \mathbf{n} \cdot \mathbf{D}_0$$

to specify the normal component of the electric displacement at a boundary.

Surface Charge

The surface charge boundary condition

$$-\mathbf{n} \cdot \mathbf{D} = \rho_s, \qquad \mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

makes it possible to specify the surface charge density at an exterior boundary or at the interior boundary between two media.

Zero Charge/Symmetry

The zero charge/symmetry boundary condition

$$\mathbf{n} \cdot \mathbf{D} = 0$$

specifies that the normal component of the electric displacement is zero.

You can also use this boundary condition at symmetry boundaries where the potential is known to be symmetric with respect to the boundary.

Electric Potential

The electric potential boundary condition

$$V = V_0$$

specifies the voltage at the boundary. Because you are solving for the potential, it is necessary to define its value at some boundary in the geometry for the potential to be fully determined.

Ground

The ground boundary condition

$$V = 0$$

is a special case of the previous one specifying a zero potential. You can also use this boundary condition at symmetry boundaries, where the potential is known to be antisymmetric with respect to the boundary.

Port

Use the port condition on the electrodes to calculate the capacitance. This condition forces the potential to one or zero depending on the settings (see the section "Lumped Parameters" on page 47).

Electric Shielding

The electric shielding boundary condition describes a thin layer of a dielectric medium that shields the electric field. See the section "Floating Potentials and Electric Shielding" on page 17 for an example.

Floating Potential

To set a floating potential with an integral constraint, use the boundary condition

$$\int_{\partial\Omega} \rho_s = Q_0$$

which sets the potential to a constant value on the boundary such that the total charge is equal to Q_0. See the section "Floating Potentials and Electric Shielding" on page 17 for an example.

Continuity

The continuity boundary condition

$$\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

specifies that the normal component of the electric displacement is continuous across the interior boundary.

Thin Low Permittivity Gap

Use the thin low permittivity gap condition

$$(\mathbf{n} \cdot \mathbf{D})_1 = \frac{\varepsilon}{d}(V_1 - V_2)$$

$$(\mathbf{n} \cdot \mathbf{D})_2 = \frac{\varepsilon}{d}(V_2 - V_1)$$

to model a thin gap of a material with a small permittivity compared to the adjacent domains. The gap has the thickness d and the relative permittivity ε_r. This boundary condition is only available at the border between parts in an assembly.

Axial Symmetry

Apply the axial symmetry boundary condition to the symmetry axis $r = 0$ in axisymmetric models.

Periodic Boundary Condition

The periodic boundary condition sets up a periodicity between the selected boundaries. See section "Periodic Boundary Conditions" on page 26 for more details on this boundary condition.

LINE SOURCES

In 3D line sources can be specified along the edges of the geometry.

Line Charge

A line charge Q_l can be applied along the edges. The source is interpreted as electric charge per unit length.

POINT SOURCES AND CONSTRAINTS

Point charges and point constraints are available in 2D and 3D.

Point Charge

A point charge Q_0 can be applied to points.

Point Constraint (Electric Potential)

The electric potential can be constrained to the value V_0 in a point.

APPLICATION MODE VARIABLES

See the section "The Electrostatics Application Mode" on page 7 in the *MEMS Module Reference Guide*.

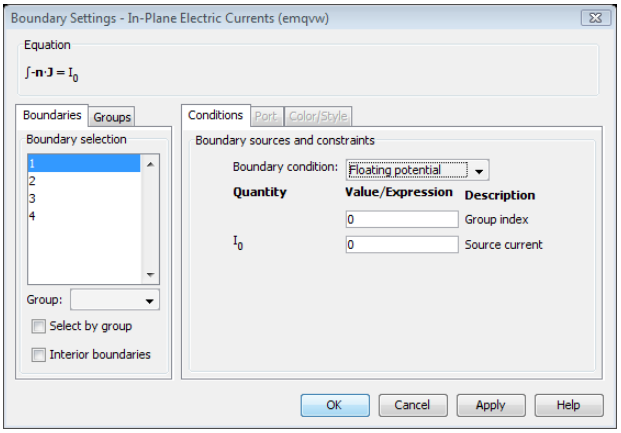
Floating Potentials and Electric Shielding

Floating Potentials

The floating potential boundary condition sets the potential on the boundary to a constant. The value of the constant is set so that the integral of the current density normal to the boundary is equal to the source current that you have specified

$$\int_{\partial\Omega} -\mathbf{n} \cdot \mathbf{J} = I_0$$

You can specify boundaries far apart to have the same potential by using the **Group index** edit field to assign these boundaries to a certain index, or boundaries adjacent to each other to have different floating potentials by specifying different group indexes. You enter the total current in the **Source current** edit field.



For the Electrostatics application mode the total charge replaces the total current, so the voltage is set so the integral of the charge density is equal to the charge specified in the **Charge** edit field.

The floating potential boundary condition is available for all application modes that solves for the electrostatic potential, which are:

- Conductive Media DC
- Electrostatics

It is also available for all analysis types.

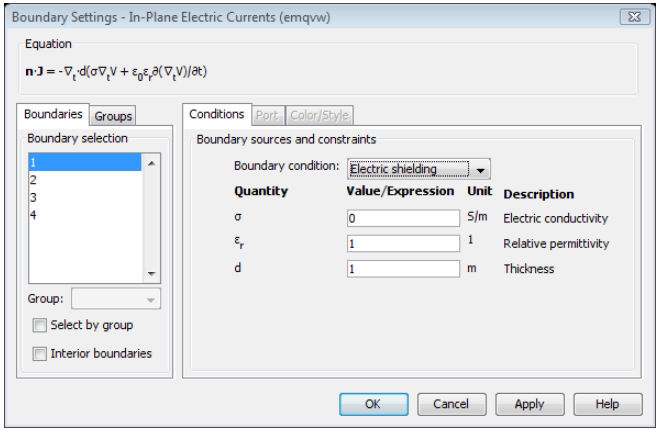
Electric Shielding

The electric shielding condition adds the same equation as in the subdomain on the boundary using tangential derivative variables. You can read more about these variables in the section "Modeling with PDEs on Boundaries, Edges, and Points" on page 294 in the *COMSOL Multiphysics Modeling Guide*. This is the equation used for the Conductive Media DC application mode.

$$-\nabla_t \cdot (\sigma d \nabla_t V) = 0$$

The variable d accounts for the thickness of the shield, but the solution is constant through the thickness. The conductivity that you enter here is the conductivity of the boundary. You can use this boundary condition when approximating a thin subdomain with a boundary to reduce the number of mesh elements.

For the Quasi-statics, Electric application mode, it is also possible to specify a dielectric constant.



The Electrostatics application mode only uses the relative permittivity material parameter. In addition, you can specify a surface charge density.

Example Model—Floating Potential

This is a tutorial model to show how to use the floating potential and electric shielding boundary conditions in the Conductive Media DC application mode. The analysis includes solving the same model changing between the two different boundary conditions and with and without weak constraints on the boundaries.

Model Definition

The modeling domain is a box filled with air containing an electrode. The sides of the box are insulated while the top has a potential and the bottom is grounded.

BOUNDARY CONDITIONS ON THE ELECTRODE

First use the electric shielding boundary condition on the electrode. Setting the conductivity of the electrode to the conductivity of a metal gives you an electrode with an almost constant potential. In this way you can set a constant potential without knowing the actual value of it.

Then set the boundary condition on the electrode to a floating potential. If the source current is zero this gives you a potential so that the flux over the boundary is zero. The integral is calculated with a coupling variable. You can keep the default solver settings.

Finally solve the model with a floating potential boundary condition and use weak constraints on the electrode. The reason to use the weak constraints is that this gives a more accurate computation of the current density normal to the boundary in the postprocessing stage. When using weak constraints you solve for the variable `lm1` (the Lagrange multiplier), which is equal to the normal flux on the boundary. The floating point boundary condition uses this variable to calculate the integral of the current flowing to the boundary. You cannot use the default AMG preconditioner when dealing with weak constraints, so therefore you need to select the Incomplete LU preconditioner instead. Note that the solution is exactly the same with and without weak constraints, it is access to the accurate flux that you get with weak constraints. For more information on weak constraints, see "Using Weak Constraints" on page 300 in the *COMSOL Multiphysics Modeling Guide*.

Results and Discussion

The differences between the three ways to model a floating electrode can be investigated by calculating the integral of the current flowing to the electrode. Ideally, this current should be zero. Using the electric shielding boundary condition, you get

a current of approximately 0.19 A. Using the floating potential boundary condition without weak constraints, you also get a current of 0.19 A, most of which is due to interpolation errors. Finally, when using weak constraints, you get a current of about 8.9 nA. The approach using weak constraints is clearly the best if you need access to the total flux. The electric shielding boundary condition does not strictly impose a fixed potential on the electrode as it allows for a small but finite tangential gradient in the potential which of course may be an advantage if you need to account for small resistive loss in the electrode. The advantage of not using weak constraints is that you can use the default solver settings.

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

1 In the **Model Navigator** select **3D** from the **Space dimension** list.

2 Open the **MEMS Module** folder, then select **Electrostatics>Conductive Media DC**.

3 Click **OK**.

GEOMETRY MODELING

1 Click the **Block** button. Set the **Base** to **Center** and the **Length** to 10 in all directions. When done, click **OK**.

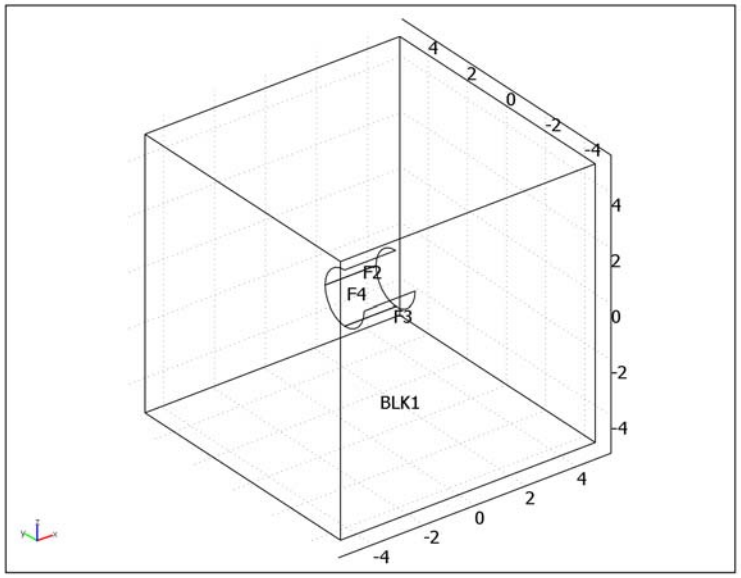
2 Click the **Cylinder** button. Set the **Style** to **Face** and enter the following settings. When done, click **OK**.

PROPERTY	VALUE
Radius	1
Height	2
Axis base point	-1, 0, 0
Axis direction vector	1, 0, 0

3 Click the **Split Object** button when the cylinder is selected.

4 Click the **Zoom Extents** button on the Main toolbar.

5 Select the face **F1** and delete it.



PHYSICS SETTINGS—ELECTRIC SHIELDING

Subdomain Settings

1 Open the **Subdomain Settings** dialog box.

2 Select Subdomain 1 and type `1` in the **σ (isotropic)** edit field for the electric conductivity.

3 Click **OK**.

Boundary Conditions

1 Open the **Boundary Settings** dialog box.

2 Select all boundaries and change boundary condition to **Electric insulation**.

3 Set an **Electric potential** boundary condition with $V_0 = 1$ on boundary 4.

4 Select Boundary 3 and set the boundary condition to **Ground**.

5 Select all the interior boundaries (Boundaries 6–8) and select the **Interior boundaries** check box.

6 Set the boundary condition to **Electric shielding**, the **Electric conductivity** to `5.99e7`, and the **Thickness** to `0.01`.

7 Click **OK**.

MESH GENERATION AND SOLUTION

Initialize the mesh and solve with the default settings.

POSTPROCESSING AND VISUALIZATION

Calculate the integral of the current flowing to the electrode.

1 Open the **Boundary Integration** dialog box from the **Postprocessing** menu.

2 Type `nJs_emdc` in the **Expression** field and click **Apply**.

The result appears in the message log at the bottom of the user interface. The current is approximately 0.19 A.

PHYSICS SETTINGS—FLOATING POTENTIAL

Boundary Conditions

1 Open the **Boundary Settings** dialog box.

2 Select Boundaries 6–8.

3 Set the boundary condition to **Floating potential**.

4 Click **OK**.

MESH GENERATION AND SOLUTION

Initialize the mesh and solve with the default settings.

POSTPROCESSING AND VISUALIZATION

Calculate the integral of the current flowing to the electrode.

1 Open the **Boundary Integration** dialog box from the **Postprocessing** menu.

2 Type `nJs_emdc` in the **Expression** field and click **Apply**.

The result appears in the message log at the bottom of the user interface. You should get a current of approximately 0.19 A. Note that this current calculation is not very accurate; the actual current is almost zero. It is the evaluation of the derivatives in the expression for `nJs_emdc` that is the source of the error.

PROPERTIES—WEAK CONSTRAINTS

1 From the **Physics** menu, choose **Properties** to open the **Application Model Properties** dialog box.

2 Select **On** from the **Weak constraints** list, then click **OK**.

PHYSICS SETTINGS—WEAK CONSTRAINTS

Boundary Conditions

1. Open the **Boundary Settings** dialog box.
2. Select Boundaries 6–8.
3. Click the **Weak Constr.** tab and make sure that **Use weak constraints** is selected.
4. Click **OK**.

MESH GENERATION

Click the **Initialize Mesh** button on the Main toolbar.

COMPUTING THE SOLUTION

1. Open the **Solver Parameters** dialog box.
2. Change the **Linear system solver** to **GMRES**.
3. Make sure that **Automatic** or **Nonsymmetric** is selected in the **Matrix symmetry** list.
4. Change the **Preconditioner** to **Incomplete LU**.
5. Click **OK**.
6. Click the **Solve** button on the Main toolbar to compute the solution.

If you selected **Automatic** in the **Matrix symmetry** list, a warning message appears. You can ignore this warning because it only tells you that the solver settings cannot take advantage of the matrix symmetry, so the solver uses the nonsymmetric setting instead.

POSTPROCESSING AND VISUALIZATION

Calculate the integral of the current flowing to the electrode.

1. Open the **Boundary Integration** dialog box from the **Postprocessing** menu.
2. Select Boundaries 6–8.
3. Type `nJs_emdc` in the **Expression** field, then click **OK**.

The result appears in the message log at the bottom of the user interface. You should get a current of approximately $8.9 \cdot 10^{-9}$ A. This is also the true value of the total current in the previous solution step without weak constraints.

To plot the potential on the electrode as a boundary plot, suppress the display of some of the outer boundaries:

1. In the **Options** menu, select **Suppress>Suppress Boundaries**.
2. Select Boundaries 6–8, then click **Apply**.

3 Click **Invert Suppression**, then click **OK**.

4 Open the **Plot Parameters** dialog box from the **Postprocessing** menu.

5 On the **Slice** page, set the **Number of levels** in the *x* direction to 1.

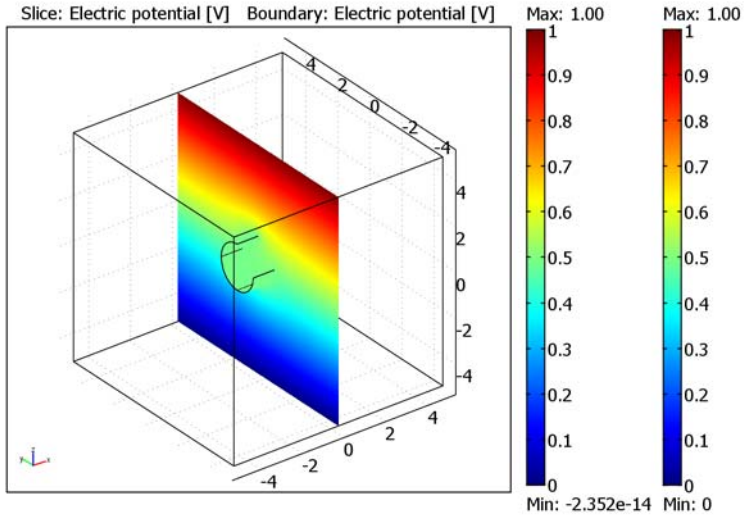
6 Click the **Boundary** tab.

7 Select the **Boundary plot** check box.

8 Click the **Range** button and clear the **Auto** check box.

9 Type 0 in the **Min** edit field and type 1 in the **Max** edit field.

10 Click **OK** twice.



Periodic Boundary Conditions

The section "Using Periodic Boundary Conditions" on page 245 in the *COMSOL Multiphysics User's Guide* presents a general description on how to define periodic Boundary conditions. The MEMS Module has an automatic **Periodic condition** accessible from the **Boundary Settings** dialog box, so it is not necessary to use the **Periodic Boundary Conditions** dialog box. Use the latter dialog box for special cases when you need full control of the periodic condition. The automatic periodic condition can identify simple mappings on plane groups of source and destination boundaries with equal shape. The destination can also be rotated with respect to the source.

User Interface for Periodic Conditions

You specify the periodic condition in the **Physics>Boundary Settings** dialog box. Select the boundaries that define one periodic condition and choose **Periodic condition** from the **Boundary condition** list. The boundaries can consist of one or more source boundaries plus one or more destination boundaries. The combined cross section of all source boundaries must be equal in shape to the combined cross section of all destination boundaries. If you want several periodic conditions with different orientations, separate them with a group index that you enter in the **Group index** edit field. If you, for example, want to set periodic boundaries on all sides of a cube, you must use three indexes to separate the three orientations of the periodic boundaries.

You select the type of periodic condition from the **Type of periodicity** list, where there are two available choices:

- **Continuity**—The solution variables are equal on the source and destination.
- **Antiperiodicity**—The solution variables on the destination have an opposite sign compared to the variables on the source.

The boundary with the lowest number becomes the source by default. It is possible to change this order by selecting the **Change source and destination order** check box.

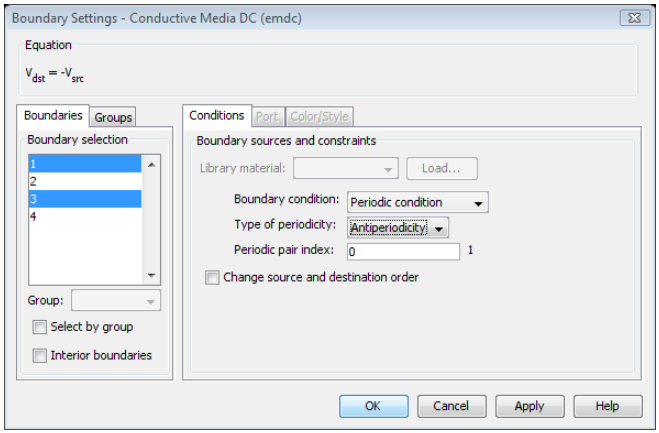


Figure 7-1: The Boundary Settings dialog box with sector antisymmetry selected for Pair 1.

Infinite Elements

Many environments that are modeled with finite elements are unbounded or open, meaning that the electromagnetic fields extend toward infinity. The easiest approach to modeling an unbounded domain is to extend the simulation domain "far enough" that the influence of the terminating boundary conditions at the far end becomes negligible. This approach can create unnecessary mesh elements and make the geometry difficult to mesh due to large differences between the largest and smallest object.

Another approach is to use *infinite elements*. There are many implementations of infinite elements available, and the one used in the MEMS Module is often referred to as *mapped infinite elements* (see Ref. 1). This implementation maps the model coordinates from the local, finite-sized domain, to a stretched domain. The inner boundary of this stretched domain is coincident with the local domain, but at the exterior boundary the coordinates are scaled toward infinity.

$$t' = t_0 \frac{\delta t}{t_0 + \delta t - t}$$

The inner coordinate, t_0, and the width of the infinite element region, δt, are input parameters for each region. The software uses default values for these properties for geometries that are Cartesian, cylindrical, or spherical. However, these default parameters might not work well for complex geometries, so it might be necessary to define other parameters. The following figures show typical examples of infinite element regions that work nicely for each of the infinite element types. These types are:

- Stretching in Cartesian coordinate directions, labeled **Cartesian**.
- Stretching in cylindrical directions, labeled **Cylindrical**.
- Stretching in spherical direction, labeled **Spherical**.
- User-defined coordinate transform for general infinite elements.

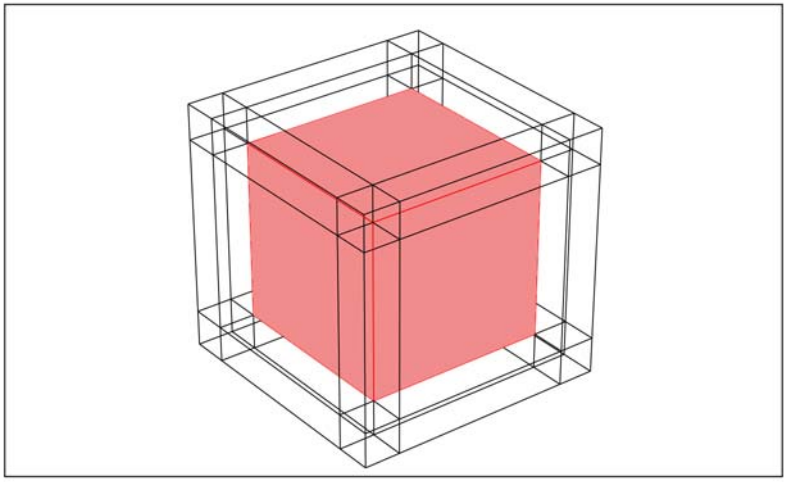


Figure 7-2: A square surrounded by typical infinite-element regions of Cartesian type.

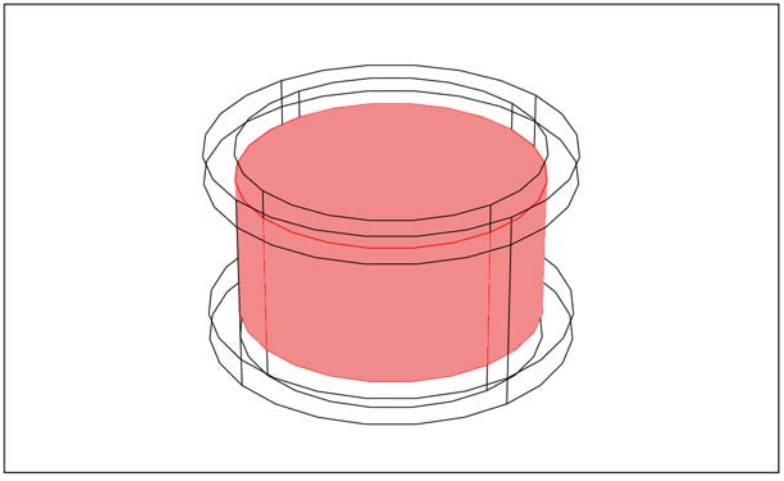


Figure 7-3: A cylinder surrounded by typical cylindrical infinite-element regions.

