COMSOL Multiphysics®

User´s Guide

VERSION 4.3

COMSOL

COMSOL Multiphysics User’s Guide

Version: May 2012 COMSOL 4.3

Contact Information

Visit www.comsol.com/contact for a searchable list of all COMSOL offices and local representatives. From this web page, search the contacts and find a local sales representative, go to other COMSOL websites, request information and pricing, submit technical support queries, subscribe to the monthly eNews email newsletter, and much more.

If you need to contact Technical Support, an online request form is located at www.comsol.com/support/contact.

Other useful links include:

- Technical Support www.comsol.com/support
- Software updates: www.comsol.com/support/updates
- Online community: www.comsol.com/community
- Events, conferences, and training: www.comsol.com/events
- Tutorials: www.comsol.com/products/tutorials
- Knowledge Base: www.comsol.com/support/knowledgebase

Part No. CM020002

Contents

Chapter 1: Introduction

Chapter 2: The COMSOL Modeling Environment

The COMSOL Desktop Environment **68**

Introduction to the Model Wizard and the Model Builder **76**

Introduction to the COMSOL Windows **95**

Chapter 3: Global and Local Definitions

Chapter 4: Visualization and Selection Tools

Chapter 5: Geometry Modeling and CAD Tools

Chapter 6: Materials

Chapter 7: Building a COMSOL Model

Chapter 8: Overview of the Physics Interfaces

Chapter 9: The AC/DC Branch

Chapter 10: The Acoustics Branch

Chapter 11: The Chemical Species Transport Branch

Chapter 12: The Fluid Flow Branch

Chapter 13: The Heat Transfer Branch

Chapter 14: The Structural Mechanics Branch

Chapter 15: Equation-Based Modeling

Chapter 16: Sensitivity Analysis

Chapter 17: Multiphysics Modeling

Chapter 18: The Deformed Geometry and Moving Mesh Interfaces

Chapter 19: Meshing

Chapter 20: Solvers and Study Types

Chapter 21: Results Evaluation and Visualization

Chapter 22: Running the COMSOL Software

Chapter 23: Glossary

1

Introduction

Welcome to COMSOL Multiphysics®! This *User's Guide* details features and techniques that help you throughout all of your COMSOL Multiphysics modeling in Version 4.3 using the COMSOL Desktop environment. For example, detailed information about how to build model geometries in COMSOL Multiphysics, how to create a mesh for the finite elements, how to create parameters and variables used within a model, how to add the physics and material properties, and how to solve and display the results, are all explained. The full set of documentation shows you, step by step, how to tap into many functions and capabilities available in the COMSOL environment.

This introductory chapter provides an overview of COMSOL Multiphysics and its product family and includes these topics:

- About COMSOL Multiphysics
- The COMSOL Modules
- Overview of the User's Guide
- The COMSOL Documentation Set

About COMSOL Multiphysics

COMSOL Multiphysics is a powerful interactive environment for modeling and solving all kinds of scientific and engineering problems. The software provides a powerful integrated desktop environment with a *Model Builder* where you get full overview of the model and access to all functionality. With COMSOL Multiphysics you can easily extend conventional models for one type of physics into multiphysics models that solve coupled physics phenomena—and do so simultaneously. Accessing this power does not require an in-depth knowledge of mathematics or numerical analysis.

Using the built-in *physics interfaces* and the advanced support for material properties, it is possible to build models by defining the relevant physical quantities—such as material properties, loads, constraints, sources, and fluxes—rather than by defining the underlying equations. You can always apply these variables, expressions, or numbers directly to solid and fluid domains, boundaries, edges, and points independently of the computational mesh. COMSOL Multiphysics then internally compiles a set of equations representing the entire model.

You access the power of COMSOL Multiphysics as a standalone product through a flexible graphical user interface (GUI) or by script programming in Java or the MATLAB® language (requires the COMSOL LiveLink for MATLAB).

Using these physics interfaces, you can perform various types of studies including:

- Stationary and time-dependent (transient) studies
- Linear and nonlinear studies
- Eigenfrequency, modal, and frequency response studies

When solving the models, COMSOL Multiphysics uses the proven *finite element method (FEM)*. The software runs the finite element analysis together with adaptive meshing (if selected) and error control using a variety of numerical solvers. The studies can make use of multiprocessor systems and cluster computing, and you can run batch jobs and parametric sweeps. A more detailed description of this mathematical and numerical foundation is in the *COMSOL Multiphysics Reference Guide*.

COMSOL Multiphysics creates *sequences* to record all steps that create the geometry, mesh, studies and solver settings, and visualization and results presentation. It is therefore easy to parameterize any part of the model: Simply change a node in the model tree and re-run the sequences. The program remembers and reapplies all other information and data in the model.

Partial differential equations (PDEs) form the basis for the laws of science and provide the foundation for modeling a wide range of scientific and engineering phenomena. You can use COMSOL Multiphysics in many application areas, for example:

- Acoustics
- Bioscience
- Chemical reactions
- Corrosion and corrosion protection
- Diffusion
- Electrochemistry
- Electromagnetics
- Fluid dynamics
- Fuel cells and electrochemistry
- Geophysics
- Heat transfer
- Microelectromechanical systems (MEMS)
- Microfluidics
- Microwave engineering
- Optics
- Particle tracing
- Photonics
- Plasma physics
- Porous media flow
- Quantum mechanics
- Radio-frequency components
- Semiconductor devices
- Structural mechanics
- Transport phenomena
- Wave propagation

Many real-world applications involve simultaneous couplings in a system of PDEs—*multiphysics*. For instance, the electric resistance of a conductor often varies with temperature, and a model of a conductor carrying current should include resistive-heating effects. The Multiphysics Modeling section discusses multiphysics modeling techniques. Many predefined multiphysics interfaces provide easy-to-use entry points for common multiphysics applications.

In its base configuration, COMSOL Multiphysics offers modeling and analysis power for many application areas. For several of the key application areas there are also optional modules (see The COMSOL Modules). These application-specific modules use terminology and solution methods specific to the particular discipline, which simplifies creating and analyzing models. The modules also include comprehensive model libraries with example models that show the use of the product within its application areas.

The COMSOL Modules

The optional modules are optimized for specific application areas and offer discipline-standard terminology and physics interfaces. For some modules, additional material libraries, specialized solvers, element types, and visualization tools are also available. This section briefly describes these modules:

- AC/DC Module
- Acoustics Module
- Batteries & Fuel Cells Module
- CFD Module (Computational Fluid Dynamics)
- Chemical Reaction Engineering Module
- Corrosion Module
- Geomechanics Module
- Heat Transfer Module
- MEMS Module
- Microfluidics Module
- Nonlinear Structural Materials Module
- Optimization Module
- Particle Tracing Module
- Pipe Flow Module
- Plasma Module
- RF Module
- Structural Mechanics Module
- Subsurface Flow Module

The CAD Import Module enables you to import CAD data using these formats: IGES, SAT (ACIS), Parasolid, STEP, SolidWorks, Pro/ENGINEER, and Inventor. An additional add-on provides support for CATIA V5. There are also LiveLinks (bidirectional interfaces) for SolidWorks, Creo Parametric, Pro/ENGINEER, Autodesk Inventor, AutoCAD, and SpaceClaim. See CAD Import and LiveLink Connections.

You can build models of all types in the COMSOL Multiphysics user interface. For additional flexibility, COMSOL also provides LiveLink for MATLAB, a seamless interface to MATLAB. This gives you the freedom to combine multiphysics modeling, simulation, and analysis with other modeling techniques. See LiveLink for MATLAB.

AC/DC Module

The AC/DC Module provides a unique environment for simulation of AC/DC electromagnetics in 2D and 3D. The module is a powerful tool for detailed analysis of coils, capacitors, and electrical machinery. With this module you can run static, quasi-static, transient, and time-harmonic simulations in an easy-to-use graphical user interface.

The available physics interfaces cover the following types of electromagnetics field simulations:

- Electrostatics
- Electric currents in conductive media
- Magnetostatics
- Low-frequency electromagnetics

Material properties include inhomogeneous and fully anisotropic materials, media with gains or losses, and complex-valued material properties. Infinite elements makes it possible to model unbounded domains. In addition to the standard results and visualization functionality, the module supports direct computation of lumped parameters such as capacitances and inductances as well as electromagnetic forces.

Like all COMSOL modules, there is a library of ready-to-run models that make it quicker and easier to analyze discipline-specific problems. In addition, any model you develop is described in terms of the underlying partial differential equations, offering a unique way to see the underlying physical laws of a simulation.

The interfaces are fully multiphysics enabled—couple them to any other interface in COMSOL Multiphysics or the other modules. For example, to find the heat distribution in a motor, first find the current in the coils using one of the quasi-static interfaces in this module and then couple it to a heat equation in the main COMSOL Multiphysics package or the Heat Transfer Module. This forms a powerful *multiphysics* model that solves all the equations simultaneously.

COMSOL Multiphysics also has an interface to the MATLAB technical computing environment. If you have a MATLAB license, save it as a Model M-file—a script file that runs in MATLAB. This module also provides interfaces for modeling electrical circuits and importing ECAD drawings.

Tip

For more information go to www.comsol.com/products/acdc

Acoustics Module

The Acoustics Module is a collection of physics interfaces for COMSOL Multiphysics adapted to a broad category of acoustics simulations in fluids and solids. This module is useful even if you are not familiar with computational techniques. It can serve equally well as an excellent tool for educational purposes.

The module supports time-harmonic, modal, and transient studies for fluid pressure as well as static, transient, eigenfrequency, and frequency-response analyses for structures. The available physics interfaces include the following functionality:

- Frequency-domain and transient pressure acoustics
- Acoustic-structure interactions
- Aeroacoustics
- Boundary mode acoustics
- Thermoacoustics
- Aeroacoustics with flow
- Compressible potential flow
- Solid mechanics
- Piezoelectricity

For the pressure acoustics applications, you can choose to analyze the scattered wave in addition to the total wave. PMLs (perfectly matched layers) provide accurate simulations of open pipes and other models with unbounded domains. The modeling domain can include dipole sources as well as monopole sources, and it is easy to specify point sources in terms of flow, intensity, or power. The module also includes modeling support for several types of damping. For results evaluation of pressure acoustics models, you can compute the far field.

Typical application areas for the Acoustics Module include:

- Automotive applications such as mufflers and car interiors
- Modeling of loudspeakers and microphones
- Aeroacoustics
- Underwater acoustics

Using the full multiphysics couplings within the COMSOL Multiphysics environment, you can couple the acoustic waves to, for example, an electromagnetic analysis or a structural analysis for acoustic-structure interaction. The module smoothly integrates with all of the COMSOL Multiphysics functionality.

Tip

For more information go to www.comsol.com/products/acoustics

Batteries & Fuel Cells Module

The Batteries & Fuel Cells Module provides customized physics interfaces for modeling of batteries and fuel cells. These physics interfaces provide tools for building detailed models of the configuration of the electrodes and electrolyte in electrochemical cells. They include descriptions of the electrochemical reactions and the transport properties that influence the performance of batteries, fuel cells, and other electrochemical cells.

The physics interfaces are organized in primary, secondary and tertiary current density distributions physics interfaces. These are available for solid nonporous electrodes and for porous electrodes. In addition to these generic physics interfaces, the module contains a dedicated physics interface for the modeling of Li-ion batteries.

The tailored physics interfaces mentioned above are also complemented with extended functionality in other physics interfaces for chemical species transport, heat transfer, and fluid flow.

The physics interfaces for chemical species transport of neutral species are extended by adding nodes that directly couple to electrochemical reactions defined in the physics interfaces for electrochemical cells. A typical example is the transport and reactions of gaseous species in gas diffusion electrodes and gas channels in fuel cells.

The heat transfer physics interfaces include heat sources that describe ohmic losses in the electrodes and electrolyte and heat sources due to electrochemical reactions in electrochemical cells.

The fluid flow capabilities are extended for laminar flow, where the chemical species transport and the energy balances influence the properties of the flow.

Tip

For more information go to www.comsol.com/products/batteries-fuel-cells

CFD Module

The CFD Module is an optional package that extends the COMSOL Multiphysics modeling environment with customized user interfaces and functionality optimized for the analysis of all types of fluid flow. Ready-to-use interfaces let you model laminar and turbulent flows in single or multiple phases. Functionality for treating coupled free and porous media flow, stirred vessels, and fluid-structure interaction are also included.

- Laminar and turbulent flow using several established turbulence models
- Single-phase and multiphase flow
- Isothermal and non-isothermal flow
- Compressible and incompressible flow
- Newtonian and non-Newtonian flow

The ready coupling of heat and mass transport to fluid flow enables modeling of a wide range of industrial applications such as heat exchangers, turbines, separations units, and ventilation systems.

Together with COMSOL Multiphysics, the CFD Module takes flow simulations to a new level, allowing for arbitrary coupling to physics interfaces describing other physical phenomena, such as structural mechanics, electromagnetics, or even user-defined transport equations.

Tip

For more information go to www.comsol.com/products/cfd

Chemical Reaction Engineering Module

The reaction engineering tools use reaction formulas to create models of reacting systems. In this context, a model means the material (mass), energy (heat), and momentum balances for a system. The Chemical Reaction Engineering Module not only defines these balances, it can also solve the material and energy balances for space-independent models (that is, for models where the composition and temperature in the reacting system vary only in time) and space-dependent models. This makes it possible to create models involving material, energy, and momentum balances in COMSOL Multiphysics directly from a set of reaction formulas.

Included in these models are the kinetic expressions for the reacting system, which are automatically or manually defined in the module. You also have access to a variety of ready-made expressions in order to calculate a system's thermodynamic and transport properties.

The module presents a powerful way of modeling equipment and processes in chemical engineering. It provides customized physics interfaces and formulations for momentum, mass, and heat transport coupled with chemical reactions for applications such as:

- Reaction engineering and design
- Heterogeneous catalysis
- Separation processes
- Fuel cells and industrial electrolysis
- Process control

COMSOL Multiphysics excels in solving systems of coupled nonlinear PDEs that can include:

- Heat transfer
- Mass transfer through diffusion, convection, and migration
- Fluid dynamics
- Chemical reaction kinetics
- Varying material properties

The multiphysics capabilities of COMSOL can fully couple and simultaneously model fluid flow, mass and heat transport, and chemical reactions.

In fluid dynamics you can model fluid flow through porous media, characterize flow with the incompressible Navier-Stokes equations. It is easy to represent chemical reactions by source or sink terms in mass and heat balances. These terms can be of arbitrary order. The physics interfaces in this module cover the following areas:

- Chemical Species Transport
  - Reaction engineering
  - Transport of diluted species through diffusion, convection, and migration in electric fields
  - Transport of concentrated species using one of the following diffusion models: mixture-averaged, Maxwell-Stefan, or Fick's law
  - Nernst-Planck transport equations
- Heat Transfer in fluids, solids and porous media
- Fluid Flow
  - Single-phase flow (incompressible Navier-Stokes equations)
  - Darcy's law
  - Brinkman equations
  - Free and porous media flow

Tip

For more information go to www.comsol.com/products/chemical-reaction-engineering

Corrosion Module

The Corrosion Module is intended for the modeling of corrosion and corrosion protection. The descriptions made available by the module are based on current and potential distribution in galvanic cells. Its modeling capabilities covers galvanic corrosion, cathodic protection, anodic protection, and sacrificial anode protection.

The module consists of a number of chemical species transport, fluid flow, heat transfer, electrochemistry, and corrosion interfaces. These physics interfaces describe the potential in the electrolyte and in the corroding or protected metallic structure. The electrode reactions can be described using arbitrary electrode kinetic expressions of the overpotential for the anodic and cathodic reactions. The potential and current distribution can also include the influence of mass transport and heat transfer, which means that the Corrosion Module makes it possible to model composition and temperature in a corrosion cell in detail. The ability to account for mass transport in the electrolyte allows for modeling of corrosion caused by variations in, for example, salt concentration, oxygen concentration, and pH.

The basis of the Corrosion Module is the mass and current balances in the electrolyte an in the corroding or protected metal structure. In the electrolyte, the current balance is defined together with individual species balances for the charged species and the electroneutrality condition. In the metallic structure, the current balance is described using Ohm's law for the current density. The electron transfer reactions, at the interface between the metallic structure and the electrolyte, couple the transport processes in the electrolyte with the current flowing in the metallic structure. The module contains predefined formulations for the above-mentioned processes. In addition, the module modifies the Heat Transfer interfaces so that predefined terms for heat sources caused by losses in the electrochemical process are described properly.

Tip

For more information go to www.comsol.com/products/corrosion

Geomechanics Module

The Geomechanics Module is an optional package that extends the Structural Mechanics Module to the quantitative investigation of geotechnical processes. It is designed for researchers, engineers, developers, teachers, and students, and suits both single-physics and interdisciplinary studies within geomechanics and soil mechanics.

The module includes an extensive set of fundamental material models, such as the Drucker-Prager and Mohr-Coulomb criteria and the Cam-Clay model in soil mechanics. These material models can also couple to any new equations created, and to physics interfaces (for example, heat transfer, fluid flow, and solute transport in porous media) already built into COMSOL Multiphysics and its other specialized modules.

Tip

For more information go to www.comsol.com/products/geomechanics

Heat Transfer Module

The Heat Transfer Module supports all fundamental mechanisms of heat transfer, including *conductive*, *convective*, and *radiative* heat transfer (both *surface-to-surface* and *surface-to-ambient* radiation). Using the physics interfaces in this module along with inherent multiphysics capabilities of COMSOL Multiphysics you can model a temperature field in parallel with other physics—a powerful combination that makes your models even more accurate and representative of the real world.

Available physics interfaces include functionality for:

- General heat transfer by conduction, convection, and surface-to-surface radiation (including external radiation sources)
- Bioheat equation for heat transfer in biomedical systems
- Heat transfer in porous media
- Heat radiation in participating media
- Highly conductive layer for modeling of heat transfer in thin structures
- Non-isothermal incompressible fluid flow
- Turbulent flow using the k-ε and k-ω turbulence models

The Heat Transfer Module Model Library contains models, many with multiphysics couplings, that cover applications in electronics and power systems, process industries, and manufacturing industries. This Model Library also provides tutorial and benchmark models.

Tip

For more information go to www.comsol.com/products/heat-transfer

MEMS Module

The MEMS Module is a collection of physics interfaces and predefined models for COMSOL Multiphysics with which you can model various MEMS devices and applications.

It includes interfaces for modeling with electrostatics, structural mechanics, piezoelectricity, film damping, and other thin-film flow. The module also includes predefined multiphysics interfaces for thermal-structural interaction, Joule heating with thermal stress or thermoelectromechanical (TEM) interactions, and fluid-structure interaction (FSI).

The MEMS Module Model Library contains a suite of models of MEMS devices such as sensors and actuator systems. The models demonstrate a variety of multiphysics couplings and techniques for moving boundaries. This module also provides interfaces for circuit modeling, a SPICE interface, and support for importing ECAD drawings.

The combination of the models and documentation provides an outline for the theoretical basis of MEMS-specific phenomena and the interfaces in the module, and it contains step-by-step instructions for creating models. The software is uniquely transparent 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, but there are also couplings across these types of physics such as thermal-electric-structural interaction. Although COMSOL Multiphysics does not pose any limitations on the number or type of such couplings, these couplings are important building blocks in complex multiphysics models containing any number of coupled physical phenomena.

Tip

For more information go to www.comsol.com/products/mems

Microfluidics Module

The Microfluidics Module is a collection of tailored physics interfaces for the simulation of microfluidic devices. It has a range of tools to address the specific challenges of modeling micro- and nanoscale flows. Physics interfaces that address laminar flow, multiphase flow, flow in porous media, and rarefied flow such as free molecular flow, slip flow, and transitional flow are available. Enhanced capabilities to treat chemical reactions between dilute species are also included. In addition to the standard tools for modeling fluid flow, interfaces between fluids can be modeled by the Level Set, Phase Field, and Moving Mesh physics interfaces, making it possible to model surface tension and multiphase flow at the microscale.

The Model Library and supporting documentation explain how to use the interfaces to model a range of microfluidic devices.

Tip

For more information go to www.comsol.com/products/microfluidics

Nonlinear Structural Materials Module

The Nonlinear Structural Materials Module is an optional package that extends the Structural Mechanics Module to studies that include structural mechanics with nonlinear material behavior. It is designed for researchers, engineers, developers, teachers, and students that want to simulate nonlinear structural materials, including a full range of possible multiphysics couplings.

The module provides an extensive set of nonlinear structural material models, including the following materials:

- Predefined and user-defined hyperelastic materials: neo-Hookean, Mooney-Rivlin, Saint-Venant Kirchhoff, and others.
- Small-strain and large-strain plasticity models using different hardening models.
- Predefined and user-defined creep and viscoplastic material models: Norton, Garofalo, Anand, potential, volumetric, deviatoric, and others.

Optimization Module

The Optimization Module can be used throughout the COMSOL Multiphysics product family—it is a general interface for calculating optimal solutions to engineering problems. Any model inputs, be it geometric dimensions, part shapes, material properties, or material distribution, can be treated as control variables, and any model output can be an objective function.

There are two optimization algorithms available in the module. The first algorithm is based on the SNOPT code developed by Philip E. Gill of the University of California San Diego, and Walter Murray and Michael A. Saunders of Stanford University. When using SNOPT, the objective function can have any form and any constraints can be applied. The algorithm uses a gradient-based optimization technique to find optimal designs and when the underlying PDE is stationary or time-dependent, analytic sensitivities of the objective function with respect to the control variables can be used.

The second algorithm is a Levenberg-Marquardt solver. When this solver is used, the objective function must be of least squares type. Also, constraints are not supported. Since the Levenberg-Marquardt method is derived to solve problems of least squares type, it typically converges faster than SNOPT for such problems.

Simulation is a powerful tool in science and engineering for predicting the behavior of physical systems, particularly those governed by partial differential equations. In many cases a single or a few simulations are not enough to provide sufficient understanding of a system. Two important classes of problems whose resolution relies on a more systematic exploratory process are:

- *Design problems* with a single objective. Here, the problem is to find the values of control variables or design variables that yield the best performance of the output of a simulation model when the latter is quantified by means of a single function. Problems of this kind arise, for example, in structural optimization, antenna design, and process optimization.
- *Inverse problems*, and in particular *parameter estimation* in multiphysics models. Here, the problem is to reliably determine the values of a set of parameters that provide simulated data which best matches measured data. Such problems arise in, for example, geophysical imaging, nondestructive testing, and biomedical imaging.

It is often possible to reformulate problems of the above type as *optimization problems*. The Optimization interface in COMSOL Multiphysics is useful for solving design problems as well as inverse problems and parameter estimation.

Tip

For more information go to www.comsol.com/products/optimization

Particle Tracing Module

The Particle Tracing Module is a general purpose, flexible tool that allows you to trace the trajectories of particles in the presence of an external field.

A custom interface is available for modeling the trajectory of electrons and ions in an electromagnetic field. This makes it possible to model devices such as magnetic lenses, electron guns, and mass spectrometers. There are a number of tools available to extract typical quantities of interest and to make plots of the particle trajectories, including phase portraits and Poincaré maps. It is also possible to model particle-field interactions provided that the particle trajectories and electromagnetic fields are computed simultaneously.

There is also a dedicated interface for tracing particle trajectories in a fluid system. There are predefined forces for drag, gravity, and other forces, which can be conveniently be added to your model.

A mathematical-based particle tracing interface gives you complete freedom and flexibility over the equations of motion governing particle trajectories. There are a number of different formulations available that makes it possible to specify, for example, a Lagrangian or Hamiltonian that dictates the motion of the particles.

All of the interfaces provide many different ways of releasing particles, either on the boundary or domain level. Particles can also be released at multiple different times. There are several options available for describing how the particles interact with the wall, including specular reflection. The interfaces can all be used with COMSOL Multiphysics but tracing charged particles benefits from the AC/DC Module, as it makes it possible to model complex alternating or direct current systems. Similarly, tracing particles in a fluid system benefits from the CFD Module or the Microfluidics Module due to the number of advanced fluid flow features available in those modules.

Tip

For more information go to www.comsol.com/products/particle-tracing

Pipe Flow Module

The Pipe Flow Module is intended for the modeling and simulation of flow of incompressible fluids in pipe and channel systems, as well as compressible hydraulic transients and acoustics waves. Typical simulations yield the velocity, pressure variation, and temperature in systems of pipes and channels. Hydraulic transients resulting from a valve that is closed rapidly in a pipe network is referred to water hammer, which can be modeled too. The module can be used to design and optimize complex cooling systems in turbines, ventilation systems in buildings, pipe systems in chemical processes, and pipelines in the oil and gas industry.

In common for pipes and channels that can be modeled is that the pipe length is large enough so that the flow inside can be considered fully developed. Piping components such as bends, valves, T-junctions, contractions/expansions and pumps are also available in the module.

The module includes seven physics:

- The *Pipe Flow* physics computes the pressure and velocity field in isothermal pipe systems.
- The *Heat Transfer in Pipes* physics computes the energy balance in pipe systems but receives the flow field as a value or as a known solved field. Wall heat transfer to the surroundings is included.
- The *Transport of Diluted Species in Pipes* physics solves a mass balance equation for pipes in order to compute the concentration distribution of a solute in a dilute solution, considering diffusion, dispersion, convection, and chemical reactions.
- The *Non-Isothermal Pipe Flow* physics is a multiphysics interface that solves the flow, pressure, and temperature simultaneously and fully coupled.
- The *Reacting Pipe Flow* physics is a multiphysics interface that solves the flow, pressure, temperature, and reacting species transport simultaneously and fully coupled.
- The *Water Hammer* physics solves rapid hydraulic transients in pipe systems, taking the elastic properties of both the fluid and pipe wall into account.
- The *Pipe Acoustics, Transient* physics models sound waves in flexible pipe systems.

The physics in the module define the conservation of momentum, energy, and mass of an fluid inside a pipe or channel system. The flow, pressure, temperature, and concentration fields across the pipe cross sections are modeled as cross-section averaged quantities, which only vary along the length of the pipes and channels. The pressure losses along the length of a pipe or in a pipe component are described using friction factor expressions. A broad range of built-in expressions for Darcy friction factors are used that cover the entire flow regime from laminar to turbulent flow, Newtonian and non-Newtonian fluids, different cross-sectional geometries, and a wide range of relative surface roughness values. In addition to the continuous frictional pressure drop along pipe stretches, pressure drops due to momentum changes in components such as as bends, contractions, expansions, T-junctions and valves are computed through an extensive library of industry standard loss coefficients. Pumps are also available as components.

Tip

For more information go to www.comsol.com/products/pipe-flow

Plasma Module

The Plasma Module is based on a series of scientific publications on numerical modeling of non-equilibrium discharges. These papers are referenced throughout the documentation. The physics interfaces can model low temperature, non-equilibrium discharges such as:

- Inductively coupled plasmas (ICP)
- Capacitively coupled plasmas (CCP)
- Microwave plasmas
- Light sources
- Electrical breakdown
- Space thrusters
- DC discharges
- Chemical vapor deposition (CVD)
- Plasma-enhanced chemical vapor deposition (PECVD)
- Dielectric barrier discharges (DBD)
- Electron cyclotron resonance (ECR)
- Reactive gas generators

The complexity of plasma modeling lies in the fact that it combines elements of reaction engineering, statistical physics, fluid mechanics, physical kinetics, heat transfer, mass transfer, and electromagnetics. The net result is a true multiphysics problem involving complicated coupling between the different physics. The module is designed to simplify the process of setting up a self-consistent model of a low-temperature plasma.

The physics interfaces include all the necessary tools to model plasma discharges, beginning with a Boltzmann Equation, Two-Term Approximation solver that computes the electron transport properties and source coefficients from a set of electron impact collision cross sections. This interface makes it possible to determine many of the interesting characteristics of a discharge by providing input properties such as the electric field and the electron impact reactions that make up the plasma chemistry, without solving a space-dependent problem.

For space-dependent models, the reactions and species which make up the plasma chemistry are conveniently managed in the Model Builder. When the fluid velocity and gas temperature are of interest, there are physics interfaces available for laminar flow and heat transfer. There are several options available when coupling the charged species transport to the electromagnetic fields. Poisson's equation for the electrostatic potential is always solved. Modeling inductively coupled plasmas where induction currents are responsible for sustaining the plasma requires the AC/DC Module. For wave-heated discharges (microwave plasmas), the RF Module is required.

Tip

For more information go to www.comsol.com/products/plasma

RF Module

The RF Module solves problems in the general field of electromagnetic waves, such as RF and microwave applications, optics, and photonics. The underlying equations for electromagnetics are automatically available in all of the physics interfaces—a feature unique to COMSOL Multiphysics. This also makes nonstandard modeling easily accessible.

The module is useful for component design in virtually all areas where you find electromagnetic waves, such as:

- Antennas
- Waveguides and cavity resonators in microwave engineering
- Optical fibers
- Photonic waveguides
- Photonic crystals
- Active devices in photonics

The physics interfaces cover the following types of electromagnetics field simulations and handle time-harmonic, time-dependent, and eigenfrequency/eigenmode problems:

- In-plane, axisymmetric, and full 3D electromagnetic wave propagation
- Full vector mode analysis in 2D and 3D

Material properties include inhomogeneous and fully anisotropic materials, media with gains or losses, and complex-valued material properties. In addition to the standard postprocessing features, the module supports direct computation of S-parameters and far-field patterns. You can add ports with a wave excitation with specified power level and mode type, and add PMLs (perfectly matched layers) to simulate electromagnetic waves that propagate into an unbounded domain. For time-harmonic simulations, you can use the scattered wave or the total wave.

Using the multiphysics capabilities of COMSOL Multiphysics you can couple simulations with heat transfer, structural mechanics, fluid flow formulations, and other physical phenomena.

This module also has interfaces for circuit modeling, a SPICE interface, and support for importing ECAD drawings.

Tip

For more information go to www.comsol.com/products/rf

Structural Mechanics Module

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

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

- Solid mechanics for 2D plane stress and plane strain, axial symmetry, and 3D solids
- Piezoelectric modeling
- Beams in 2D and 3D, Euler theory

- Truss and cable elements
- Shells and plates

The module's study capabilities include static, eigenfrequency, time dependent (transient), frequency response, and parametric studies, as well as contact and friction. There are also predefined interfaces for linear elastic and viscoelastic materials. Materials can be isotropic, orthotropic, or fully anisotropic, and you can use local coordinate systems to specify material properties. Large deformations can also be included in a study.

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

Tip

For more information go to www.comsol.com/products/structural-mechanics

Subsurface Flow Module

The earth and surrounding planets are a giant laboratory filled with an unlimited array of basic physics and multiphysics interactions. Whether in concert or alone, these physical phenomena alter our access to important resources, affect the quality of the environment, shape the ground beneath our feet on earth, and form other planets.

The Subsurface Flow Module is used in the quantitative investigation of geophysical and environmental phenomena, especially within the area of subsurface flow. The module combines physics interfaces for fundamental processes and links to COMSOL Multiphysics and the other modules for structural mechanics and electromagnetics analyses. New physics represented include heating from radiogenic decay that produces the geotherm, which is the increase in background temperature with depth. You can use the variably saturated flow interfaces to analyze unsaturated zone processes (important to environmentalists) and two-phase flow (of particular interest in the petroleum industry as well as steam-liquid systems). Important in subsurface flow, the heat transfer and chemical transport interfaces explicitly account for physics in the liquid, solid, and gas phases.

The interfaces, options, and functions are tailored to account for subsurface flow and geologic process terminology. The Heat Transfer branch of the Model Wizard, for example, include features to superpose a background geotherm without incorporating it as a boundary condition. These interfaces also include options to automate the calculation of effective thermal properties for multicomponent systems.

The Fluid Flow branch represent a wide range of possibilities. The Richards' Equation interface describes nonlinear flow in variably saturated porous media. The options for saturated porous media include the Darcy's Law interface for slow flow and the Brinkman Equations interface where shear is non-negligible. The Single-Phase Flow, Laminar interface uses the Navier-Stokes equations to cover free flows and the Fracture Flow interface is used for modeling flow in fractures.

The module also treats the transport of chemicals and their reactions with the Chemical Species Transport branch. The Solute Transport interface account for chemical transfer in solid, liquid, and gas phases for free, saturated, and variably saturated fluid flows. The Subsurface Flow Module Model Library has a number examples linking these interfaces together.

Tip	For more information go to www.comsol.com/products/subsurface-flow

CAD Import and LiveLink Connections

COMSOL includes packages for importing CAD files into COMSOL Multiphysics and LiveLinks, which provide bidirectional interfaces to SolidWorks®, Creo™ Parametric, Pro/ENGINEER®, Autodesk Inventor®, AutoCAD®, and SpaceClaim®. The following products are available:

- CAD Import Module, for file import of CAD drawings in Parasolid, SAT (ACIS®), STEP, IGES, SolidWorks, Pro/ENGINEER, and Inventor formats.
- CATIA V5 Import Module, which requires the CAD Import Module, imports CAD drawings in the CATIA V5 format.

- COMSOL LiveLink for SolidWorks, which provides a fully-associative bidirectional link to SolidWorks.
- COMSOL LiveLink for Creo Parametric, which provides a fully-associative bidirectional link to Creo Parametric.
- COMSOL LiveLink for Pro/ENGINEER, which provides a fully-associative bidirectional link to Pro/ENGINEER.
- COMSOL LiveLink for Inventor, which provides a fully-associative bidirectional link to Autodesk Inventor.
- COMSOL LiveLink for AutoCAD, which provides a fully-associative bidirectional link to AutoCAD.
- COMSOL LiveLink for SpaceClaim, which provides a fully-associative bidirectional link to SpaceClaim.

LiveLink for MATLAB

The LiveLink for MATLAB® gives you access to all modeling functionality in COMSOL and its add-ons through MATLAB. The interface in the LiveLink is based on the COMSOL API. The product includes utility functions for producing graphics in MATLAB figure windows, accessing COMSOL data, and performing general results analysis and other postprocessing.

Overview of the User's Guide

The *COMSOL Multiphysics User's Guide* gets you started with modeling using COMSOL Multiphysics. See the individual module User's Guides for information specific to a specialized module (see The COMSOL Modules for a list).

Tip

As detailed in the section Where Do I Access the Documentation and Model Library? this information is also searchable from the COMSOL Multiphysics software **Help** menu.

TABLE OF CONTENTS, GLOSSARY, AND INDEX

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

ENVIRONMENT

The COMSOL Modeling Environment chapter provides an overview of the COMSOL modeling environment as controlled by COMSOL Desktop and the tools and windows it provides. Topics include the The COMSOL Desktop Environment and an Introduction to the Model Wizard and the Model Builder.

DEFINITIONS

The chapter describes the global and local (model) definitions features. Depending on the geometric scope, you add the feature nodes described in this section to either the Global Definitions node or under the Definitions node for a particular model. Topics include Operators, Functions, and Variables Reference, Variables, Global and Local Functions, and Coordinate Systems.

VISUALIZE

The Visualization and Selection Tools chapter describes the tools used to visualize and control how you view models and select parts of the model geometry in the Graphics window and the settings windows. Important topics include Working with Geometry, User-Defined Selections, and User-Defined Views.

GEOMETRY

The Geometry Modeling and CAD Tools chapter covers geometry modeling in 1D, 2D, and 3D with examples of solid modeling, boundary modeling, Boolean operators, and other CAD tools in COMSOL Multiphysics. In addition, it shows how to use the tools for exploring geometric properties, such as volumes and surfaces. There is also information about using external CAD data. Topics include Creating a Geometry for Successful Analysis, The COMSOL Multiphysics Geometry and CAD Environment, and Creating a 3D Geometry Model.

MATERIAL

The Materials chapter introduces you to the material databases included with COMSOL Multiphysics. Topics include a Materials Overview, Adding Predefined Materials, User-Defined Materials and Libraries, and Using Functions.

MODELING

Building a COMSOL Model explains a range of methods and topics used when building models in COMSOL Multiphysics: Topics include more information about The Model Wizard and Model Builder, details about Model Administration, The Physics Feature Nodes, Specifying Model Equation Settings, Periodic Boundary Conditions, Calculating Accurate Fluxes. and Using Units.

PHYSICS INTERFACES

The Overview of the Physics Interfaces chapter briefly describes The COMSOL Multiphysics Physics Interfaces as well as some general Modeling Guidelines for effective modeling. The last section describes the Equations for the Physics Interfaces.

AC/DC

The AC/DC Branch chapter explains the physics interfaces available for modeling electromagnetics, which you find under the AC/DC branch ( ) in the Model Wizard. It also contains sections about general fundamentals and theory for electric fields.

ACOUSTICS

The Acoustics Branch chapter describes how to use the Pressure Acoustics, Frequency Domain interface, found under the Acoustics branch ( ) in the Model Wizard, for modeling and simulation of acoustics and vibrations.

CHEMICAL SPECIES TRANSPORT

The Chemical Species Transport Branch chapter explains how to use the Transport of Diluted Species interface, found under the Chemical Species Transport branch ( ) in the Model Wizard, to model and simulate mass transfer by diffusion and convection based on Fick’s law of diffusion.

FLUID FLOW

The Fluid Flow Branch chapter explains how to use the Laminar Flow interface, found under the Fluid Flow>Single-Phase Flow branch ( ) in the Model Wizard, to model and simulate fluid mechanics for laminar, incompressible fluids.

HEAT TRANSFER

The Heat Transfer Branch chapter describes the different types of Heat Transfer interfaces (Heat Transfer in Solids and Heat Transfer in Fluids), and the Joule Heating interface, all found under the Heat Transfer branch ( ) in the Model Wizard.

SOLID MECHANICS

The Structural Mechanics Branch chapter explains how to use the Solid Mechanics interface, found under the Structural Mechanics branch ( ) in the Model Wizard, to simulate and analyze applications involving solid mechanics. The interface is used for stress analysis and general solid mechanics simulation.

EQUATION BASED

The Equation-Based Modeling chapter describes the use of the mathematics interfaces, found under the Mathematics branch (Δu) in the Model Wizard, which are used for equation-based modeling. With those interfaces you can solve various types of PDEs using different formulations. You can also solve ODEs and other global equations.

SENSITIVITY

The Sensitivity Analysis chapter describes how to perform sensitivity analysis using the Sensitivity interface, found under the Mathematics>Optimization and Sensitivity ( ) branch in the Model Wizard.

MULTIPHYSICS

Multiphysics Modeling discusses how to use COMSOL Multiphysics for multiphysics modeling and coupled-field analyses. It describes various ways to build multiphysics models and includes an overview of the available predefined multiphysics interfaces in COMSOL Multiphysics and the add-on modules. Topics include Multiphysics Modeling Approaches, Using Predefined Multiphysics Interfaces, Adding Physics Sequentially, Building a Multiphysics Model Directly, Adding Multiphysics Couplings, and Deleting Physics Interfaces.

DEFORMED MESHES

The Deformed Geometry and Moving Mesh Interfaces chapter explains how to use the modeling interfaces that control mesh deformation. It also contains fundamentals about deformed meshes and information about the Eulerian and Lagrangian formulations of the physics, the frame types that support these formulations, and the arbitrary Lagrangian-Eulerian (ALE) method.

MESH

The Meshing chapter summarizes how to create and control your mesh for 1D, 2D, and 3D geometries in COMSOL Multiphysics. It also explains the possibilities for importing and exporting meshes in different formats. Topics include Creating Meshes, Free Meshing, Structured Meshes, and Clearing Meshes and Deleting Meshing Sequences.

SOLVER

The Solvers and Study Types chapter lists the various types of solvers and studies in COMSOL Multiphysics and explains the study steps and solver configurations. It also describes the major solvers and settings as well as batch jobs, parametric sweeps, and cluster computing. See also the *COMSOL Multiphysics Reference Guide* and the *Optimization Module User's Guide* for other supplementary information

RESULTS

The Results Evaluation and Visualization chapter helps you analyze results in COMSOL Multiphysics and describes numerous result-evaluation and visualization tools, including advanced graphics, data display, and export functions. Topics include Processing and Analyzing Results, Defining Data Sets, Creating 1D, 2D, 3D, and Polar Plots, Creating Cross-Section Plots and Combining Plots, Defining Derived Values and Tables, and Exporting Data and Images.

RUNNING

The last chapter, Running the COMSOL Software, provides an overview of the different ways that you can run the COMSOL software in addition to the COMSOL Desktop graphical user interface. The topics include Running COMSOL, COMSOL Client/Server Architecture, Running COMSOL Client/Server, and Running COMSOL in Parallel.

The COMSOL Documentation Set

The full documentation set that ships with COMSOL Multiphysics consists of the following titles:

- *COMSOL Quick Installation Guide*—basic information for installing the COMSOL software and getting started. Included in the DVD package.
- *Introduction to COMSOL Multiphysics*—information about version 4.3 and how to build models using the desktop environment. Included in the DVD package.
- *COMSOL License Agreement*—the license agreement. Included in the DVD package.
- *COMSOL Installation and Operations Guide*—besides covering various installation options, it describes system requirements and how to configure and run the COMSOL software on different platforms, including client/server architectures as well as shared-memory and distributed (cluster) parallel versions.
- *COMSOL Multiphysics User's Guide*—the book you are reading, it covers the functionality of COMSOL Multiphysics across its entire range from geometry modeling to results evaluation and visualization, including the interfaces for physics and equation-based modeling. It serves as a tutorial and a reference guide to using COMSOL Multiphysics.
- *COMSOL Multiphysics Reference Guide*—this book reviews geometry, mesh, solver, and results features and provides detailed information about their settings and options. Additionally, it describes some advanced functionality and settings in COMSOL Multiphysics and provides background material and references.
- *COMSOL Java API Reference Guide*—this book provides details about features and techniques that help you control COMSOL Multiphysics using its application programming interface (API). The COMSOL Java API can be used from a standalone Java application as well as from the LiveLink™ for MATLAB® interface.

In addition, each of the optional modules has a *User's Guide* as described in The COMSOL Modules. The documentation for the optional CAD Import Module and LiveLinks to CAD packages is available in separate *User's Guides*, and the documentation for the optional Material Library in the *Material Library User's Guide*.

The *COMSOL LiveLink™ for MATLAB® User's Guide* shows how to access all of COMSOL Multiphysics' capabilities from the MATLAB programming environment.

Where Do I Access the Documentation and Model Library?

A number of Internet resources provide more information about COMSOL Multiphysics, including licensing and technical information. The electronic documentation, Dynamic Help, and the Model Library are all accessed through the COMSOL Desktop.

Important

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

THE DOCUMENTATION

The *COMSOL Multiphysics User's Guide* and *COMSOL Multiphysics Reference Guide* describe all interfaces and functionality included with the basic COMSOL Multiphysics license. These guides also have instructions about how to use COMSOL Multiphysics and how to access the documentation electronically through the COMSOL Multiphysics help desk.

To locate and search all the documentation, in COMSOL Multiphysics:

- Press F1 for Dynamic Help,
- Click the buttons on the toolbar, or
- Select **Help>Documentation** (?) or **Help>Dynamic Help** (?) from the main menu

and then either enter a search term or look under a specific module in the documentation tree.

THE MODEL LIBRARY

Each model comes with documentation that includes a theoretical background and step-by-step instructions to create the model. The models are available in COMSOL as MPH-files that you can open for further investigation. You can use the step-by-step instructions and the actual models as a template for your own modeling and applications.

SI units are used to describe the relevant properties, parameters, and dimensions in most examples, but other unit systems are available.

To open the Model Library, select **View>Model Library** ( ) from the main menu, and then search by model name or browse under a module folder name. Click to highlight any model of interest, and select **Open Model and PDF** to open both the model and the documentation explaining how to build the model. Alternatively, click the **Dynamic Help** button ( ) or select **Help>Documentation** in COMSOL to search by name or browse by module.

The model libraries are updated on a regular basis by COMSOL in order to add new models and to improve existing models. Choose **View>Model Library Update** ( ) to update your model library to include the latest versions of the model examples.

If you have any feedback or suggestions for additional models for the library (including those developed by you), feel free to contact us at info@comsol.com.

CONTACTING COMSOL BY EMAIL

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

To receive technical support from COMSOL for the COMSOL products, please contact your local COMSOL representative or send your questions to support@comsol.com. An automatic notification and case number is sent to you by email.

COMSOL WEB SITES

Main Corporate web site	www.comsol.com
Worldwide contact information	www.comsol.com/contact
Technical Support main page	www.comsol.com/support
Support Knowledge Base	www.comsol.com/support/knowledgebase
Product updates	www.comsol.com/support/updates
COMSOL User Community	www.comsol.com/community

Checking for Software Updates

COMSOL provides software updates that improve the software and correct issues found. To check is an update is available, choose **Check For Updates** from the **Help** menu. The program then checks if an update that is applicable but not yet installed is available from the COMSOL website. If an update is available, an **Update** dialog box appears where you click **Download** to download the update directly or click **Browse Update** to open the COMSOL website where you can read about and download the update. If no updates are available, the **Update** dialog box reports that your COMSOL installation is up-to-date. To check for updates automatically each time that you start COMSOL Multiphysics, select the **Check for updates at launch** check box under **Product updates** on the **Updates** page in the **Preferences** dialog box.

Typographical Conventions

All COMSOL user's guides use a set of consistent typographical conventions that make it easier to follow the discussion, understand what you can expect to see on the graphical user interface (GUI), and know which data must be entered into various data-entry fields.

In particular, these conventions are used throughout the documentation:

CONVENTION	EXAMPLE
text highlighted in blue	Click text highlighted in blue to go to other information in the PDF. When you are using the online help desk in COMSOL Multiphysics, these links also work to other modules, model examples, and documentation sets.
boldface font	A **boldface** font indicates that the given word(s) appear exactly that way on the COMSOL Desktop (or, for toolbar buttons, in the corresponding tip). For example, the **Model Builder** window () is often referred to and this is the window that contains the model tree. As another example, the instructions might say to click the **Zoom Extents** button (), and this means that when you hover over the button with your mouse, the same label displays on the COMSOL Desktop.
Forward arrow symbol **>**	The forward arrow symbol **>** is instructing you to select a series of menu items in a specific order. For example, **Options>Preferences** is equivalent to: From the **Options** menu, choose **Preferences**.
`Code` (monospace) font	A `Code` (monospace) font indicates you are to make a keyboard entry in the user interface. You might see an instruction such as "Enter (or type) `1.25` in the **Current density** field." The monospace font also is an indication of programming code or a variable name.

CONVENTION	EXAMPLE
Italic *`Code`* (monospace) font	An italic *`Code`* (monospace) font indicates user inputs and parts of names that can vary or be defined by the user.
Arrow brackets <> following the `Code` (monospace) or *`Code`* (italic) fonts	The arrow brackets included in round brackets after either a monospace `Code` or an italic *`Code`* font means that the content in the string can be freely chosen or entered by the user, such as feature tags. For example, `model.geom(<`*`tag`*`>)` where <*`tag`*> is the geometry's tag (an identifier of your choice). When the string is predefined by COMSOL, no bracket is used and this indicates that this is a finite set, such as a feature name.

KEY TO THE GRAPHICS

Throughout the documentation, additional icons are used to help navigate the information. These categories are used to draw your attention to the information based on the level of importance, although it is always recommended that you read these text boxes.