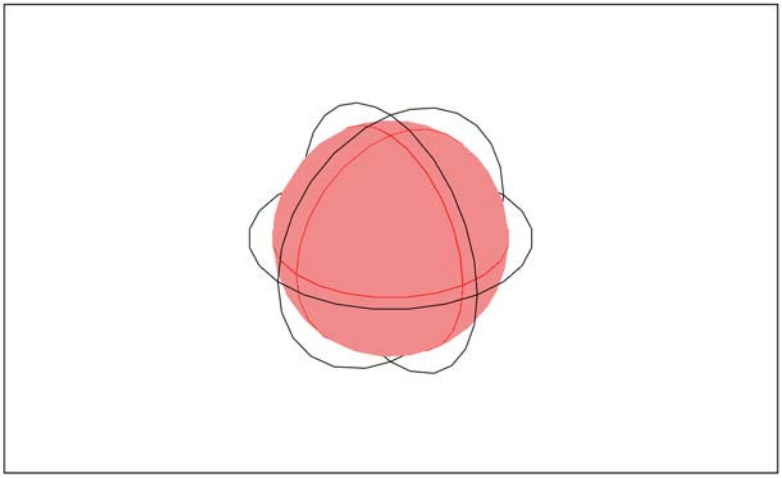
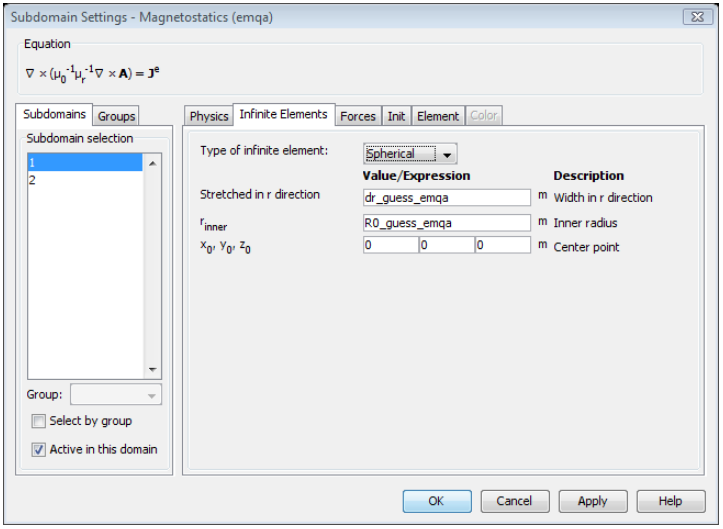
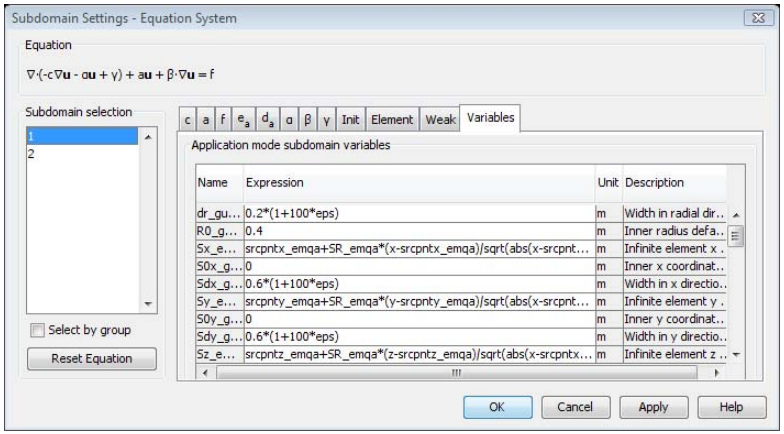


Figure 7-4: A sphere surrounded by a typical spherical infinite-element region.

If you use other shapes for the infinite element regions not similar to the shapes shown in the previous figures, it might be necessary to define the infinite-element parameters manually. The software stores the default parameters in variables with the naming convention *param*`_guess_`*suffix*, where *param* is the name of the parameter, and *suffix* is the application mode suffix.



You can check their values by choosing **Equation System>Subdomain Settings** from the **Physics** menu. Click the **Variables** tab and look for variables with `_guess_` in the name.



1

Click the **Solve** button to start solving.

This model uses an efficient approach for handling the gauge fixing with the SOR gauge smoothers for the geometric multigrid preconditioner; see "Solver Settings for Numerical Gauge Fixing in Magnetostatics" on page 92 in the *AC/DC Module User's Guide* for more information.

2 In the **x-axis data** area, click first the lower option button then the **Expression** button.

3 In the **X-Axis Data** dialog box, type z in the **Expression** edit field, then click **OK**.

Known Issues When Modeling Using Infinite Elements

When modeling with infinite elements you should be aware of the following:

- The expressions resulting from the stretching get quite complicated for spherical and cylindrical infinite elements in 3D. This increases the time for the assembly stage in the solution process. After the assembly, the computation time and memory consumption is comparable to a problem without infinite elements. The number of iterations for iterative solvers might increase if the infinite element regions have a coarse mesh.
- Infinite element regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the

infinite element parameter to give erroneous result. Enter the parameter values manually if you find that this is the case.

- The infinite element region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an infinite element region.

Reference

1. O.C. Zienkiewicz, C. Emson, and P. Bettess, "A Novel Boundary Infinite Element," *Int. J. Num. Meth. Engrg*, vol. 19(3), pp. 393–404, 1983.

Force and Torque Computations

Computing Electromagnetic Forces and Torques

To compute electromagnetic forces and torques in COMSOL Multiphysics, two methods are available. This section describes one of them, which uses Maxwell's stress tensor. There are also two functions, `cemforce` and `cemtorque`, which you can use when running COMSOL Multiphysics together with COMSOL Script or MATLAB. These functions use the method of virtual displacement to calculate the force and torque.

Force and torque calculations using Maxwell's stress tensor are available in the application modes for electrostatics, magnetostatics, and quasi-statics. In electrostatics the force is calculated by integrating

$$\mathbf{n}_1 T_2 = -\frac{1}{2}\mathbf{n}_1(\mathbf{E}\cdot\mathbf{D}) + (\mathbf{n}_1\cdot\mathbf{E})\mathbf{D}^T \tag{7-1}$$

on the surface of the object that the force acts on. In magnetostatics and quasi-statics the expression

$$\mathbf{n}_1 T_2 = -\frac{1}{2}\mathbf{n}_1(\mathbf{H}\cdot\mathbf{B}) + (\mathbf{n}_1\cdot\mathbf{H})\mathbf{B}^T \tag{7-2}$$

is integrated on the surface to obtain the force. $\mathbf{E}$ is the electric field, $\mathbf{D}$ the electric displacement, $\mathbf{H}$ the magnetic field, $\mathbf{B}$ the magnetic flux density, and $\mathbf{n}_1$ the outward normal from the object.

The naming syntax for the surface variables is

<given name>`_nT`*<independent variable name>_<application mode name>*

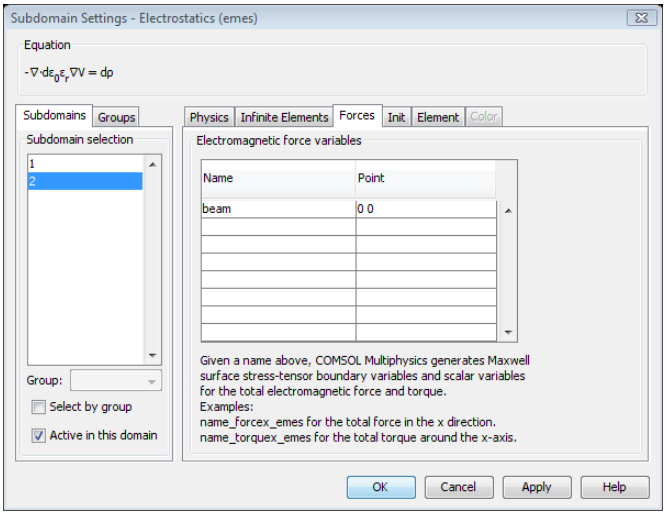
For example, `iron_nTx_emqa`. Similarly, the variables for the total force are named as

<given name>`_force`*<independent variable name>_<application mode name>*

For example, `iron_forcex_emqa`.

DEFINING FORCE AND TORQUE VARIABLES

To define force variables, use the **Subdomain Settings** dialog box. On the **Forces** tab there is a table where you define the variables. Select the two subdomains representing the structure and enter a name in the **Name** column, for example, `beam`.



This generates a set of variables. The variables `beam_nTx_emes` and `beam_nTy_emes` are defined on the exterior boundary of the piece of iron. The suffix `_emes` corresponds to the Electrostatics application mode. These are the components in the *x* and *y* directions, respectively, of the contraction of the Maxwell stress tensor given in Equation 7-1. Together, they represent the surface force density. In addition, the variable definition generates two force scalar variables, `beam_forcex_emes` and `beam_forcey_emes`, which are the total force components on the iron in the *x* and *y* directions, respectively. Finally, a torque variable is generated: `beam_torque_z`, which represents the torque around the *z*-axis going through the point defined in the **Point** column. For 3D models, there is also an **Axis** column. Here you define an axis direction. The torque computation then finds the total torque around this axis.

NAME	DIMENSION	GENERATED VARIABLE	DESCRIPTION
`beam`	all	beam_nTx_emes	Surface force density in the x direction
	all	beam_nTy_emes	Surface force density in the y direction
	all	beam_forcex_emes	Total force in the x direction
	all	beam_forcey_emes	Total force in the y direction
	2D, 3D	beam_torquez_emes	Total torque in the z direction, around the specified point

NAME	DIMENSION	GENERATED VARIABLE	DESCRIPTION
	3D	beam_torquex_emes	Total torque in the x direction, around the specified point
	3D	beam_torquey_emes	Total torque in the y direction, around the specified point
	3D	beam_torqueax_emes	Total torque around the specified axis and point

The table on the **Forces** tab makes it possible to define multiple variables. The set of subdomains where each name appears in the table specifies where the force should be calculated. The surface variables exist on the exterior boundaries of the subdomains where the variable name is given.

The naming syntax for the surface variables is

*<given name>*_nT*<independent variable name>*_*<application mode name>*

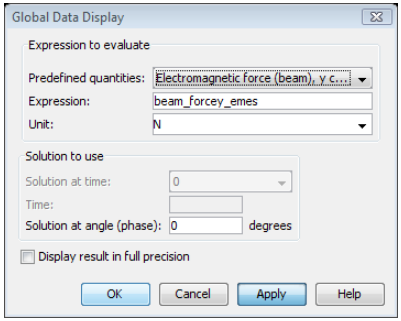
For example, `beam_nTx_emes`. Similarly, the variables for the total force are named as

*<given name>*_force*<independent variable name>*_*<application mode name>*

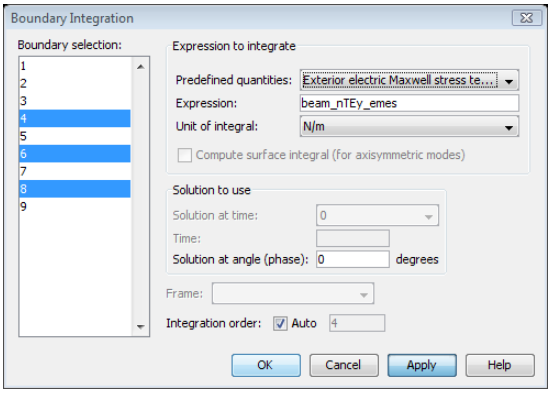
For example, `beam_forcex_emes`.

The torque variables are constructed in a similar manner. Note that for 2D only one nonzero torque component exists, because all the forces are in the plane. For models using axial symmetry no nonzero components exist, so no torque variables are computed, and the table only contain the **Name** column.

To display the forces on the structure (named as `beam`) you can use the **Global Data Display** dialog box. Enter the expression `beam_forcey_emes` and click **OK** to obtain the force in the *x* direction in the message log at the bottom of the main window. Because the variable `beam_forcey_emes` is a scalar it is possible to evaluate it in any point.



The result is equal to the boundary integral of `beam_nTy_emes` on the exterior boundary of the structure `beam`. You can verify this using the **Boundary Integration** dialog box to make this integration.



Lumped Parameters

Lumped parameters are matrices describing electromagnetic properties such as resistance, capacitance, and inductance. In the time-harmonic case the lumped parameter matrix is either an impedance matrix or an admittance matrix. In a static calculation you only get the resistive, capacitive, or inductive part of the lumped parameter matrix.

Calculating Lumped Parameters with Ohm's Law

To calculate the lumped parameters, there must be at least two electrodes in the system, where one is grounded. You can force either a voltage or a current on the electrodes. After the simulation you can extract the other property or you can extract the energy and use it when calculating the lumped parameter.

FORCED VOLTAGE

If voltages are applied between the electrodes, the extracted currents represent elements in the admittance matrix, **Y**. This matrix determines the relation between the applied voltages and the corresponding currents with the formula

$$\begin{bmatrix} I_1 \\ I_2 \\ I_4 \\ I_4 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$

so when V_1 is one (in some unit system) and all other voltages are zero, the vector I is equal to the first column of **Y**.

FIXED CURRENT

It might be necessary to calculate the **Z**-matrix in a more direct way. Similar to the **Y** calculation, the **Z** calculation can be done by forcing the current through one electrode at the time to one while the others are set to zero, and then extracting the voltages on all electrodes. Then, the columns of the impedance matrix are the voltage values:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} \\ Z_{21} & Z_{22} & Z_{23} & Z_{24} \\ Z_{31} & Z_{32} & Z_{33} & Z_{34} \\ Z_{41} & Z_{42} & Z_{43} & Z_{44} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}$$

FIXED CURRENT DENSITY

An alternative approach for calculating the **Z**-matrix is to force the current density to a uniform value across each electrode. The total current through the electrode is the area times the current density, which is selected to be one or zero. The voltage can then vary across the port, so averaging is necessary. With this approach the **Z**-matrix is calculated with the formula

$$\begin{bmatrix} \frac{1}{A_1}\int_{A_1} V_1 dA \\ \frac{1}{A_2}\int_{A_2} V_2 dA \\ \frac{1}{A_3}\int_{A_3} V_3 dA \\ \frac{1}{A_4}\int_{A_4} V_4 dA \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} \\ Z_{21} & Z_{22} & Z_{23} & Z_{24} \\ Z_{31} & Z_{32} & Z_{33} & Z_{34} \\ Z_{41} & Z_{42} & Z_{43} & Z_{44} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}.$$

Here A_n represents the area and I_n the normal current density of port n. The current density is just the current divided by the area of that electrode.

$$J_n = \frac{I_n}{A_n}$$

Calculating Lumped Parameters Using the Energy Method

When using this method the potential or the current is one on one or two ports at a time and you extract the energy density integrated over the whole geometry. The following formulas show how to calculate the capacitance matrix from the integral of the electric energy density.

$$C_{ii} = 2\int_{\Omega} W_e d\Omega \qquad V_j = \begin{cases} 0 & j \neq i \\ 1 & j = i \end{cases}$$

$$C_{ij} = \int_{\Omega} W_e d\Omega - \frac{1}{2}(C_{ii} + C_{jj}) \qquad V_k = \begin{cases} 0 & k \neq i, j \\ 1 & k = i, j \end{cases}$$

You can calculate the inductance matrix in the same way from the magnetic energy density:

$$L_{ii} = 2\int_{\Omega} W_m d\Omega \qquad I_j = \begin{cases} 0 & j \neq i \\ 1 & j = i \end{cases}$$

$$L_{ij} = \int_{\Omega} W_m d\Omega - \frac{1}{2}(L_{ii} + L_{jj}) \qquad I_k = \begin{cases} 0 & k \neq i, j \\ 1 & k = i, j \end{cases}$$

Lumped Parameters in the MEMS Module

To study lumped parameters you use the Port boundary condition for each electrode. This boundary condition is available in the Conductive Media DC application mode, and the Electrostatics application mode.

The MEMS Module includes the two different approaches explained above to calculate the lumped parameters. The static Electrostatics application mode uses the energy method, and the method using Ohm's law is used in the Conductive Media DC application mode.

THE PORT PAGE

To specify the properties for the port, click the **Port** tab. Each port must have a unique port number. On the ports where you want to force the value of the input parameter to be one, select the **Use port as inport** check box. In some application modes you can choose which property (for example, a forced voltage or a fixed current) to use as input from the **Input property** list.

ACCURACY

To get a good accuracy when calculating the total current over the boundary you need to use weak constraints. This is necessary for the forced voltage input property.

When the current density is fixed the weak constraints are unnecessary, which makes this the preferred method in large problems where you need iterative solvers. The only

requirement for this method is that the port has a small variation in potential across its surface, which generally is the case for metal electrodes.

The fixed current property performs a coupling that guarantees that the total current is equal to one, although you cannot verify this without adding weak constraints. The result is the same with and without weak constraints, but a model without weak constraints is beneficial for iterative solvers. The fine tuning of the iterative solver settings is slightly more complicated than the fixed current density input property.

VARIABLES

Depending on which application mode you are working with and what input property you choose you get different postprocessing variables.

APPLICATION MODE	FORCED VOLTAGE		FIXED CURRENT OR CURRENT DENSITY	
Conductive Media DC	`Gij`	Conductance matrix element (i, j)	`Rij`	Resistance matrix element (i, j)
Electrostatics	`Cii`	Capacitance matrix element (i, i)	-	-
Electrostatics	`intWeij`	Integrated energy input ports i and j	-	-

In an application mode using the energy method, i and j are the input port numbers. If there is only one input port the software calculates the diagonal matrix element C_{ii}. In the other application modes j is the input port number and i is the port number. The software generates the variables for all port numbers for the input port in use.

If you, for example, in the Conductive Media DC application mode have two ports, Port 1 and Port 2, and use Port 1 as an input port you get two variables.

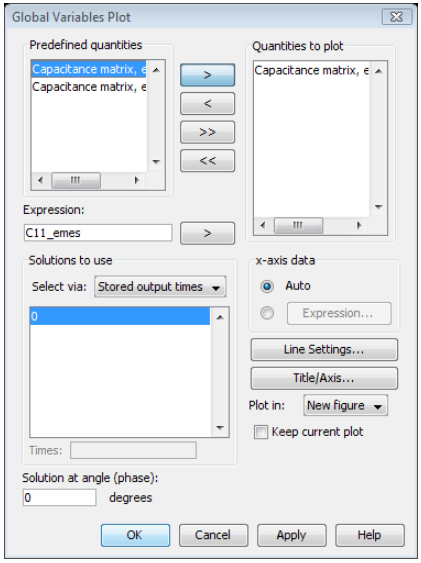
- `G11_emdc` and `G21_emdc` if voltage is the input property
- `R11_emdc` and `R21_emdc` if current or current density is the input property.

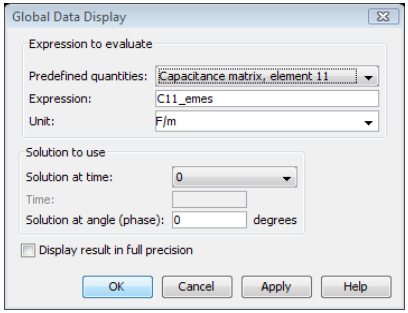
If you have two ports in the Electrostatics application mode, Port 1 and Port 2, and Port 1 is an input port, the software generates the variable `C11_emes` and `intWe11_emes`. If you use both Port 1 and Port 2 as input ports, the software generates the variables `intWe12_emes` and `intWe21_emes` instead.

POSTPROCESSING

The lumped parameters are defined as global variables, so you can access them from the **Postprocessing** menu, under either of the menu options **Global Plot Variables** or **Data Display>Global**. Use the **Global Plot Variables** dialog box if you want to plot one or

several lumped parameters as a function of a swept parameter. In the **Global Data Display** dialog box, you can get the value of a lumped parameter (real or complex) printed at the bottom of the COMSOL Multiphysics window. The defined lumped parameters are available from the **Predefined quantities** list under the application mode to which they belong.





You have access to global variables at any level in a geometry, and the lumped parameters are also available from the **Predefined quantities** list in the **Point Evaluation** and **Domain Plot Parameters** dialog boxes, which you open from the **Postprocessing** menu. For the **Domain Plot Parameters** dialog box you access the lumped parameters on the **Point** page.

Note: Because the lumped parameters are global variables, a conflict in variable names can occur if two application modes of the same type have the same suffix. Always make sure that the application mode suffixes are different when you add extra application modes in other geometries.

PHYSICS SETTINGS

Subdomain Settings

1. Open the **Subdomain Settings** dialog box.
2. Select Subdomain 1 and click the **Electric Parameters** tab.
3. Set the relative permittivity to `3.25`, then click **OK**.

Boundary Conditions

1. Open the **Boundary Settings** dialog box.
2. Select Boundaries 1 and 8 and click the **Electric Parameters** tab.
3. Set the boundary condition to **Electric insulations**.
4. Set the boundary condition on Boundaries 4–7 to **Port**.
5. Click the **Port** tab and select the **Use port as input** check box.
6. Click **OK**.

Application Mode Properties

1. Open the **Application Mode Properties** dialog box by selecting **Properties** from the **Physics** menu.
2. Set the **Weak constraints** to **Ideal**.
3. Click **OK**.

COMPUTING THE SOLUTION

Click the **Solve** button on the Main toolbar.

POSTPROCESSING

1. Select **Data Display>Global** from the **Postprocessing** menu.
2. Type `imag(Y11_emqap)/omega_emqap` in the **Expression to evaluate** edit field and click **OK** to see the capacitance in the message log.

8

Microfluidics Application Modes

The MEMS Module provides the following application modes for studying fluidic problems on the microscale:

- The General Laminar Flow application mode
- The Incompressible Navier-Stokes application mode
- The Weakly Compressible Navier-Stokes application mode
- The Stokes Flow application mode
- The Weakly Compressible Stokes Flow application mode
- The Level Set Two-Phase Flow application mode
- The Convection and Diffusion application mode
- The Electrokinetic Flow application mode

The first five application modes in this list share many features and, apart from slight differences in their user interfaces and variables, are basically the same. The main difference is in the equations each one uses. They all model flow in the laminar regime, but the simpler forms of the Stokes flow and weakly compressible Stokes flow equations mean that the corresponding application modes are applicable only for very low velocities. Another difference is that while the Stokes flow and incompressible Navier-Stokes equations assume constant density, the Weakly

Compressible Stokes flow and the Weakly Compressible Navier-Stokes application modes allow changes in density. The General Laminar Flow application mode is general purpose and can take any of the four equations.