ICON	NAME	DESCRIPTION
⚠	Caution	A Caution icon is used to indicate that the user should proceed carefully and consider the next steps. It might mean that an action is required, or if the instructions are not followed, that there will be problems with the model solution.
❗	Important	An Important icon is used to indicate that the information provided is key to the model building, design, or solution. The information is of higher importance than a note or tip, and the user should endeavor to follow the instructions.
📝	Note	A Note icon is used to indicate that the information may be of use to the user. It is recommended that the user read the text.
💡	Tip	A Tip icon is used to provide information, reminders, shortcuts, suggestions of how to improve model design, and other information that may or may not be useful to the user.
🔍	See Also	The See Also icon indicates that other useful information is located in the named section. If you are working on line, click the hyperlink to go to the information directly. When the link is outside of the current PDF document, the text indicates this, for example See The Laminar Flow Interface in the *COMSOL Multiphysics User's Guide*. Note that if you are in COMSOL Multiphysics' online help, the link will work.

ICON	NAME	DESCRIPTION
▥	Model	The Model icon is used in the documentation as well as in COMSOL Multiphysics from the View>Model Library menu. If you are working online, click the link to go to the PDF version of the step-by-step instructions. In some cases, a model is only available if you have a license for a specific module. These examples occur in the COMSOL Multiphysics User's Guide. The Model Library path describes how to find the actual model in COMSOL Multiphysics, for example If you have the RF Module, see Radar Cross Section: Model Library path **RF_Module/Tutorial_Models/radar_cross_section**
Space Dimension		Another set of icons are also used in the Model Builder—the model space dimension is indicated by 0D, 1D, 1D axial symmetry, 2D, 2D axial symmetry, and 3D icons. These icons are also used in the documentation to clearly list the differences to an interface, feature node, or theory section, which are based on space dimension.

2

The COMSOL Modeling Environment

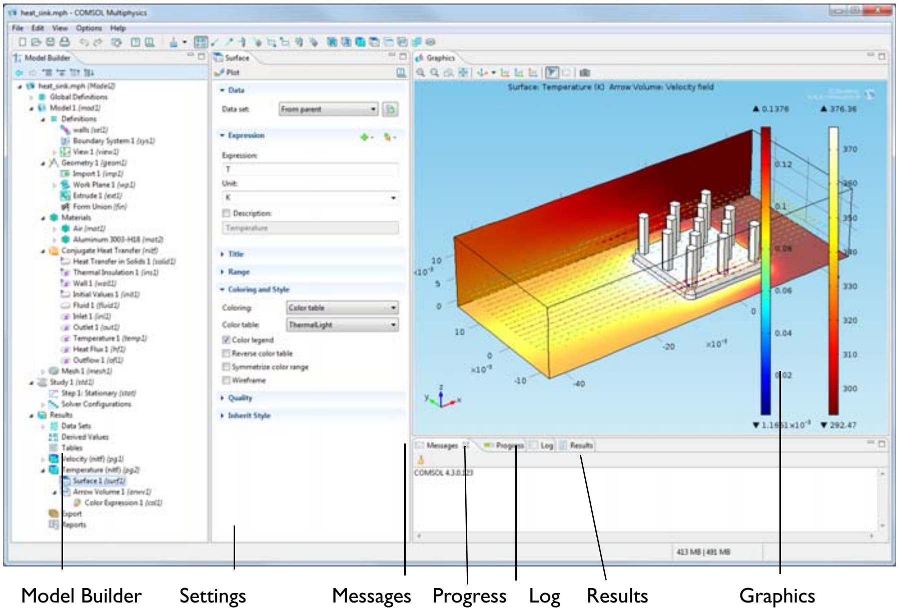
The COMSOL Desktop provides a complete and integrated modeling environment for creating, analyzing, and visualizing multiphysics models. This chapter provides an overview of the COMSOL modeling environment as controlled by COMSOL Desktop and the tools and windows it provides.

In this chapter:

The COMSOL Desktop Environment

Introduction

This section describes the major components in the COMSOL Multiphysics environment. This is completely controlled by the *COMSOL Desktop*, which you can personalize to your own modeling needs and preferences. Primarily consisting of the *Model Builder*, *node settings*, and *Graphics* windows, other dockable windows can be opened, closed, and placed according to the modeling settings you need to access and the GUI configuration you want to work in. You can save these configurations, and the last opened configuration is always displayed when you open COMSOL Multiphysics again.



Surface

Plot

- Data
  - Data set: From parent
- Expression
  - Expression: T
  - Unit: K
  - ☐ Description: Temperature
- Title
- Range
- Coloring and Style
  - Coloring: Color table
  - Color table: ThermalLight
  - ☑ Color legend
  - ☐ Reverse color table
  - ☐ Symmetrize color range
  - ☐ Wireframe
- Quality
- Inherit Style

Figure 2-1: The COMSOL Desktop with its major windows in a widescreen layout.

The following sections describe how to configure the COMSOL Desktop layout and language and the main functionality of the major windows, most of which appear in Figure 2-1. These topics are included:

- Changing the COMSOL Desktop Layout

- Moving Between Windows and Sections on the COMSOL Desktop
- Changing the COMSOL Desktop Language
- Printing from the COMSOL Desktop
- The Main Menu and Toolbar

Changing the COMSOL Desktop Layout

To customize the COMSOL Desktop environment, you rearrange the windows by moving, resizing, detaching, or docking each window. Predefined layouts are also available from the **View>Desktop Layout** menu.

For any window, perform the following tasks as required.

MOVING OR RESIZING THE WINDOW

- Click-and-drag the window tab (the tab is where the window name, **Model Builder** ( ) for example, displays) to where you want it.
- Right-click the window tab and select **Move>View** (to move a separate window) Move the mouse to where you want to the view or tab to display and left-click to confirm the move.
- Right-click the window tab and select **Move>Tab Group** (to move several tabbed windows) from the drop-down list. Move the mouse to where you want to the view or tab to display and left-click to confirm the move.

MAXIMIZING AND RESTORING A WINDOW'S ORIGINAL POSITION

Double-click a window tab to maximize the window and then double-click it again to restore it.

DETACHING A WINDOW FOR MOVING AND RESIZING

Right-click the window tab and select **Detached** so that you can move it and resize it as a separate window. Select **Detached** again to dock it on the COMSOL Desktop.

MINIMIZING OR MAXIMIZING THE WINDOW

- Click the **Minimize** or **Maximize** button in the top-right corner.
- Right-click the window tab and select **Minimize** or **Maximize** from the list.

RESIZING THE WINDOWS

Hover your mouse over the window borders until a double arrow displays. Click-and-drag the borders between windows until the layout is how you want it.

SETTING THE LAYOUT TO WIDESCREEN OR REGULAR SCREEN

Select **Options>Desktop Layout>**

- **Widescreen Layout**: suitable for widescreen monitors.

  The **Model Builder** window and the settings window display side by side.

- **Regular Screen Layout**: suitable for monitors with a regular screen (4:3).

  The **Model Builder** window displays on top of the settings window.

RESETTING THE DESKTOP TO DEFAULT SETTINGS

Select **View>Desktop Layout>Reset Desktop** or click the **Reset Desktop** button ( ) on the main toolbar. The default settings are restored either for a widescreen layout or a regular screen layout depending on your monitor.

CHANGING THE FONT FOR PLOT LABELS AND TITLES

COMSOL Multiphysics uses a default font for the axis labels and titles in plots. You can change the font family and font size in the **Preferences** dialog box. This might be necessary to display non-Latin characters such as Chinese and Japanese characters, for example. To change the font:

1 From the **Options** menu, select **Preferences**.

2 In the list to the left in the **Preferences** dialog box, click **Graphics**.

3 Under **Font**, select a font **Family** from the list and enter a font **Size** (in points). The default is to use a predefined **default** font with a font **Size** of 9 points. Depending on the operating system and the installed fonts on the computer, you can select from a number of other font families.

4 Click **OK**.

   The program stores the specified font family and size as a preference setting, so you only have to change it once.

See Also

- Editing Preferences Settings
- Key to the Nodes and Toolbar Buttons

Moving Between Windows and Sections on the COMSOL Desktop

You can use the mouse and the options described in Changing the COMSOL Desktop Layout to select, move, and resize windows. In addition, these keyboard shortcuts provide quick ways to navigate between the windows on the COMSOL Desktop and to switch focus from one window to another or between sections in the settings window.

- Press Ctrl+Tab to switch focus to the next window on the desktop.
- Press Ctrl+Shift+Tab to switch focus to the previous window in the desktop.
- Press Ctrl+Alt+left arrow to switch focus to the **Model Builder** window ( ).
- Press Ctrl+Alt+right arrow to switch focus to the settings window.
- Press Ctrl+Alt+up arrow to switch focus to the previous section in the settings window.
- Press Ctrl+Alt+down arrow to switch focus to the next section in the settings window.

See Also

- Keyboard Shortcuts
- Key to the Nodes and Toolbar Buttons

Changing the COMSOL Desktop Language

1. Start COMSOL Multiphysics.
2. Select **Options>Preferences**.
3. Click **General** and select an available **Language** for the graphical user interface (GUI). The following languages are available: Traditional Chinese, Simplified Chinese, English, French, German, Italian, Japanese, Korean, or Spanish.
4. Click **OK**. A message displays indicating COMSOL Multiphysics must be restarted for the changes to take effect.
5. Click **OK,** exit and re-open COMSOL to display the GUI in the selected language.

See Also

Editing Preferences Settings

Printing from the COMSOL Desktop

To print the contents in the **Graphics** window, click the **Print** button (🖨) on the main toolbar, select **Print** from the **File** menu, or press Ctrl+P. Then **Print** dialog box then opens. In the **Print** dialog box, follow these steps:

1. Under **Image**, from the **Size** list select **Manual** (the default) to define the print size manually using the settings below, or select **Current** to use the current size of the **Graphics** window. Of the settings below, only the **Antialiasing** check box is then available.
2. Select a **Unit** to define the image size—**Millimeters (mm)**, **Inches (in)**, or **Pixels (px)**.
3. Select the **Lock aspect ratio** check box to maintain the calculation of the width and height (if one or the other is changed).
4. Enter the **Width** and **Height** in the units selected for the image.
5. Enter the **Resolution** for the image in DPI (dots per inch). The default value is 300 DPI.

   Under these settings, the dialog box shows the resulting image size and size on the screen in pixels.
6. The **Antialiasing** check box is cleared by default. Click to select if required. Antialiasing minimizes distortion such as jagged edges in the image.
7. Under **Layout**, the **Title**, **Legend**, **Axes**, and **Logotype** check boxes are selected by default to display the information on the screenshot if you select the **Include** check box. You can then also edit the selections for including these parts of the plot.
8. Enter a **Font** size in points (pt). The default value is 9 pt.
9. Select a **Background**—**Current** or **Color**. Current is the background color in the plot window on the COMSOL Desktop. If **Color** is chosen, click the **Color** button to select a custom color from the color palette that opens.
10. Click **OK** to print the contents of the **Graphics** window. Typically the operating system's **Print** dialog box first opens for selecting a printer, the number of copies, and other printer settings.

For generating snapshots of the **Graphics** window to a file or the clipboard, click the **Image Snapshot** button (📷) on the Graphics toolbar.

See Also

Capturing and Copying Screenshots

The Main Menu and Toolbar

At the top of every window is the main menu and toolbar, which has a drop down menu and standard buttons for frequently used actions. See Table 2-1 and Figure 2-2.

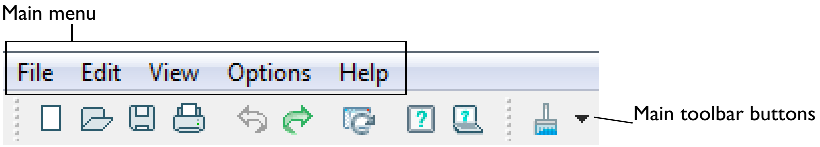


Figure 2-2: The main menu and toolbar buttons.

TABLE 2-1: THE MAIN MENU AND TOOLBAR

NAME	BUTTON	DESCRIPTION OR LINK TO MORE INFORMATION
File	—	Create new models, open existing models, save, print, and create a model thumbnail. • Model Administration • Printing from the COMSOL Desktop • Capturing and Copying Screenshots
Edit	—	Undo or redo operations; copy, paste, duplicate, and delete nodes; clear all meshes or clear all solutions. • Keyboard Shortcuts • Copying, Pasting, and Duplicating Nodes • Clearing All Meshes and Clearing all Solutions
View	—	Introduction to the COMSOL Windows

TABLE 2-1: THE MAIN MENU AND TOOLBAR

NAME	BUTTON	DESCRIPTION OR LINK TO MORE INFORMATION
Options	—	• Editing Preferences Settings • Checking and Controlling Products and Licenses Used
Help	—	Open Help and other online resources, check for updates, and view product information. • The Help Window • The About COMSOL Multiphysics Box
New or File>New	□	Create a new model file and open the Model Wizard.
Open	📂	Open an existing model file located on the computer.
Save or File>Save	💾	Saving COMSOL Multiphysics Model Files
Print	🖨	Printing from the COMSOL Desktop
Undo or Edit>Undo	↶	Undoing and Redoing Operations
Redo or Edit>Redo	↷	Undoing and Redoing Operations
Reset Desktop	⟳	Changing the COMSOL Desktop Layout
Documentation or Help>Documentation	?	The Help Window
Dynamic Help or Help>Dynamic Help	?	The Help Window
Toolbar	🖌	Dynamic list of available toolbars based on the window open or the stage of the modeling process. For example, when building a Geometry.

THE DIFFERENCE BETWEEN NODES, BUTTONS, AND ICONS

Node

A node is located in the **Model Builder** and has an icon image to the left of it. Right-click a node to open a context menu and to perform actions.

Button

Click a button to perform an action. Usually located on a toolbar (the main toolbar or the Graphics toolbar, for example), or in the top part of a settings window.

Icon

An icon is an image that displays on a window (for example, the **Model Wizard** or **Model Library**) or displays in a context menu when a node is right-clicked. Sometimes selecting an item with an icon from a node's context menu adds a node with the same image and name, sometimes it simply performs the action indicated (for example, **Delete**, **Enable**, or **Disable**).

Introduction to the Model Wizard and the Model Builder

In this section:

- The Model Wizard
- Basic Steps to Build a Model
- The Model Builder Window
- About Sequencing in COMSOL Multiphysics
- Opening a Context Menu to Add Nodes
- Moving Nodes in the Model Tree
- Going to the Source Node
- Copying, Pasting, and Duplicating Nodes
- Expanding and Collapsing All Nodes
- Undoing and Redoing Operations
- Clearing All Meshes and Clearing all Solutions
- Sorting Nodes by Space Dimension and Type

See Also	- Introduction to the COMSOL Windows - Key to the Nodes and Toolbar Buttons - Building a COMSOL Model

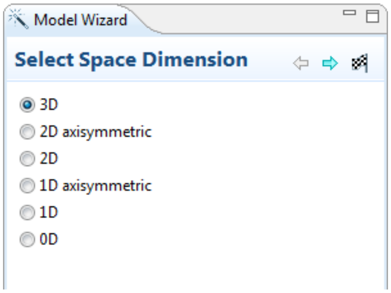
The Model Wizard

When COMSOL Multiphysics is first opened, or when you start creating a new model, the GUI displays the Model Builder, Graphics, and Model Wizard windows. The *Model Wizard* window ( ) contains a series of pages to help you in starting to build a model—*Select Space Dimension*, *Select Physics*, and *Select Study Type*.

See Also	The Model Wizard and Model Builder

THE SELECT SPACE DIMENSION PAGE

Specify the final geometry dimension on the **Select Space Dimension** page—**3D**, **2D axisymmetric**, **2D**, **1D axisymmetric**, **1D**, or **0D**. 0D is used for *physics interfaces* modeling spatially homogeneous systems such as chemical reacting systems and electrical circuits. If you want to import a geometry, this is done in the **Model Builder** ( ), but make sure you choose spatial dimensions that this geometry exists in.



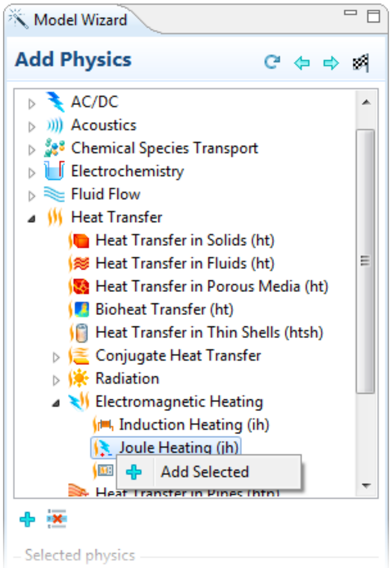
Model Wizard

Select Space Dimension

● 3D
○ 2D axisymmetric
○ 2D
○ 1D axisymmetric
○ 1D
○ 0D

THE ADD PHYSICS PAGE

On the **Add Physics** page, select one or more physics interfaces to include in the model. Use the **Add Selected** (✚) or **Remove Selected** (✖) buttons to easily add or remove physics interfaces for *multiphysics models.*



Model Wizard

Add Physics

- AC/DC
- Acoustics
- Chemical Species Transport
- Electrochemistry
- Fluid Flow
- Heat Transfer
  - Heat Transfer in Solids (ht)
  - Heat Transfer in Fluids (ht)
  - Heat Transfer in Porous Media (ht)
  - Bioheat Transfer (ht)
  - Heat Transfer in Thin Shells (htsh)
  - Conjugate Heat Transfer
  - Radiation
  - Electromagnetic Heating
    - Induction Heating (ih)
    - Joule Heating (ih)
  - Heat Transfer in Pipes (htp)

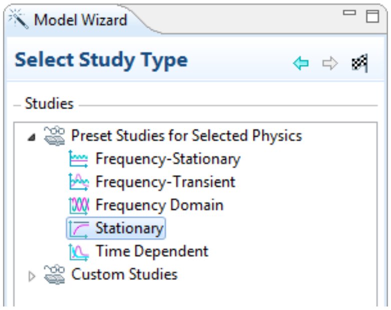
The tree organizes the available interfaces on the basis of application areas such as fluid flow, heat transfer, and structural mechanics. The physics found in the modules your license supports display in the different application areas. In some cases, licensing of a module adds physics interfaces to these application areas as well as attributes to existing physics interfaces, which are enhanced with additional functionality. The **Recently Used** branch ( ) lists the last five physics interfaces used in recent modeling sessions.

When selecting and adding a physics interface to the **Selected physics** section, you can review and optionally modify any dependent variable names in the **Dependent variables** section and, for some physics interfaces and PDE interfaces, specify the number of dependent variables. For other physics interfaces you can edit both the name of the field and the field components. Examples of fields with components are the displacement field in structural mechanics and the velocity field in fluid flow.

When you are finished adding the physics, click the **Next** button (⇨) to go to the **Select Study Type** page.

THE SELECT STUDY TYPE PAGE

From the branches under **Studies** (), you select the type of study you want to perform. The available options on the **Select Study Type** page depend on the set of physics interfaces (and mathematics interfaces) included in the model. Some study types are applicable to all physics for which you choose to solve, while others are not; but all are in some way available. You can select the study type from one of the following branches:



Model Wizard

Select Study Type

Studies

- Preset Studies for Selected Physics
  - Frequency-Stationary
  - Frequency-Transient
  - Frequency Domain
  - Stationary
  - Time Dependent
- Custom Studies

- **Preset Studies**—The study types suggested by a single physics interface (or mathematics interface) if only one has been chosen.
- **Preset Studies for Selected Physics**—The study types applicable to all of the interfaces which you have chosen to solve for.
- **Custom Studies**—This branch contains study types for which not all physics solved for can generate suitable equations:
  - **Preset Studies for Some Physics**—The study types recognized by some, but not all, of the interfaces being solved for.
  - **Other studies**—Any fundamental study types (**Stationary**, **Time Dependent**, **Eigenfrequency**, **Eigenvalue** and **Frequency Domain**) which are not applicable to any of the interfaces being solved for. There is also an empty study type.

The **Selected Physics** table shows all physics interfaces included in the model. You can choose not to solve for a particular physics in the selected study by clicking in the **Solve for** column. This updates the contents of the different branches under **Studies** to reflect applicable studies only for the physics being solved for.

Basic Steps to Build a Model

1 In the **Model Wizard**, **Select a Space Dimension** for the model: **3D**, **2D axisymmetric**, **2D**, **1D axisymmetric**, **1D**, or **0D**. Remember, not all physics interfaces are available for all space dimensions.

2 Click the **Next** button (⇨).

The **Add Physics** page opens.

3 There are several ways to select one or several physics interfaces to add to the model. There are also mathematics interfaces (for PDEs, ODEs, and DAEs) in the **Mathematics** branch. Double-click the physics interface, click the **Add Selected** (✚) or **Remove Selected** (✖) buttons, or right-click and select **Add Selected**. The **Selected physics** section lists your choices.

Tip	When selecting and adding physics or mathematics interfaces in the **Selected physics** section, you can review and optionally modify its dependent variable names in the **Dependent variables** section. For some interfaces, also specify the number of dependent variables.

4 Click the **Next** button (⇨).

5 On the **Select Study Type** page, select a study type.

6 Click the **Next** button (⇨) (when available), otherwise click the **Finish** button (🏁) to confirm the selections and close the **Model Wizard**.

After clicking **Finish**, other nodes, such as **Materials** (▨) and **Mesh** (▨) automatically display in the **Model Builder** in the default sequence.

Tip	Click the **Finish** button (🏁) anytime, and any operation selected can be added to, changed, or deleted in the **Model Builder**.

Model Nodes by Space Dimension

When you click the **Finish** button ( ) on the **Model Wizard** window, a **Model** node is added to the **Model Builder** based on the space dimension selected in the first step. The Model node has different icons as shown in Table 2-2.

TABLE 2-2: THE MODEL ICONS BY SPACE DIMENSION

ICON	SPACE DIMENSION
•	no space dimension (0)
—	1D
┤	1D axisymmetric
▱	2D
▱	2D axisymmetric
◼	3D

See Also

- Geometry Modeling and CAD Tools
- The Model Wizard

The Model Builder Window

The modeling procedure is controlled through the **Model Builder** window ( ), which includes a *model tree* with all the functionality and operations for building and solving models and displaying the results. These are introduced to your modeling procedure by adding a *branch*, such as the *Geometry* branch ( ). Branches can have further nodes that relate to their parent node. A *leaf* is considered to be the final node in a branching structure (tree) and usually has a number of *attributes* with their own *settings* that are characteristic to it. Branches and subbranches can also contain attributes and settings. See Figure 2-1.

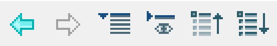


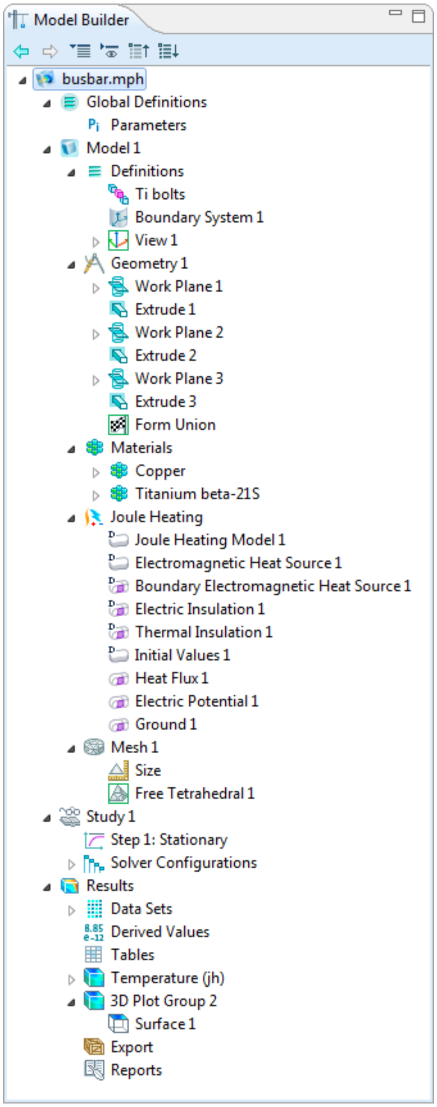
Figure 2-3: The buttons located on the Model Builder window.

TABLE 2-3: BUTTONS ON THE MODEL BUILDER WINDOW

NAME	ICON	DESCRIPTION OR LINK TO MORE INFORMATION
Previous Node	⇦	Navigate back to the last node selected. The button is enabled once you start to work in the Model Builder and navigate between nodes. See also Keyboard Shortcuts.
Next Node	⇨	Navigate to the next node in the sequence. See also Keyboard Shortcuts.
Show	☰	Showing and Expanding Advanced Physics Sections
Expand Sections	☰	Showing and Expanding Advanced Physics Sections
Collapse All	☰↑	Expanding and Collapsing All Nodes
Expand All	☰↓	Expanding and Collapsing All Nodes
Minimize and Maximize	▭ □	Available on most windows to maximize and minimize the window.

BRANCHES IN THE MODEL TREE

You can proceed through your modeling in the Model Builder by selecting the branches in the order suggested by their default positions, from the top down, or selecting and defining each branch as needed.



Model Builder

- busbar.mph
  - Global Definitions
    - Parameters
  - Model 1
    - Definitions
      - Ti bolts
      - Boundary System 1
      - View 1
    - Geometry 1
      - Work Plane 1
      - Extrude 1
      - Work Plane 2
      - Extrude 2
      - Work Plane 3
      - Extrude 3
      - Form Union
    - Materials
      - Copper
      - Titanium beta-21S
    - Joule Heating
      - Joule Heating Model 1
      - Electromagnetic Heat Source 1
      - Boundary Electromagnetic Heat Source 1
      - Electric Insulation 1
      - Thermal Insulation 1
      - Initial Values 1
      - Heat Flux 1
      - Electric Potential 1
      - Ground 1
    - Mesh 1
      - Size
      - Free Tetrahedral 1
  - Study 1
    - Step 1: Stationary
    - Solver Configurations
  - Results
    - Data Sets
    - Derived Values
    - Tables
    - Temperature (jh)
    - 3D Plot Group 2
      - Surface 1
    - Export
    - Reports

The main branches include:

- The *Global Definitions branch* ( ): for global definitions, for example, defining parameters and functions that you can use everywhere.
- The *Model branches*: for defining models. A Model branch includes the associated subbranches of *Definitions* ( ), for locally defining parts of your model, Geometry, Materials, Physics, and Meshes. A Model branch includes functionality for local Definitions because several models can separately be defined in one multiphysics file, for example, when treating certain parts of the model in 2D and other parts in 3D, or when setting up a system model with several components.
- The *Study branch* ( ): where you can set up study steps and solver configurations for solving a model using one or more study types for different analyses.
- The *Results branch* ( ): for presenting and analyzing results.

See Also

- The Model Wizard and Model Builder
- Key to the Nodes and Toolbar Buttons
- Global and Local Definitions
- Building a COMSOL Model
- Solver Studies and Study Types
- Results Evaluation and Visualization
- Visualization and Selection Tools

SUBBRANCHES IN THE MODEL TREE

One level below, each Model branch contains the following subbranches:

- Just as the Global Definitions branch collects parameters, variables, and functions accessible at all levels in the Model Builder, the Definitions branch (one per Model) collects the definitions of variables, functions, and other objects whose geometric scope is restricted to a single model. An example of the type of objects you can add under the Definitions branch is the *Selection* feature ( ), which saves selections of geometric entities (boundaries, for example) that relate to a region or part of the overall geometry for reuse in multiple operations later in the modeling process.
- The *Geometry* branch ( ) contains the definition of the model's geometry.
- Under the *Materials* branch ( ), you can collect all material properties organized in Materials nodes with a defined geometric scope. Material properties required by

any of the physics interfaces show up automatically in the defined material's settings window.

- Each *Physics Interface* you include in a model forms its own subbranch placed below the Materials branch.
- The *Mesh* branch ( ) collects all meshes defined for a model. If there is only a single mesh in a model, its Mesh node appears directly under the corresponding Model node.

See Also

- Geometry Modeling and CAD Tools
- Global and Local Definitions
- Materials
- Overview of the Physics Interfaces
- Key to the Nodes and Toolbar Buttons
- Meshing

NAVIGATING IN THE MODEL TREE

The following methods are available to select nodes, expand and contract branches, and move up and down the nodes in the model tree:

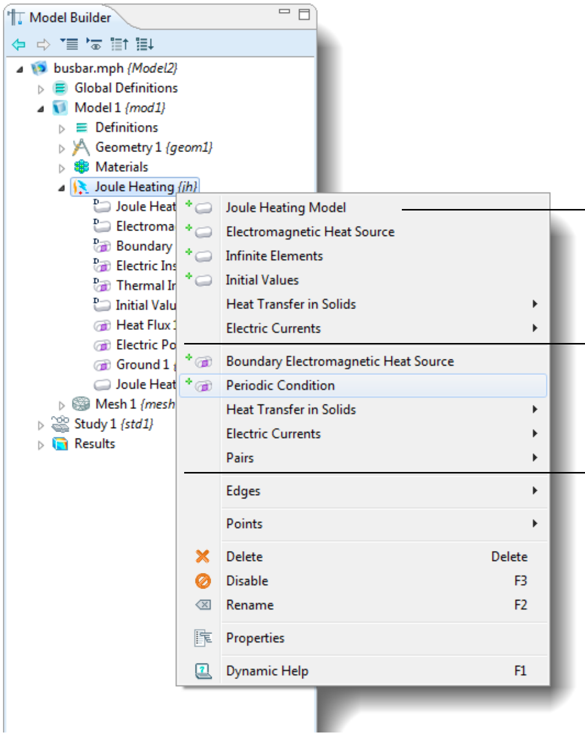
- To select a node and open its settings window, click the node in the model tree.
- To move to the node above, use the up arrow key; to move to the node below, use the down arrow key.
- To expand a branch to display all nodes in the branch, click the small plus sign next to the branch icon in the model tree, or press the right arrow key. To collapse a branch to display only the main branch node, click the small minus sign next to the branch icon in the model tree, or press the left arrow key.

About Sequencing in COMSOL Multiphysics

COMSOL Multiphysics operates through *sequencing*. By adding nodes to a model branch in the Model Builder in a certain order, you set up sequences of operations such as geometry, mesh, solver, and results sequences. The software records a sequence as a macro in the background. A solver sequence, for example, could define your model with one solver and then, using the returned solution, solve it with an alternative solver. If you make changes to a node, the software automatically updates other parts of the sequence that are dependent on this change. For most sequences, you run the sequence by right-clicking the top node for the branch and selecting **Build All** (geometry and mesh), **Compute** (studies), or **Plot** (plot groups), or by pressing F8.

Opening a Context Menu to Add Nodes

Right-click nodes in the **Model Builder** to open a *context menu* and add additional, and relevant, functionality, operations, or attributes to the model sequence. After selecting an option from the list, a settings window opens to the right (by default) of the **Model Builder** window ( ). In the context menu, a plus sign next to the icon means a node of that type is added to the Model Builder. On Windows, you can also open the context menu using the Menu key or by pressing Shift+F10, on Linux by pressing Shift+F10, and on the Mac by pressing Ctrl+F10.



Domain level settings for unsorted nodes

Boundary level settings for unsorted nodes

Edges and points submenus

Note

The layout of the context menu depends whether the nodes are organized by space dimension. The default is not to sort the nodes. See Sorting Nodes by Space Dimension and Type for an example comparing the different context menus.

See Also

- Keyboard Shortcuts
- Physics Node Context Menu Layout
- Key to the Nodes and Toolbar Buttons

Moving Nodes in the Model Tree

The order of the nodes in some of the branches affects the evaluation of the sequence that they define. In the following branches it is possible to move nodes up and down to control the evaluation of the sequence or the order in which they appear:

- The Selection nodes in the Definitions branch ( ).
- The Geometry nodes in the Geometry branch ( )
- The Material nodes in the Materials branch ( ).
- The nodes for physics interfaces (such as material models, boundary conditions, and sources) in the Physics branch ( ), except the default nodes.
- The mesh nodes in the Mesh branch ( ).
- The study step nodes in the Study branch ( ).

  In addition, the order of the nodes can be rearranged in the following subbranches under Results ( ):
- The derived values nodes in the Derived Values branch (8.85 e-12).
- The table nodes in the Tables branch ( ).
- The plot nodes in the Plot Group branches ( ).
- The data, image, and movie export nodes in the Export branch ( )
- The Report nodes in the Reports branch ( ).

To move nodes use one of the following methods:

- Select the nodes and use the mouse to drop them in another applicable position in the model tree. A horizontal line indicates where in the model tree the moved (or copied) nodes get inserted when releasing the mouse.
- Right-click the selected nodes and select **Move Up** (⇧) or **Move Down** (⇩).
- Use the keyboard shortcuts Ctrl+up arrow or Ctrl+down arrow to move nodes up or down.

Note

For physics interface nodes it is not possible to move the default nodes (for the default boundary condition, for example). It is possible to create a copy of a default node, which initially has no selection. To "click-and-drag" a default node creates a copy regardless if the Ctrl key is pressed or not.

Going to the Source Node

In the settings window for many nodes, other nodes can be referenced in the model tree such as a model, solution, study or study step, or data set.

Nodes where you refer to other nodes include plot groups, data sets, and solvers; in such nodes' settings windows, click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

Copying, Pasting, and Duplicating Nodes

It is possible to copy and paste many of the nodes in the Model Builder to create additional nodes with identical settings. Some nodes can also be duplicated underneath the original node. You can also move, copy, and duplicate nodes using "drag-and-drop" of nodes in the Model Builder.

Tip

After duplicating or copying nodes such as functions and selections, it is a good idea to change the name of the function or selection so that it is different from the original.

To select **Copy** , **Paste** , and **Duplicate** , either right-click selected nodes or select the options from the main **Edit** menu.

Tip

Duplicate is a convenient way to copy and paste in one step.

USING COPY-PASTE FOR NODES

After selecting **Copy** , right-click the parent node (or from the main menu select **Edit>Paste**) and select **Paste Analytic** , for example, or press Ctrl+V to paste a copied node (an **Analytic** function node in this case) to the parent node's branch. You can also create a copy of a node by Ctrl-clicking it and dragging a copy to an applicable location. A small plus sign at the cursor indicates that you drag a copy of the selected node.

Nodes that can be copied (and duplicated) include the following:

- Functions, which are possible to copy from one **Definitions** or **Global Definitions** branch to another.

- Physics interface nodes, which can be copied within the same physics interface or to another identical physics interface.
- Study steps, which are possible to copy from one **Study** branch to another.
- Plot nodes, which are possible to copy from one plot group to another.

See Also	- Global and Local Functions - Solver Studies and Study Types - About Plot Groups and Plotting - Copy in the *COMSOL Multiphysics Reference Guide*

DUPLICATING NODES

After right-clicking to select **Duplicate** (or from the main menu selecting **Edit>Duplicate**), COMSOL Multiphysics adds identical nodes underneath the original nodes on the same branch. Duplicate nodes by Ctrl-clicking and dragging duplicates to an applicable location. A small plus sign at the cursor indicates that you drag duplicates of the selected nodes. Nodes that can be duplicated include the following:

- Global and Local Functions
- Selections (see User-Defined Selections)
- Model Couplings
- Coordinate Systems
- Probes
- Geometry objects (see Copying and Pasting Geometry Objects)
- Study steps (see Adding Study Steps)
- Plot Groups (see About Plot Groups and Plotting)
- Tables (see Defining Derived Values and Tables)
- Integrations in the Derived Values branch under **Results**
- Images, animations, plots, data, and meshes in the **Report** branch under **Results** (see Exporting Data and Images)

Expanding and Collapsing All Nodes

Using these buttons you can quickly see all current nodes in the **Model Builder** and also quickly collapse the tree to start from a minimal view of the model tree.

- To expand all nodes in the model tree, click the **Expand All** button ( ) on the **Model Builder**’s toolbar.
- To collapse all nodes in the model tree, except the top nodes on the main branch, click the **Collapse All** button ( ) on the **Model Builder**’s toolbar.

Undoing and Redoing Operations

Note

Undo is not possible for nodes that are built directly, such as geometry objects, solutions, meshes, and plots.

It is possible to undo the last operation for operations like adding, disabling, moving, and deleting nodes in the **Model Builder** as well as changing values in the settings window. You can undo or redo several successive operations.

- To undo the last operation, from the main menu, select **Edit>Undo**, or click the **Undo** button ( ) on the main toolbar, or press Ctrl+Z.
- To redo an undone operation, from the main menu, select **Edit>Redo**, or click the **Redo** button ( ) on the main toolbar, or press Ctrl+Y.

Clearing All Meshes and Clearing all Solutions

Use a **Clear** function to keep the features and be able to recreate the mesh or solution by building the sequence again. Use a **Delete** function to completely remove a mesh or solution.

Note

Undo is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

If you have a model geometry with several meshes you can clear all meshes at the same time. From the main menu, select **Edit>Clear All Meshes** ( ).

If you have a model geometry with several solutions you can clear all solutions at the same time. From the main menu, select **Edit>Clear All Solutions**.

See Also

- Clearing Meshes and Deleting Meshing Sequences
- Clearing, Deleting, Disabling, and Enabling a Solver or Solutions

Sorting Nodes by Space Dimension and Type

Note

The default setting, found under **Model Builder** in the Editing Preferences Settings section, disables sorting in new models. Go to the preference settings to change the default as required.

The **Sort Nodes by Space Dimension** (for physics interface nodes), **Sort Nodes by Type** (for **Definitions** nodes), or **No Sorting** options are available from the context menu for these features:

- **Global Definitions**—nodes can be organized by type: **Variables**, **Functions**, or **Groups**.
- **Definitions**—nodes can be organized by type: **Functions**, **Variables**, **Selections**, **Probes**, **Model Couplings**, **Coordinate Systems**, **Pairs**, and **Domain Properties** (this includes Perfectly Matched Layers and Infinite Elements).
- For any physics interface, nodes can be organized by space dimension, that is, by geometric entity level: **Domains**, **Boundaries**, **Edges** (3D only), or **Points**.

Tip

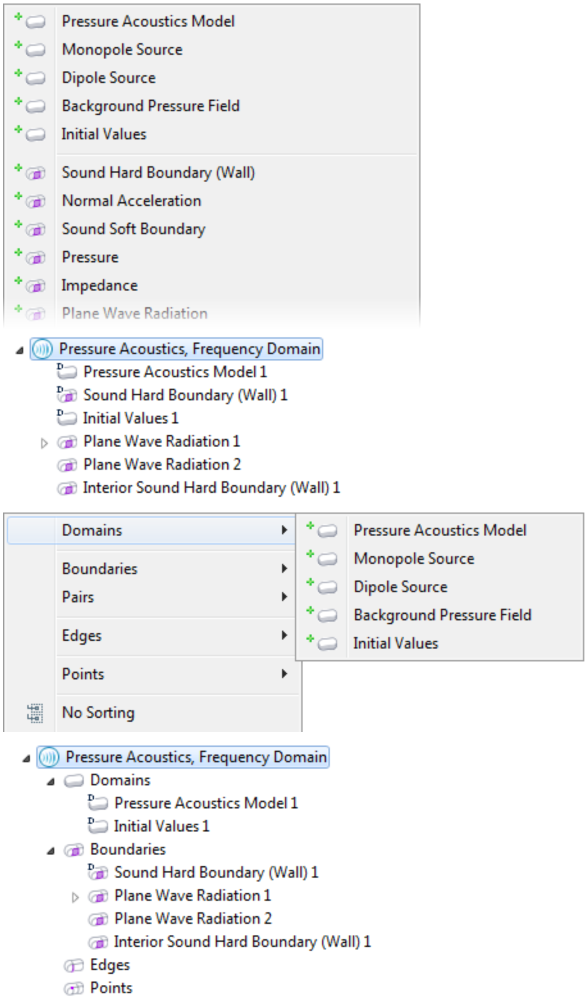
For physics interfaces, when either of these options is selected, there are changes to the way the nodes are organized in both the Model Builder as well as when you right-click to view the context menu.

ORGANIZING NODES BY SPACE DIMENSION

The default is **No Sorting** in the **Model Builder** for all new models. Right-click and select **Sort Nodes by Space Dimension** ( ) from the context menu to both sort the nodes in the Model Builder and in the context menu (see Figure 2-4).

In Figure 2-4 for the **Pressure Acoustics, Frequency Domain** physics interface and when **Sort by Space Dimension** is selected, the default **Pressure Acoustic Model** and **Initial Values** subnodes are included under the **Domains** node, and four boundary level nodes are included under **Boundaries**. However, there are no nodes under **Edges** or **Points** even though these nodes display in the **Model Builder**.

When **No Sorting** (▦) is selected from the context menu (or the default is kept), the tree is flattened and all nodes are unsorted. Only default physics feature nodes and user-added nodes are included in the **Model Builder** sequence.



No sorting of nodes in the context menu

No sorting of nodes in the Model Builder

Pressure Acoustics, Frequency Domain
Pressure Acoustics Model 1
Sound Hard Boundary (Wall) 1
Initial Values 1
Plane Wave Radiation 1
Plane Wave Radiation 2
Interior Sound Hard Boundary (Wall) 1

Sorting nodes by space dimension in the context menu

Sorting nodes by space dimension in the Model Builder

Pressure Acoustics, Frequency Domain
Domains
Pressure Acoustics Model 1
Initial Values 1
Boundaries
Sound Hard Boundary (Wall) 1
Plane Wave Radiation 1
Plane Wave Radiation 2
Interior Sound Hard Boundary (Wall) 1
Edges
Points

Figure 2-4: The context menu and Model Builder sequence for physics interface nodes when Sort Nodes by Space Dimension or No Sorting is selected. For the Global Definitions and Definitions nodes, the same principles apply but the nodes are organized by type instead.

Organizing Nodes by Type

The same principles apply for the **Global Definitions** and **Definitions** nodes as for the physics interface nodes, except the nodes are organized by type, that is:

- Under **Global Definitions** the types are **Variables**, **Functions**, or **Groups**.
- Under **Definitions** the types are **Functions**, **Variables**, **Selections**, **Probes**, **Model Couplings**, **Coordinate Systems**, **Pairs**, and **Domain Properties**.

The default is **No Sorting** in the **Model Builder** for all new models. Right-click and select **Sort Nodes by Type** ( ) from the context menu to both sort the nodes in the Model Builder and in the context menu.

When **No Sorting** ( ) is selected from the context menu (or the default is kept), the tree is flattened and all nodes are unsorted. Only default **Definitions** or **Global Definitions** nodes and user-added nodes are included in the **Model Builder** sequence.

See Also

- Overview of the Physics Interfaces
- Building a COMSOL Model

Introduction to the COMSOL Windows

In addition to The Model Wizard and The Model Builder Window, there are several other COMSOL windows that are integral to building your model.

In this section:

- The Node Settings Windows
- The Graphics Window
- The Messages Window
- The Progress Window
- The Log Window
- The Results Window
- The Help Window
- The Model Library Window
- The Material Browser Window
- The External Process Window

This section also contains information about these desktop topics:

- Errors and Warnings
- The About COMSOL Multiphysics Box

See Also

- Introduction to the Model Wizard and the Model Builder
- Building a COMSOL Model
- Information About Memory Use

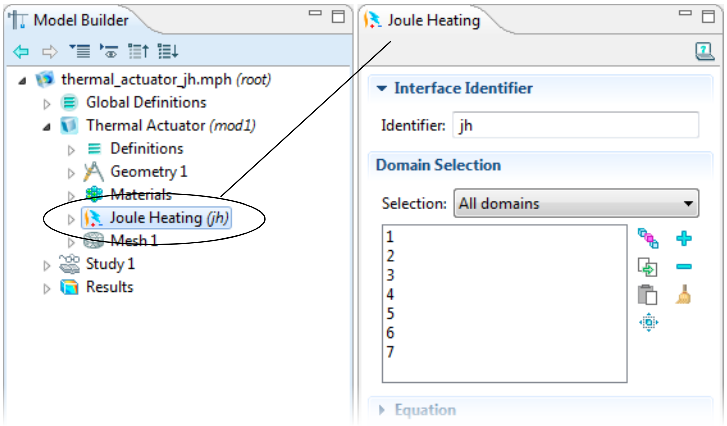
The Node Settings Windows

When any node is clicked in the Model Builder, the corresponding settings window opens with the same name as the node. The settings window is an interactive environment for defining an operation or model property while the model and changes are displayed in the Graphics window.

When an operation or property is updated in the settings window, its effect on the model is displayed in the Graphics window either instantaneously or by clicking the

Build Selected button ( ) or the **Plot** button ( ) available in some of the settings window toolbars.

To select the parts of the model to define in a specific settings window, select the relevant geometric entities either directly in the displayed model in the Graphics window or from the Selection List window.



Identifier: jh

Selection: All domains

See Also

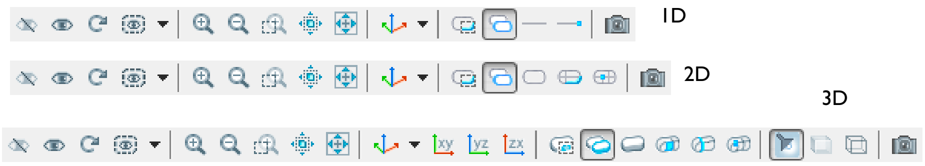
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

The Graphics Window

The **Graphics** window ( ) presents a graphical view of the geometry, mesh, and results of the model. See Figure 2-1 for an example.

The Graphics window has useful tools for changing the view and selecting multiple entities—geometry objects as well as domains, boundaries, edges, and points for defining the physics or selecting geometric entities for fine-tuning the mesh or evaluating quantities in a certain part of the model, for example.

The Graphics toolbar at the top of the Graphics window contains a set of tools for changing the visualization (for example, to zoom in or out or to add transparency) and making selections. See Figure 2-1.



See Also

- Visualization and Selection Tools
- Selecting and Deselecting Geometric Entities
- The Graphics Toolbar Buttons
- Creating 1D, 2D, 3D, and Polar Plots

Plot Windows

Plot windows are closable graphics windows. They are automatically generated for displaying results while solving if your model contains probes; see the section Getting Results While Solving for details. You can also create plot windows manually by choosing **Plot In>New Window** from the context menu for a plot group.

The Messages Window

The **Messages** window (✉) contains information useful to you after an operation is performed. The window displays by default, but it can also be opened from the main menu by selecting **View>Messages**. Click the **Clear** button (🧹) to clear the **Messages** window from all messages. The information in this window includes:

- The number of mesh elements and degrees of freedom in your model.

- Solution times.
- Error messages. The messages are in chronological order and can be scrolled through.

See *Also*

- Meshing
- Solvers and Study Types

The Progress Window

The **Progress** window ( ) displays the progress of the solver or mesher during the process, including a progress bar and progress information for each solver or mesher. You can cancel or stop the solver process if the model's solving time or likelihood for convergence is not progressing as expected using the **Cancel** button or the **Stop** buttons (one for each solver). Use the horizontal or vertical scrollbars if needed.

This window is always available but is empty when no progress information is available. You can make it the active window from the main menu by selecting **View>Progress**. For a log of the progress information, see the **Log** window.