The Level Set Two-Phase Flow application mode build on one of the flow application modes just described. But it has an additional degree of freedom, that solves for the separating boundary between the two different fluids.

The last two application modes do not model flow as such, but they model transport of species within the fluid.

The following sections describe all eight listed applications modes in detail. This includes the description theory, equations, and boundary conditions for each application mode, as well as the descriptions of their user interfaces.

In addition to the application modes listed above, the MEMS Module provides various predefined multiphysics couplings for microfluidics modeling. These predefined couplings combine one of the application modes listed above to another application mode that affects or is affected by the flow field. For more information about these application modes, see the final section in this chapter, "Predefined Multiphysics Couplings" on page 305.

Laminar Flow Theory

The flow application modes for microfluidics model laminar flows in the microscale. They are based on four different equations for computing the solution: the incompressible Navier-Stokes, Weakly Compressible Navier-Stokes, Stokes Flow, and Weakly Compressible Stokes Flow equations. All these application mode share the same boundary conditions. This chapter describes the theory background for these application modes.

In addition, the MEMS Module includes the Convection and Diffusion application mode (for convection and diffusion of species) and the Level Set Two-Phase Flow application mode (for modeling of separated two-phase flow using a level set method). The theory background for those application modes is given in other specific chapters below.

The units for the physical properties in this chapter are SI unit, but you can use any units system in COMSOL Multiphysics; the user interface then displays the units in the corresponding unit system

PDE Formulations

THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

The Navier-Stokes equations describe flow in viscous fluids with momentum balances for each component of the momentum vector in all spatial dimensions. They also assume that the density and viscosity of the modeled fluid are constant, which gives rise to a continuity condition.

The Incompressible Navier-Stokes application mode in COMSOL Multiphysics is somewhat more general than this and can account for arbitrary variations in viscosity and small variations in density, for example, through the Boussinesq approximation.

The Navier-Stokes equations with the total stress tensor formulation are

$$\rho\frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot [-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{F}$$
$$\nabla \cdot \mathbf{u} = 0 \tag{8-1}$$

where

- ρ is the fluid's density (kg/m^3)

- $\mathbf{u}$ represents the velocity vector (m/s)
- p equals the pressure (Pa)
- η denotes the dynamic viscosity (Pa·s)
- $\mathbf{F}$ is a body force term (N/m^3)
- $\mathbf{I}$ is the identity matrix

WEAKLY COMPRESSIBLE NAVIER-STOKES EQUATIONS

The weakly compressible Navier-Stokes equations are suitable where small density variations arise due to temperature differences. They are not suitable for high velocities or where fluid compression or expansion results in substantial internal work or heat effects.

These equations are almost identical to the incompressible Navier-Stokes equations except for some extra terms. The momentum balance includes an extra contribution, $(2\eta/3 - \kappa)(\nabla \cdot \mathbf{u})\mathbf{I}$, in the viscous-stress tensor. This contribution is zero in the incompressible Navier-Stokes equations because the incompressibility assumption states that $\nabla \cdot \mathbf{u} = 0$.

This application mode also modifies the equation of continuity equation to take into account changes in the fluid density, ρ.

The weakly compressible Navier-Stokes equations in the *total stress tensor* formulation are

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho\mathbf{u} \cdot \nabla\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - (2\eta/3 - \kappa_{dv})(\nabla \cdot \mathbf{u})\mathbf{I}] + \mathbf{F}$$
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0 \qquad (8\text{-}2)$$

where

- ρ is the density (kg/m^3)
- $\mathbf{u}$ represents the velocity field (m/s)
- p denotes the pressure (Pa)
- η equals the dynamic viscosity (Pa·s)
- κ_{dv} is the dilatational viscosity (Pa·s)
- $\mathbf{F}$ is a volume force field such as gravity (kg/m^3)
- $\mathbf{I}$ is the identity matrix

You can define the density as a function of any of the other dependent variables. Typically you couple this application mode to a heat equation where the density is a function of the temperature. COMSOL Multiphysics then solves the coupled equations simultaneously.

STOKES FLOW EQUATIONS

The Stokes flow equations are almost the same as the Navier-Stokes equations with the exception that they assume the inertial term $\rho\mathbf{u}\cdot\nabla\mathbf{u}$ is zero. Therefore these equations describe flow whose Reynolds number is very low and where the inertial forces in the fluid are very small. Otherwise you can apply these equations to the same problems as the Navier-Stokes equations, but the system of equations is almost linear.

The Stokes flow equations in the *total stress tensor* formulation are

$$\rho\frac{\partial\mathbf{u}}{\partial t} - \nabla\cdot[-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] = \mathbf{F}$$
$$\nabla\cdot\mathbf{u} = 0 \tag{8-3}$$

where

- ρ is the fluid's density (kg/m^3)
- $\mathbf{u}$ represents the velocity vector (m/s)
- p equals the pressure (Pa)
- η denotes the dynamic viscosity (Pa·s)
- $\mathbf{F}$ is a body force term (N/m^3)
- $\mathbf{I}$ is the identity matrix

WEAKLY COMPRESSIBLE STOKES FLOW EQUATIONS

The weakly compressible Stokes flow equations are almost identical to the general non-isothermal flow equations except that the inertial term $\rho\mathbf{u}\cdot\nabla\mathbf{u}$ is zero. Therefore these equations describe flow whose Reynolds number is very low and where the inertial forces in the fluid are very small. Otherwise you can apply these equations to same problems as the non-isothermal flow equations, but the system of equations is less nonlinear.

The weakly compressible Stokes flow equations using the *total stress tensor* formulation are

$$\rho\frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot [-p\mathbf{I} + \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - (2\eta/3 - \kappa_{dv})(\nabla \cdot \mathbf{u})\mathbf{I}] + \mathbf{F}$$
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0 \qquad (8\text{-}4)$$

where

- ρ is the density (kg/m^3)
- $\mathbf{u}$ represents the velocity field (m/s)
- p denotes the pressure (Pa)
- η equals the dynamic viscosity (Pa·s)
- κ_{dv} is the dilatational viscosity (Pa·s)
- $\mathbf{F}$ is a volume force field such as gravity (kg/m^3)
- $\mathbf{I}$ is the identity matrix

BOUNDARY FORCES

For the weakly compressible equations, the *viscous stress tensor*, τ, is

$$\tau = \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - (2\eta/3 - \kappa)(\nabla \cdot \mathbf{u})\mathbf{I}.$$

In addition to the stress due to viscous forces, a fluid can also have internal stresses due to the fluid pressure, p. The *total stress tensor*, σ, is

$$\sigma = -p\mathbf{I} + \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - (2\eta/3 - \kappa)(\nabla \cdot \mathbf{u})\mathbf{I}\,,$$

where $\mathbf{I}$ is the identity matrix or unit diagonal matrix.

For the incompressible equations, the stress tensors are simpler due the incompressibility condition $\nabla \cdot \mathbf{u} = 0$. Thus, for these equations, the *viscous stress tensor*, τ, is defined as

$$\tau = \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\,,$$

and the *total stress tensor*, σ, is

$$\sigma = -p\mathbf{I} + \eta\,(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\,.$$

From these stress tensors you can define the *viscous boundary force* per unit area, $\mathbf{K}$, as

$$\mathbf{K} = \tau\mathbf{n}\,,$$

and the *total boundary force* per unit area, $\mathbf{T}$, as

$$\mathbf{T} = \sigma\mathbf{n}$$

where $\mathbf{n}$ is the outward normal vector on the boundary.

SHALLOW CHANNEL APPROXIMATION

In microfluidics applications, channels often have an almost rectangular cross section where the thickness is much less than the channel width. For this type of geometry, simple 2D models often fail to give correct results because they exclude the boundaries that have greater effect on the flow. In 2D, however, you can approximate the effect of these boundaries by adding a drag term as a volume force to any of the relationships in Equations 8-1 to 8-4. The form of this term is

$$\mathbf{F}_\eta = -12\frac{\eta\mathbf{u}}{h^2} \tag{8-5}$$

where η is the fluid's dynamic viscosity (kg/(m·s)), $\mathbf{u}$ is its velocity field (m/s), and h is the channel thickness (m).

This term represents the resistance that the parallel boundaries place on the flow. However, they do not account for any changes in velocity due to variations in the cross-sectional area when the channel thickness differs.

Boundary Conditions

To fully define the flow problem you need to define boundary conditions for the flow. For easier usability, the available boundary conditions have been divided into six groups or condition types: **Wall**, **Inlet**, **Outlet**, **Symmetry**, **Open boundary**, and **Stress**.

The following chapters describes these conditions one by one.

BOUNDARY TYPE: WALL

These boundary conditions describe the existence of a solid wall.

No Slip

Boundary conditions that sets the fluid's velocity to zero at the boundary,

$$\mathbf{u} = \mathbf{0}.$$

Moving/Leaking Wall

If the wall moves, so must the fluid. Hence, this boundary condition prescribes

$$\mathbf{u} = \mathbf{u}_w$$

Note that setting this boundary condition does not automatically cause the associated wall to move. The section The Moving Mesh Application Mode on page 401 of the COMSOL Multiphysics Modeling Guide describes how to set up a model with moving boundaries.

You can also use the Moving/perforated wall boundary condition to simulate a wall where fluid is leaking into or leaving through a perforated wall.

Sliding Wall

If you use this boundary condition, the wall is assumed to behave like a conveyor belt, that is, that the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

In two space dimensions (2D), the tangential direction is unambiguously defined by the direction of the boundary. However, the situation becomes more complicated in 3D. For this reason, this boundary condition has slightly different definitions in the different space dimensions.

2D and Axial Symmetry The velocity is given as a scalar U_w and the condition prescribes

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{u} \cdot \mathbf{t} = U_w$$

where $\mathbf{t} = (-n_y, n_x)$ for 2D and $\mathbf{t} = (-n_z, n_r)$ for axial symmetry.

3D The velocity is set equal to a given vector $\mathbf{u}_w$ projected onto the boundary plane:

$$\mathbf{u} = \mathbf{u}_w - (\mathbf{n} \cdot \mathbf{u}_w)\mathbf{n}$$

Slip

The Slip condition sets the normal component of the velocity to zero

$$\mathbf{n} \cdot \mathbf{u} = 0.$$

It also assumes that the tangential component of the viscous force is zero

$$\mathbf{t} \cdot (-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n}) = 0.$$

Electroosmotic Velocity

Most solid surfaces acquire a surface charge when brought into contact with an electrolyte. In response to the spontaneously formed surface charge, a charged solution forms close to the liquid-solid interface. This is known as an electric double layer. If you apply an electric field to the fluid, this very narrow layer starts to move along the boundary.

It is possible to model the fluid's velocity near the boundary with the Helmholtz-Smoluchowski relationship between electroosmotic velocity and applied electric field. Thus, electroosmotic velocity is

$$u = \mu_{eo}\mathbf{E}_t = -\varepsilon_r\varepsilon_0\frac{\zeta}{\eta}\mathbf{E}_t \tag{8-6}$$

where

- μ_{eo} is electroosmotic mobility ($m^2/(s{\cdot}V)$)
- $\mathbf{E}_t$ equals the electric field in the fluid (V/m) tangential to the wall.
- ε_r represents the fluid's relative permittivity
- ε_0 denotes the permittivity of the free space (F/m)
- ζ is the fluid's zeta potential (V)
- η equals the fluid's dynamic viscosity (Pa·s)

Slip Velocity

In the microscale, flow at a boundary is seldom strictly no slip or slip. Instead, the boundary condition is something in between, and there is a slip velocity at the boundary. Two phenomena account for this velocity: the noncontinuum effect in viscosity and flow induced by a thermal gradient along the boundary.

The following equation relates the viscosity-induced jump in tangential velocity to the tangential shear stress along the boundary:

$$\Delta u = \frac{1}{\beta}\tau_{n,t}\,.$$

For gaseous fluids, the coefficient β equals

$$\beta = \frac{\eta}{\left(\frac{2-\alpha_v}{\alpha_v}\right)\lambda}$$

where η is the fluid's dynamic viscosity, α_v represents the tangential momentum-accommodation coefficient, and λ is the molecules' mean free path.

A simpler expression for β is

$$\beta = \frac{\eta}{L_s}$$

where L_s (the slip length) is a measure that says for a straight channel, the flow profile extrapolates to zero at distance L_s away from the boundary. This equation holds for both liquids and gases.

Thermal creep results from a temperature gradient along the boundary. The following equation relates the thermally-induced jump in tangential velocity to the tangential gradient of the natural logarithm of the temperature along the boundary:

$$\Delta u = \sigma_T \frac{\eta}{\rho} \nabla_t \log T$$

where σ_T is the thermal slip coefficient, η represents the fluid's dynamic viscosity, and ρ is the fluid's density.

Combining the previous relationships results in the following equation:

$$\mathbf{u} - \mathbf{u}_{T,\,\mathrm{wall}} = \frac{L_s}{\eta} \tau_{n,\,t} + \sigma_T \frac{\eta}{\rho T} \nabla_t T$$

where $\mathbf{u}$ is the flow vector, $\mathbf{u}_{T,\mathrm{wall}}$ gives the tangential wall velocity, L_s represents the slip length, η is the dynamic viscosity, $\tau_{n,t}$ denotes the tangential shear stress, σ_T is the thermal slip coefficient (often close to 0.75), ρ is the fluid density, and T is the fluid temperature.

You can either define L_s directly or calculate it from

$$L_s = \left(\frac{2 - \alpha_v}{\alpha_v}\right)\lambda\,,$$

where α_v and λ are the same as before.

Referring to "Boundary Forces" on page 252, you can relate the tangential shear stress to viscous boundary force by

$$\tau_{n,\,t} = \mathbf{K} - (\mathbf{n} \cdot \mathbf{K})\mathbf{n}\,,$$

but instead of using the viscous stress tensor, the component of $\mathbf{K}$ are the Lagrange multipliers that are used to implement the boundary condition.

Similarly, the tangential temperature gradient results from the difference of the gradient and its normal projection:

$$\nabla_t T = \nabla T - (\mathbf{n} \cdot \nabla T)\mathbf{n} .$$

BOUNDARY TYPE: INLET

This boundary type contains different ways to specify conditions on a boundary where the fluid is supposed to enter the domain. Notice that the formulations contained in this boundary type all appear, some of them slightly modified, in the Outlet boundary type as well. Hence, there is nothing in the mathematical formulations that prevents a fluid from leaving the domain through boundaries where you have specified the Inlet boundary type.

Velocity

For inlet boundaries you can specify the fluid's velocity equal to a given vector $\mathbf{u}_0$:

$$\mathbf{u} = \mathbf{u}_0$$

You can also specify the normal inflow velocity (the speed in the inward direction normal to the boundary), that is,

$$\mathbf{u} = -\mathrm{U}_0\mathbf{n} .$$

Pressure, No Viscous Stress

This boundary condition states that the pressure is p_0 and that the viscous boundary force is zero:

$$p = p_0$$
$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

It is a numerically stable boundary condition that admits total control of the pressure level along the entire boundary. However, if the inflow is not normal to the boundary, this condition is an overspecification. In the case that your solution turns out to have a non-normal inflow velocity, there are two choices. Either, move the boundary farther away to a location where the inflow is normal to the boundary or, use a stress type boundary condition described on page 262.

Laminar Inflow

In the microscale, flows often have a very low Reynolds number and a fully developed laminar profile. Thus it is reasonable to assume this profile in any inlet boundary of a microscale model.

You can compute the laminar flow profile by adding a special formulation of the stationary Stokes flow equations to boundary as in

$$-L_{\text{entr}}\nabla_t \cdot [-p\mathbf{I} + \eta(\nabla_t \mathbf{u} + (\nabla_t \mathbf{u})^T)] = -\mathbf{n}p_{\text{entr}}$$
$$\nabla_t \cdot \mathbf{u} = 0 \qquad (8\text{-}7)$$

where L_{entr} gives the entrance length, that is, the length of the fictious inlet channel outside of the model, p is the pressure, η denotes the dynamic viscosity, $\mathbf{u}$ is the velocity field, p_{entr} is the entrance pressure at the end of the inlet channel, $\mathbf{I}$ is the identity matrix, and $\mathbf{n}$ is the outward normal vector. The subscript in ∇_t emphasizes that this equation is solved for tangential components of the boundary. The equation's dimension is one less than the model dimension.

The Equation 8-7 simulate the effect of a fictious channel outside of the true geometry. The software solves it together with any of Equations 8-1 to 8-4, and they share the variables p, η, and $\mathbf{u}$. The pressure p_{entr} has either a user-defined value or it is solved so that the average velocity or the volume flow through the boundary meets the user-defined value. Be sure to set the inlet length, L_{entr}, large enough so that the flow can develop its laminar form:

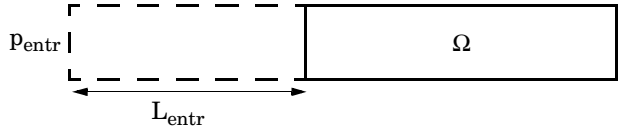


Figure 8-1: Sketch of the physical situation simulated when using the Laminar inflow boundary condition. Ω is the actual computational domain while the dashed domain is a fictitious domain.

The default operation of Equation 8-7 assumes that the conditions on the inlet channel are the same as in the adjacent domain Ω. For example, if the walls of Ω have electroosmotic then the inlet channel will behave similarly. The velocity components can be forced to zero in the outer sides of the inlet boundary. which corresponds to inlet channel with no-slip condition.

BOUNDARY TYPE: OUTLET

This boundary type contains different ways to specify conditions on a boundary where the fluid exits the domain. Note that all of the formulations in this type can be found, possibly slightly modified, in other boundary types as well. Hence, there is nothing in the mathematical formulations that prevent a fluid from entering the domain through boundaries where you have set the Outlet boundary type.

Velocity

Also for outlets, you can specify the fluid's velocity equal to a given vector $\mathbf{u}_0$:

$$\mathbf{u} = \mathbf{u}_0$$

Alternatively, you can specify the outward normal velocity (the speed in the outward direction normal to the boundary), that is,

$$\mathbf{u} = U_0\mathbf{n}\,.$$

Pressure, No Viscous Stress

This boundary condition states that the pressure is p_0, and that the viscous force is zero:

$$p = p_0$$
$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

It is a numerically stable boundary condition that admits total control of the pressure level at the whole boundary. However, if the outflow is not normal to the boundary, this condition is an overspecification. In the case that your solution turns out to have a non-normal outflow velocity, there are two choices. Either move the boundary farther away to a location where the outflow is normal to the boundary or use a stress type boundary condition described on page 262.

Note that this condition is identical to the Pressure, no viscous stress condition for Inflow boundaries. Hence, depending on the pressure field in the rest of the subdomain, a boundary with this condition can become an inflow boundary.

Pressure

This boundary condition prescribes only a Dirichlet condition for the pressure:

$$p = p_0$$

Use this boundary condition only for high Reynolds number outflow boundaries, that is $\mathrm{Re}^c = \rho|\mathbf{u}|h/(2\eta) >> 1$. It is far less stable than the Pressure, no viscous stress boundary condition, but it is consistent with a non-normal outflow velocity.

No Viscous Stress

Prescribes vanishing viscous stress:

$$\eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

This condition can be useful in some situations because it does not impose any constraint on the pressure. A typical example is a model with volume forces that give rise to pressure gradients that are hard to prescribe in advance. It should however be combined with a point constraint on the pressure to be numerically stable (see "Point Settings" on page 276).

Normal Stress

The total stress on the boundary is set equal to a stress vector of magnitude, f_0, oriented in the negative normal direction:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta\frac{\partial u_n}{\partial n} + f_0 \qquad (8\text{-}8)$$

If $\partial u_n/\partial n$ is small, Equation 8-8 can be interpreted as $p \approx f_0$.

Laminar Outflow

The Laminar outflow boundary condition is similar to the Laminar inflow boundary condition (see page 258), but you define the exit pressure and the exit length instead of the entrance pressure and entrance length.

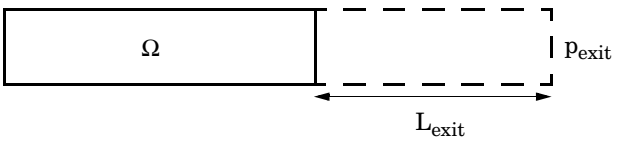


Figure 8-2: Sketch of the physical situation simulated when using Laminar outflow boundary condition. Ω is the actual computational domain while the dashed domain is a fictitious domain

BOUNDARY TYPE: SYMMETRY BOUNDARY

Symmetry Boundary

Prescribes no penetration and vanishing shear stresses:

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{t} \cdot (-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T))\mathbf{n} = 0$$

The terms inside the parenthesis are the stress tensor of the main equation. Thus for the weakly compressible formulations (Equation 8-2 and Equation 8-4) extra terms appear.

Axial Symmetry

This boundary condition is only available in 2D Axial Symmetry. Use it on all boundaries with coordinate $r = 0$. It prescribes $u_r = 0$ and vanishing stresses in the z direction.

BOUNDARY TYPE: OPEN BOUNDARY

You can use this boundary type on boundaries that are open to large volumes of fluid. Fluid can both enter and leave the domain on boundaries with this type of condition.

No Viscous Stress

Prescribes vanishing viscous stress:

$$\eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

This condition can be useful in some situations because it does not impose any constraint on the pressure. A typical example is a model with volume forces that give rise to pressure gradients that are hard to prescribe in advance. It should however be combined with a point constraint on the pressure to be numerically stable (see "Point Settings" on page 276).

Normal Stress

The total stress on the boundary is set equal to a stress vector of magnitude, f_0, oriented in the negative normal direction:

$$(-p\mathbf{I} + \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

This implies that the total stress in the tangential direction is zero. This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta \frac{\partial u_n}{\partial n} + f_0 \tag{8-9}$$

If $\partial u_n/\partial n$ is small, Equation 8-9 can be interpreted as $p \approx f_0$.

BOUNDARY TYPE: STRESS

This type of boundary condition represents a very general class of conditions also known as traction boundary conditions.

General Stress

With this boundary condition, you can specify the total stress at the boundary:

$$-p\mathbf{n} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{F}$$

This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta\frac{\partial u_n}{\partial n} - \mathbf{n}\cdot\mathbf{F} \tag{8-10}$$

If $\partial u_n/\partial n$ is small, Equation 8-10 can be interpreted as $p \approx -\mathbf{n}\cdot\mathbf{F}$.

Normal Stress

If the total stress is directed normal to the boundary, you can use this boundary condition, which sets

$$-p\mathbf{n} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n} = -f_0\mathbf{n}$$

This boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta\frac{\partial u_n}{\partial n} + f_0 \tag{8-11}$$

If $\partial u_n/\partial n$ is small, Equation 8-11 can be interpreted as $p \approx f_0$.