You can also view the progress from the status bar in the lower-right corner of the COMSOL Desktop. On the status bar you can follow the progress, cancel the solver process by clicking the **Cancel** button ( ), or open the **Progress** window by clicking the **Progress** button ( ).

See *Also*

- Progress and Log Information
- Meshing
- Solvers and Study Types

The Log Window

The **Log** window ( ) contains information from previous solver runs. A horizontal divider (===============) indicates the start of a new solver progress log. To differentiate logs from different models, the log contains a horizontal divider displaying the name of the Model MPH-file each time a model is opened. For example,

```
=============== Opened thin_layer_diffusion.mph ================
```

It also contains a similar divide when you save a model to a new file (using **Save As**):

```
================= Saved thin_layer_diffusion.mph =================
```

You can scroll the contents of the **Log** window to display information from earlier runs. Click the **Scroll Lock** button (🔒) to stop the window from scrolling the log during a solver call, for example. Click the **Scroll Lock** button again to resume scrolling. Click the **Clear** button (🧹) to clear the **Log** window from all information.

This window is always available. Make it the active window from the main menu by selecting **View>Log**. For progress information during a solver or mesher process, see the **Progress** window.

By default, the buffer size of the **Log** window is limited to 300,000 characters. Change the size in the **Preferences** dialog box, which is opened from the **Options** menu. Click **General** in the list, and then enter a maximum buffer size (in characters) in the **Log window size (characters)** field.

See Also

Progress and Log Information

The Results Window

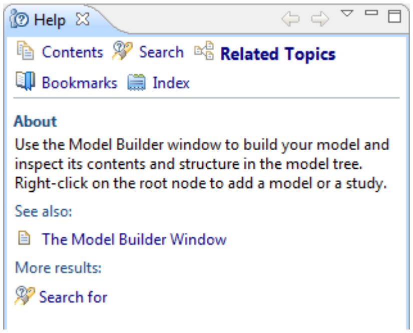
The **Results** window (🗒) displays the results from integral and variable evaluations defined in Derived Values features and stored in Tables features. The window displays by default, but it can also be opened from the main menu by selecting **View>Results**.

See Also

- Processing and Analyzing Results
- Defining Derived Values and Tables

The Help Window

The **Help** window ( ) provides access to *Dynamic Help* ( ) in the COMSOL Desktop and has these pages: Related Topics, Contents, Search, Bookmarks and Index



About

Use the Model Builder window to build your model and inspect its contents and structure in the model tree. Right-click on the root node to add a model or a study.

See also:

The Model Builder Window

More results:

Search for

See Also

- The Help Documentation Window
- Table 2-4 for descriptions of the Help buttons

Contents

The **Contents** section is a tree-based menu structure of COMSOL Multiphysics' documentation.

Search

The **Search** option is a search engine for returning text strings and combinations of same. Entering two text strings in the **Search expression** field returns all instances where both strings display in the same document (such as a section or example model in a manual). Use quotation marks around a text string to search for exactly that phrase, that is, the words in the order given within the quotation marks. The @ character ("at sign") works as a wild-card.

Tip

Specify the search scope by expanding the **Scope** section, where you can select to include the local help, the COMSOL website on www.comsol.com, or a custom search scope.

The default setting is to search all local help files and the COMSOL website (including the support knowledge base, discussion forum, model gallery, and papers & presentations sections of the website). See Selecting the Scope From the Documentation Window and Changing the Search Scope From the Dynamic Help Window for details.

Selecting the Scope From the Documentation Window

Note

This is available from both the **Help>Documentation** and **Help>Dynamic Help** Search windows, but with slightly different options.

1 Select **Help>Documentation** and click **Scope**. The **Select Scope** window opens.

2 The default is to **Show all topics**. To refine the search, click **Show only the following topics** and then either add **New** topics, or **Edit** or **Remove** existing topics from the list.

- Click **New** to add topics. The **New Scope** window opens where you can define a **List name** and add as many topics as you want from the lists.
- Once a **Scope** list is created, click **Edit** to make changes or **Remove** to delete the list.

3 When done with the Scope lists, click **OK** to save or **Cancel** to exit.

Changing the Search Scope From the Dynamic Help Window

Note

This is available from both the **Help>Documentation** and **Help>Dynamic Help** Search windows, but with slightly different options.

1 Select **Help>Dynamic Help.** Click **Search**.

2 From the **Search** window, click **Search scope**.

3 The **Default** searches the **Local Help** and the **COMSOL Support Knowledge Base**. Click to clear the check boxes as required.

4 To create a customized **Search Scope** set, click **Default** next to **Search scope** to add **New** topics, or **Rename**, **Edit**, or **Remove** existing topics from the list.

- Click **New** to add topics. The **New Scope Set** window opens where a **Scope Set Name** can be defined.
- Once a **Scope Set** is created, click **Edit** to make changes or **Rename** it, or **Remove** to delete the set.

5 When done with creating or editing a **Scope Set**, click **OK** to save or **Cancel** to exit.

6 Click **Advanced Settings** on the **Search** window (or **Edit** from the Scope Set window) to open the **Search Scope** window. This is where you can customize the Local Help or COMSOL Support Knowledge Base searches.

- The default **Local Help** setting is **Search all topics**. To refine the search, click **Search only the following topics** and select the check boxes as required. The **COMSOL Support Knowledge Base** is set to search the Support Knowledge Base at www.comsol.com. Click to clear the **Enable search engine** check box if required.

RELATED TOPICS

Related Topics has a brief description of the respective branch, sub-branch, or leaf node that is active in the Model Builder. This is an interactive environment. Click a node followed by the Related Topics window to update the contents instantly.

To view more detailed information, including summaries of the different attributes and settings related to the node, click on one or more of the topics found under the **See also** section. This takes you into COMSOL Multiphysics' documentation.

The **More results** section, if present, gives a shortcut to the Search page with a search string related to the active Model Tree node or COMSOL Desktop component.

BOOKMARKS AND INDEX

The **Bookmarks** section has a list of saved search criteria and the **Index** is a tree-based menu structure of the documentation indexes.

OPENING OR REFRESHING THE HELP WINDOW

There are different ways to open or refresh the **Help** window:

- Click the **Dynamic Help** button ( ) in the active window's toolbar (not all windows have this toolbar).
- From the main menu, select **Help>Dynamic Help**.
- Press F1.

- Right-click a node in the **Model Builder** and select **Dynamic Help** ( ) (the last item in the menu).

  By default, the **Help** window is added to the right of the **Graphics** window.

Sticky Help

With the **Sticky help** option cleared (the default) from the **Options>Preferences>General** menu, the **Help** window always updates its contents to shows the relevant context help information when clicking a window tab or on a model tree node in the **Model Builder**, regardless of which part of the help system that is displayed in the **Help** window.

If the **Sticky help** check box is selected (see Editing Preferences Settings), the contents of the Help window are no longer updated when the active node or window is changed once entering the COMSOL Multiphysics documentation. To return to the Related Topics page, either click the **Back** button (the yellow left arrow button) or refresh the Help window.

Note	The setting for **Sticky help** is preserved between modeling sessions.

Important	The first time the **Help** window is opened on your computer, the operating system might issue a firewall security warning. To use Dynamic Help, allow COMSOL Multiphysics access through the firewall.

Preload Help Files

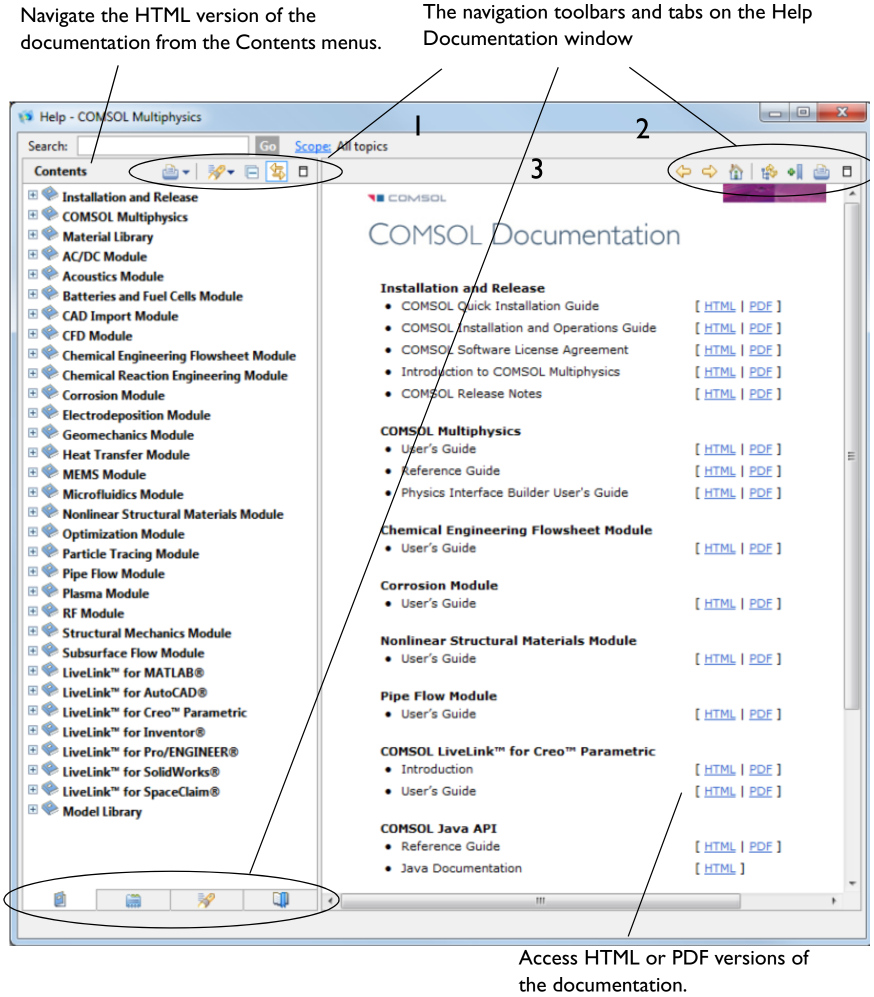
With the **Preload help files** option selected from the **Options>Preferences>General** menu, it means that the help files are preloaded when launching COMSOL Multiphysics. By preloading the help files the context help opens faster when used for the first time. To disable this feature, see Editing Preferences Settings and clear the check box.

THE HELP DOCUMENTATION WINDOW

When **Help>Documentation** is selected, a different Help window opens than the one described in The Help Window although the functionality is the same and all bookmarks are saved in both places.

From this window you can navigate to PDF or HTML versions of the User's Guides (availability is based on your license), as well as Search, Bookmark, Print Topics, and

Link with Contents. You can also narrow the **Scope** of the topics searched. See Table 2-4 for descriptions of the buttons shown in Figure 2-5 and Figure 2-6.



Navigate the HTML version of the documentation from the Contents menus.

The navigation toolbars and tabs on the Help Documentation window

Help - COMSOL Multiphysics

Search: Go Scope: All topics

Contents

- Installation and Release
- COMSOL Multiphysics
- Material Library
- AC/DC Module
- Acoustics Module
- Batteries and Fuel Cells Module
- CAD Import Module
- CFD Module
- Chemical Engineering Flowsheet Module
- Chemical Reaction Engineering Module
- Corrosion Module
- Electrodeposition Module
- Geomechanics Module
- Heat Transfer Module
- MEMS Module
- Microfluidics Module
- Nonlinear Structural Materials Module
- Optimization Module
- Particle Tracing Module
- Pipe Flow Module
- Plasma Module
- RF Module
- Structural Mechanics Module
- Subsurface Flow Module
- LiveLink™ for MATLAB®
- LiveLink™ for AutoCAD®
- LiveLink™ for Creo™ Parametric
- LiveLink™ for Inventor®
- LiveLink™ for Pro/ENGINEER®
- LiveLink™ for SolidWorks®
- LiveLink™ for SpaceClaim®
- Model Library

COMSOL

COMSOL Documentation

Installation and Release
- COMSOL Quick Installation Guide [HTML | PDF]
- COMSOL Installation and Operations Guide [HTML | PDF]
- COMSOL Software License Agreement [HTML | PDF]
- Introduction to COMSOL Multiphysics [HTML | PDF]
- COMSOL Release Notes [HTML | PDF]

COMSOL Multiphysics
- User's Guide [HTML | PDF]
- Reference Guide [HTML | PDF]
- Physics Interface Builder User's Guide [HTML | PDF]

Chemical Engineering Flowsheet Module
- User's Guide [HTML | PDF]

Corrosion Module
- User's Guide [HTML | PDF]

Nonlinear Structural Materials Module
- User's Guide [HTML | PDF]

Pipe Flow Module
- User's Guide [HTML | PDF]

COMSOL LiveLink™ for Creo™ Parametric
- Introduction [HTML | PDF]
- User's Guide [HTML | PDF]

COMSOL Java API
- Reference Guide [HTML | PDF]
- Java Documentation [HTML]

Access HTML or PDF versions of the documentation.

Figure 2-5: Overview of the Help Documentation window.

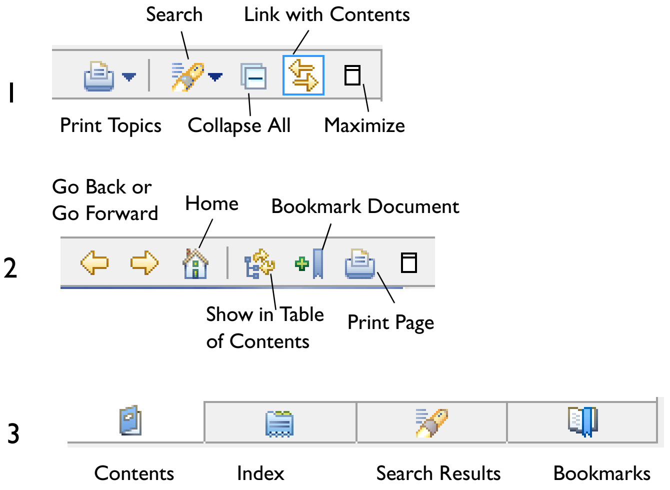


Figure 2-6: The Help Documentation toolbars. The numbers correspond to the toolbars in Figure 2-5.

THE HELP DOCUMENTATION TOOLBAR BUTTONS AND WINDOWS

TABLE 2-4: THE HELP DOCUMENTATION TOOLBAR BUTTONS AND TABS

WINDOW NAME AND BUTTONS	DESCRIPTION AND TASK
DOCUMENTATION	The window where the documentation contents are displayed.
Maximize and Minimize	Available on most windows to maximize and minimize the window.
Go Forward	Navigates forward within a topic, but only to the end of the current list.
Go Back	Navigates backwards to the topics previously selected.
Show in Table of Contents	Opens the **Contents** window to display where the topic is located within the tree structure.
Home	Returns to the COMSOL Documentation Home page.
Print Page	Prints the current page.
Bookmark Document	Adds the current page to all Help **Bookmarks** windows.
Next	Navigates forward to the next topic in the order displayed.

TABLE 2-4: THE HELP DOCUMENTATION TOOLBAR BUTTONS AND TABS

WINDOW NAME AND BUTTONS	DESCRIPTION AND TASK
Previous	Navigates backwards to the previous contents topic.
Show in External Window	Available from the Dynamic Help window only, this button opens the same page as when you select **Help>Documentation** from the menu.
CONTENTS	Displays the contents of the User's Guides in a tree structure.
Print Topics	After selecting a branch in the **Contents** window, select **Print selected topic** or **Print selected topic and all subtopics**.
Search	After selecting a branch in the **Contents** window, select **Search selected topic** or **Search selected topic and all subtopics** and enter a **Search Expression**. The results display in the **Search Results** window.
Collapse All	Collapses (closes) all branches in the Contents window, except the top ones.
Link with Contents	Click to activate (turn ON) to link the Contents window to the topic you are navigating. Click to deactivate (turn OFF).
SEARCH RESULTS	Displays links to the Search results. You can also display the results categories and descriptions.
Show result categories	Click to activate (turn ON) to display result categories by module. Click to deactivate (turn OFF).
Show result descriptions	Click to activate (turn ON) to display result descriptions Click to deactivate (turn OFF).
BOOKMARKS	Displays any bookmarks to topics added. You can also delete one or all of the bookmarks saved to this window.
Delete selected bookmark	This button is available on the Bookmarks window. Click a bookmark then click this button to remove it.
Delete all bookmarks	This button is available on the Bookmarks window. Click to remove all bookmarks.
INDEX	Search the index and display all the results.

The Model Library Window

Select **View>Model Library** ( ). Browse and load model files to the COMSOL Desktop. To locate any models using a specific feature, type a suitable keyword or phrase in the field at the top of the **Model Library** window and click **Search**.

The **Model Library** displays all the models included with an installation. When you open the **Model Library** window, the folders each have a unique icon and contain the models specific to the module installed.

Some models may have been delivered with an MPH-file that contains no stored meshes or solutions. Such models are indicated by a gray icon in the **Model Library** window (●). You can download the complete MPH-files via **Model Library Update** or generate them by building the models' mesh sequences and computing the studies. Note that the procedure for restoring the solutions of a model may involve other steps, such as adjusting physics settings; see the model's documentation for details if the naive approach does not work.

See Also

To set up some additional search options, see Selecting the Scope From the Documentation Window as described for the **Help** window.

The **Model Library** window's toolbar contain the buttons as listed in Table 2-5.

TABLE 2-5: MODEL LIBRARY TOOLBAR BUTTONS

NAME	BUTTON	DESCRIPTION
Refresh	↻	Click to update the Model Library tree, for example if a file has been saved in a folder under the Model Library root directory since opening the Model Library window.
Set Model Library Root		Click to launch the Model Library Root window where you can enter a new root directory for the Model Library or just inspect its current location.
Dynamic Help	?	Select a model in the Model Library tree and click this button to read a brief description of the model in the Help window. To read the full model documentation, including step-by-step modeling instructions, click the link under See also in the Help window.
Add User Model Library	+	Click to add a new folder to the Model Library and store your own model files.
Delete Selected	✕	This button is enabled after a User Model Library folder is created. Click any model or folder and then click this button to remove it from the Model Library.

To read the model documentation in PDF format, select the model in the tree and then click the **Open PDF** button. Alternatively, right-click the model node and select **Open PDF** from the context menu.

You can load a model to the COMSOL Desktop from the Model Library in three different ways:

- Double-click the model node in the tree.
- Select the model node, then click the **Open Model** button.
- Right-click the model node, then select **Open Model** from the context menu.

To load the model and open the PDF documentation file in a single operation, click the **Open Model and PDF** button.

See Also

- Model Administration
- Opening and Searching the Model Library
- Changing the Model Library Root Folder
- Updating the Model Library Using Model Library Update
- Adding a User Model Library
- For examples of model plots, see Results Evaluation and Visualization

The Material Browser Window

In the **Material Browser** window ( ) you can browse the material libraries and load materials into your models. The window can be opened from the main menu by selecting **View>Material Browser**.

See Also

Materials

The External Process Window

Use the **External Process** window ( ) to follow external processes (such as distributed batch jobs) that have been started. The window updates when you are attached. You can do operations that are performed in the **External Process** nodes under **Job Configurations** by selecting an External Process from the list and select the operation. When detached, you can reattach by pressing the **Attach Job** button. The window can be opened from the main menu by selecting **View>External Process**. It opens automatically when you start a batch process.

See Also

In the *COMSOL Multiphysics Reference Guide*:

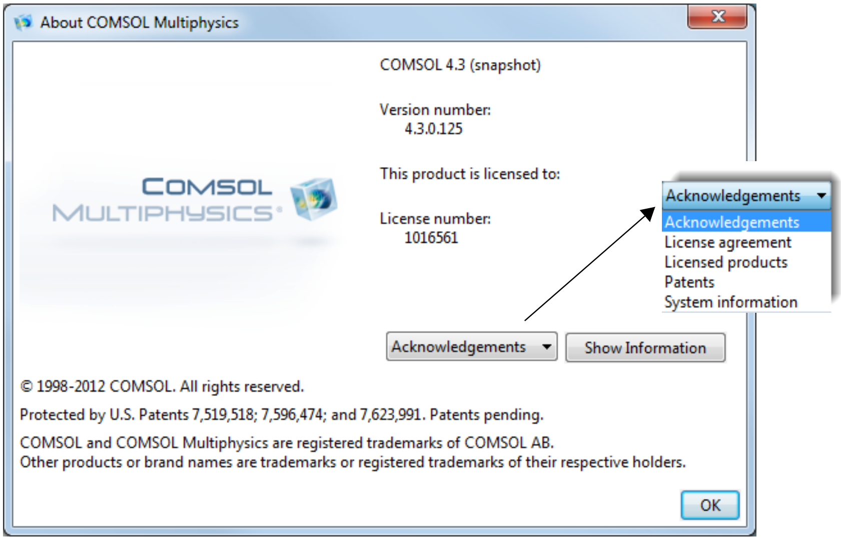
- Batch (study type)
- Cluster Sweep (study type)
- Batch Sweep (study type)
- Cluster Computing (study type)
- Batch (job configurations)
- Cluster Computing (job configurations)
- External Process

Errors and Warnings

COMSOL Multiphysics reports problems of two types:

- Errors, which prevent the program from completing a task. For errors, a **COMSOL Error** window appears with a brief error description and, in some cases, an **Open Log File** button for additional information. Under the node where the error occurred there is, in most cases, also an **Error** subnode (✖) that contains an error message that generally provide additional information. Also, for many error types, the icon for the node where the error occurred appears with a red cross in the lower-right corner.
- Warnings, which are problems that do not prevent the completion of a task but that might affect the accuracy or other aspects of the model. Warnings typically appear in the **Log** window (▤). The warning message also appears as a **Warning** subnode (⚠) under the node from which the warning was sent.

The About COMSOL Multiphysics Box



COMSOL 4.3 (snapshot)

Version number:
4.3.0.125

This product is licensed to:

License number:
1016561

© 1998-2012 COMSOL. All rights reserved.

Protected by U.S. Patents 7,519,518; 7,596,474; and 7,623,991. Patents pending.

COMSOL and COMSOL Multiphysics are registered trademarks of COMSOL AB.
Other products or brand names are trademarks or registered trademarks of their respective holders.

Figure 2-7: The About COMSOL Multiphysics box with an expanded Acknowledgements list.

From the **Help** menu, choose **About COMSOL Multiphysics** to open the **About COMSOL Multiphysics** box, where you can find the following information about the program and the license—the version number, the user or company the product is licensed to, and the license number.

From the list below, you can show one of the following types of information:

- Select **Acknowledgements** to show information about third-party software components, including license notices required by the software component authors.
- Select **License agreement** to show the COMSOL software license agreement.
- Select **Licensed products** to show the licensed COMSOL products, including the number of used licenses and the total number of licenses for each product.
- Select **Patents** to show the patents that the COMSOL program is protected by.
- Select **System information** to show a list of system properties, which can be useful for troubleshooting purposes, for example.

Click the **Show Information** button to display the information.

Keyboard Shortcuts

The following table summarizes the available keyboard shortcuts on Windows and Linux and on the Mac:

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MAC)	ACTION
F1	F1	Display dynamic help for the selected node
F2	F2	Rename the selected node, file, or folder
F3	F3	Disable selected nodes
F4	F4	Enable selected nodes
F5	F5	Update solution
F7	F7	Build the selected node in the geometry and mesh branches, or compute to the selected node in the solver sequence.
F8	F8	Build the geometry, build the mesh, compute entire solver sequence, update results data, or update the plot
Del	Del	Delete selected nodes
Left arrow (Windows); Shift + left arrow (Linux)	Left arrow	Collapse a branch in the Model Builder
Right arrow (Windows); Shift + right arrow (Linux)	Right arrow	Expand a branch in the Model Builder
Up arrow	Up arrow	Move to the node above in the Model Builder
Down arrow	Down arrow	Move to the node below in the Model Builder
Alt+left arrow	Ctrl+left arrow	Move to the previously selected node in the Model Builder
Alt+right arrow	Ctrl+right arrow	Move to the next selected node in the Model Builder
Ctrl+A	Command+A	Select all domains, boundaries, edges, or points; select all cells in a table
Ctrl+D	Command+D	Deselect all domains, boundaries, edges, or points
Ctrl+C	Command+C	Copy text in fields
Ctrl+V	Command+V	Paste copied text

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MAC)	ACTION
Ctrl+Z	Command+Z	Undo the last operation
Ctrl+Y	Ctrl+Shift+Z	Redo the last undone operation
Ctrl+up arrow	Command+up arrow	Move a definitions node, geometry node, physics node (except default nodes), material node, mesh node, study step node, or results node up one step
Ctrl+down arrow	Command+down arrow	Move a definitions node, geometry node, physics node (except default nodes), material node, mesh node, study step node, or results node down one step
Ctrl+Tab	Ctrl+Tab	Switch focus to the next window on the desktop
Ctrl+Shift+Tab	Ctrl+Shift+Tab	Switch focus to the previous window on the desktop
Ctrl+Alt+left arrow	Command+Alt+left arrow	Switch focus to the Model Builder window
Ctrl+Alt+right arrow	Command+Alt+right arrow	Switch focus to the settings window
Ctrl+Alt+up arrow	Command+Alt+up arrow	Switch focus to the previous section in the settings window
Ctrl+Alt+down arrow	Command+Alt+down arrow	Switch focus to the next section in the settings window
Shift+F10 or (Windows only) Menu key	Ctrl+F10	Open context menu
Ctrl+Space	Ctrl+Space	Open list of predefined quantities for insertion in Expression fields for plotting and results evaluation.
Ctrl+N	Command+N	Create a new model
Shift+Ctrl+N	Shift+Command+N	Create a new Physics Builder (the option must be enabled in Preferences)
Ctrl+O	Command+O	Open a model file
Ctrl+S	Command+S	Save a model file
Ctrl+P	Command+P	Print a model file

Key to the Nodes and Toolbar Buttons

MODEL BUILDER NODES

The **Model Builder** has many types of nodes to help you visualize the model structure. These nodes are described throughout this guide. Table 2-6 is a key to the main nodes.

TABLE 2-6: THE MAIN MODEL BUILDER NODES

FEATURE NAME	NODE	LINKS TO MORE INFORMATION
Data Sets	▦	Defining Data Sets
Definitions	≡	About Global and Local Definitions
Derived Values	8.85 e-12	Defining Derived Values and Tables
Equation View	$\frac{\partial u}{\partial t}=f$	• The Equation View Node • Physics Nodes—Equation Section
Export	▤	Exporting Data and Images
Geometry	✐	Geometry Modeling and CAD Tools
Global Definitions	≡	Global and Local Definitions
Materials	❖	Materials
Mesh	▩	Meshing
Model Builder	⊤	• The Model Builder Window • The Model Wizard and Model Builder • The Model Nodes in the Model Builder

TABLE 2-6: THE MAIN MODEL BUILDER NODES

FEATURE NAME	NODE	LINKS TO MORE INFORMATION
Physics		• Overview of the Physics Interfaces • Physics Nodes by Space Dimension • Physics Node Context Menu Layout • Multiphysics Modeling • Equation-Based Modeling
Plots		• About Plot Groups and Plotting • Color Coding for Plot Groups and Plot Types
Reports		Creating Reports
Results		Results Evaluation and Visualization
Study		Solvers and Study Types
Tables	▦	Defining Derived Values and Tables

DYNAMIC NODES

Table 2-7 lists generic examples and links to the dynamic visual aids that are used to help you identify nodes that change status in the Model Builder. Some nodes also change in the Solver settings windows.

TABLE 2-7: DYNAMIC NODES - VISUAL AIDS TO IDENTIFICATION

TYPE	ICON	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
Error	✕	For example, on a Material node . See Errors Relating to the Material Nodes.
Error node	⊗	Errors and Warnings
Current node, not built		For example, on a Geometry node. This node is also displaying the asterisk indicating the node is being Edited. The asterisk also appears on plot nodes when the plot has not been updated to reflect changes in the data or settings (for example, after re-solving),

TABLE 2-7: DYNAMIC NODES - VISUAL AIDS TO IDENTIFICATION

TYPE	ICON	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
Current node	□	A current node is used for Geometry and Meshing nodes and indicates that the feature or sequence of steps has been built. It is a green line on the left and upper edges of the node. For example, on a Geometry node , after building.
Harmonic Perturbation	~	For example, on a boundary level node for the Electric Currents interface, Electric Ground node . See Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.
Warning	⚠	For example, on a Mesh node . This node is also displaying the green outline indicating the node is Current.
Editing, or in process of editing, a node	*	For example, on a Mesh node . This node is also displaying the asterisk indicating the node is being Edited. Also indicates physics interface nodes that have been disabled in a Study Step.
Pairs	⋈	For example, on a 3D Boundary Level node . See Specifying Boundary Conditions for Identity Pairs and Identity and Contact Pairs.
Contributing node	●	For example, on a 3D boundary level node . • Physics Exclusive and Contributing Feature Node Types • Physics Node Status
Default node	D	For example, on a 3D boundary level node. See Physics Default Feature Nodes.
Override	▲	For example, on a 3D boundary level node . See Physics Exclusive and Contributing Feature Node Types. Physics Node Status
Overridden	▼	For example, on a 3D boundary level node . See Physics Exclusive and Contributing Feature Node Types
STUDY STEPS ANALYSIS		

TABLE 2-7: DYNAMIC NODES - VISUAL AIDS TO IDENTIFICATION

TYPE	ICON	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
Solve For	🟢	For example, a Laminar Flow interface () where the green dot in the lower-right corner indicates that the study solves for the degrees of freedom in this physics interface. See Selecting Physics and Variables in the Study Steps in the COMSOL Multiphysics Reference Guide.
Provide Degrees of Freedom	🟡	For example, a Laminar Flow interface () is enabled (not greyed out), shows that the study step provides degrees of freedom (the yellow dot in the lower-right corner), and has a change of state indicated by the asterisk. The yellow dot means that the study step provides degrees of freedom but does not solve for the physics interface. See Selecting Physics and Variables in the Study Steps in the COMSOL Multiphysics Reference Guide.
Change of State (editing)	*	An asterisk appears in the upper-right corner of nodes for which you change their state in the study step's selection tree compared to their state in the main model tree in the Model Builder.
Disabled, provides no degrees of freedom, and shows a change of state	🔴	In this example, a Transport in Diluted Species interface () is disabled (greyed out), provides no degrees of freedom (red dot in the lower-right corner), and has a change of state indicated by the asterisk. See Selecting Physics and Variables in the Study Steps in the COMSOL Multiphysics Reference Guide.

TOOLBARS AND BUTTONS

There are many toolbars and buttons available and these are described throughout this guide. Table 2-8 is a list of links to the information

TABLE 2-8: LINKS TO THE COMSOL MULTIPHYSICS TOOLBARS AND BUTTONS

CATEGORY	GO TO THIS SECTION
Model Builder	Buttons on the Model Builder Window
Main menu and toolbar	The Main Menu and Toolbar

TABLE 2-8: LINKS TO THE COMSOL MULTIPHYSICS TOOLBARS AND BUTTONS

CATEGORY	GO TO THIS SECTION
Help Documentation	The Help Documentation ToolBar Buttons and Tabs
Model Library	Model Library Toolbar Buttons
Dynamic visual feedback	Dynamic Nodes - Visual Aids to Identification, for generic examples
Geometry and Graphics windows	• The Geometry Toolbar Buttons • Geometric Entity Selection List Buttons • The Graphics Toolbar Buttons • Different ways to select geometric entities
Show and Expand Sections buttons	Showing and Expanding Advanced Physics Sections
Plotting and Results	• Capturing and Copying Screenshots • Plotting and Cross-Section Interactive Toolbar • Results Table Editing Buttons • Export Feature Types • Model Report Components
Meshes	The Meshing Toolbar and Features
Physics interface nodes	• Dynamic Nodes - Visual Aids to Identification • Physics Nodes by Space Dimension • Color Coding for Plot Groups and Plot Types
Replace Expression button	Expressions and Predefined Quantities

3

Global and Local Definitions

This chapter describes the global and local (model) definitions features. Depending on the geometric scope, you add the feature nodes described in this section to either the Global Definitions node or under the Definitions node for a particular model. In this chapter:

- About Global and Local Definitions
- Operators, Functions, and Variables Reference
- Global Parameters
- Variables
- Global and Local Functions
- Model Couplings
- Coordinate Systems
- Identity and Contact Pairs
- Probes

See also Visualization and Selection Tools and Using Units.

About Global and Local Definitions

Depending on the geometric scope, there are two types of definitions that can be defined in the Model Builder—**Global Definitions** and local **Definitions**.

In this section:

- Global Definitions
- Local Definitions
- Load Group
- Constraint Group
- About Building Expressions

Global Definitions

Right-click **Global Definitions** ( ) to add the following:

- **Parameters** ( ): global, scalar values that are used to parameterize any part of the model.
- **Variables** ( ): expression variables that are used anywhere to simplify the specifications of some properties.
- **Functions** are predefined function templates for common function types such as step functions, ramps, and random functions.
- **Groups** ( ): Add **Load Group** ( ) and **Constraint Group** ( ) nodes for use in load cases.

See Also

- Global Parameters
- Variables
- Global and Local Functions
- Load Group
- Constraint Group
- About Building Expressions
- Sorting Nodes by Space Dimension and Type

Local Definitions

After adding any Model node to the Model Builder, right-click **Definitions** ( ) to add definitions with a local scope that are applied to that specific model. Add:

- **Variables** (a=): expression variables used to simplify the specifications of some properties.
- **Selection** ( ): create a user-defined set of geometric entities for reuse throughout the model.
- **View** ( ): create a user-defined view to visualize the model.
- **Contact Pair** ( ): specify two selections of boundaries that cannot penetrate each other under deformation. Requires the use of pairs and is only available if a license includes the Structural Mechanics Module or the MEMS Module.
- **Identity Pair** ( ): specify two boundary selections that overlap but belong to different parts of a pair. Then assign a boundary condition to connect the physics in the two parts in an interface. Identity pairs are only available and necessary when the model geometry is an assembly.
- **Functions** are predefined function templates for common function types such as step functions, ramps, and random functions.
- **Model Couplings**. The coupling operators are useful for creating integrated quantities and identity mappings, adding nonlocal couplings, and many other applications.
- **Coordinate Systems**. Create different coordinate systems for use in some of the physics interfaces.
- **Probes**. To monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency, parametric).
- **Perfectly Matched Layer**. To apply a coordinate scaling to a layer of virtual domains surrounding the physical region of interest. For stationary and transient problems, these virtual domains can be stretched out towards infinity, giving rise to infinite elements.
- **Infinite Element Domain.** To apply a coordinate scaling to a layer of virtual domains surrounding the physical region of interest. For stationary and transient problems, these virtual domains can be stretched out towards infinity, giving rise to infinite elements. To absorb outgoing waves in a frequency-domain problem, you must instead stretch the virtual domains into the complex plane, creating perfectly matched layers (PMLs).

See Also

- Variables
- User-Defined Selections
- User-Defined Views
- Identity and Contact Pairs
- Global and Local Functions
- Model Couplings
- Coordinate Systems
- Probes
- Load Group
- Constraint Group
- About Building Expressions
- Sorting Nodes by Space Dimension and Type

Load Group

Add a **Load Group** ( ) to the **Global Definitions** branch (under **Loads and Constraints Groups** node ( ) if you use sorting by type) to create a load group to which you can assign one or more loads. You can then activate the load group in one or more *load cases* for efficiently solving a structural mechanics model to analyze the effects of various loads (and, optionally, constraints). In the **Identifier** field you can modify the default group identifier (`lg1`, for example) if you, for example, want to use a more descriptive name. You can also create load groups from physics interface features for structural mechanics that represent loads and support load cases: Right-click the physics interface feature node and choose **Load Group>New Load Group**.

See Also

Using Load Cases

Model

- Tapered Cantilever with Two Load Cases: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/tapered_cantilever**
- If you have the Structural Mechanics Module, also see Pratt Truss Bridge: Model Library path **Structural_Mechanics_Module/ Civil_Engineering/pratt_truss_bridge**

Constraint Group

Add a **Constraint Group** ( ) to the **Global Definitions** branch (under **Loads and Constraints Groups** node ( ) if you use sorting by type) to create a constraint group to which you can assign one or more constraints. You can then activate the constraint group in one or more *load cases* for efficiently solving a model to analyze the effects of various constraints (and loads). In the **Identifier** field you can modify the default group identifier (`lg1`, for example) if you, for example, want to use a more descriptive name. You can also create constraint groups from physics interface features for structural mechanics that represent constraints and support load cases: Right-click the physics interface feature node and choose **Constraint Group>New Constraint Group**.

See Also

Using Load Cases

About Building Expressions

When defining any expression, whether it is global or local, there are many available variables, mathematical functions, physical constants, and operators that can be used to define the expression.

Many common mathematical and logical operators and functions can also be used along with constants to specify parameters and variables as well as equation coefficients and material properties. The following components frequently appear in expressions:

- Variables
- Physical constants, which are predefined constants for universal constants such as the universal gas constant and the speed of light in vacuum.
- Operators
- Mathematical and logical functions

If the expression contains a syntax error, it appears in red. Syntax errors can be due to illegal characters, mismatched parentheses, and other syntactic errors. The **Error** node (☒), which occurs when trying to solve a model with a syntax error, typically contains information about the position and expression where the syntax error is located.

Operators, Functions, and Variables Reference

Many common mathematical and logical operators, functions, and constants can be used to specify parameters, variables, equation coefficients, and material properties. These tables list the unary and binary operators (Table 3-1 and Table 3-2), special operators (Table 3-3), mathematical functions and constants (Table 3-4), and physical constants predefined as variables (Table 3-5) that can be used in COMSOL. See also Model Couplings for information about coupling operators.

In this section:

- Unary and Binary Operators
- Special Operators
- Mathematical Functions
- Physical Constants

Unary and Binary Operators

TABLE 3-1: UNARY OPERATORS

OPERATOR	DESCRIPTION
+	Unary plus
-	Unary minus
!	Logical not

TABLE 3-2: BINARY OPERATORS

OPERATOR	DESCRIPTION
+	Plus
-	Minus
*	Multiply
/	Divide
^	Power
==	Equal
!=	Not equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
&&	Logical and
||	Logical or

Special Operators

There are special operators available for modeling and for evaluating results. Many physics interfaces use these operators to implement equations and special functionality. See Table 3-3 and the detailed descriptions that follow.

TABLE 3-3: SPECIAL OPERATORS

OPERATOR	DESCRIPTION
`d(f,x)`	Differentiation operator. Differentiation of `f` with respect to `x`. See The Differentiation Operators for more information.
`pd(f,x)`	Differentiation operator. Differentiation of `f` with respect to `x`. No chain rule for dependent variables. See The Differentiation Operators for more information.

TABLE 3-3: SPECIAL OPERATORS

OPERATOR	DESCRIPTION
`dtang(f,x)`	Tangential differentiation of an expression `f` defined on a boundary with respect to a spatial dimension `x`. See The Differentiation Operators for more information.
`test(`*expr*`)`	Test function operator. See The Test Operator for more information.
`var(`*expr*`,`*fieldname1*`,` *fieldname2*`, ...)`	Variation operator. See The Var Operator for more information.
`nojac(`*expr*`)`	No contribution to the Jacobian. See The Nojac Operator for more information.
`up(`*expr*`)`	Evaluate expression as defined in adjacent upside. See The Up, Down, and Mean Operators for more information.
`down(`*expr*`)`	Evaluate expression as defined in adjacent downside. See The Up, Down, and Mean Operators for more information.
`mean(`*expr*`)`	Mean value of expression as evaluated on adjacent boundaries. See The Up, Down, and Mean Operators for more information.
`depends(`*expr*`)`	True if expression depends on the solution. See The Depends Operator for more information.
`isdefined(`*variable*`)`	Returns one where the variable is defined and zero where it is not defined.
`dest(`*expr*`)`	Evaluate parts of an integration coupling expression on destination side. See The Dest Operator for more information.
`if(`*cond*`,`*expr1*`,`*expr2*`)`	Conditional expression evaluating the second or third argument depending on the value of the condition. See The If Operator for more information.
`isinf(`*expr*`)`	True if expression evaluates to infinity. See The Isinf and Isnan Operators for more information.
`islinear(`*expr*`)`	True if expression is a linear function of the solution. See The Islinear Operator for more information.
`isnan(`*expr*`)`	True if expression evaluates to NaN (not-a-number). See The Isinf and Isnan Operators for more information.
`with`	Access any solution. See The With Operator for more information.

TABLE 3-3: SPECIAL OPERATORS

OPERATOR	DESCRIPTION
`at`	Access the solution at any time. See The At Operator for more information.
`timeint`	Integrate time-dependent expression over a time interval. See The Timeint Operator for more information.
`timeavg`	Compute average of time-dependent expression over a time interval. See The Timeavg Operator for more information.
`linpoint`	Access the linearization point (when a linearization point is stored in the solution). See The Linpoint Operator for more information.
`lindev`	Evaluate an expression linearized at the linearization point (when a linearization point is stored in the solution). See The Lindev Operator for more information.
`lintotal`	Access the sum of the linearization point and linear perturbation. See The Lintotal Operator for more information.
`lintotalavg`	Evaluate average of `lintotal(`*expr*`)` over all phases. See The Lintotalavg Operator for more information.
`lintotalrms`	Evaluate RMS of `lintotal(`*expr*`)` over all phases. See The Lintotalrms Operator for more information.
`lintotalpeak`	Evaluate maximum of `lintotal(`*expr*`)` over all phases. See The Lintotalpeak Operator for more information.
`linsol`	Access the standard solution (for example inside `linpoint` or `lintotal`). See The Linsol Operator for more information.
`linzero`	Evaluate expression with zero solution. See The Linzero Operator for more information.
`linper`	Marks a load term to be used in a Linear perturbation solver. See The Linper Operator for more information.
`ppr`	Accurate derivative recovery. See The Ppr and Pprint Operators for more information.
`pprint`	Accurate derivative recovery within each domain group. See The Ppr and Pprint Operators for more information.
`reacf`	Accurate integrals of reaction forces and fluxes. See The Reacf Operator for more information.

TABLE 3-3: SPECIAL OPERATORS

OPERATOR	DESCRIPTION
`adj(`*`expr`*`)`	Evaluate expression using the adjoint sensitivity. See The Adj Operator for more information.
`fsens(`*`expr`*`)`	Evaluate expression using the functional sensitivity. See The Fsens Operator for more information.
`sens(`*`expr`*`,`*`i`*`)`	Evaluate expression using the forward sensitivity for the parameter given by the second argument. See The Sens Operator for more information.
`realdot(a,b)`	Treat complex numbers *a* and *b* as real 2-vectors and return their dot product. See The Realdot Operator for more information.
`shapeorder(`*`variable`*`)`	The element order used for discretization of a variable. See The Shapeorder Operator for more information.
`prev(`*`expr`*`,`*`i`*`)`	Evaluate expression at the *`i`*th previous time step. See The Prev Operator for more information.
`bdf(`*`expr`*`,`*`i`*`)`	Apply backward differentiation formula of order *`i`* on expression. See The Bdf Operator for more information.

`subst(`*`expr`*`,`
`expr1_orig,`` ,`
`expr1_subst``,...)`
	Substitute variables in an expression with other variables or expressions. See The Subst Operator for more information

`circint(`*`r`*`,`*`expr`*`),`
`circavg(`*`r`*`,`*`expr`*`),`
`diskint(`*`r`*`,`*`expr`*`),`
`diskavg(`*`r`*`,`*`expr`*`),`
`sphint(`*`r`*`,`*`expr`*`),`
`sphavg(`*`r`*`,`*`expr`*`),`
`ballint(`*`r`*`,`*`expr`*`),`
`ballavg(`*`r`*`,`*`expr`*`)`
	Evaluates the integral or average of the expression on the specified shape with radius *r*. See The Circle, Disk, Sphere, and Ball Integration and Average Operators for more information.

`scope.ati(`*`coordinate exprs`*`,`*`expr`*`)`	Evaluates the expression *`expr`* at an *`i`*-dimensional entity in the point with coordinates given by the coordinate expressions *`coordinate exprs`*. See The Spatial At Operators for more information.

THE DIFFERENTIATION OPERATORS

- Use the `d` operator to differentiate a variable with respect to another variable. For example, `d(T,x)` means differentiation of T with respect to x.

- Use the d operator in the model settings. If it contains any of the dependent variables, it typically makes the model nonlinear.
- Use the d operator during results evaluation to visualize and compute derivatives. Some space derivatives are also available using predefined variables. For example, uxx, d(ux,x), and d(d(u,x),x) are equivalent.
- The pd operator works in a similar way to the d operator but does not apply the chain rule for dependent variables.
- The expression d(E,TIME) computes the reference time derivative of the expression E.
- Use the dtang operator to compute derivatives in the tangential direction along a boundary of expressions that are only defined on the boundary and therefore cannot be differentiated by the d operator. In a 3D model, (dtang(f,x), dtang(f,y), dtang(f,z)) is a vector in the tangent plane of a boundary at the point where it is evaluated, and similarly in a 2D model, (dtang(f,x), dtang(f,y)) is a vector in the tangent line of a boundary point. When evaluated in a domain, dtang(f,x) is the same as d(f,x). The second argument of dtang(f,x) must be one of the spatial coordinates. Not all quantities have rules for evaluating tangential derivatives. Applying dtang(f,x) to an expression with no tangential derivative rule results in an error message.

Examples of Using the Differentiation Operators

The expressions d(u^2,u) and pd(u^2,u) both equal 2*u—d also takes the spatial and time variables into account and treats their derivatives accordingly. In other words, if u is the dependent variable and x and t are the spatial coordinate and time, respectively, then d(u+x,x) equals ux+1 (ux is the spatial derivative of u with respect to x), while pd(u+x,x) equals 1 because u is considered to be independent of anything but itself in the case of pd. Equivalently, d(u,t) equals ut, while pd(u,t) is zero.

If u is a dependent variable defined only on a boundary, d(u,x) is not defined, but the tangential derivative dtang(u,x) can be evaluated on the boundary. The tangential derivative obeys most of the common differentiation rules, such as the product rule and the chain rule. It is worth pointing out, however, that dtang(x,x) is not always equal to 1.

THE TEST OPERATOR

- The test operator is available for modeling using the weak formulation of the equations.
- This operator creates the test function for the variable that it operates on. For an expression, test($F(u, \nabla u)$), the test operator is equivalent to:

$$\sum_i \mathrm{test}(u_i)\frac{\partial}{\partial u_i}F(u_i, \nabla u_i) + \mathrm{test}(\nabla u_i)\frac{\partial}{\partial \nabla u_i}F(u_i, \nabla u_i)$$

  for all dependent variables u_i.

THE VAR OPERATOR

- The var operator (variation operator) is available for modeling using the weak formulation of the equations.
- The var operator has the same function as the test operator but is limited to the specified set of fields.
- This operator creates the test function for the variable that it operates on. For an expression, var($F(u, \nabla u, v, \nabla v), a$), where the dependent variable u is in the field named a and the dependent variable v is not, the var operator is equivalent to:

$$\sum_i \mathrm{test}(u_i)\frac{\partial}{\partial u_i}F(u_i, \nabla u_i, v_i, \nabla v_i) + \mathrm{test}(\nabla u_i)\frac{\partial}{\partial \nabla u_i}F(u_i, \nabla u_i, v_i, \nabla v_i)$$

for all dependent variables u_i.