Normal Stress, Normal Flow

Boundary condition similar to *Normal stress*, which in addition assumes the velocity vector is directed normal to the boundary:

$$(-p\mathbf{I} + \eta(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}, \quad \mathbf{t}\cdot\mathbf{u} = \mathbf{0}\ \cdot$$

Also this boundary condition implicitly sets a constraint on the pressure that for 2D flows can be written

$$p = 2\eta\frac{\partial u_n}{\partial n} + \mathrm{f}_0 \tag{8-12}$$

If $\partial u_n/\partial n$ is small, Equation 8-10 can be interpreted as $p \approx \mathrm{f}_0$.

Application Modes for Laminar Flow

The microfluidics application modes for laminar flow share many common features, and the main difference is in the formulation of the PDE equation they use. This section describes the common structure of the General Laminar Flow, Incompressible Navier-Stokes, Non-Isothermal Flow, Stokes Flow, and Non-Isothermal Stokes Flow application modes.

Variables and Space Dimension

The microfluidics application modes solve for the pressure and the velocity vector components. They are available for 2D, 2D axisymmetric, and 3D geometries.

The dependent variables in this application mode appear in the following table:

TABLE 8-1: MICROFLUIDICS APPLICATION MODES—DEPENDENT VARIABLES

DEPENDENT VARIABLE	DESCRIPTION
u, v, w	Velocity in the x_1, x_2, and x_3 direction
p	Pressure

Application Mode Properties

The following figure shows the **Application Mode Properties** dialog box of the microfluidics application modes. Here you control global settings for a model. To open it, choose **Physics>Properties**.

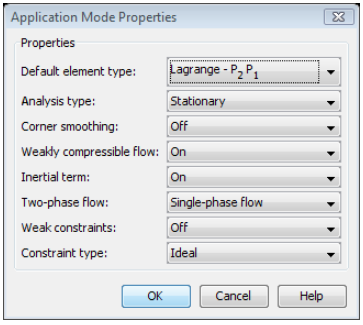


Figure 8-3: Application mode properties of the microfluidics application modes.

Among the properties you control are:

Default element type The selected element type becomes the default for all new subdomains. However, it does not affect existing subdomains.

Analysis type Possible choices are either **Stationary** or **Transient** analyses. The choice affects both the equations and the solver settings in the **Solver Parameters** dialog box.

Weakly compressible flow Possible choices are **On** or **Off**. This selection defines whether the equation is incompressible or weakly compressible. See Table 8-2 for how the value of this setting corresponds to the equations.

Inertial term Possible choices are **On** or **Off**. This setting sets or removes the inertial component from the equations. See Table 8-2 for how the value of this setting corresponds to the equations.

TABLE 8-2: APPLICATION MODE PROPERTY AND EQUATION CORRESPONDENCE

WEAKLY COMPRESSIBLE FLOW	INERTIAL TERM	EQUATION
Off	On	Incompressible Navier-Stokes
On	On	Weakly compressible Navier-Stokes
Off	Off	Stokes flow
On	Off	Weakly compressible Stokes flow

Scalar Variables

The microfluidics application modes have only one scalar variable: the electric permittivity of free space, which the electroosmotic velocity boundary condition uses. The software appends an underscore followed by the application mode abbreviation (`mmglf` by default) to the variable name.

TABLE 8-3: APPLICATION SCALAR VARIABLE

PROPERTY	VARIABLE	VALUE	DESCRIPTION
ε_0	epsilon0	$8.854187817 \cdot 10^{-12}$ F/m	Electric permittivity of free space

Subdomain Settings

You enter the fluid's properties using the two or three pages of the **Subdomain Settings** dialog box: On the **Physics** page (Figure 8-4) you define general properties, and on the **Microfluidic** page (Figure 8-5) you define special parameters meaningful only in the microscale. If you have defined **Weakly compressible flow** = **On** and **Inertial term** = **On** (th Weakly Compressible Navier-Stokes application mode) then there is also an additional **Density** page (Figure 8-6) where you define the fluid's density. Otherwise

these pages have only slight differences between the application modes as pointed out in the text.

THE PHYSICS PAGE

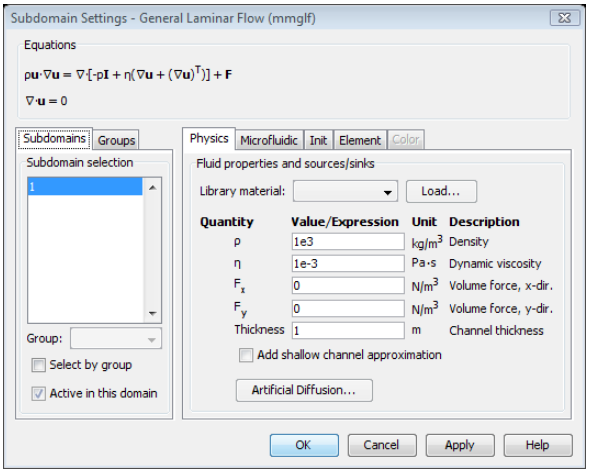


Figure 8-4: The Physics page of the Subdomain Settings dialog box.

In the Physics page you define the following settings:

Density, ρ This setting appears for other than the Weakly Compressible Navier-Stokes application mode. This material property specifies the fluid density. In the Stokes Flow application mode ρ has an effect only in a transient analysis.

Dynamic Viscosity, η This term describes the relationship between the shear stresses in a fluid and the shear rate. Intuitively, water and air have a low viscosity, and substances often described as thick, such as oil, have a higher viscosity. You can describe a non-Newtonian fluid by defining a shear-rate dependent viscosity. Examples of non-Newtonian fluids include yogurt, paper pulp, and polymer suspensions.

Dilatational Viscosity, κ You can set this coefficient to zero for most fluid-dynamics applications. This coefficient exists only for the Non-Isothermal and Non-Isothermal Stokes Flow application modes.

Volume Force, $\mathbf{F}$ The volume force vector, $\mathbf{F} = (F_x, F_y, F_z)$, describes a distributed force field such as gravity. The unit is force/volume.

Thickness This coefficient exists only in 2D and gives the thickness of a channel with a rectangular cross section. The solver uses this value in the drag term if you select the **Add shallow channel approximation** check box.

Add Shallow Channel Approximation This check box exists only in 2D. When selected, it adds the drag term from Equation 8-5 to the application mode equations.

Artificial Diffusion Click this button to define the artificial diffusion settings. The microfluidics application modes support artificial diffusion using the following methods:

- Isotropic diffusion
- Streamline diffusion
- Crosswind diffusion
- Pressure stabilization

See "Stabilization Techniques" on page 433 in the *COMSOL Multiphysics Modeling Guide* for more information about artificial diffusion and numerical stabilization techniques.

THE MICROFLUIDIC PAGE

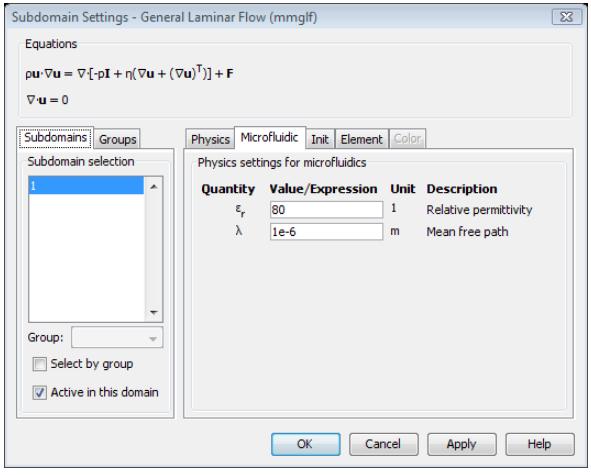


Figure 8-5: The Microfluidic page of the Subdomain Settings dialog box.

In the Microfluidics page you define the following settings:

Relative Permittivity, ε_r This coefficient defines the fluid's relative electric permittivity. It is needed if you apply the electroosmotic velocity boundary condition to a model.

Mean Free Path, λ If the fluid is a gas, you can define the mean free path of the fluid molecules. The model uses this information if you want to calculate the slip length for the slip velocity boundary condition.

THE DENSITY PAGE

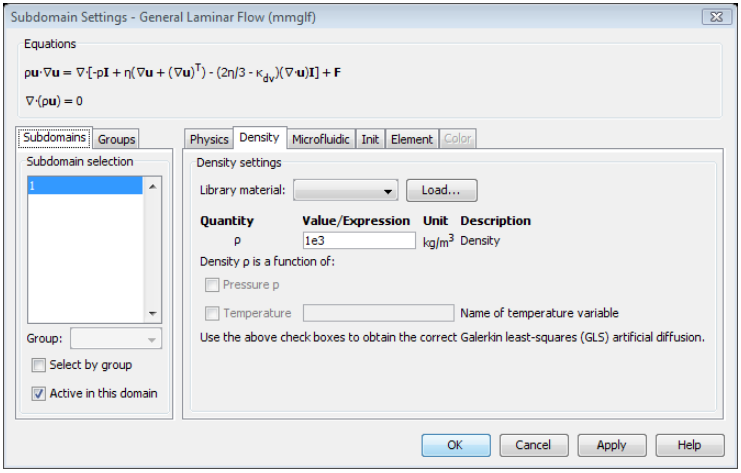


Figure 8-6: The Density page of the Subdomain Settings dialog box.

The Density page only appear if you have defined **Weakly compressible flow** = **On** and **Inertial term** = **On** (the Weakly Compressible Navier-Stokes application mode). On this page you define the following parameters:

Density, ρ This setting appears for other than the Weakly Compressible Navier-Stokes application mode. This material property specifies the fluid density. In the Stokes Flow application mode ρ has an effect only in a transient analysis.

Also, if you have activated the GLS artificial stabilization (To activate GLS artificial stabilization go to **Physics** page and click **Artificial Diffusion**, select **Streamline Diffusion** checkbox and then select **Galerkin Least-Squares (GLS)**), then you have additional settings:

Pressure This defines whether the density is a function of pressure.

Temperature This defines whether density is a function of temperature. Also define the dependent variable for temperature.

Note that you should use the above settings only if you have typed an expression into the density field that makes the density a function of either pressure or temperature. These settings then control the exact weak expression for the Galerkin Least-Squares (GLS) artificial diffusion. This method (See "Stabilization Techniques" on page 433 in the *COMSOL Multiphysics Modeling Guide* for more information) is necessary to get a stable finite element discretization of the Weakly Compressible Navier-Stokes equations. You can also define the density as a function of any other dependent variable, but GLS can only stabilize with respect to pressure and temperature, which are the most common variables.

The GLS artificial stabilization is not active by default for two reasons. In microfluidic systems the flow often has very low Reynolds number because the viscous effects dominate over the inertial effects. In these applications the stabilization is not needed, and they would produce unnecessary computational load. Another reason is that the stabilization technique have been designed to operate best on certain velocity ranges, and for very slow velocity it will produce too much stabilization.

Boundary Conditions Settings

You select and specify boundary conditions in the **Boundary Settings** dialog box.

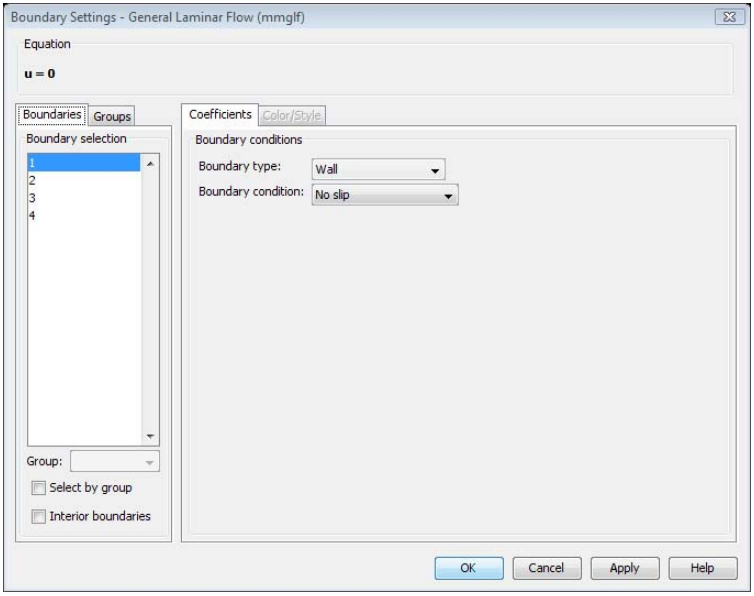


Figure 8-7: The Boundary Settings dialog box.

Different types of boundary conditions are available by first selecting a **Boundary type**: **Wall**, **Inlet**, **Outlet**, **Symmetry**, **Open boundary**, or **Stress**, then you select the actual condition from the **Boundary conditions** list. See "Boundary Conditions" on page 253 for an overview and theory of all available boundary conditions. The following describes the user interfaces for each condition.

BOUNDARY TYPE: WALL

Slip

Defines that fluid flows freely and only along the tangent of the boundary. No additional parameters are needed to define this condition.

Sliding Wall

Tangential wall velocity U_w Defines the velocity of the tangentially moving wall.

Moving/Leaking Wall

Wall velocity, u_w, v_w Define the *x*- and *y*-components of the boundary (or through boundary) velocity.

No Slip

Defines that fluid is not moving on the boundary. No additional parameters are needed to define this condition.

Electroosmotic Velocity

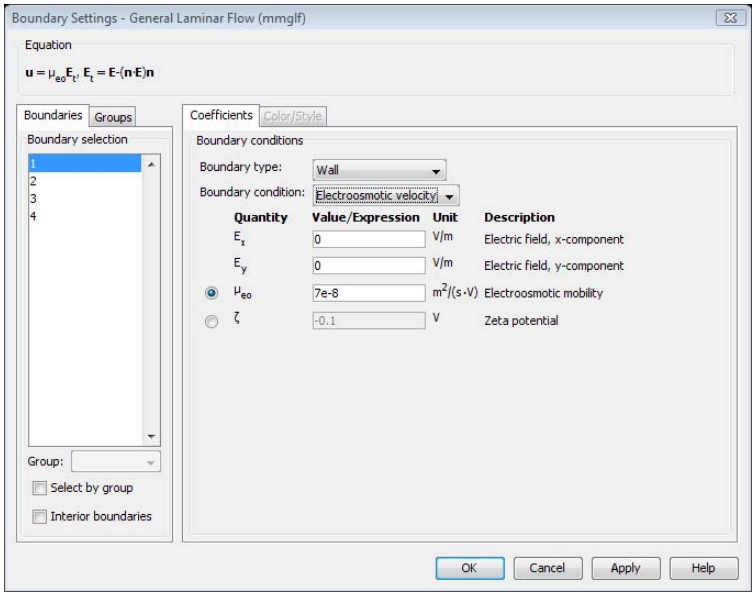


Figure 8-8: The Electroosmotic velocity settings in the Boundary Settings dialog box.

Use the following settings to specify the Electroosmotic velocity boundary condition:

Electric field, E_x, E_y, E_z, E_r These coefficients define components of the electric field in the fluid near the boundary.

Electroosmotic mobility, μ_{eo} This coefficient defines the fluid's electroosmotic mobility.

Zeta potential, ζ This coefficient defines the boundary's zeta potential if you want to calculate electroosmotic mobility according to Equation 8-6. You also need the dynamic viscosity in this equation.

Slip Velocity

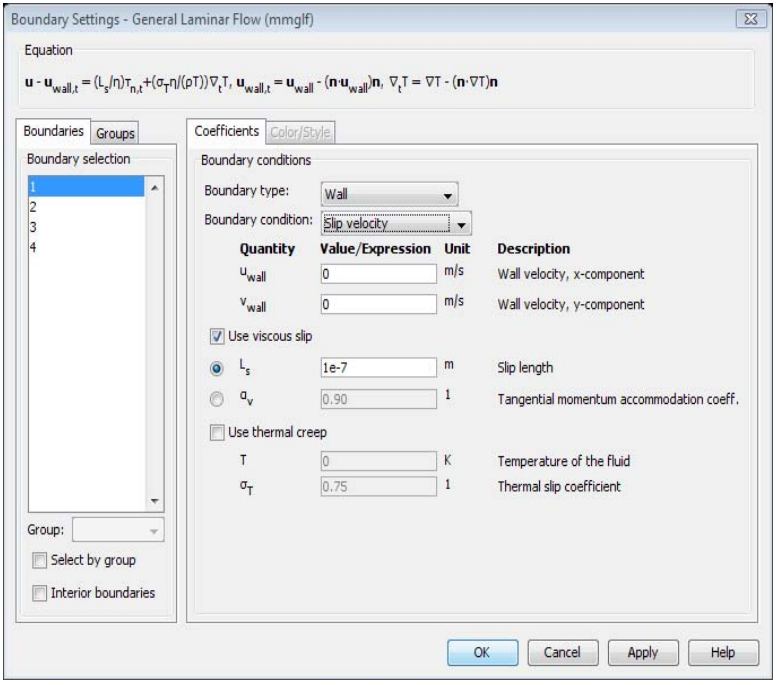


Figure 8-9: The Slip velocity settings in the Boundary Settings dialog box.

Use the following settings to specify the Slip velocity boundary condition:

Wall velocity, u_{wall}**,** v_{wall}**,** w_{wall} These coefficients define the components of the tangential wall velocity. The application modes recompute the tangential components so the given velocity need not be exactly tangential.

Use viscous slip Selecting this check box activates the viscous slip component to the boundary condition.

Slip length, L_s This coefficient defines the fluid's slip length on the boundary.

Momentum accommodation coefficient, α_v This coefficient defines the momentum accommodation coefficient if you wish to calculate the slip length for a gaseous fluid. You must also define the mean free path of the gas molecules in the subdomain settings.

Use thermal creep Selecting this check box activates the thermal creep component to the boundary condition.

Temperature of the fluid, T This coefficient defines the fluid's temperature.

Thermal slip coefficient, σ_T This coefficient defines the thermal slip coefficient.

WALL TYPE: INLET

Velocity

Defines fluid's velocity at the boundary.

Velocity, u_0, v_0 Define the fluid velocity at the boundary

Normal velocity, U_0 Defines the fluids normal inflow velocity at the boundary.

Pressure, No Viscous Stress

Defines the fluid's pressure and vanishing viscous stress at the boundary.

Pressure, p_0 Defines the fluid pressure at the boundary.

Laminar Inflow

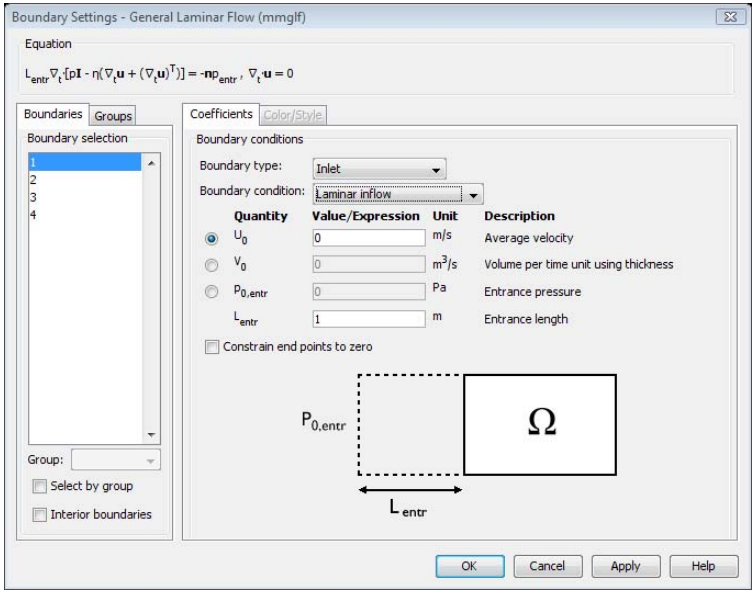


Figure 8-10: The Laminar inflow settings in the Boundary Settings dialog box.

Use the following settings to specify a Laminar inflow or Laminar outflow boundary condition:

Average velocity, U_0 This coefficient defines the average velocity through the inlet boundaries.

Volume flow, V_0 This coefficient defines the volume flow through the inlet in time unit. In 2D, the application mode uses thickness information from the subdomain settings.

Entrance pressure, $p_{0,entr}$ This coefficient defines pressure at the end of the inlet. For Outflow boundaries, specify the exit pressure, $p_{0,exit}$ instead.

Entrance length, L_{entr} This coefficient defines the length of the inlet channel outside the model domain. This value must be large enough so that flow can reach a laminar profile. For Outflow boundaries, specify the exit length, L_{exit}.

Constrain end points to zero (2D) This check box forces the laminar profile to go to zero at the outer sides of the inlet channel. Otherwise the velocity is defined by the boundary condition of the adjacent boundary in the model. For example, if one end of a boundary with a Laminar inflow condition connects to a Slip boundary condition, then the laminar profile has a maximum at that end.

Constrain outer edges to zero (3D) Selecting this check box has the same effect as constraining an end point to zero in 2D.

Note: All boundaries that are connected and have a laminar inflow boundary condition and also have same settings and group name form a single inlet channel. This affects a model if you define volume flow or choose to constrain end points or outer edges to zero.

WALL TYPE: OUTLET

Velocity

Defines the velocity at the boundary.

Velocity, u_0**,** v_0**,** w_0 Define the fluid velocity at the boundary

Normal velocity, U_0 Defines the fluids normal outflow velocity at the boundary.

Pressure, No Viscous Stress
Defines the pressure and vanishing viscous stress at the boundary.

Pressure, p_0 Defines the fluid pressure at the boundary.

Pressure

Pressure, p_0 Defines the fluid pressure at the boundary.

No Viscous Stress
Defines vanishing viscous stress at the boundary. No additional parameters are needed to define this condition.

Normal Stress
Defines stress on the boundary.

Normal stress, f_0 Defines normal stress on the boundary.

Laminar Outflow
Defines the outflow with fully developed laminar profile. Settings for this condition are the same as for Laminar Inflow on page 258, with the difference that average velocity U_0 and volume flow V_0 define outflow, and $p_{0,exit}$ and L_{exit} defines the exit pressure and exit length

WALL TYPE: SYMMETRY

In 2D and 3D the Symmetry wall type defines the boundary condition directly. In Axial Symmetry

Symmetry
Defines symmetry boundary. No additional parameters are needed to define this condition.

Axial Symmetry
Defines the symmetry condition on boundary at $r = 0$.No additional parameters are needed to define this condition.

WALL TYPE: OPEN BOUNDARY

Normal Stress
Defines stress on the boundary.

Normal stress, f_0 Defines normal stress on the boundary.

No Viscous Stress
Defines vanishing viscous stress at the boundary. No additional parameters are needed to define this condition.

WALL TYPE: STRESS

General Stress
Defines stress on the boundary.

Stress, $\mathbf{F}$ Defines the x- and y- components of the fluid stress on the boundary

Normal Stress
Defines stress on the boundary.

Normal stress, f_0 Defines the outward directed normal stress of the fluid on the boundary

Normal Stress, Normal Flow
Defines stress and constraints flow to normal direction on the boundary.

Normal stress, f_0 Defines the outward directed normal stress of the fluid on the boundary

Point Settings

If you do not specify the pressure using a Pressure boundary condition, the flow model might not be sufficiently specified because there is no restriction on the pressure variable. This, in turn, can lead to problems when solving the model.

One way to specify a pressure reference without changing the boundary conditions is to set the pressure, p_0, at a single point (vertex) using the **Point Settings** dialog box:

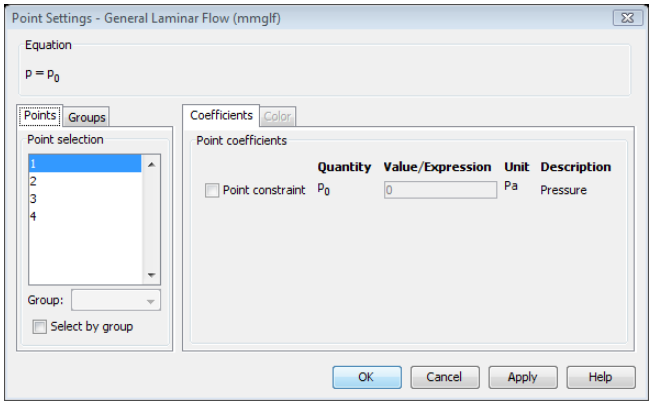


Figure 8-11: The Point Settings dialog box for setting pressure at a single point.

Application Mode Variables

A number of variables and physical quantities are available for postprocessing and for use in equations and boundary conditions. For information of these variables see "Application Modes for Laminar Flow" on page 73 in the *MEMS Module Reference Guide*.

General Laminar Flow

The General Laminar Flow application mode is a general-purpose application mode for modeling laminar flows in viscous fluids. By changing the application mode properties, you can use any of the equations from the Incompressible Navier-Stokes, Weakly Compressible Flow, Stokes Flow, and Weakly Compressible Stokes Flow application modes.

With the General Laminar Flow application mode you can solve transient and steady-state models of incompressible fluid dynamics. When started, this application mode uses the most general equations of the four non-isothermal flow equations given in Equation 8-4 on page 252.

For detailed information about the equations and boundary condition for this application mode see "Laminar Flow Theory" on page 249, and for instructions how to use this application mode see the section "Application Modes for Laminar Flow" on page 264.

Incompressible Navier-Stokes

The Navier-Stokes equations describe laminar flow in viscous fluids through momentum balances for each of the components of the momentum vector in all spatial dimensions. Originally they assume that the density and viscosity of the modeled fluid are constant, which gives rise to a continuity condition.

The Incompressible Navier-Stokes application mode in COMSOL Multiphysics is somewhat more general and can account for arbitrary variations in viscosity and small variations in density, for example, through the Boussinesq approximation.

With the Incompressible Navier-Stokes application mode you can solve transient and steady-state models of incompressible fluid dynamics. When started, this application mode uses the Incompressible Navier-Stokes relationships given in Equation 8-1 on page 249. If you change the equation by changing either of the Non-isothermal flow or Inertial term application mode properties, the application mode changes and always corresponds to the equation your are using.

For detailed information about the equations and boundary condition for this application mode, see the section "Laminar Flow Theory" on page 249, and for instructions how to use this application mode, see the section "Application Modes for Laminar Flow" on page 264.

Weakly Compressible Navier-Stokes

The Weakly Compressible Navier-Stokes application mode is an extension of the Navier-Stokes equations for laminar flow that allows for small changes in fluid density. For many fluids, changes in density arise from temperature changes. This phenomenon is of particular interest when modeling free-convection problems where density differences drive the fluid flow. Note that the application mode is not suitable for high velocities or when compression or expansion of the fluid results in substantial internal work or heat effects.

With the Weakly Compressible Navier-Stokes application mode you can solve transient and steady-state models of incompressible fluid dynamics. When started, this application mode uses the weakly compressible flow relationships from Equation 8-2 on page 250. If you change the equation by changing the Weakly compressible flow or the Inertial term application mode property, the application mode changes and always corresponds to the equation your are using.

For detailed information about the equations and boundary condition for this application mode see the section "Laminar Flow Theory" on page 249, and for instructions how to use this application mode see the section "Application Modes for Laminar Flow" on page 264.

Stokes Flow

The Stokes flow equations describe laminar flow in viscous fluids. Thus this application mode is intended for flow with very low velocity, creeping flows whose Reynolds number is very low. In this case the inertial forces of the flow are very small so you can simplify the incompressible Navies-Stokes equations by removing the inertial term. Keeping in mind the restriction to creeping flows, you can use this application mode in same applications as the Incompressible Navier-Stokes application mode except the equations are less nonlinear.

With the Stokes Flow application mode you can solve transient and steady-state models of incompressible fluid dynamics. When started, this application uses the Stokes flow relationships given in Equation 8-3 on page 251. If you modify the equation by changing either of the Non-isothermal flow or Inertial term application mode properties, the application mode changes and always corresponds to the equation your are using.

For detailed information about the equations and boundary condition for this application mode see the section "Laminar Flow Theory" on page 249, and for instructions how to use this application mode see the section "Application Modes for Laminar Flow" on page 264.

Weakly Compressible Stokes Flow

The weakly compressible Stokes flow equations describe laminar flow in viscous fluids, whose density may change due to external heating, for example. This application mode is intended for modeling flows with very low velocity, or creeping flows whose Reynolds number is very low. In this case the inertial forces of the flow are very small, and thus you can simplify the weakly compressible flow equations by removing the inertial term. Keeping in mind the restriction to creeping flows, you can use this application mode for the same applications as the Weakly Compressible Navier-Stokes application mode, but the equations are less nonlinear.

With the Weakly Compressible Stokes Flow application mode you can solve transient and steady-state models of weakly compressible fluid dynamics. When started, this application uses the weakly compressible Stokes flow relationships in Equation 8-4 on page 252. If you change the equation by changing either of the Weakly compressible flow or Inertial term application mode properties, the application mode changes and always corresponds to the equation your are using.

For detailed information about the equations and boundary condition for this application mode see the section "Laminar Flow Theory" on page 249, and for instructions how to use this application mode see the section "Application Modes for Laminar Flow" on page 264.

The Level Set Method

Flow problems with moving interfaces or boundaries occur in a number of different applications, such as fluid-structure interaction, multiphase flows, and flexible membranes moving in a liquid. One possible way to track moving interfaces is to use a level set method. A certain contour line of the globally defined function, the level set function, then represents the interface between phases. With the Level Set application mode you can move the interface within any velocity field.

For two-phase flow problems modeled with the level set method, use the Level Set Two-Phase Flow application mode, see "The Level Set Method for Two-Phase Flow" on page 289.

Subdomain Equations

The level set method is a technique to represent moving interfaces or boundaries using a fixed mesh. It is useful for problems where the computational domain can be divided into two domains separated by an interface. Each of the two domains can consist of several parts. Figure 8-12 shows an example where one of the domains consists of two separated parts. The interface is represented by a certain level set or isocontour of a globally defined function, the level set function, ϕ. In COMSOL Multiphysics, ϕ is a smooth step function that equals zero in a domain and one in the other. Across the interface, there is a smooth transition from zero to one. The interface is defined by the 0.5 isocontour, or the level set, of ϕ. Figure 8-13 shows the level set representation of the interface in Figure 8-12.

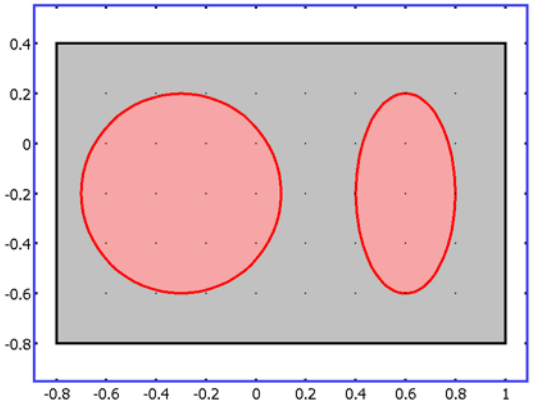


Figure 8-12: Example of two domains divided by an interface. In this case, one of the domain consists of two parts. Figure 8-13 shows the corresponding level set representation.

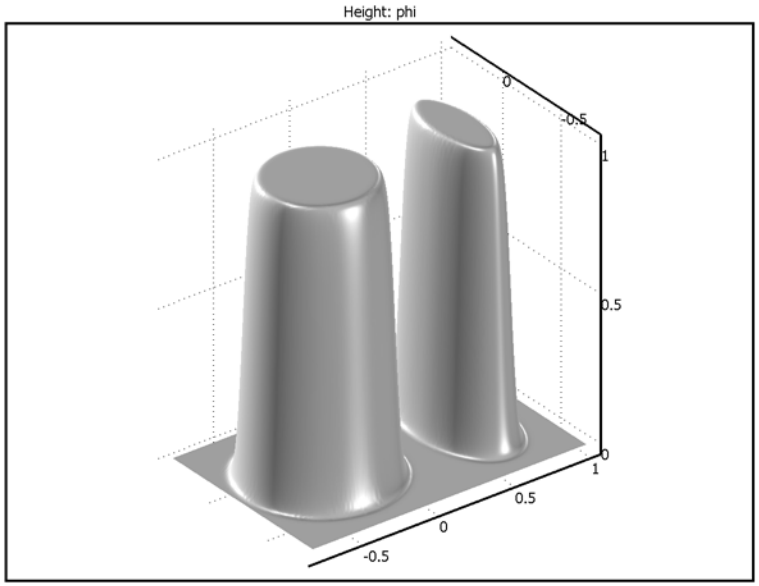


Figure 8-13: Surface plot of the level set function corresponding to Figure 8-12.

The application mode solves the following equations in order to move the interface with the velocity field, $\mathbf{u}$:

$$\phi_t + \mathbf{u} \cdot \nabla\phi = \gamma\nabla \cdot \left(\varepsilon\nabla\phi - \phi(1-\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{8-13}$$

The terms on the left hand side gives the correct motion of the interface, while the right hand side terms are necessary for numerical stability. The parameter, ε, determines the thickness of the region where ϕ goes smoothly from zero to one and should be of the same order as the size of the elements of your mesh. By default, ε is constant within each subdomain and equals the largest value of the mesh size, h, within the subdomain. The parameter γ determines the amount of reinitialization or stabilization of the level set function. It needs to be tuned for each specific problem. If γ is too small, the thickness of the interface might not remain constant, and oscillations in ϕ may appear because of numerical instabilities. On the other hand, if γ is too large, the interface moves incorrectly. A suitable value for γ is the maximum magnitude of the velocity field $\mathbf{u}$.

CONSERVATIVE AND NONCONSERVATIVE FORM

If the velocity is divergence free, that is, if

$$\nabla \cdot \mathbf{u} = 0 \tag{8-14}$$

the volume (area for 2D problems) bounded by the interface should be conserved if there is no inflow or outflow through the boundaries. To obtain exact numerical conservation, you can switch to the conservative form

$$\phi_t + \nabla \cdot (\mathbf{u}\phi) = \gamma\nabla \cdot \left(\varepsilon\nabla\phi - \phi(1-\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{8-15}$$

using the **Application Mode Properties** dialog box (see Figure 8-14).

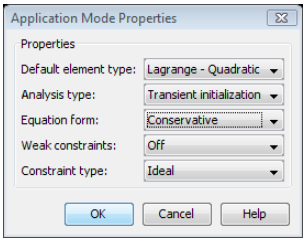


Figure 8-14: In the Application Mode Properties dialog box, you can switch between the conservative and the nonconservative form of the level set equation.

Using the conservative level set form you obtain exact numerical conservation of the integral of ϕ. Note, however, that the nonconservative form is better suited for numerical calculations and usually converges more easily. The integral of the level set function is then only approximately conserved, but this is sufficient for most applications.

INITIALIZING THE LEVEL SET FUNCTION

Before you can solve Equation 8-13 or Equation 8-15, you must initialize the level set function such that it varies smoothly from zero to one across the interface. Do so by letting ϕ_0 be zero on one side of the interface and one on the other. Then solve

$$\phi_t = \gamma\nabla \cdot \left(\varepsilon\nabla\phi - \phi(1-\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{8-16}$$

using ϕ_0 as the initial condition from $t = 0$ to $t \approx 5\varepsilon/\gamma$. The resulting ϕ is smooth across the interface and a suitable initial condition to the level set equation. The Level Set application mode automatically sets up Equation 8-16 if you select **Transient initialization** from the **Analysis type** list in the **Application Mode Properties** dialog box (Figure 8-15). You then solve the equation, store the solution, change the analysis type to **Transient**, and finally use the stored solution as the initial condition.

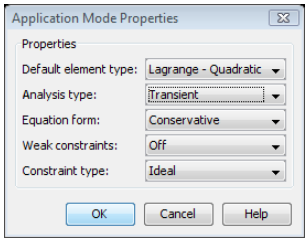


Figure 8-15: By default, the Analysis type is set to Transient initialization. Note that you must switch it to Transient after you have initialized the level set function.

VARIABLES FOR GEOMETRIC PROPERTIES OF THE INTERFACE

Geometric properties of the interface are often needed. The unit normal to the interface is given by

$$\mathbf{n} = \left.\frac{\nabla\phi}{|\nabla\phi|}\right|_{\phi = 0.5} \tag{8-17}$$

The curvature is defined as

$$\kappa = -\nabla \cdot \mathbf{n}|_{\phi = 0.5} \tag{8-18}$$

These variables are available in the application mode.

Note: It is only possible to compute the curvature explicitly when using second-order or higher-order elements.

Boundary Conditions

Inflow/Phi At inflow boundaries you must specify a value of the level set function. Typically you set ϕ to either 0 or 1.

Outflow For outflow boundaries no boundary condition is imposed on the level set function.

Insulation/Symmetry Use this boundary condition when there should be no flow across the boundary or if the boundary is a symmetry line or plane.

Axial Symmetry Boundary condition at $r = 0$ for axisymmetric problems.

Initial interface Defines the boundary as the initial position of the interface.

Reference

1. E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," *J. Comput. Phys.*, vol. 210, pp. 225–246, 2005.

The Level Set Method for Two-Phase Flow

To model the flow of two different, immiscible fluids, where the exact position of the interface is of interest, you can use the Level Set Two-Phase Flow application mode. The application mode tracks the fluid-fluid interface using a level set method. It accounts for differences in the two fluids' densities and viscosities and includes the effect of surface tension and gravity.

Subdomain Settings

For incompressible fluids, the application mode uses the equations

$$\rho\frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \eta(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{F}_{\mathrm{st}} + \mathbf{F}_{\mathrm{g}} + \mathbf{F} \tag{8-19}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{8-20}$$

$$\frac{\partial \phi}{\partial t} + \nabla \phi \cdot \mathbf{u} = \gamma \nabla \cdot \left(\varepsilon \nabla \phi - \phi(1-\phi)\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{8-21}$$

The density is a function of the level set function according to

$$\rho = \rho_1 + (\rho_2 - \rho_1)\phi$$

and the dynamic viscosity is

$$\eta = \eta_1 + (\eta_2 - \eta_1)\phi$$

where ρ_1 and ρ_2 are the constant densities of Fluid 1 and Fluid 2, respectively, and η_1 and η_2 are the dynamic viscosities of Fluid 1 and Fluid 2, respectively. Here, Fluid 1 corresponds to the domain where $\phi < 0.5$, and Fluid 2 corresponds to the domain where $\phi > 0.5$.

The surface tension force acting at the interface between the two fluids is

$$\mathbf{F}_{\mathrm{st}} = \sigma \kappa \delta \mathbf{n}$$

where σ is the surface tensions coefficient (N/m), κ is the curvature, and $\mathbf{n}$ is the unit normal to the interface, as defined in Equation 8-17 and Equation 8-18 in the section

"Variables For Geometric Properties of the Interface" on page 287. δ (1/m) is a Dirac delta function concentrated to the interface. κ depends on second derivatives of the level set function ϕ. This can lead to poor accuracy of the surface tension force. Therefore, the application mode uses the alternative formulation

$$\mathbf{F}_{st} = \nabla \cdot (\sigma(\mathbf{I} - (\mathbf{n}\mathbf{n}^T))\delta)$$

In the weak formulation of the momentum equations, it is possible to move the divergence operator, using integration by parts, to the test functions for the velocity components.

The δ-function is approximated by a smooth function according to

$$\delta = 6|\nabla\phi||\phi(1-\phi)|$$

In addition to the surface tension force, there is also a term that accounts for gravity. The gravity force is

$$\mathbf{F}_g = \rho\mathbf{g}$$

where $\mathbf{g}$ is the gravity vector.

Because the velocity field is divergence free, you can use either the conservative or the nonconservative form of the level set equation (see the section "Conservative and Nonconservative form" on page 285). The conservative form perfectly conserves the mass of each fluid, but the computational time is in general longer. Note that when you use the conservative form, the default element type for the velocity and pressure automatically changes to P_2 P_{-1} in 2D and P_2+ P_{-1} in 3D, since they result in a better approximation of the continuity equation.

You can modify the model in a number of different ways from the **Application Mode Properties** dialog box.

INITIALIZING THE LEVEL SET FUNCTION

Before you can solve Equation 8-19–Equation 8-21, you must initialize the level set function such that it varies smoothly from zero to one across the interface. Do so by specifying which fluid each subdomain is initially filled with. Then solve

$$\phi_t = \gamma\nabla \cdot \left(\varepsilon\nabla\phi - \phi(1-\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{8-22}$$

using ϕ_0 as the initial condition from $t = 0$ to $t \approx 5\varepsilon/\gamma$. The resulting ϕ is smooth across the interface and a suitable initial condition to the level set equation. The Level

Set Two-Phase Flow application mode automatically sets up Equation 8-22 if you select **Transient initialization** from the **Analysis type** list in the **Application Mode Properties** dialog box (see Figure ?-???). You then solve the equation, store the solution, change the analysis type to **Transient**, and finally use the stored solution as the initial condition.

Boundary Conditions

All the boundary conditions available for single-phase flow are also available for the Level Set Two-Phase Flow application mode. The boundary condition used on the level set function ϕ depends on whether the boundary is an inlet, an outlet, or a wall or symmetry boundary. For inlets, you must specify the value of the level set function. Typically, you set it to either zero or one, depending on which fluid enters the boundary. At outlets, no boundary conditions on the level set function is imposed. For walls and symmetry lines, the application mode sets the flux of the level set function to zero. The boundary types Open boundary and Stress are not available for two-phase flows.

There are also two boundary conditions that are only available for the Level Set Two-Phase Flow application mode: the Initial fluid interface and the Wetted wall boundary conditions.

THE INITIAL FLUID INTERFACE CONDITION

To specify the initial position of the interface, use the Initial fluid interface boundary condition on interior boundaries. During the initialization step, the boundary condition sets the level set function to 0.5. For the transient simulation of the fluid flow, the boundary is treated as an interior boundary.

THE WETTED WALL CONDITION

The Wetted wall boundary condition is suitable for walls in contact with the fluid interface. It enforces the slip condition

$$\mathbf{u} \cdot \mathbf{n}_{\text{wall}} = 0$$

and adds a frictional force of the form

$$\mathbf{F}_{\text{fr}} = -\frac{\eta}{\beta}\mathbf{u}$$

where β is the slip length. For numerical calculations, it is suitable to set $\beta = h$, where h is the mesh element size. The boundary condition does not set the tangential velocity

component to zero. However, the extrapolated tangential velocity component is 0 at the distance β outside the wall, as Figure 8-16 illustrates.

Finally, the boundary condition adds the following weak boundary term:

$$\int_{\partial\Omega} \text{test}(\mathbf{u}) \cdot [\sigma(\mathbf{n}_{\text{wall}} - (\mathbf{n}\cos\theta))\delta]dS$$

The boundary term is a result of the partial integration of the surface tension force in the Navier-Stokes equations. It allows you to specify the contact angle θ, that is, the angle between the fluid interface and the wall. Figure 8-18 illustrates how the contact angle is defined.

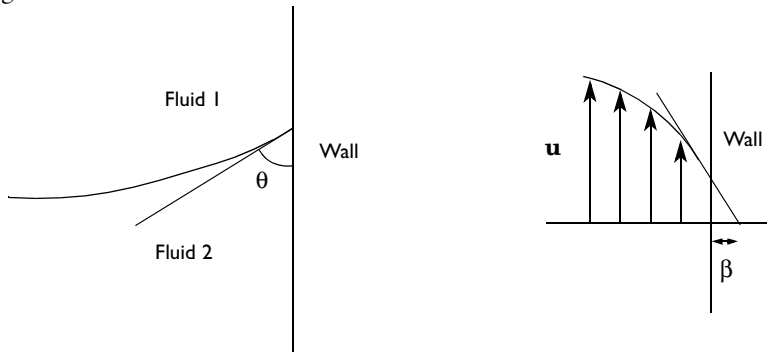


Figure 8-16: Definition of the contact angle θ at interface/wall contact points (left) and illustration of the slip length β (right).

If you use the Wetted wall boundary condition, the interface can move along the wall. For applications where the interface is fixed on the wall, the No slip condition is suitable.

Example Model

For an example model that uses the Level Set Two-Phase Flow application mode, see "Filling of a Capillary Channel" on page 404 in the *MEMS Module Model Library.*

Numerical Stabilization

In most cases the application mode works best if no stabilization is used. However, for convection-dominated flow some stabilization might be necessary. If you encounter convergence problems, make a surface plot of the Cell Reynolds number. If it is larger

than 2, switch on streamline diffusion of the anisotropic diffusion type in the **Artificial Diffusion** dialog box.

The choice of the level set parameter, γ, is also important for stability. The maximum speed of the flow is usually a suitable value.

Whenever the effect of surface tension is negligible, set the surface tension coefficient, σ, to 0 for increased stability.

Reference

1. E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," *J. Comput. Phys.*, vol. 210, pp. 225–246, 2005.

Application Mode Variables

A number of variables and physical quantities are available for postprocessing and for use in equations and boundary conditions. For information of these variables see "The Level Set Two Phase Flow Application Mode" on page 77 in the *MEMS Module Reference Guide*.

Convection and Diffusion

In the Convection and Diffusion application mode, Fick's law describes the diffusive transport in the flux vector. Fick's law is adequate when the diffusing species is dilute with respect to a solvent.

The Convection and Diffusion Application Mode

THE CONVECTION AND DIFFUSION EQUATION

The convection and diffusion equation is:

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (-D_i \nabla c_i + c_i \mathbf{u}) = R_i \tag{8-23}$$

where c_i is the concentration of species i (mol/m^3), D_i denotes its diffusion coefficient (m^2/s), $\mathbf{u}$ refers to the velocity (m/s), and R_i denotes the reaction term (mol/(m^3·s)). The velocity can be expressed analytically or be obtained by coupling a microfluidics application mode that models the momentum balance to the Convection and Diffusion application mode.