THE NOJAC OPERATOR

- The nojac operator makes sure that any expression that it operates on is excluded from the Jacobian computation. This is useful if a Jacobian contribution is not strictly necessary and the computational requirements for it are high.
- The k-ε turbulence model is an example where a built-in use of the nojac operator improves performance.

THE UP, DOWN, AND MEAN OPERATORS

- COMSOL Multiphysics can evaluate expressions on both sides of a domain. These operators are available only on boundaries.

- For an expression or a variable that is discontinuous across a boundary, the value is different on either side, and COMSOL Multiphysics normally displays the mean values on the boundary.
- Use the `up`, `down`, and `mean` operators to evaluate an expression on the upside or downside of the boundary or the mean value on the boundary. If the upside or downside is outside of the geometry, or if the variables in the expression are not active on that side, the `up` or `down` operator returns 0.

For more information about the upside and downside of a boundary, see Tangent and Normal Variables.

THE DEPENDS OPERATOR

- The `depends` operator returns 1 if the expression that it operates on depends on the solution; otherwise it returns 0.
- Use this operator to check user-defined expressions for dependency on the solution.

THE DEST OPERATOR

The `dest` (destination) operator is available for use in integration coupling expressions. This operator forces the expression that it operates on to be evaluated on the destination points instead of the source points. This means that the destination operator can be used to create convolution integrals and other integral transforms. For instance, integrating the expression `u/((dest(x)-x)^2+(dest(y)-y)^2)` gives the following function of x and y:

$$f(x,y) = \int \frac{u(x',y')}{(x-x')^2+(y-y')^2} dx'dy'$$

THE IF OPERATOR

- The `if(`*`cond`*`,`*`expr1`*`,`*`expr2`*`)` operator implements a conditional expression.
- The first argument is a condition that COMSOL Multiphysics treats as a Boolean expression. If—at a particular evaluation point—`cond` is true, then the second argument is evaluated, otherwise the third argument is evaluated. That is, only one branch is evaluated at any evaluation point.
- Use the `if` operator to avoid illegal operations. For example, `if(x==0,1,sin(x)/x)` is a valid and continuous expression for all values of x, including $x = 0$.

THE ISLINEAR OPERATOR

- The `islinear` operator returns 1 if the expression that it operates on is a linear function of the solution; otherwise it returns 0.
- Use this operator to check user-defined expressions for linearity with respect to the solution. The stationary solver does this automatically to choose between a linear or a nonlinear solver.

THE ISINF AND ISNAN OPERATORS

- The `isinf` operator returns 1 if the expression that it operates on evaluates to infinity (or minus infinity); otherwise it returns 0.
- The `isnan` operator returns 1 if the expression that it operates on evaluates to NaN (not-a-number); otherwise it returns 0.

THE WITH OPERATOR

- The `with` operator can access any solution during results evaluation.
- For time-dependent problems, parametric problems, and eigenvalue problems, this makes it possible to use the solution at any of the time steps, any parameter value, or any eigensolution in an expression used for plotting or data evaluation.
- Use the solution number as the first input argument. The second input argument is the expression that you want to evaluate using this solution. For example, `with(3,u^2)` provides the square of the third eigensolution for an eigenvalue problem.
- For example, use the `with` operator to verify that two eigensolutions are orthogonal.
- The `with` operator can only be used during results evaluation, so it should not be used when setting up the model.

THE AT OPERATOR

- The `at` operator can access a solution to a time dependent problem at any time. COMSOL Multiphysics provides the solution at that time using interpolation.
- The first input argument is the time. The second input argument is the expression that you want to evaluate using this solution. For example, `at(12.5,u)` is the solution at 12.5 s.
- The `at` operator can only be used during results evaluation, so it should not be used when setting up the model.

THE TIMEINT OPERATOR

- The timeint operator integrates a time dependent expression over a time interval. timeint(t1,t2,expr) computes the integral of expr over the interval t=t1 to t=t2. The first two arguments must be real scalars. The integral is computed by numerical integration, subdividing the interval until the required accuracy is reached.
- timeint(t1,t2,expr,tol) sets the relative tolerance in the numerical integration procedure to tol. The tolerance must be a positive real scalar. The default tolerance (used when the fourth argument is omitted) is 1e-8.
- timeint(t1,t2,expr,tol,minlen) sets the smallest length of the subintervals used in numerical integration as a fraction of the length of the whole integration interval. Subintervals smaller than this length are not further subdivided even if that means that the required accuracy is not reached. minlen must be a positive real scalar. The default value of minlen (used when the fifth argument is omitted) is 1e-6.
- The timeint operator can only be used during results evaluation, so it should not be used when setting up the model.

THE TIMEAVG OPERATOR

- The timeavg operator computes the average of a time-dependent expression over a time interval. timeavg(t1,t2,expr) computes the average of expr over the interval t=t1 to t=t2. The first two arguments must be real scalars. The timeavg operator numerically integrates the expression in the same way as timeint and then divides the result by t2-t1.
- timeavg(t1,t2,expr,tol) sets the relative tolerance in the numerical integration procedure to tol. The tolerance must be a positive real scalar. The default tolerance (used when the fourth argument is omitted) is 1e-8.
- timeavg(t1,t2,expr,tol,minlen) sets the smallest length of the subintervals used in numerical integration as a fraction of the length of the whole integration interval. Subintervals smaller than this length are not further subdivided even if that means that the required accuracy is not reached. minlen must be a positive real scalar. The default value of minlen (used when the fifth argument is omitted) is 1e-6.
- The timeavg operator can only be used during results evaluation, so it should not be used when setting up the model.

THE LINPOINT OPERATOR

The linpoint operator can access the linearization point in a solution with a stored linearization point. If the solution does not have a stored linearization point, using linpoint causes an error message.

THE LINDEV OPERATOR

The lindev operator evaluates its argument in the following way when the solution has a stored linearization point: The expression is first linearized at the linearization point, and then evaluated at the current solution. In particular, if f depends linearly on the solution, lindev(f) is the same as f. If f does not depend on the solution, lindev(f) is 0. If the solution does not have a stored linearization point, using lindev causes an error message.

THE LINTOTAL OPERATOR

The lintotal operator can access the sum of the linearization point and the linear perturbation in a solution with a stored linearization point. If the solution does not have a stored linearization point, using lintotal causes an error message.

THE LINTOTALAVG OPERATOR

The lintotalavg operator evaluates the average of an expression over all phases. lintotalavg(f) is evaluated by taking the average of lintotal(f) with the solution (but not the linearization point) multiplied by $e^{i\varphi}$ for a number of phases φ. The number of phases is automatically selected to achieve an accurate value. lintotalavg(f,n) uses n equidistantly spaced phases.

THE LINTOTALRMS OPERATOR

The lintotalrms operator evaluates the RMS of an expression over all phases. lintotalrms(f) is the same as sqrt(lintotalavg(abs(f)^2)).

THE LINTOTALPEAK OPERATOR

The lintotalpeak operator evaluates the maximum of an expression over all phases. lintotalpeak(f) is evaluated by taking the maximum of real(lintotal(f)) with the solution (but not the linearization point) multiplied by $e^{i\varphi}$ for a number of phases φ. The number of phases is automatically selected to achieve an accurate value. lintotalpeak(f,n) uses n equidistantly spaced phases.

THE LINSOL OPERATOR

The linsol operator evaluates an expression using the standard solution. This is the default, so in most contexts the operator is not very useful. However, it can for example be used inside the argument of `linpoint` to evaluate a part of the expression with the standard solution instead of the linearization point.

THE LINZERO OPERATOR

The `linzero` operator evaluates an expression using a zero solution. This is mostly used internally in the definitions of some other operators.

THE LINPER OPERATOR

The `linper` operator has one single use: To indicate load terms that should be used by stationary solvers setting **Linearity** to **Linear perturbation** in the **Stationary Solver** node's settings. Terms not enclosed by `linper` are ignored by such solvers. On the other hand, terms inside `linper` are ignored by all other solvers. Evaluating the `linper` operator always gives the result 0.

THE PPR AND PPRINT OPERATORS

- When the `ppr` operator is applied on an expression, COMSOL uses polynomial-preserving recovery to evaluate all variables in the expression that are discretized using Lagrange shape functions; see Accurate Derivative Recovery.

  For example, if `e = ux + vy`, then `ppr(e^2)=(ppr(ux)+ppr(vy))^2`.

- The `pprint` operator similarly applies polynomial-preserving recovery within each group of domains with equal settings. Use these operators to get an estimate of the discretization error in the gradient.

  For example `ux-pprint(ux)` in a 1D model.

Note

If these operators are applied on expressions that are used when solving the model, COMSOL computes the Jacobian approximately by ignoring the operator. For example, the Jacobian of `ux-pprint(ux)` is 0.

THE REACF OPERATOR

- The reaction force operator (`reacf`) is available when calculating integrals of reaction forces or fluxes.
- Apply the `reacf` operator on the names of dependent variables when doing a surface integration. For example, in structural mechanics, with dependent variables u and v corresponding to x- and y-displacements, use `reacf(u)` and `reacf(v)` to access integrals of the reaction forces in the x- and y-direction, respectively.

- Disable access to the operator by clearing the **Reaction forces** check box in the **Output** section in the solver's settings window.
- The reaction force operator is not available when using weak constraints.

THE ADJ OPERATOR

- When the adjoint sensitivity operator (`adj`) is applied to an expression, COMSOL uses the adjoint sensitivity solution instead of the primal solution for the evaluation.
- The adjoint sensitivity solution is available for results when the sensitivity solver has been used with the adjoint sensitivity method, and for the dependent variables that have been solved for.

THE FSENS OPERATOR

- When the functional sensitivity operator (`fsens`) is applied to an expression, COMSOL uses the functional sensitivity solution for the evaluation.
- The functional sensitivity solution is available for analysis when the sensitivity solver has been used with either the adjoint or the forward sensitivity method, and for the sensitivity variables that have been solved for.

THE SENS OPERATOR

- When the forward sensitivity operator (`sens`) is applied to an expression, COMSOL uses the forward sensitivity solution instead of the primal solution for the evaluation.
- The forward sensitivity solution is available for analysis when the sensitivity solver has been used with the forward sensitivity method, and for the dependent variables that have been solved for.
- For scalar sensitivity variables, access the corresponding forward sensitivity solution by giving the sensitivity variable name as the second argument to this operator. For example, with the dependent variable u and the scalar sensitivity parameter q, access the forward sensitivity solution $\partial u/\partial q$ as `sens(u,q)`.
- For a sensitivity parameter that is not scalar, a more elaborate syntax specifying a unique degree of freedom must be used. This is done by giving an integer as the second argument, corresponding to the global degree of freedom number for the requested sensitivity parameter.

THE REALDOT OPERATOR

- The expression `realdot(a,b)` treats complex numbers a and b as if they were real-valued vectors of length 2 and returns their dot product. Also think of the operator call as a shorthand form of `real(a*conj(b))`. This expression, however, is not an analytical function of its complex arguments and therefore has no unique partial derivatives with respect to a and b.

- The difference between `realdot(a,b)` and `real(a*conj(b))` is that the partial derivatives of the former with respect to a and b are defined as conj(b) and conj(a), respectively, while for the latter expression, the partial derivatives are real(conj(a)) and real(a).

Note

The difference between the partial derivative definitions is important during sensitivity analysis of frequency-response problems (scalar or vector Helmholtz equations).

- Common objective function quantities like power and energy must be redefined in terms of `realdot(a,b)` rather than `real(a*conj(b))` for the sensitivity solver to compute correct derivatives. This applies also to the absolute value, `abs(a)`, via the definition $|a|^2 = \mathrm{realdot}(a, a)$.

THE SHAPEORDER OPERATOR

- The expression `shapeorder(`u`)` gives the element order used for discretization of the variable u.
- The argument u must be a dependent variable or a partial derivative of a dependent variable. In the latter case, note that the order returned is the order of the dependent variable itself and not the order of its derivative.
- It is an error to apply the `shapeorder` operator to, for example, an expression, a constant, or a spatial coordinate.

THE PREV OPERATOR

- When the time discrete solver is used, it stores the solution at a number of previous time steps.
- The expression `prev(`expr,i`)` evaluates expr using the solution obtained i time steps before the current time step.
- The operator can be used in equations as well as for results evaluation.
- When used in equations, the `prev` operator makes it possible to discretize time derivatives. For example, to discretize `ut` (the time derivative of u) with the formula known as the *backward Euler method*, use the expression `(u-prev(u,1))/ timestep`. Here, `timestep` is the size of the time step used to reach the current solution `u`. The `prev` operator is also applicable for `timestep`. For example,

`prev(timestep,1)` is the size of the time step used to reach the solution at the previous time step.

- When using the `prev` operator, sufficiently many previous time steps must be stored. Specify the number of previous time steps to store in the time discrete solver (time discrete levels) in the **Number of time discrete levels** field in the **General** section of the **Time Discrete Solver** node's settings window. Evaluating an expression at a previous time step that has not been stored results in an error.

THE BDF OPERATOR

- Use the `bdf` operator to approximate time derivatives when the time discrete solver is used.
- The expression `bdf(`*expr*, i`)` results in a discretization of the time derivative of *expr* using a backward differentiation formula.
- The second argument, i, determines the order of accuracy of the discretization. Currently, first order and second order is available, so allowed values are $i = 1$ and $i = 2$. A second-order formula requires access to two previous time steps. Because this is not possible at the initial step, the evaluation at the initial step always uses the first-order formula.
- The `bdf` operator can be implemented using the `prev` operator. For example, obtain the first-order backward differentiation formula, also known as the backward Euler method, through `bdf(u,1) = (u-prev(u,1))/timestep`.

THE SUBST OPERATOR

- The `subst` operator takes a variable or expression as its first argument, followed by one or more argument pairs, each consisting of a variable name and an expression. The first argument in each pair is an original variable that appears in the variable or expression that you specify as the first argument, and the second argument in each pair is the variable or expression that you want to substitute the original variable with. This can be useful, for example, for replacing the variable for temperature in a temperature-dependent expression for some quantity by a fixed initial temperature for use as an initial condition.
- As an example, the expression `subst(hmnf.nutildeinit,p,pin_stat)` (taken from the Sajben Diffuser model in the CFD Module's model library) substitutes the dependent variable for pressure, `p`, with a user-defined variable `pin_stat` for the inlet static pressure. The evaluation of the variable `hmnf.nutildeinit` (for the

undamped turbulent kinematic viscosity) then takes the value of pin_stat instead of p.

- The unit of the output from the subst operator is the same as the unit for its first input argument.

THE CIRCLE, DISK, SPHERE, AND BALL INTEGRATION AND AVERAGE OPERATORS

- The circint(*r*, expr) operator integrates the expression expr on a circle with radius *r* around the point in which it is evaluated. The circint operator can be evaluated on all entities in 2D and on edges in 3D, when used in 3D the integration is done on the circle in the normal plane to the edge.
- The circavg(*r*, expr) operator is defined as circint(*r*, expr)/circint(*r*, 1).
- The diskint(*r*, expr) is similar to the circint operator but calculates the integral on a disk instead.
- The diskavg(*r*, expr) operator is defined as diskint(*r*, expr)/diskint(*r*, 1).
- The sphint(*r*, expr) operator integrates the expression on a sphere with radius *r* around the point in which it is evaluated. The sphint operator can be evaluated on all entities in 3D.
- The sphavg(*r*, expr) operator is defined as sphint(*r*, expr)/sphint(*r*, 1).
- The ballint(*r*, expr) is similar to the sphint operator but calculates the integral on a ball instead.
- The ballavg(*r*, expr) operator is defined as ballint(*r*, expr)/ballint(*r*, 1).
- All of operators can be used with a third argument *N* that approximately specifies the number of integration points used—for example, circint(*r*, expr,100).
- To all operators you can add a suffix ("*_frameId*") that specifies the frame in which the integration is done—for example, circint_spatial(*r*, expr).

THE SPATIAL AT OPERATORS

- The scope.ati(<*coordinate expressions*>, expr) operator evaluates the expression expr in the geometry with the given model scope on an i-dimensional entity in the point given by the coordinate expressions. For example, root.mod1.at1(0,y,dom) evaluates dom in the two-dimensional geometry on an edge in the point $(0, y)$.
- To all operators you can add a suffix ("_frameId") that specifies the frame in which the coordinate expressions are used, for example, at1_spatial(*x*,*y*, expr).

Mathematical Functions

Note

The following functions are only available when solving and not when evaluating parameters and variables in the user interface: `acosh`, `acoth`, `acsch`, `asech`, `asinh`, `atanh`, `besselj`, `bessely`, `besseli`, `besselk`, and `erf`.

TABLE 3-4: MATHEMATICAL FUNCTIONS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
`abs`	Absolute value	`abs(x)`
`acos`	Inverse cosine (in radians)	`acos(x)`
`acosh`	Inverse hyperbolic cosine (in radians)	`acosh(x)`
`acot`	Inverse cotangent (in radians)	`acot(x)`
`acoth`	Inverse hyperbolic cotangent (in radians)	`acoth(x)`
`acsc`	Inverse cosecant (in radians)	`acsc(x)`
`acsch`	Inverse hyperbolic cosecant (in radians)	`acsch(x)`
`arg`	Phase angle (in radians)	`arg(x)`
`asec`	Inverse secant (in radians)	`asec(x)`
`asech`	Inverse hyperbolic secant (in radians)	`asech(x)`
`asin`	Inverse sine (in radians)	`asin(x)`
`asinh`	Inverse hyperbolic sine (in radians)	`asinh(x)`
`atan`	Inverse tangent (in radians)	`atan(x)`
`atan2`	Four-quadrant inverse tangent (in radians)	`atan2(y,x)`
`atanh`	Inverse hyperbolic tangent (in radians)	`atanh(x)`
`besselj`	Bessel function of the first kind	`besselj(a,x)`
`bessely`	Bessel function of the second kind	`bessely(a,x)`
`besseli`	Modified Bessel function of the first kind	`besseli(a,x)`
`besselk`	Modified Bessel function of the second kind	`besselk(a,x)`
`ceil`	Round up to nearest integer	`ceil(x)`
`conj`	Complex conjugate	`conj(x)`
`cos`	Cosine	`cos(x)`
`cosh`	Hyperbolic cosine	`cosh(x)`

TABLE 3-4: MATHEMATICAL FUNCTIONS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
cot	Cotangent	cot(x)
coth	Hyperbolic cotangent	coth(x)
csc	Cosecant	csc(x)
csch	Hyperbolic cosecant	csch(x)
eps	Floating point relative accuracy (machine epsilon, 2^{-52} or about $2.2204\cdot10^{-16}$, for double floating point numbers)	eps
erf	Error function	erf(x)
exp	Exponential	exp(x)
floor	Round down to nearest integer	floor(x)
gamma	Gamma function	gamma(x)
i, j	Imaginary unit	i
imag	Imaginary part	imag(u)
inf	Infinity	inf
log	Natural logarithm	log(x)
log10	Common logarithm (base 10)	log10(x)
log2	Base-2 logarithm	log2(x)
max	Maximum of two arguments	max(a,b)
min	Minimum of two arguments	min(a,b)
mod	Modulo operator	mod(a,b)
NaN, nan	Not-a-number	nan
pi	Pi	pi
psi	Psi function and its derivatives	psi(x,k)
range	Create a range of numbers	range(a,step,b)
real	Real part	real(u)
round	Round to closest integer	round(x)
sec	Secant	sec(x)
sech	Hyperbolic secant	sech(x)
sign	Signum function	sign(u)
sin	Sine	sin(x)
sinh	Hyperbolic sine	sinh(x)
sqrt	Square root	sqrt(x)

TABLE 3-4: MATHEMATICAL FUNCTIONS

FUNCTION	DESCRIPTION	SYNTAX EXAMPLE
tan	Tangent	tan(x)
tanh	Hyperbolic tangent	tanh(x)

Physical Constants

Physical constants are fundamental, universal constants that represent physical quantities. COMSOL Multiphysics includes the most widely used physical constants as predefined variables.

See Also

Variables

Table 3-5 lists all supported physical constants with their names, symbol (variable name), value, and SI unit. When defined as variables, all variable names get the suffix _const, which indicates that they are physical constants. The values are taken from Ref. 1 and include the SI unit.

Important

The suffix _const must be added to the symbols in Table 3-5 when using the physical constants as predefined variables in models. For example, h_const is the variable name for Planck's constant.

TABLE 3-5: PHYSICAL CONSTANTS

NAME	SYMBOL	VALUE
Acceleration of gravity	g	9.80665[m/s^2]
Avogadro constant	N_A	6.02214179e23[1/mol]
Boltzmann constant	k_B	1.3806504e-23[J/K]
Characteristic impedance of vacuum (impedance of free space)	Z0	376.730313461...[ohm] (mu0*c)
Electron mass	me	9.10938215e-31[kg]
Elementary charge	e	1.602176487e-19[C]
Faraday constant	F	96485.3399[C/mol]
Fine-structure constant	alpha	7.2973525376e-3
Gravitational constant	G	6.67428e-11[m^3/(kg*s^2)]

TABLE 3-5: PHYSICAL CONSTANTS

NAME	SYMBOL	VALUE
Molar volume of ideal gas (at 273.15 K and 1 atm)	V_m	22.413996e-3[m^3/mol]
Neutron mass	mn	1.674927211e-27[kg]
Permeability of vacuum (magnetic constant)	mu0	4*pi*1e-7[H/m]
Permittivity of vacuum (electric constant)	epsilon0	8.854187817e-12[F/m]
Planck's constant	h	6.62606896e-34[J*s]
Planck's constant over 2 pi	hbar	1.054571628e-34[J*s]
Proton mass	mp	1.672621637e-27[kg]
Speed of light in vacuum	c	299792458[m/s]
Stefan-Boltzmann constant	sigma	5.670400e-8[W/(m^2*K^4)]
Universal gas constant	R	8.314472[J/(mol*K)]
Wien displacement law constant	b	2.8977685e-3[m*K]

Reference

1. *The NIST Reference on Constants, Units, and Uncertainty*, http://physics.nist.gov/cuu/Constants/index.html

Global Parameters

In this section:

- About the Global Parameters
- Defining, Saving, or Importing Global Parameters
- Loading Parameters from a Text File

About the Global Parameters

Under the **Global Definitions** node (≡), enter values in the **Parameters** table to define parameters used throughout the model making it possible to parameterize, for example, a geometric dimension. Parameters are scalar numbers that are the same for all geometries and domains, can depend on other parameters, and can contain any built-in mathematical function.

Model	Process Control Using a PID Controller: Model Library path **COMSOL_Multiphysics/Multidisciplinary_Models/pid_control**

Use parameters in the following contexts:

- In dimensions for geometric primitives or other geometry features
- As parameters for the mesh generators
- As parameters to control some aspects of the solution process
- To quickly evaluate a mathematical expression, including unit conversion
- Dimensions in the geometry
- As mesh-size parameters
- In physics settings, expressions, and coupling operators
- In expressions when evaluating results

Defining, Saving, or Importing Global Parameters

1 In the **Model Builder**, right-click **Global Definitions** and select **Parameters** (Pi).

2 Enter parameters manually or import from a text file (see Loading Parameters from a Text File).

   a In the **Parameters** table or the field under the table, enter a parameter **Name**.

   b In the **Expression** column or field, enter an expression that defines the parameter value—a scalar number.

   The parameter value can be a function of other parameters as well as built-in mathematical constants and functions, and it can use unit syntax.

   The **Value** column displays the value of the parameter in the base unit system.

   c In the **Description** column or field, enter an optional description.

3 Add more parameters as required to the rows in the table.

   Organize the table with the **Move Up** (↑), **Move Down** (↓), or **Delete** (✕) buttons or right-click the cell and select **Move Up**, **Move Down**, or **Delete**.