You can introduce arbitrary kinetic expressions of the reactants and products in the reaction term. The expression within brackets represents the mass flux vector

$$\mathbf{N}_i \equiv -D_i \nabla c_i + c_i \mathbf{u}\,, \tag{8-24}$$

where the first term describes the transport by diffusion, and the second represents the convective flux. The diffusion coefficient for the dissolved species accounts exclusively for the interaction between the solute and the solvent.

BOUNDARY CONDITIONS

The available boundary conditions in the Convection and Diffusion application mode include a Flux condition for the inward flux

$$\mathbf{N}_i \cdot \mathbf{n} = N_0 \tag{8-25}$$

where the boundary source term, N_0, can be arbitrarily defined to represent flux into an infinite medium, or material change, most often through chemical reaction. The Insulation/Symmetry condition has the source term in Equation 8-25 set to zero:

$$\mathbf{N}_i \cdot \mathbf{n} = 0 \tag{8-26}$$

There is also a Concentration boundary condition:

$$c_i = c_{i,0} \tag{8-27}$$

The Convective flux boundary condition assumes that the mass flow through the boundary is convection dominated. This assumes that any mass flux due to diffusion across this boundary is zero

$$\mathbf{n} \cdot (-D_i \nabla c_i) = 0 \tag{8-28}$$

so that

$$\mathbf{N}_i \cdot \mathbf{n} = c_i \mathbf{u} \cdot \mathbf{n} \tag{8-29}$$

This is a useful boundary condition, particularly in convection-dominated mass balances where the outlet concentration is unknown.

Boundary Conditions for Interior Boundaries

The following boundary conditions are available on interior boundaries and pair boundaries in assemblies:

Continuity is the default boundary condition on interior boundaries and pair boundaries; it is not applicable to exterior boundaries.

$$\mathbf{n} \cdot (\mathbf{N}_1 - \mathbf{N}_2) = 0$$

The Flux discontinuity boundary condition represents a discontinuity in the mass flux across am interior boundary or a border between parts in an assembly:

$$-\mathbf{n} \cdot (\mathbf{N}_1 - \mathbf{N}_2) = N_0$$

You can use the Thin boundary layer condition to model a thin layer of a material with a small diffusion coefficient compared to the adjacent domains:

$$\mathbf{n}_1 \cdot (-D\nabla c + c\mathbf{u})_1 = \frac{D}{d}(c_1 - c_2)$$

$$\mathbf{n}_2 \cdot (-D\nabla c + c\mathbf{u})_2 = \frac{D}{d}(c_2 - c_1)$$

The layer has the thickness d and the diffusion coefficient D. This boundary condition is only available at the border between the parts in an assembly.

CONSERVATIVE AND NONCONSERVATIVE FORMULATIONS

In modeling mass balances in the Convection and Diffusion application mode, there are two mass balance formulations available: a conservative and a nonconservative formulation. The conservative formulation is given by Equation 8-23.

The nonconservative formulation removes the convective term from the divergence operator in Equation 8-23, which gives more stability to systems coupled to a momentum balance. Use it only for systems that contain incompressible fluids. In such fluids, the following continuity equation is satisfied:

$$\nabla \cdot \mathbf{u} = 0 \tag{8-30}$$

This condition means that it is possible to move the convective term in Equation 8-23 outside of the divergence operator and to the right side:

$$\frac{\partial c_i}{\partial t} + \nabla \cdot (-D_i \nabla c_i) = R_i - \mathbf{u} \cdot \nabla c_i \tag{8-31}$$

This reformulation provides stability because it does not require COMSOL Multiphysics to additionally calculate a function similar to Equation 8-30 in the mass balance. Equation 8-31 is the default in the Convection and Diffusion application mode.

APPLICATION MODE VARIABLES

A number of variables and physical quantities are available for postprocessing and for use in equations and boundary conditions. The type of the variable denotes where the variable can be used:

- B for boundaries
- S for subdomains
- V for vector expressions

Most of the labels of the postprocessing variables are followed by the code (application mode name) given to the respective application mode. For example, the gradient of the concentration c in the x direction, denoted `grad_c_x`, is called `grad_c_x_chcd` in the Convection and Diffusion application mode. Yet, this can change if you have renamed the application mode, for example. Always check the names of the variables before using them in equation-based modeling.

They are given in the following table:

TABLE 8-4: CONVECTION AND DIFFUSION—AVAILABLE APPLICATION MODE VARIABLES

LABEL	TYPE	DESCRIPTION	EXPRESSION
c	S/B	Concentration	c
grad_	S/V	Concentration gradient	$\lvert\nabla c\rvert,\ \frac{\partial c}{\partial x_i}$
dflux_	S	Diffusive flux	$\lvert\underline{D}\nabla c\rvert$
cflux_	S	Convective flux	$\lvert c\mathbf{u}\rvert$
tflux_	S	Total flux	$\lvert-\underline{D}\nabla c + c\mathbf{u}\rvert$
ndflux_	B	Normal diffusive flux	$\mathbf{n}\cdot(-\underline{D}\nabla c)$
ncflux_	B	Normal convective flux	$c\mathbf{n}\cdot\mathbf{u}$
ntflux_	B	Normal total flux	$\mathbf{n}\cdot(-\underline{D}\nabla c + c\mathbf{u})$
dflux_	V	Diffusive flux	$\sum_j -D_{ij}\frac{\partial c}{\partial x_j}$
cflux_	V	Convective flux	cu_i
tflux_	V	Total flux	$\sum_j -D_{ij}\frac{\partial c}{\partial x_j} + cu_i$
cellPe_	S	Cell Peclet number	$\left\lvert\frac{\mathbf{u}h}{D}\right\rvert$
Dts_	S	Time-scaling coefficient	δ_{ts}
udl_	S	Dimensionless velocity	u_{dl}
D_	S	Diffusion coefficient	D, D_{ij}
R_	S	Reaction rate	R
u_, v_, w_	S	Velocity of c	u_i
N_	B	Inward flux	N_0
c0_	B	Concentration	c_0
beta_*i*	S	Convective field	u_i
Dm_	S	Mean diffusion coefficient	$\frac{\sum_{i,j} D_{ij}\beta_i\beta_j}{\lvert\vec{\beta}\rvert}$

TABLE 8-4: CONVECTION AND DIFFUSION—AVAILABLE APPLICATION MODE VARIABLES

LABEL	TYPE	DESCRIPTION	EXPRESSION
res_	S	Equation residual	$\nabla \cdot (-\underline{D}\nabla c + c\mathbf{u}) - R$
res_sc_	S	Shock capturing residual	$\nabla \cdot (c\mathbf{u}) - R$
da_	S	Total time-scale factor	δ_{ts}

ARTIFICIAL DIFFUSION

The Convection and Diffusion application mode supports artificial diffusion using the following methods:

- Isotropic diffusion
- Streamline diffusion
- Crosswind diffusion

See “Stabilization Techniques” on page 433 in the *COMSOL Multiphysics Modeling Guide* for more information.

Electrokinetic Flow

In ionic solutions subjected to electric fields, the things that drive transport are diffusion, convection, and migration. Migration implies that positive ions migrate from a positive potential to a negative potential along the direction of the electric field and vice versa for negatively charged ions. Thus given by the flux expression

$$\mathbf{N}_i = -D_i \nabla c_i - z_i u_i F c_i \nabla \phi + c_i \mathbf{u}$$

gives the flux of every ion in the solution. In this equation, $\mathbf{N}$ denotes the flux vector, D is the diffusivity, c is the concentration, z equals the charge of the ionic species, u represents the mobility, F is Faraday's constant, ϕ equals the electric potential, and $\mathbf{u}$ is the velocity. The index, i, denotes the specific species transported.

The MEMS Module includes an applications mode for this type of transport: the *Electrokinetic Flow* application mode. It relies on the assumption that the contribution of the modeled species to the total transport of current is negligible.

Transport Balances

The Electrokinetic Flow application mode addresses the transport of ionic species that are present at very low concentrations in an otherwise moderately concentrated electrolyte. To study the electric current and potential distribution, it is natural to look first at the species present at a high concentration.

Assuming that species present in moderate concentration exhibit current transport and that the concentration gradients of these species are small, you can work with the following expression for the transport of the dominating species

$$\mathbf{N}_i = -z_i u_i F c_i \nabla \phi + c_i \mathbf{u} \ .$$

The current density comes from Faraday's law

$$\mathbf{i} = -F \sum z_i \mathbf{N}_i$$

which, in combination with the electroneutrality condition for the dominating species, yields

$$\mathbf{i} = -F \sum -z_i^2 u_i F c_i \nabla \phi \ .$$

Here the electroneutrality condition eliminates the convective term; the fact that convection cannot transport current is reasonably intuitive, otherwise it would be possible to transfer current with a pump. The previous equation is actually Ohm's law for ionic current transport, which simplifies to

$$\mathbf{i} = -\kappa\nabla\phi\,.$$

A current balance gives the current and potential density in the cell

$$\nabla\cdot\mathbf{i} = 0\,,$$

which in combination with Ohm's law yields

$$\nabla\cdot(-\kappa\nabla\phi) = 0\,.$$

The Conductive Media DC application mode solves this equation easily. The potential distribution from that equation solution gives the migration term for species present at low concentrations. For these species, the mass balance comes from the full transport and mass-balance equation

$$\frac{\partial c_i}{\partial t} + \nabla\cdot(-D_i\nabla c_i - z_i u_i F c_i\nabla\phi + c_i\mathbf{u}) = R_i\,.$$

You can take the velocity vector, $\mathbf{u}$, from the equation of motion or a function of the space variables x, y and z, while the potential comes from the current balance already described; that current balance inherently includes electroneutrality.

In combination with proper boundary conditions, the current- and mass-balance equations give the current, potential, and mass distribution in electrolytic systems. This section of the manual addresses only the mass-balance equation and its boundary conditions.

One important boundary condition states that the flux of the diluted species is known and is a function f:

$$\mathbf{N}_i\cdot\mathbf{n} = f(c,\phi)\,.$$

The Electrokinetic Flow Application Mode

This application mode models a common transport mechanism in electrochemical systems, which use convection, diffusion, and migration. It can treat transport in 1D, 2D, and 3D plus axisymmetric systems in 1D and 2D.

The MEMS Module formulates this application mode to support electrolytes, which implies that the main transport of ionic current is migration and that the system automatically satisfies electroneutrality. Species transported by diffusion and migration are present in very small concentrations and do not contribute to the transport of current. You must solve for the mass balances for the transport of species in combination with a balance of current based on Ohm's law. Thus you can enter the potential field as an input in this application mode.

The dependent variables in the application mode are the mass concentrations c_1, c_2, ... c_n.

PDE Formulation

The equations for the non-conservative and conservative formulations for a species, c, are:

$$\delta_{ts}\frac{\partial c}{\partial t} + \nabla \cdot (-D\nabla c - zuFc\nabla V) = R - \mathbf{u} \cdot \nabla c \qquad \text{(non–conservative)}$$

$$\delta_{ts}\frac{\partial c}{\partial t} + \nabla \cdot (-D\nabla c - zuFc\nabla V + c\mathbf{u}) = R \qquad \text{(conservative)}$$

with F being Faraday's constant equal to 96485.3415 As/mol in SI units. To change this value use the **Application Scalar Variables** dialog box, which you open by choosing **Physics>Scalar Variables**.

The non-conservative formulation is the default for advection and diffusion types of equations in COMSOL Multiphysics. The reason for this default setting is that it assumes an incompressible fluid, hence the term $c\nabla \cdot \mathbf{u}$ is zero, and thus it does not appear in the non-conservative formulation. This omission ensures that no unphysical source term arises from a flow field that does not absolutely fulfill the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$.

Subdomain Parameters

The coefficients in the equations appear in the following table:

COEFFICIENT	VARIABLE	DESCRIPTION
δ_{ts}	`Cts_c`	Time-scaling coefficient
D	`D_c`	Diffusion coefficient tensor
D_{ij}	`Dxixj_c`	Diffusion-coefficient tensor, $x_i x_j$ component
R	`R_c`	Reaction rate
u	`um_c`	Mobility
z	`z_c`	Charge number

COEFFICIENT	VARIABLE	DESCRIPTION
u, v, w	`u_c, v_c, w_c`	Velocity in the x_1, x_2, and x_3 directions
V	`pot_c`	Potential

Note that the time-scaling coefficient and velocities are present only in the standard formulations.

For equations in space dimensions higher than one, COMSOL Multiphysics expands the isotropic diffusion coefficient, D, if you have selected it in the **Subdomain Settings** dialog box, to the diagonal of the diffusion-coefficient tensor; that is, Dx_ix_i equals D. In 1D, the diffusion coefficient tensor is identical to the diffusion coefficient.

Boundary Conditions

The available boundary conditions for this application mode are:

BOUNDARY CONDITION	DESCRIPTION
$c = c_0$	Concentration
$-\mathbf{n} \cdot (-D\nabla c - zu_m Fc\nabla V + c\mathbf{u}) = N_0$	Flux
$\mathbf{n} \cdot (-D\nabla c - zu_m Fc\nabla V + c\mathbf{u}) = 0$	Insulation/Symmetry
$\mathbf{n} \cdot (-D\nabla c - zu_m Fc\nabla V) = 0$	Convective flow
$\mathbf{n} \cdot (-D\nabla c - zu_m Fc\nabla V + c\mathbf{u}) = 0$	Axial symmetry
$\mathbf{n}_1 \cdot (-D\nabla c - zu_\mathrm{m} Fc\nabla V + c\mathbf{u})_1 = \frac{D}{d}(c_1 - c_2)$ $\mathbf{n}_2 \cdot (-D\nabla c - zu_\mathrm{m} Fc\nabla V + c\mathbf{u})_2 = \frac{D}{d}(c_2 - c_1)$	Thin boundary layer
$-\mathbf{n} \cdot (\mathbf{N}_1 - \mathbf{N}_2) = N_0$	Flux discontinuity
$-\mathbf{n} \cdot (\mathbf{N}_1 - \mathbf{N}_2) = 0$	Continuity

For the concentration boundary condition, c_0 is a user-specified concentration. Similarly, for the flux condition, N_0 is an arbitrary user-specified flux expression.

The axial symmetry condition is identical to the insulation/symmetry condition, and it is available only for relevant coordinate systems.

Use the Thin boundary layer condition to model a thin layer of a material with a small diffusion coefficient compared to the adjacent domains. The layer has the thickness d and the diffusion coefficient D. This boundary condition is only available at the border between the parts in an assembly.

The Flux discontinuity boundary condition represents a discontinuity in the mass flux across an interior boundary or a border between parts in an assembly. It is not applicable to exterior boundaries

Continuity is the default boundary condition on interior boundaries and pair boundaries; it is not applicable to exterior boundaries.

Application Mode Variables

The Electrokinetic Flow application mode uses the following expressions and coefficients in boundary conditions, equations, and for postprocessing:

NAME	TYPE	DESCRIPTION	EXPRESSION
c	S/B	Concentration	c
grad_c	S/V	Concentration gradient	$\lvert\nabla c\rvert$, $\frac{\partial c}{\partial x_i}$
dflux_c	S	Diffusive flux	$\lvert\underline{D}\nabla c\rvert$
cflux_c	S	Convective flux	$\lvert c\mathbf{u}\rvert$
mflux_c	S	Electrophoretic flux	$\lvert zuFc\nabla V\rvert$
tflux_c	S	Total flux	$\lvert -\underline{D}\nabla c - zuFc\nabla V + c\mathbf{u}\rvert$
ndflux_c	B	Normal diffusive flux	$\mathbf{n}\cdot(-\underline{D}\nabla c)$
ncflux_c	B	Normal convective flux	$\mathbf{n}\cdot c\mathbf{u}$
nmflux_c	B	Normal electrophoretic flux	$\mathbf{n}\cdot(-zuFc\nabla V)$
ntflux_c	B	Normal total flux	$\mathbf{n}\cdot(-\underline{D}\nabla c - zuFc\nabla V + c\mathbf{u})$
dflux_c_x*i*	V	Diffusive flux, x_i component	$\sum_j -D_{ij}\frac{\partial c}{\partial x_j}$
cflux_c_x*i*	V	Convective flux, x_i component	cu_i
mflux_c_x*i*	V	Electrophoretic flux, x_i component	$-zuFc\frac{\partial V}{\partial x_i}$
tflux_c_x*i*	V	Total flux, x_i component	$\sum_j -D_{ij}\frac{\partial c}{\partial x_j} - zuFc\frac{\partial V}{\partial x_i} + cu_i$

NAME	TYPE	DESCRIPTION	EXPRESSION
cellPe_c	S	Cell Peclet number	$\left\|\frac{(\mathbf{u} - zuF\nabla V)h}{\underline{D}}\right\|$
Cts_c	S	Time-scaling coefficient	δ_{ts}
udl_c	S	Dimensionless velocity	u_{dl}
D_c, Dxixj_c	S	Diffusion coefficient	D, D_{ij}
R_c	S	Reaction rate	R
um_c	S	Mobility	u
z_c	S	Charge number	z
u_c, v_c, w_c	S	Velocity of c, x_i component	u_i
V_c	S	Potential	V
N_c	B	Inward flux	N_0
c0_c	B	Concentration	c_0
gradpot_c_xi	S	Potential gradient, x_i component	$\frac{\partial V}{\partial x_i}$
Dm_c	S	Mean diffusion coefficient	$\frac{\sum_{i,j} D_{ij}\beta_i\beta_j}{\lvert\bar{\beta}\rvert}$
res_c	S	Equation residual	$\nabla \cdot (-\underline{D}\nabla c - zuFc\nabla V + c\mathbf{u}) - R$
res_sc_c	S	Shock capturing residual	$\nabla \cdot (c\mathbf{u} - zuFc\nabla V) - R$
da_c	S	Total time-scale factor	δ_{ts}

ARTIFICIAL DIFFUSION

The Electrokinetic Flow application mode supports artificial diffusion using the following methods:

- Isotropic diffusion
- Streamline diffusion
- Crosswind diffusion

See "Stabilization Techniques" on page 433 in the *COMSOL Multiphysics Modeling Guide* for more information about artificial diffusion.

Predefined Multiphysics Couplings

MEMS Module provides a number of predefined multiphysics couplings for microfluidics modeling. They appear in named folders under the **MEMS Module>Microfluidics** branch in the Model Navigator.

The predefined multiphysics couplings consist of two application modes that the MEMS Module adds to the model. In addition, default settings provide the typical coupled fields for the multiphysics application.

You can find other predefined multiphysics couplings in the COMSOL Multiphysics product line. For more information about predefined multiphysics couplings and multiphysics modeling in general, see "Multiphysics Modeling" on page 317 of the *COMSOL Multiphysics Modeling Guide*.

Flow with Species Transport

The entries in the **MEMS Module>Microfluidics>Flow with Species Transport** folder initialize models with a microfluidics application mode coupled to an application mode for modeling transport (that is, convection and diffusion) of species within the fluid.

The application modes within this folder are named according to the microfluidics application mode they use: Incompressible Navier-Stokes, Non-Isothermal Flow, Stokes Flow, Non-Isothermal Stokes Flow, and General Laminar Flow. The application mode for transport modeling is the Convection and Diffusion application mode.

The interaction between the application modes is a one-way coupling: The velocity described by the microfluidics application mode affects the convective transport of the species. The components of the velocity vector **u** from the microfluidics application mode appear in the **u**, **v**, and **w** edit fields in the **Subdomain Settings** dialog box for the Convection and Diffusion application mode.

For an example of using these predefined multiphysics couplings, see "Microchannel Cell" on page 376 in the *MEMS Module Model Library*.

Electroosmotic Flow

The entries in the **MEMS Module>Microfluidics>Electroosmotic Flow** folder initialize models with a microfluidics application mode coupled to the Conductive Media DC

application mode that models electric fields in conductive fluids and more generally in conductive materials.

The entries in this folder are named according to the microfluidics application mode they use: Incompressible Navier-Stokes, Non-Isothermal Flow, Stokes Flow, Non-Isothermal Stokes Flow, and General Laminar Flow. The Conductive Media DC application mode is the one located under **MEMS Module** in the **Model Navigator**.

This predefined multiphysics coupling differs from many other multiphysics couplings, because the coupling occurs at the boundary level only. The interaction between the application modes is a one-way coupling: The electric field described by the Conductive Media DC application mode drives the electroosmotic flow in the boundaries of the microfluidics application mode. This coupling appears in the **Electroosmotic** tab of the **Boundary Settings** dialog box for the microfluidics application modes, where components of the electric field (typically `Ex_emdc`, `Ey_emdc`, and `Ez_emdc`) appear in the $\mathbf{E_x}$, $\mathbf{E_y}$, and $\mathbf{E_z}$ edit fields.

When you create a model with this predefined multiphysics coupling, you define the boundary condition using named groups of predefined boundary conditions. Both application mode have the same group names, but the actual boundary conditions are different. For example, the *electroosmosis* group defines the *electroosmotic velocity* boundary condition in the microfluidics application mode, whereas it defines it as *electric insulation* in the Conductive Media DC application mode. Using the same group names, the logical structure of the model is the same in both application modes.

Table 8-5 shows the predefined groups and the corresponding boundary conditions for both application modes.

For an example of using these predefined multiphysics couplings, see "Low-Voltage Electroosmotic Micropump" on page 335 in the *MEMS Module Model Library*.

TABLE 8-5: PREDEFINED GROUPS FOR ELECTROOSMOTIC FLOW

GROUP NAME	BOUNDARY CONDITIONS AND SETTINGS	
	MICROFLUIDICS	CONDUCTIVE MEDIA DC
Electrode	No slip	Electric Potential • V_0 = 1
Ground	No slip	Ground
Electroosmosis	Electroosmotic velocity • E_x = Ex_emdc • E_y = Ey_emdc • E_z = Ez_emdc	Electric insulation

TABLE 8-5: PREDEFINED GROUPS FOR ELECTROOSMOTIC FLOW

GROUP NAME	BOUNDARY CONDITIONS AND SETTINGS	
	MICROFLUIDICS	CONDUCTIVE MEDIA DC
Inlet	Laminar inflow • $U_0 = 0.001$	Electric insulation
Outlet	Pressure • $p_0 = 0$	Electric insulation
Symmetry	Symmetry boundary	Electric insulation

9

The Materials/Coefficients Library

This chapter describes how to use the material libraries for MEMS materials.

Using the MEMS Material Libraries

A useful feature in COMSOL Multiphysics is the Materials/Coefficients library. In addition to the Basic Material Properties library the MEMS Module extends this library with three extra material libraries:

- Liquids and Gases, a specialized material library for use in heat transfer and fluid mechanical models. See "The Liquids and Gases Library and Material Property Functions" on page 313.
- Piezoelectric Material Properties, a material library with 23 common piezoelectric materials. See "Piezoelectric Material Properties Library" on page 321.
- MEMS Material Properties, an extended solid material library for MEMS applications. See "MEMS Material Properties Library" on page 322.

The Basic Material Properties library is included with COMSOL Multiphysics and contains properties for a limited number of basic solid materials, given as constants, and temperature-dependent properties for air and water, given as functions.

Loading Material Properties from the Library

For most application modes in COMSOL Multiphysics, you can load a material and its accompanying properties directly from within the **Subdomain Settings** dialog window; simply click the **Load** button as indicated in Figure 9-1.

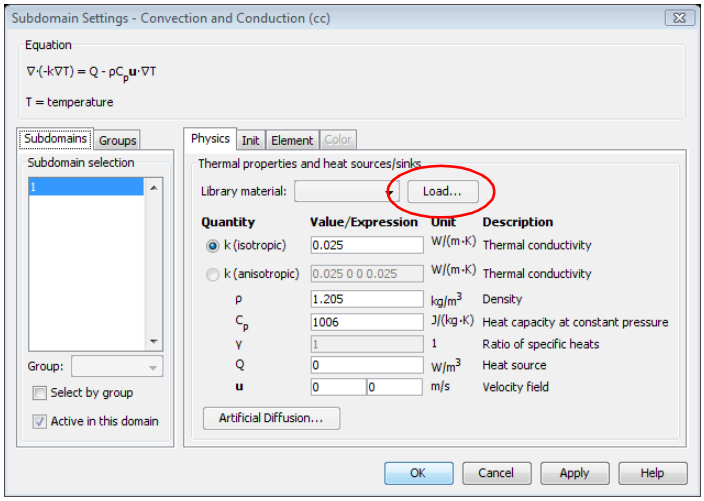


Figure 9-1: The Load button associated with library materials in the Subdomain Settings dialog box.

This action opens the **Materials/Coefficients Library** dialog box (see Figure 9-2). From there you can select a material to load.

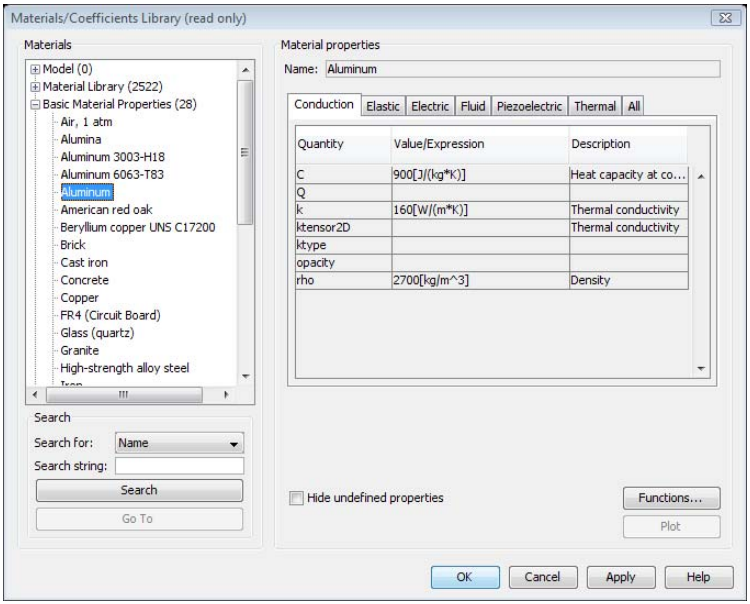


Figure 9-2: The Materials/Coefficients Library dialog box.

The **Materials** list on the left side contains the installed library folders plus a model folder that contains already-selected materials and coefficients. To load a material into the **Model** folder, select it from the **Materials** list and click **Apply** or **OK**; it now appears in the **Model** folder, which shows all the materials available for the model.

After clicking **OK**, the **Materials/Coefficients Library** dialog box closes and you return to the **Subdomain Settings** dialog box. Now the software has loaded the selected material properties into the corresponding edit fields of that dialog box. The edit fields that

contain data taken from the materials library you just selected appear in bold (see Figure 9-3).

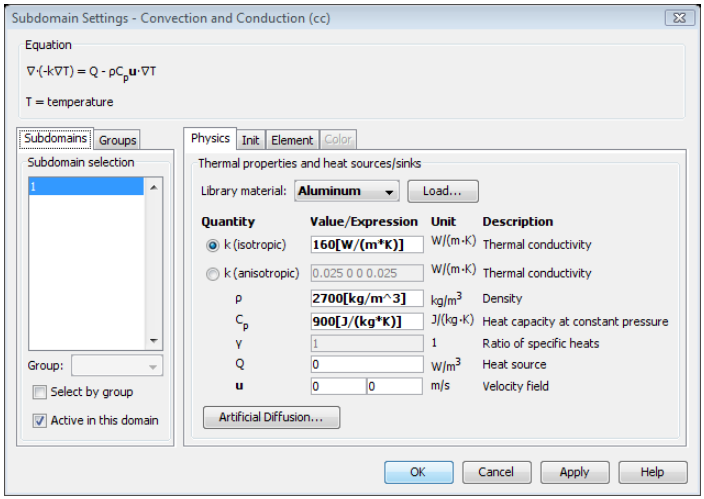


Figure 9-3: The Subdomain Settings dialog box after loading aluminum from the material library. Boldface characters indicate that the library material is active.

The Material Libraries

As noted earlier, the **Materials** list in the **Materials/Coefficients Library** dialog box contains multiple expandable folders. The list includes **Model**, **Basic Material Properties**, and additional libraries depending on the modules that your license includes.

Also, if your license includes the COMSOL Material Library, this also appears as a separate folder in the **Materials** list.

Note: The COMSOL Material Library is a separate add-on product that includes over 2500 materials and about 20,000 properties, most of which are temperature-dependent functions.

The Liquids and Gases Library and Material Property Functions

In many cases of modeling the material properties varies with the dependent variables describing the state (temperature, pressure, concentration, potential, stress, and so

on). COMSOL Multiphysics is capable of describing material properties as functions of the dependent variables. The following discussion exemplifies use and access a library with material property functions (in this case Liquids and Gases), and also how to edit it and create your own material property functions.

The Liquids and Gases material library contains thermal and fluid dynamic properties for a set of common fluids. The properties are described as temperature-dependent functions. These functions are based on data collected from scientific publications. If you click the **Load** button in the **Subdomain Settings** dialog box of any application mode and select to expand **Liquids and Gases** you can choose from various fluids, both liquids or gases (as depicted in Figure 9-4).Their properties are described at atmospheric pressure conditions.

Using Material Property Functions

In many modeling situations, the material properties vary with the dependent variables describing the state (temperature, pressure, concentration, potential, stress, and so on). In COMSOL Multiphysics you can describe material properties as functions of the dependent variables. The following discussion exemplifies how to use a library with material property functions (in this case Liquids and Gases), and also how to edit it and create your own material property functions.

The Liquids and Gases material library contains thermal and fluid-dynamic properties for a set of common fluids. All properties are given as functions of temperature and at atmospheric pressure, except the density, which for gases is also a function of the local pressure.The functions are based on data collected from scientific publications.

Note: The data-fitted functions expect temperature and pressure arguments to be expressed in the SI units *kelvin* (K) and *pascal* (Pa), respectively, and return values in appropriate SI units. Unit expressions are automatically inserted to handle the conversions to and from the model's base unit system.

If you click the **Load** button in the **Subdomain Settings** dialog box of any application mode and select to expand **Liquids and Gases** you can choose from various fluids, both liquids and gases (see Figure 9-4).

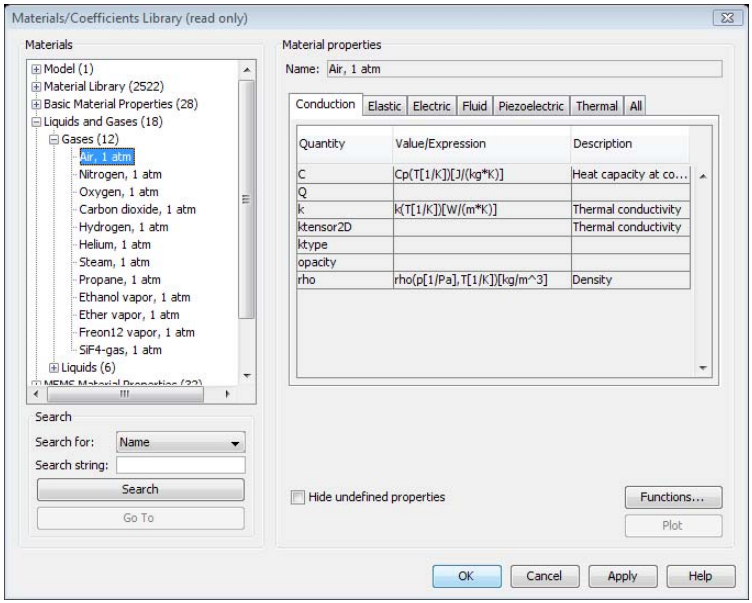


Figure 9-4: The Materials/Coefficients Library dialog box, with Liquids and Gases expanded.

If you load a material that uses a function such as those in the Liquids and Gases library, the **Subdomain Settings** dialog box looks like Figure 9-5, with function calls in the edit fields for the material properties.

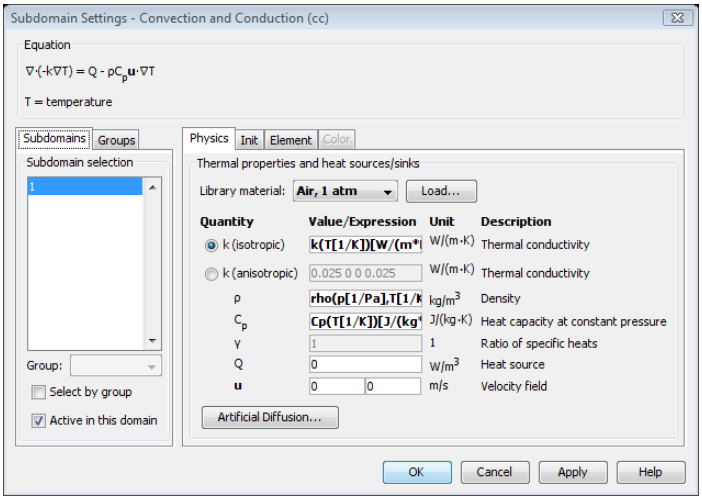


Figure 9-5: The Subdomain Settings dialog box after loading a material that uses function calls, for example, the materials from the fluid library.

In this example, the software specifies the material property for density with the function call

```
rho(p[1/Pa],T[1/K])[kg/m^3]
```

which is a function call to the material loaded, in this case **Air**. The function uses two inputs: pressure, `p`, and temperature, `T`. The default settings are based on the assumption that the temperature variable in the model is `T` and that there is a pressure variable named `p`. Being dependent variables, these are expressed in the model's selected base unit system. The unit expressions inside the function calls convert the values from the model's unit system to nondimensional numbers corresponding to SI units, while the expression between the last brackets makes sure the returned SI value is interpreted correctly in the model's unit system.

In many cases you must change these function inputs. For example, if you model only heat transfer, there is no variable for pressure. In that case you must either specify the pressure directly in the function input or set up a constant or expression for the variable *p*. This constant or expression variable must have the dimension of pressure, which you achieve by adding a pressure unit to the expression, such as `135[kPa]`.

It is easy to alter the function input values and variables: simply click inside the parentheses delimiting the function argument and replace the default symbol with the desired value or variable. In the following figure you can see such a modification where a numerical value replaces the pressure variable.

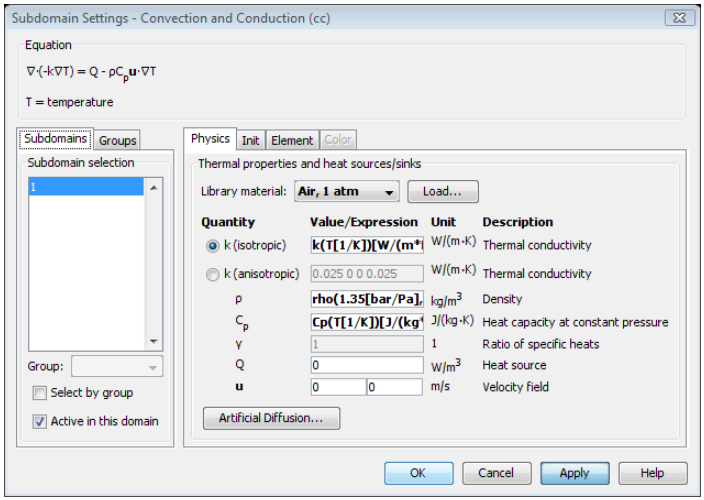


Figure 9-6: Modifying the function inputs.Notice how the unit syntax can convert argument values from any pressure unit to pascal.

In this case the function output is the density for air at the specified pressure and at the temperature given by the variable `T`.

Note: If you override one property defined by a library material by changing anything outside the function arguments, all other property expressions loose their connection to the material library. In particular, material functions appearing in other edit fields (now in plain text instead of in bold face) stop working. The proper way to edit one property of a loaded material is to change the material functions, as described later in this section, rather than editing the edit field in the **Subdomain Settings** dialog box.

USING MATERIAL PROPERTY FUNCTION CALLS OUTSIDE THE SUBDOMAIN SETTINGS

You can also use a library material property function in a model in places other than the **Subdomain Settings** dialog box. One example might be to combine several

properties in an expression in the **Scalar Expressions** dialog box. To do so, you must first load the library material into the model using the **Materials/Coefficients Library** dialog box. This dialog box opens either, as described above, from the **Subdomain Settings** dialog box (then in read only mode), or directly from the **Options** menu. The currently loaded materials and their assigned names are listed in the **Model** folder in the **Materials** tree. (Figure 9-7).

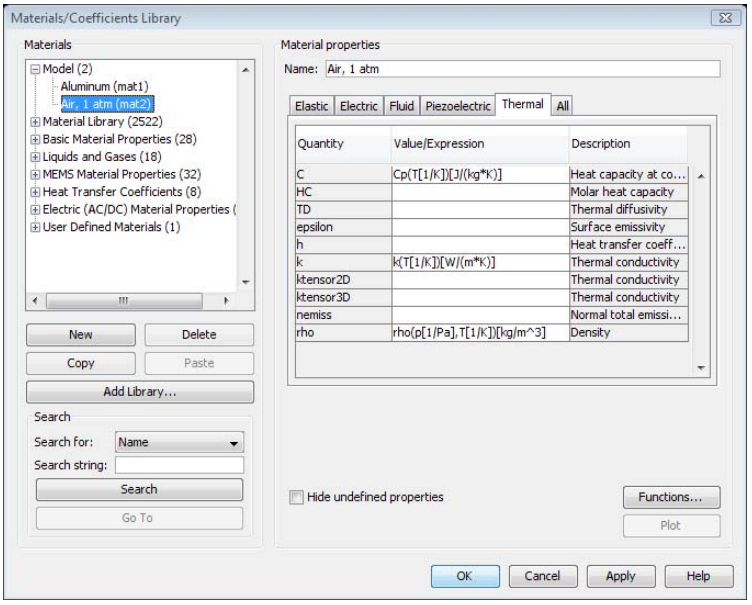


Figure 9-7: Viewing the materials loaded into a model along with their names in COMSOL Multiphysics.

In the above figure, two materials are already loaded into the model. COMSOL Multiphysics allocates local material names on the form `matX`, where `X` is a running number assigned in the order in which materials were loaded into the model. In this case, the user loaded **Aluminum** first, so it takes the name `mat1`.

To use a particular material-property function, you can start by copying the syntax shown in the **Value/Expression** column in the **Materials/Coefficients Library** dialog box. Then you must add `matX_` in front of the function call. Thus, referring to Figure 9-7, to evaluate the function for thermal conductivity of air at 350 K, the syntax is

```
mat2_k(350)[W/m*K]
```

Editing Material Properties

To change a property of a loaded material, choose **Materials/Coefficients Library** from the **Options** menu. This opens the dialog box in edit mode. Then select the desired material in the **Materials** list, click the **Value/Expression** field of interest, and change the expression as depicted in Figure 9-8. In this example the user is adding `k_turb` to the original function for thermal conductivity, where `k_turb` is a variable that must be defined elsewhere in the model.

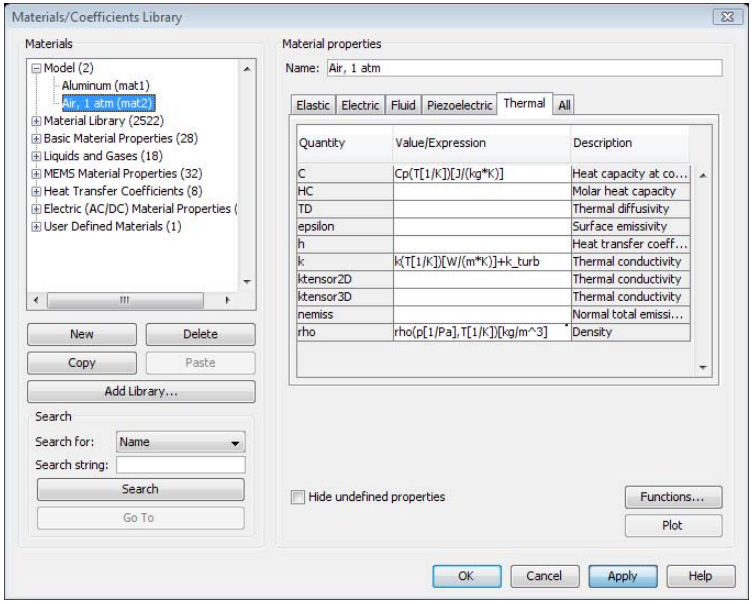


Figure 9-8: Editing a material property.

If you want to edit the function describing some material property, click the **Functions** button. This opens the **Functions** dialog box, where you can view and edit any function describing the material.

After changing a material property in this way, you need to reload all subdomain settings that use the material; otherwise the function call will not work. To reload the material, simply go to **Physics>Subdomain Settings** and select the modified material from the **Library material** list. The new expression then shows up in the edit field for

the corresponding material property in the **Subdomain Settings** dialog box (see Figure 9-9).

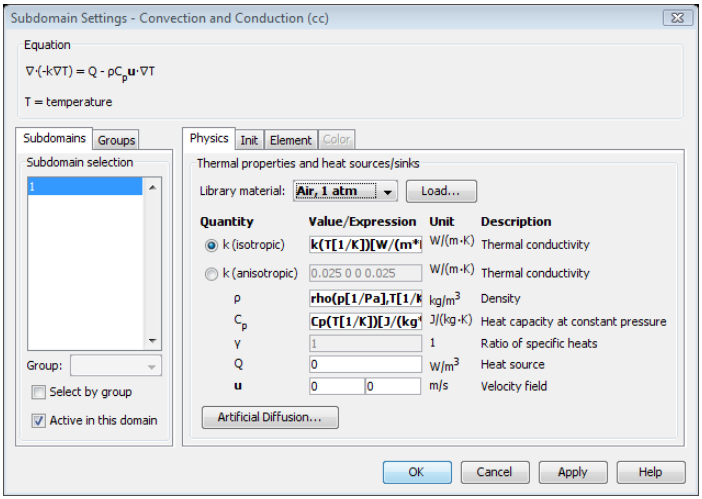


Figure 9-9: The update of a material property expression in the Subdomain Settings dialog box.

Note that you can only edit materials currently in the **Model** folder in this way and that any changes are local to the current model. The original material in the library file remains intact. Within the **Materials/Coefficients Library** dialog box you can, however, change a material's name and then **Copy** and **Paste** it into one of the other libraries. Afterwards, clicking **OK** saves the new material for future use in the corresponding library text file.

COMSOL Multiphysics also allows you to set up new materials by creating a new library file, as well as change the existing files using any text editor. A detailed description of this process appears in the *COMSOL Multiphysics User's Guide*, where you also find complete documentation on the functionality of the Materials/ Coefficients Library.

Piezoelectric Material Properties Library

The Piezoelectric Material Properties library ships with the Acoustics Module, MEMS Module, and Structural Mechanics Module. It contains the following piezoelectric materials:

MATERIAL
Barium Sodium Niobate
Barium Titanate
Barium Titanate (poled)
Lithium Niobate
Lithium Tantalate
Lead Zirconate Titanate (PZT-2)
Lead Zirconate Titanate (PZT-4)
Lead Zirconate Titanate (PZT-4D)
Lead Zirconate Titanate (PZT-5A)
Lead Zirconate Titanate (PZT-5H)
Lead Zirconate Titanate (PZT-5J)
Lead Zirconate Titanate (PZT-7A)
Lead Zirconate Titanate (PZT-8)
Quartz
Rochelle Salt
Bismuth Germanate
Cadmium Sulfide
Gallium Arsenide
Tellurium Dioxide
Zinc Oxide
Zinc Sulfide
Ammonium Dihydrogen Phosphate
Aluminum Nitride

All materials define the following material properties needed for piezoelectric modeling:

MATERIAL PROPERTY	DESCRIPTION
c_E	Elasticity matrix
e	Coupling matrix, stress-charge
ε_{rS}	Relative permittivity, stress-charge
s_E	Compliance matrix
d	Coupling matrix, strain-charge
ε_{rT}	Relative permittivity, strain-charge
ρ	Density

MEMS Material Properties Library

The MEMS Material Properties library ships with the Acoustics Module, MEMS Module, and Structural Mechanics Module. It contains 33 materials commonly used in MEMS applications. The materials are divided into the following groups: Metals, Semiconductors, Insulators, and Polymers.

The basic structure of this library comes from the book *Microsensors, MEMS, and Smart Devices* (Ref. 3). The material properties come from two primary sources: the *CRC Handbook of Chemistry and Physics* (Ref. 1) and *MacMillan's Chemical and Physical Data* (Ref. 2). Some of the mechanical properties in the library are instead more MEMS-specific values from *The MEMS Handbook* (Ref. 4), and most of the semiconductor properties are values from Ref. 5. Ref. 6 provides a valuable resource for cross-checking the insulation material properties.

The table below lists the materials and their corresponding groups:

MATERIAL	GROUP
Aluminium (Al)	Metals
Silver (Ag)	Metals
Gold (Au)	Metals
Chrome (Cr)	Metals
Indium (In)	Metals
Titanium (Ti)	Metals
Iron (Fe)	Metals
Nickel (Ni)	Metals

MATERIAL	GROUP
Lead (Pb)	Metals
Palladium (Pd)	Metals
Platine (Pt)	Metals
Antimon (Sb)	Metals
Tungsten (W)	Metals
C [100]	Semiconductors
GaAs	Semiconductors
Ge	Semiconductors
InSb	Semiconductors
Si(c)	Semiconductors
Poly-Si	Semiconductors
Silicon (single-crystal)	Semiconductors
Al2O3	Insulators
SiC (6H)	Insulators
Si3N4	Insulators
SiO2	Insulators
ZnO	Insulators
Borosilicate	Insulators
Nylon	Polymers
PMMA	Polymers
Polymide	Polymers
Polyethylene	Polymers
PTFE	Polymers
PVC	Polymers

REFERENCES

1. D.R. Lide (Editor-in-chief), *CRC Handbook of Chemistry and Physics*, 84th edition, CRC Press, 2003.

2. A.M. James and M.P. Lord, *MacMillan's Chemical and Physical Data*, MacMillan's Press, 1992.

3. J.W. Gardner, V.K. Varadan, and O.O. Awadelkarim, *Microsensors, MEMS, and Smart Devices*, John Wiley & Sons, 2001.

4. M. Gad-el-Hak (editor), *The MEMS Handbook*, CRC Press, 2002.

5. *New Semiconductor Materials. Characteristics and Properties*, http://www.ioffe.ru/SVA/NSM, 2003.

6. *Ceramics WebBook*, http://www.ceramics.nist.gov/srd/scd/scdquery.htm, 2003.

10

Glossary

This glossary contains finite-element modeling terms specific to the MEMS Module and its applications. For mathematical terms as well as geometry and CAD terms specific to the COMSOL Multiphysics software and its documentation, please see the glossary in the *COMSOL Multiphysics User's Guide*. To find references in the documentation set where you can find more information about a given term, see the index.

Glossary of Terms

absorption Uptake of a gas into the bulk of a liquid. Gas absorption takes place, for example, in the liquid of a scrubber tower where an up-streaming gas is washed by a down-going flow of a scrubber solution.

accelerometer A sensor that measures acceleration or gravitational force. See also *sensor*.

actuator A device that by means of an agent, such as electric field or heat, produces a desired effect on a target or on the surroundings.

adsorption Attachment of a molecule or atom to a solid surface. Adsorption involves a chemical bond between the adsorbed species and the surface.

ALE See *arbitrary Lagrangian-Eulerian method*.

arbitrary Lagrangian-Eulerian (ALE) method A technique to formulate equations in a mixed kinematical description. An ALE referential coordinate system is typically a mix between the material (Lagrangian) and fixed (Eulerian) coordinate systems.

aspect ratio The ratio of the dimensions of a device in different directions, for example, the ratio of height to width. See also *lateral aspect ratio* and *vertical aspect ratio*.

biosensor A general term for sensor devices that either detect biological substances or use antibodies, enzymes, or other biological molecules in their operation. Biosensors are a subcategory of chemical sensors.

bonding A process by which one type of substrate is firmly attached to the surface of another.

buckling The sudden collapse or reduction in stiffness of a structure under a critical combination of applied loads.

cantilever beam A beam with one end fixed and one end free.

capacitive sensor A sensor that produces a signal due to a change in its capacitance.

comb drive A MEMS device consisting of inter-digitated fingers similar to a comb.

convection Transport of molecules or heat due to the movement of the surrounding medium.

damping An effect that attenuates or reduces mechanical oscillations or vibrations.

dielectrophoresis Migration of polarizable particles in an electrolyte in a nonuniform applied electric field.

diffusion Transport of material resulting from the random motion of molecules in the presence of a concentration gradient.

EDL See *electric double layer.*

elastic deformation A nonpermanent deformation that recovers its shape completely upon the release of an applied stress.

electric double layer (EDL) At the contact of a solid and a polar fluid (such as water), the solid acquires an electric charge. This charge attracts ions within the fluid, and a narrow fluid layer of opposite charge, the Stern layer, forms on the boundary. In addition, adjacent to the Stern layer, a wider layer with the same charge as in the Stern layer forms in the fluid. Together, the Stern layer and the wider layer (called the diffuse or Gouy-Chapman layer) form the electric double layer. Due to the close distance between the charges, the Stern layer is fixed on the surface, but the more distant diffuse layer can move.

electrokinetics Study of the motion of charged particles under an applied electric field in moving substances such as water.

electrokinetic flow Transport of fluid or charged particles within a fluid by means of electric fields. See also *electroosmosis, electrophoresis, electrothermal flow*, and *dielectrophoresis.*

electrolyte A solution that can carry an electric current through the motion of ions.

electroosmosis Fluid flow in a narrow channel produced by the movement of the *electric double layer* (EDL) along the channel boundary under the influence of an applied electric field. Also, fluid flow through a membrane under the influence of an applied electric field. See also *electric double layer.*

electroosmotic flow See *electroosmosis*.

electrophoresis Migration of charged electrolyte ions in an applied electric field.

electrothermal flow Fluid flow resulting from an applied non-uniform AC electric field on a fluid. The Joule heating changes the fluid's electrical properties locally, and that effect, together with the power gradient of the AC electric field, results in fluid motion.

Eulerian Model described and solved in a coordinate system that is fixed. See also *Lagrangian* and *arbitrary Lagrangian-Eulerian method*.

frequency-response analysis An analysis solving for the steady-state response from a harmonic excitation. Typically a frequency sweep is performed, solving for many excitation frequencies.

fully developed laminar flow Laminar flow along a channel or pipe that has velocity components only in the main direction of the flow. The velocity profile perpendicular to the flow does not change downstream in the flow.

Green-Lagrange strain A measure of nonlinear strain 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.

Helmholtz-Smoluchowski equation Gives the velocity of a parallel electroosmotic flow for an applied electric field.

initial strain The strain in a stress-free structure before it is loaded. See also *strain* and *residual strain*.

initial stress The stress in a non-deformed structure before it is loaded. See also *stress* and *residual stress*.

Joule heating The increase in temperature of a medium as a result of resistance to an electric current flowing through it.

Knudsen number A dimensionless number that provides a measure of how rarefied a gas flow is, in other words, the average distance between the gas molecules compared

to the length scale of the flow. The following equation defines the Knudsen number Kn where λ is the mean free path of the molecules and L is a length scale characteristic to the flow.

$$\mathrm{Kn} = \frac{\lambda}{L}$$

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.

lateral aspect ratio The ratio of the length of a structure in the plane of a wafer to its width in that plane. See also *aspect ratio* and *vertical aspect ratio*.

MEMS Short for microelectromechanical systems, an acronym derived from the words Micro Electro Mechanical System. More generally, *MEMS* refers to systems, devices, and components in microscale size and where the physics are not limited only to electrical or mechanical phenomena.

microfluidics Study of the behavior of fluids at the micro scale. Also refers to MEMS fluidic devices.

microsystem See *MEMS*.

migration The nonrandom movement of particles under an external force.

mobility The relation between the drift velocity of a molecule within a fluid and the applied electric field.

piezoelectricity The ability of certain crystalline materials to produce an electric voltage when subjected to mechanical stress. Inversely, the material's ability to change shape when an external voltage is applied.

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 where all out-of-plane stress components are assumed to be zero.

proof mass A predetermined test mass in a measurement device or machine that serves as the reference mass for the quantity to be measured.

residual strain Strain remaining in a structure after some operation, for example, the strain resulting from cooling a system after high-temperature bonding. It often appears as the initial strain for any consequent tasks. See also *strain* and *initial strain*.

residual stress Stress remaining in a structure after some operation, for example, stress resulting from cooling a system after high-temperature bonding. It often appears as the initial stress for any consequent tasks. See also *stress* and *initial stress*.

Reynolds number A dimensionless number classifying how laminar or turbulent a flow is. The *Reynolds number* Re is a measure of the relative magnitude of the flow's viscous and inertial forces. It is defined by the following equation where ρ is the fluid density, η is the dynamic viscosity, ν is its kinematic viscosity, u is a velocity characteristic to the flow, and L is a length scale characteristic to the flow.

$$\mathrm{Re} = \frac{\rho u L}{\eta} = \frac{u L}{\nu}$$

sensor A device that measures a physical variable such as temperature or pressure and converts it (usually) to an electrical signal.

squeezed-film damping The damping effect of a fluid between two solid surfaces when the distance between them is small compared to their area. Also referred to as *thin-film damping*.

strain Relative change in length, a fundamental concept in structural mechanics.

stress Internal forces in a 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.

transducer A device that converts one type of energy to another. It often refers to a device that responds to a physical parameter and converts it to an electrical signal.

thin-film damping See *squeezed-film damping*.

vertical aspect ratio The ratio of the height of a structure perpendicular to a wafer's surface to its depth in the wafer's plane.

INDEX

3D Conductive Media DC
application mode 212
3D Electrostatics
application mode 217
3D electrostatics
application mode 222

A
absolute tolerances 154
absorbing boundaries 112
AC electric field 27
AC electroosmosis 25
accommodation coefficient 205
actuator models
microresistor beam 30
ALE method 17
analysis capabilities 53
analysis types 55, 90, 165
damped eigenfrequency 86
for piezoelectric modeling 165
frequency response 84
harmonic 84
static 83
static elasto-plastic material 90
time-dependent 84
transient 84
anisotropic material 69, 94
antiperiodic boundaries 230
application mode
Conductive Media DC (3D) 212
Electrokinetic Flow 22, 299
Electrostatics 26, 217, 222
Heat Transfer by Conduction 22
Heat Transfer by Conduction and Convection 22
application mode properties 56
analysis type 90, 165
create frame 117
large deformation 91
weak constraints 91, 165
Application Mode Properties dialog box 90, 164
application mode variables 56
application modes
Axial Symmetry, Stress-Strain 63
continuum 64
list of 52
Piezo Axial Symmetry 187
Piezo Plane Strain 187
Piezo Plane Stress 186
Piezo Solid 186
piezoelectric 164
Plane Strain 62
Plane Stress 61
application scalar variables 55, 56, 301
Application Scalar Variables dialog box 55
arbitrary Lagrangian-Eulerian method 17
area effects, in FSI models 129
assembly 117
`auglagiter` variable 122
augmented Lagrangian method 86, 133, 155
axial symmetry
symmetry axis 63, 188
Axial Symmetry, Stress-Strain application mode 63

B
boundary condition
electric shielding 21, 216, 219
floating potential 21, 215, 219
slip 23
boundary conditions
for electric currents 178
in the Level Set Two-Phase Flow appli-

cation mode 291
Boundary Settings dialog box 55
Boussinesq approximation 249, 279
bulk modulus 71

C
cantilever beam 17
capacitance 20
Cauchy stress 66
Cauchy-Green tensor 66, 76
charge
line 220
point 221
charges 182
chemical reactions 27
Clausius-Mosotti factor 26
cohesion sliding resistance 87
complementarity conditions 73
complex modulus 146
complex notation 85
complex permittivity 26
compliance matrix 67, 170, 171
COMSOL Material Library 313
conduction
in heat transfer 21
Conductive Media DC application mode 21, 212
conductivity
in Joule heating 213
temperature-dependent 213
conservation of level set function 285
constitutive equation 67
constitutive form
piezoelectric material 167
strain-charge 168
stress-charge form 167
constitutive relations 78
constraints 101
coordinate systems for 103, 181
for piezoelectricity 180
general notation for 104
standard notation for 104
symbols for 150
contact angle 292
contact map operator 86
Contact Modeling 117
contact modeling 155
friction 88
manual scaling 155
theory for 86
contact pairs 117, 133
continuity in periodic boundaries 230
continuum application modes 64
Axial Symmetry, Stress-Strain 63
Plane Strain 62
Plane Stress 61
using damping in 110
convection 294
in heat transfer 21
coordinate system 136
local geometrical 137
material 97
user-defined 138
work plane 141
Coordinate System Settings dialog box 139
coordinate systems
defining loads in 107
for constraints 181
for loads 183
for material properties 170, 174
in constraints 103
Coulomb friction 88
coupling
thermal-expansion 22
coupling operators 120
current flow 214
current source 212, 215

line 217
point 217
curvature 287
cylindrical coordinates 63, 187

D damped eigenfrequency analysis 86
damping 145
equivalent viscous 147, 184
explicit 147
factor 145
in continuum application modes 110
loss factor 110, 146, 184, 185
mass 145
no damping 110
page for specifying 110, 184
piezoelectric 184
Rayleigh 84, 110, 145, 184
slide film 193
squeezed film 27, 193
stiffness 110, 145, 185
viscous 147
Damping page 110, 184, 185
Debye length 24
decay factor 86
deformation gradient 65
deformed frame 81
density 97, 100
in fluids 250, 252
dependent variables 52
dialog box
Application Mode Properties 90, 164
Contact Pairs 119
Coordinate System Settings 139
Elasticity matrix 95, 168
Elasto-plastic material 96
H Matrix 105
Relative permittivity 168
Solver Parameters 152
dielectrophoresis 26
diffusion coefficient 294
dilatational viscosity 266
direct piezoelectric effect 158
displacement gradient 77
distributed loads 106
distributed resistance 214
dynamic friction coefficient 88
dynamic viscosity 250, 251, 266

E Edge Settings dialog box 55
effective plastic strain 75
elasticity matrix 67, 100
Elasticity matrix dialog box 95, 168
elasto-plastic material 73, 90, 95
Elasto-plastic material dialog box 96
`elcontact` element 86
Electric BC page 176
electric boundary conditions 176
electric currents
boundary conditions for 178
electric displacement 177
electric double layer 24, 255
electric insulation 214
electric potential 177, 178
electric shielding 223
Electrokinetic Flow
application mode 299
electrokinetic flow 22, 24
electrolyte 301
electroosmosis 24
AC 25
electroosmotic flow 24
electrophoresis 25
electrophoretic velocity 25
electroquasistatic criterion 19
Electrostatics
application mode 217, 222
electrostatics 20, 212
Electrostatics application mode 21, 26

electrothermal force 26
electrothermally driven flow 26
elements, mixed 72
engineering strain 66
entropy 79
equivalent viscous damping 147, 184
excitation frequency 84, 108, 165
explicit damping 147
extended Reynolds equation 198

F Faraday's constant 301
Faraday's law 299
film damping 191
Film Damping application mode 28
film-damping multiphysics couplings 208
first Piola-Kirchhoff stress 67, 76
fixed current
  in port boundary condition 241
fixed current density
  in port boundary condition 242
flexibility matrix 67
floating potential 224
  boundary condition 215, 219
flow
  electrothermally driven 26
flow rate, relative 198
fluid loads 129
fluid properties 309
flux discontinuity 295, 303
follower loads 66, 81, 106
force computations 237
force variables
  naming of 239
forced voltage
  in port boundary condition 241
frame 117
  deformed 81
  reference 81
frequency response analysis 84, 108
  phase 84
friction 88

G gap distance 87
General Laminar Flow application mode 278
general notation for constraints 104
geometric nonlinearity 65
gravity force 290
Green strains 65, 66
Green-Lagrange strains 65
ground potential 215, 219
guess variables 113

H H Matrix dialog box 105
hardening function 101
hardening model
  isotropic 74
  kinematic 74
harmonic loads 84
heat capacity 79
heat dissipation 147
heat transfer 21
  by conduction 21
  by convection 21
  by radiation 21
Helmholtz-Smoluchowski equation 24, 25, 255
hyperelastic material models
  Mooney-Rivlin 76
  Neo-Hookean 76
  strain energy function 77
hyperelastic materials 76

I ideal plastic material 74
ideal plasticity 75
IEEE standard, for piezo theory 159
incompressibility 76
infinite elements 232
initial

strain 80
stress 80
initial fluid interface, boundary condition for 291
initial stresses 111
initializing the level set function 286, 290
invariants 76
inverse piezoelectric effect 158
isotropic hardening 74, 75
isotropic material 67, 93
isotropic tangent modulus 100

J Joule heating 26, 213

K kinematic hardening 74, 75
kinematic tangent modulus 101
kinetic expressions 294
Knudsen number 23, 197

L large deformation 91
large deformation option 18
Level 289
Level Set and Laminar Flow application mode 289
level set function
initializing 286, 290
level set method 289
for two-phase flow 289
level set methods 283
Level Set Two-Phase Flow 289
line charge 220
line current source 217
linear elastic material 67
Liquids and Gases library 313
load symbols 149
loads 182
coordinate systems for 107, 183
distributed 106
page for specifying 105
phase 108
thermal 108
units for 107, 183, 184
local coordinate system 137
loss factor 110, 111, 146, 185
loss factor damping 72, 110, 146, 184
loss modulus 146
lumped parameters 241

M manual scaling 155
map operator 120
mapped infinite elements 232
mass conservation, for level set equations 290
mass damping parameter 110, 145, 185
mass flux vector 294
master boundary 87
master domains 133
material
anisotropic 69, 94
coordinate system 97
elasto-plastic 73, 95
ideal plastic 74
isotropic 67, 93
linear elastic 67
mixed formulation 70
model 93
Mooney-Rivlin 76
Neo-Hookean 76
orthotropic 68, 93
material libraries 57, 309, 313
material orientation
in piezoelectric application modes 168
Material page 93
material properties
coordinate system for 170
from material libraries 311
settings for 166
materials
hyperelastic 76

piezoelectric 158
matrix
compliance 67, 170, 171
elasticity 67
flexibility 67
piezo coupling 171
maximum friction traction 87
Maxwell stress tensor 237
Maxwell's stress tensor 20
mean free path 205, 268
MEMS Material Properties library 322
MEMS Module 3.4
new features in 7
mesh
scale factor 13
mesh mapping 13
meshing 13
microfluidics 22
microresistor 22
microresistor beam 30
microvalve 21, 22
migration 299
mixed formulation 70, 72, 77
mobility 301, 304
modified Reynolds equation 194
for perforated structures 198
selecting damping type for 204
Mooney-Rivlin material model 76
moving interfaces 283
multiphysics contact 135
multiphysics couplings
for electroosmotic flow 25
predefined 305

N Navier-Stokes equations 23, 249, 251, 279
Neo-Hookean 76
new features in MEMS Module 3.4 7
no slip boundary condition 253
non-conservative formulation 296
nonlinear geometric effects 19
nonlinear geometry 65
non-Newtonian fluid 266
non-Newtonian fluids 266
normal stress 66

O Ohm's law 212, 300
orthotropic material 68, 93

P page
Constraint 101, 180
Damping 110, 184
Initial Stress and Strain 111
Load 105
Material 93
parametric analysis 15
parametric solver 42
penalized friction traction 87
penalty factors 134
penalty parameter 122
perfectly matched layers 112, 232
perfectly plastic material 75
perforation admittance 199
Perforation Profile Reynolds method 199
phase, in harmonic loads 84
phases 108
Piezo Axial Symmetry application mode 187
piezo coupling matrix 171
Piezo Plane Strain application mode 187
Piezo Plane Stress application mode 186
Piezo Solid application mode 186
piezoelectric
analysis types 165
piezoelectric application modes
application mode properties 164
piezoelectric effect 158
Piezoelectric Material Properties library 321

piezoelectric materials 158
plane strain 62
plane stress 61
plasticity 73
Plot Parameters dialog box 55
PMLs
see perfectly matched layers
point charge 221
point current source 217
Point Settings dialog box 55
Poisson's ratio 97, 99
predefined multiphysics couplings 305
pressure 250, 252, 258
as dependent variable 77
as Vanka variable 154
defined as negative mean stress 70
specifying a reference for 277
pressure sensor 22
pressure-driven flow 23
principle of virtual work 82

Q quality factor 86

R radiation
in heat transfer 21
rarefaction 197
Rayleigh damping 84, 110, 145, 184
Reaction Engineering Lab 27
reaction term 294
reference frame 81
reference temperature 78
reinitialization, of level set function 285
relative flow rate 198
relative permittivity 268
Relative permittivity dialog box 168
residual stress 18
resistive heating 213
resonator 22
Reynolds equation, modified 194

S scalar variables 91, 165
second Piola-Kirchhoff stress 67, 76, 78, 147
segregated solver 86
shallow channel approximation 253, 267
shear modulus 99, 172
shear strain 64
shear stress 66
shear-rate dependent viscosity 266
SI units 316
slave boundary 87
slave domains 133
slide film damping 28, 193
slip boundary condition 23
slip effects 197
solution method
augmented Lagrangian 155
solver manager 15
solver method
augmented Lagrangian 86
solver parameters 14
absolute tolerance 154
linear system solver 153
manual scaling 155
symmetric matrices 152
Solver Parameters dialog box 152
solver settings
for contact modeling 155
squeezed film damping 27, 193
squeezed-film damping 23
standard notation for constraints 104
static analysis 83
static coefficient of friction 88
stiffness damping 110, 185
stiffness damping parameter 111, 145, 185
Stokes flow 281
weakly compressible 251, 282
storage modulus 146

strain 64, 66
axial symmetry 64, 188
effective plastic 75
elastic 64
engineering 66
engineering form 64
Green 66
initial 80, 111
invariants 76
shear 64
tensor form 64
thermal 64, 78
strain energy 81
strain energy function 77
strain reference temperature 109
strain temperature 109
strain tensor 64
strain-charge form 168
strain-displacement relation 64
large displacement 65
small displacement 64
stress 66
Cachy 66
first Piola-Kirchhoff 67, 76
initial 80, 111
normal 66
second Piola-Kirchhoff 67, 76, 78, 147
shear 66
stiffening 18
tensor 66
stress vector 79
stress-charge form 167
stress-strain relation 66
structural damping 184
Subdomain Settings dialog box 55
surface charge 177, 218
surface tension force 289
symbols
for constraints 150
for loads 149
symmetric matrices 152
symmetry axis 63, 188

T temperature coefficient 213
temperature, reference 78
temperature-dependent conductivity 213
theory
for contact modeling 86
thermal effects 21
thermal expansion 30
thermal expansion coefficient 100
anisotropic 98
isotropic 97
orthotropic 98
thermal expansion vector 78, 100
thermal loads 108
thermal strain 64, 78
thermal-expansion coupling 22
thermoelastic damping 22
thickness of cross section 267
thin boundary layer
for diffusion 295, 302
time-dependent analysis 84
torque computations 237
total force 253
transient analysis 84
two-phase flow 289
typographical conventions 3

U unbounded domains, modeling of 232
undamped model 148
unit normals 287
user-defined coordinate system 138

V variables
application mode 56
application scalar 55, 56

dependent 52
velocity
boundary condition for 257, 259
in fluids 250, 252, 253, 258
viscosity 249, 279
dilatational 266
dynamic 266
shear-rate dependent 266
viscous damping 147
viscous force 252
volume force field 250, 252
volume forces 266
volume ratio 65
volumetric heat capacity 79

W weak constraints 91, 165
Weakly Compressible Navier-Stokes application mode 280
weakly compressible Stokes flow 251, 282
wetted wall, boundary condition for 291

Y yield function 100
yield stress level 100
Young's modulus 68, 97, 98, 99

Z zeta potential 25