4 *Optional:* Save the parameters to a text file to reuse in other models. Click the **Save to File** button ( ) and enter a **File name**, including the extension .txt. Click to **Save** the text file.

   The information is saved in space-separated columns in the same order as displayed on screen.

Loading Parameters from a Text File

1 In the **Model Builder**, right-click **Global Definitions** and select **Parameters** (Pi).

2 Create data from a spreadsheet program or other format by saving the file with a .txt extension. Data must be separated by tabs or colons. Rows that begin with # or % are treated as comments and do not appear in the table.

3 In the settings window, click the **Load from File** button ( ).

4 Navigate to the text (.txt) file to load and click **Open**.

   The data loads to the table. If there is other data already in the table, it adds it after the last row. Move or edit as required.

Variables

Using the **Variables** (a=) node's settings window create symbolic *variables* that are defined using an expression. Depending on the type of variable, use a variable in a local or global context to simplify model definitions and for convenient access to quantities related to the expression that defines the variable. This section describes the syntax and names for variables that you define or that are predefined in a COMSOL Multiphysics model. Variables can be applied both at the global and local model levels.

In this section:

- About Global and Local Variables
- Predefined Physics Interface Variables
- Variable Naming Convention and Scope
- Variable Classification and Geometric Scope
- Specifying Varying Coefficients and Material Properties
- Variables for Time, Frequency, and Eigenvalues
- Geometric Variables
- Shape Function Variables
- Solver Variables
- Entering Ranges and Vector-Valued Expressions
- Adding Global Variables to the Model Builder
- Adding Local Variables to Individual Models
- Assigning Geometric Scope to a Variable
- Adding Variable Definitions
- Editing Variable Definitions
- Saving Variable Definitions to a Text File
- Loading Variable Definitions from a Text File
- Summary of Common Predefined Variables

About Global and Local Variables

You can define *global variables* or *local variables* in the same way as parameters and functions. Global variables are applied to all the models in a file, whereas local variables are valid in the entire model (geometry) or for specific domains, boundaries, edges, or points. This provides the flexibility to create variables with different meanings in different parts of the model. It often beneficial to define variables locally if they refer only to local quantities and built-in variables. Then access the variables using a more compact syntax (see Variable Naming Convention and Scope), and you do not make use of the global namespace. Global variables are primarily useful for defining expressions with variables that do not depend on the geometry, such as time and the dependent variables in an ODE or algebraic equation.

THE BENEFITS OF DEFINING VARIABLES

Defining variables simplifies model specification and results analysis. For example, variables can make a model easier to understand by using short names for complicated expressions. You can also use expressions that contain other variables, which are then computed when solving the model.

Using variables can also improve visualization and results analysis. For example, if you need to view a field variable throughout a model, but it has different names in different domains, you can create a common unique name for the field variable contained in different domains and then plot the created variable to show the entire field in all domains. If you have provided a description for a variable that you use as the expression in a plot, it becomes the default description for the plot. The variables that you define, including the descriptions, become available as predefined expressions in plots and results evaluation settings (in the **Definitions** list) when you have re-solved the model or updated the solution by right-clicking the **Solver** node ( ) under **Solver Configurations** ( ) in the **Study** branch ( ) and selecting **Solution>Update** ( ).

To specify a coefficient or a material property, enter a value or an expression directly in the corresponding field. Such expressions can contain:

- Numerical values.
- Units (see Using Units)
- Mathematical Functions
- Spatial coordinates, time, and the dependent variables in any physics interface in the model as well as their spatial derivatives and time derivatives.
- Physical Constants—predefined universal constants.
- Parameters, expression variables, coupling operators, and functions you have defined, including external functions and MATLAB functions (requires the

COMSOL LiveLink for MATLAB). See Operators, Functions, and Variables Reference.

- COMSOL Multiphysics functions and operators such as `d` and `mean`.

Use these types of variables, constants, and expressions in all features for the physics interfaces; many types of variables are also available anywhere in the model.

FIELD VARIABLES AND SCALAR VARIABLES

Most variables for material properties, coordinates, and other quantities defined on the computational domain are field variables—that is, they are defined so that they can vary in space and time. For example, a variable for the density in a Solid Mechanics interface, `solid.rho`, defines a density ρ as ρ(x,y,z,t) in 3D (but it is often defined as a scalar value in a domain with a homogeneous material). This means that you can plot it as a surface or volume plot, for example, but not use it directly where a scalar value is used, such as in an ODE or a global evaluation. Instead, use it in connection with an operator that computes an integral, average, or interpolated quantity, which then becomes a scalar variable. If you, for example, have defined an integration operator `intop1` on the domain, you can use the expression `intop1(1*solid.rho)` to define a scalar variable for the total mass in the domain. If you instead use `intop1(1)*solid.rho`, it does not define a scalar variable—it defines another field variable that is the density multiplied with the volume of the domain.

Predefined Physics Interface Variables

Physics interface variables are predefined variables that the physics interfaces introduce. They are typically functions of the dependent variables. Many of these variables are available in the **Predefined quantities** lists in the settings for plots and other results features.

Physics interface variables are physics-interface specific and consist of physically meaningful combinations of material properties and other variables. These variables are predefined for each physics interface.

To access physics interface variables, use a variable scoping syntax that uses the interface identifier to indicate the physics interface that they belong to.

> *Tip*
>
> The **Equation View** subnode is available for all physics feature nodes and contains a table with the names, expressions, units, and descriptions for the physics interface variables that the node defines. To display the **Equation View** subnodes, click the **Show** button ( ) and select **Equation View** from the **Model Builder**.

Variable Naming Convention and Scope

COMSOL Multiphysics uses variable scoping to control the access to variables within a model and variables in other models within the same MPH-file, for example. To access variables, use the following scoping mechanisms:

- To refer to the top level of the model tree, use `root`.
- To refer to variables in a model branch, use its model identifier, such as `mod1`.
- To refer to variables in a physics interface, use its interface identifier, such as `solid`.
- To refer to material properties, use the material node's tag, such as `mat1`, and the tag of the property group for the material property group, which is typically `def` for the Basic property group. For example, access the density ρ in Material 1 using `mat1.def.rho` (or using the full scope for Model 1, `root.mod1.mat1.def.rho`; see below). Use this pattern when referencing to other material properties too.
- COMSOL evaluates the physics interface variables in the model scope so you need to prepend the interface identifier to access these variables. For example, `solid.disp` refers to the total displacement in a Solid Mechanics interface with the interface identifier `solid`.
- The dependent variables (field variables) are unique within a model, and you do not need the interface identifier to access them. For example, enter `T` to access the temperature in a Heat Transfer interface using the default name for the temperature.
- When referring to a variable you only need to provide the part of the full scope that makes the variable unique. For example, within a Solid Mechanics interface `solid` in Model 1 `mod1`, it is sufficient to type `solid.disp`, but `mod1.solid.disp` and the full scope `root.mod1.solid.disp` are also correct. To access the same variable from another model, use `mod1.solid.disp` or `root.mod1.solid.disp`. The same mechanism applies to variables defined within a model. To access a global parameter `param1`, you can use `param1` directly or `root.param1`.

These variable naming conventions mean that the syntax becomes shorter when defining variables locally in a model branch instead of globally. For example, to access the x-component of the electric field, E_x, in an Electrostatics interface with the interface identifier `es` in a Model 1 with the name model identifier `mod1`, you can use `es.Ex` in a variable defined in Model 1, but for a variable defined globally, the syntax is `mod1.es.Ex`.

For information about the interface identifiers, see the sections about the physics interfaces in this book and the *User's Guides* for the modules.

Variable Classification and Geometric Scope

COMSOL Multiphysics provides a set of variables you can use in expressions when specifying a model and for visualizing and analyzing the solution. A number of variables are common to all physics interfaces in a model, for example, the spatial coordinate variables x, y, and z (for 3D and planar 1D and 2D geometries).

Every physics interface also has its own set of variables to represent quantities relevant to the physics or equations that it covers. Characteristics of variables include:

- Parameters and geometric variables are always available.
- The choice of physics interfaces and the dimension of the geometries in the model affect the set of available field variables and special variables.
- Equations can be active in different domains, which also affects the set of available variables. Variables corresponding to certain equation terms are available only in the particular part of the geometry (such as domains, boundaries, or points) where the equation is active.
- Variables defined on boundaries, edges, and points are active if the adjacent domain is active.

Variables are divided into the following general categories:

- Parameters (see Global Parameters) and *expression variables* (user-defined parameters)
- Variables for Time, Frequency, and Eigenvalues
- Physical Constants (predefined universal constants)
- Geometric Variables (variables that characterize geometric properties).
- *Field variables* (dependent variables and variables derived from them). For example, Shape Function Variables and Predefined Physics Interface Variables.

- Model Couplings
- Solver Variables (available only during the solution process)

Specifying Varying Coefficients and Material Properties

When assigning varying material properties, for example, you can enter expressions that use:

- Space and time variables
- Results from other physics interfaces in the model
- Results from the current physics interface
- Interpolation from gridded data that is entered into a table or stored in a file
- Interpolation from measured data

There are three categories of properties described next—space-varying, time-varying, and nonlinear. These are the most common for coefficients and material properties that vary during analysis.

SPACE-VARYING PROPERTIES

Model properties can vary in space for all study types and solvers (except for applications that do not involve a geometry).

3D

The spatial variables (independent variables) to use in these properties are normally x, y, and z for 3D geometries. If the model contains a Solid Mechanics, Moving Mesh or Deformed Geometry interface, you may also use X, Y and Z to refer to the material reference configuration or Xg, Yg and Zg to refer to the original geometry.

2D

The spatial variables (independent variables) to use in these properties are normally x and y for 2D geometries. If the model contains a Solid Mechanics, Moving Mesh or Deformed Geometry interface, you may also use X and Y to refer to the material reference configuration or Xg and Yg to refer to the original geometry.

2D Axi

1D Axi

The spatial variables (independent variables) to use in these properties are normally r and z for axisymmetric geometries. If the model contains a Solid Mechanics, Moving Mesh or Deformed Geometry interface, you may also use R and Z to refer to the material reference configuration or Rg and Zg to refer to the original geometry.

- Use space variables in expression variables and coupling operators. For example, you might enter the expression `K0+0.1*x` in a field if thermal conductivity varies with x.
- If the model contains a Solid Mechanics, Moving Mesh or Deformed Geometry interface, the spatial coordinate variables may be functions of the solution. A space-varying expression may in that case make the property nonlinear.
- Use space variables when using interpolation commands with gridded and measured data sets.

TIME-VARYING PROPERTIES

Model properties can vary in time for time dependent (transient) analyses. Use the reserved variable name for time, `t`. See The Time Variable.

NONLINEAR PROPERTIES

In a nonlinear model, the model properties depend on the variables for which you solve (the dependent variables). For example, you might enter an expression like `K0+0.1*T^2` in a field if thermal conductivity varies quadratically with temperature T.

COMSOL Multiphysics automatically detects if a model is nonlinear and then uses a nonlinear solver.

Variables for Time, Frequency, and Eigenvalues

The following variables are predefined to represent time (`t`), frequency (`freq`), and eigenvalue (`lambda`).

THE TIME VARIABLE

- For time dependent problems use the time variable (`t`) with the unit seconds (s).
- It can be part of any expression in the point, edge, boundary, and domain settings, as well as during analysis.
- It is always scalar, even when the solution contains more than one output time.

- For stationary models, the value of `t` is 0.
- The value of `t` for results evaluation corresponds to the selection made in the **Time** list in the **Data** sections for the visualization and data evaluation nodes in the **Results** branch in the **Model Builder**. See the Results Evaluation and Visualization section.

THE FREQUENCY VARIABLE

The frequency variable (`freq`) is the global frequency for models in the frequency domain (time-harmonic models and frequency response analysis, for example).

THE EIGENVALUE VARIABLE

- When specifying an eigenvalue problem, use the eigenvalue variable (`lambda`) like any other variable.
- The eigenvalue solver extracts the Jacobian, the damping matrix, and the mass matrix through Taylor expansion of the problem with respect to the eigenvalue variable around a specified eigenvalue linearization point (which is zero by default).
- Other solvers treat the eigenvalue variable as a constant with value zero, unless it is set by an eigenvalue solution used as initial solution.
- After solving an eigenvalue problem, the eigenvalue name is available as a scalar variable for use in expressions.
- To choose between different eigenvalues, select one from the **Eigenvalue** list in the **Data** sections for the visualization and data evaluation features in the **Results** section of the **Model Builder**. The value of the eigenvalue variable corresponds to the selection made in the **Eigenvalue** list. See the Results Evaluation and Visualization section.
- For many physics interfaces, the default is to use an eigenfrequency study and compute and display the eigenfrequencies rather than the eigenvalues.

Geometric Variables

The variables that characterize geometric properties are listed in Table 3-6, with detailed descriptions for some of the variables following the table.

TABLE 3-6: GEOMETRIC VARIABLES

VARIABLE	DESCRIPTION
`dom`	The domain number, the boundary number, the edge number, or the vertex (point) number. This variable takes only integer values.
`dvol`	The volume scale factor variable, `dvol`, is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates. For 3D domains, this is the factor that the software multiplies volumes by when moving from local coordinates to global coordinates. In 2D and 1D domains, it is an area scaling factor and length scaling factor, respectively. If a moving mesh is used, `dvol` is the mesh element scale factor for the undeformed mesh. The corresponding factor for the deformed mesh is named `dvol_spatial`.
`h`	Available on all geometric entities, the variable `h` represents the mesh element size in the material/reference frame (that is, the length of the longest edge of the element).
`linearizedelem`	In some calculations COMSOL may force mesh elements to become linear. This variable returns one inside such an element and zero otherwise. Note that the faces of the linearized mesh elements are not considered to be linearized themselves. You can use this variable to identify mesh elements with linearized elements.
`meshtype`	The mesh type index for the mesh element. This is the number of edges in the element.
`meshelement`	The mesh element number.
`meshvol`	Volume of the (linearized) mesh element.
`nx`, `ny`, `nz`	See Normal Variables.
`qual`	A mesh quality measure.

TABLE 3-6: GEOMETRIC VARIABLES

VARIABLE	DESCRIPTION
reldetjac reldetjacmin	The determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates, normalized to that of a first-order element. It is the determinant of the Jacobian matrix for the mapping from the straight mesh element to the possibly curved element used when solving. Use this variable to measure the difference in shape between a curved element and the corresponding straight element. The variable reldetjacmin is a scalar for each element defined as the minimum value of the reldetjac variable for the corresponding element. A reldetjacmin value less than zero for an element means that the element is wrapped inside-out; that is, the element is an *inverted mesh element*.
s, s1, s2	See Parameterization Variables.
t*x* and t*y* (2D) t1*x*, t1*y*, t1*z* (3D edges) t2*x*, t2*y*, t2*z* (3D surfaces)	See Tangent Variables.
x, y, z r, z	See Spatial Coordinate Variables.

Important

When entering the spatial coordinate, parameterization, tangent, and normal geometric variables, replace the letters highlighted below in *italic font* with the actual names for the dependent variables (solution components) and independent variables (spatial coordinates) in the model.

For example, replace u with the names of the dependent variables in the model, and replace x, y, and z with the first, second, and third spatial coordinate variable, respectively. x_i represents the ith spatial coordinate variable. If the model contains a deformed mesh or the displacements control the spatial frame (in solid mechanics, for example), you can replace the symbols x, y, and z with either the spatial coordinates (x, y, and z by default) or the material (reference) coordinates (X, Y, and Z by default).

The variables dvol, h, qual, reldetjac, and reldetjacmin are based on the mesh viewed in the material (reference) frame. If you have a moving mesh, the corresponding variables for the mesh viewed in the spatial frame have a suffix _spatial (that is, dvol_spatial, h_spatial, and so on). If you use a deformed geometry, the corresponding variables for the original, undeformed mesh have a suffix _mesh (for example, h_mesh).

SPATIAL COORDINATE VARIABLES

- The spatial coordinate variables (independent variables) x, y, z are available for all domain types.
- For a Cartesian geometry the default names for the spatial coordinates are x, y, z.
- For axisymmetric geometries the default names for the spatial coordinates are r, φ , and z.
- If a deformed mesh is used, x, y, z can be both the *spatial coordinates* (x, y, z) and the *material/reference coordinates* (X, Y, Z); see Mathematical Description of the Mesh Movement.
- If the model includes a deformed mesh, the variables xTIME, yTIME, zTIME represent the mesh velocity. To access these variables, replace x, y, and z with the names of the spatial coordinates in the model (x, y, and z).

PARAMETERIZATION VARIABLES

The surface-boundary parameterization variables can be useful for defining distributed loads and constraints such as a parabolic velocity profile. The available parameterization variables are:

2D	The curve parameter s (or s1) in 2D. Use a line plot to visualize the range of the parameter, to see if the relationship between x and y (the spatial coordinates) and s is nonlinear, and to see if the curve parameterization is aligned with the direction of the corresponding boundary. In most cases it runs from 0 to 1 in the direction indicated by the arrows shown on the boundaries when in the boundary selection mode. You can use s on boundaries when specifying boundary conditions.

3D

- The arc length parameter `s1` available on edges in 3D. It is approximately equivalent to the arc length of the edge. Use a line plot to visualize to see the values of `s1`.
- The surface parameters `s1` and `s2` in 3D are available on boundaries. In many cases they can be difficult to use because the relationship between x, y, and z (the spatial coordinates) and `s1` and `s2` is nonlinear. Often it is more convenient to use expressions with x, y, and z for specifying distributed boundary conditions. To see the values of `s1` and `s2`, plot them using a surface plot.

TANGENT AND NORMAL VARIABLES

The tangent and normal variables are components of the tangential and normal unit vectors.

Tangent Variables

2D

In 2D, `tx` and `ty` define the curve tangent vector associated with the direction of the boundary.

3D

In 3D, the tangent variables `t1x`, `t1y`, and `t1z` are defined on edges. The tangent variables `t1x`, `t1y`, `t1z`, `t2x`, `t2y`, and `t2z` are defined on surfaces according to

$$(t_{ix}, t_{iy}, t_{iz}) = k_i\left(\frac{\partial x(s_1, s_2)}{\partial s_i}, \frac{\partial y(s_1, s_2)}{\partial s_i}, \frac{\partial z(s_1, s_2)}{\partial s_i}\right), \quad i = 1, 2$$

These most often define two orthogonal vectors on a surface, but the orthogonality can be ruined by scaling geometry objects. The vectors are normalized; k_i is a normalizing parameter in the expression just given.

If a deformed mesh is used, the tangent variables are available only for the undeformed mesh. In that case, replace x, y, and z with the names of the reference coordinates (`X`, `Y`, and `Z` by default).

Normal Variables

1D

In 1D, `nx` is the outward unit normal pointing out from the domain.

2D

In 2D, `nx` and `ny` define a normal vector pointing outward relative to the domains.

3D

In 3D, `nx`, `ny`, and `nz` define a normal vector pointing outward relative to the domains.

Direction of the Normal Component on Interior Boundaries

To get control of the direction of the normal component on interior boundaries, the following variables are available:

1D

In 1D:

- `unx`, the outward unit normal seen from the upper domain
- `dnx`, the outward unit normal seen from the lower domain

2D

In 2D:

- `unx` and `uny` for the up direction
- `dnx` and `dny` for the down direction

The upside is defined as the left side with respect to the direction of the boundary.

3D

In 3D:

- `unx`, `uny`, and `unz` for the up direction
- `dnx`, `dny`, and `dnz` for the down direction

To visualize any of these vector variables use arrow plots on surfaces or lines.

If a deformed mesh is used, the normal variables are available both for the deformed configuration and for the undeformed configuration. In the first case, replace x, y, and z with the spatial coordinate names (`x`, `y`, and `z` by default). In the second case, replace x, y, and z with the material/reference coordinate names (`X`, `Y`, and `Z` by default).

Normal Vector Variables Representing Element Surface Normals

A similar set of variables—n*x*mesh, un*x*mesh, and dn*x*mesh, where x is the name of a spatial coordinate—use the element shape function and are normal to the actual element surfaces rather than to the geometry surfaces.

Shape Function Variables

Shape function variables and boundary-coupled shape variables are related to the *shape functions* of the finite elements. The shape function variables are related to the degrees of freedom defined by a finite element.

A finite element defines a number of variables, typically a dependent variable and its derivatives. Such variables are called shape functions variables because they are computed directly from shape functions and the degrees of freedom.

When a physics interface is selected, names are entered for the dependent variables—these names are used to construct the finite elements. The dependent variable name is the basis for additional variable names that the finite elements generate.

Important

When entering the shape function variables, replace the letters highlighted below in *italic font* with the actual names for the dependent variables (solution components) and independent variables (spatial coordinates) in the model.

For example, replace u with the names of the dependent variables in the model, and replace x, y, and z with the first, second, and third spatial coordinate variable, respectively. x_i represents the ith spatial coordinate variable. If the model contains a deformed mesh or the displacements control the spatial frame (in solid mechanics, for example), you can replace the symbols x, y, z with either the spatial coordinates (`x`, `y`, and `z` by default) or the material/reference coordinates (`X`, `Y`, and `Z` by default).

AN EXAMPLE OF LAGRANGE ELEMENT VARIABLES

For the Lagrange element, which is the element type used by most physics interfaces, Table 3-7 lists the available variable names, assuming you gave the name *u* as the argument to the shape function, and that the names *x*, *y*, and *z* are provided for the independent variables.

TABLE 3-7: LAGRANGE ELEMENT VARIABLE NAMES

ENTITY TYPE\ SPACE DIMENSION	1D	2D	3D
POINT		u	u
EDGE			u, uTx, uTy, uTz
BOUNDARY	u, uTx, ut, uTxt	u, uTx, uTy, ut, uTxt, uTyt	u, uTx, uTy, uTz, ut, uTxt, uTyt, uTzt
DOMAIN	u, ux, uxx, ut, uxt, uxxt, utt, uxtt, uxxtt,	u, ux, uy, uxx, uxy, uyx, uyy, ut, uxt, uyt, uxxt, uxyt, uyxt, uyyt, utt, uxtt, uytt, uxxtt, uxytt, uyxtt, uyytt	u, ux, uy, uz, uxx, uxy, uxz, uyx, uyy, uyz, uzx, uzy, uzz, ut, uxt, uyt, uzt, uxxt, uxyt, uxzt, uyxt, uyyt, uyzt, uzxt, uzyt, uzzt, utt, uxtt, uytt, uztt, uxxtt, uxytt, uxztt, uyxtt, uyytt, uyztt, uzxtt, uzytt, uzztt

For example, with a fluid-flow interface, you get the set of variables indicated in Table 3-7 for u, v, w, and p, respectively.

- The variables *ux*, *uy*, and *uz* are the components of the gradient ∇u, that is, the 1st-order spatial derivatives.
- The variables *uxx*, *uxy*, *uxz*, *uyx*, *uyy*, *uyz*, *uzx*, *uzy*, and *uzz* are the 2nd-order space derivative components. They are meaningful only if the degree of the polynomial shape function used in an element is high enough.
- For elements with 2nd-degree polynomial shape functions (2nd-order elements), the polynomial degree of the 2nd-order derivatives is zero; that is, the second derivatives are constant in each element.
- For element orders lower than two, the second derivatives evaluate to zero regardless of the values of the 2nd-order derivatives of the true solution.

If the model uses a deformed mesh, each finite element is associated with a certain frame (the spatial frame or the material frame). The frame determines the names of the variables generated by the finite element. For instance, if the spatial frame is used, the Lagrange element computes derivatives with respect to the spatial coordinates, `ux`, `uy`, and `uz`. If the material frame is used, the Lagrange element computes derivatives with respect to the material coordinates `uX`, `uY`, and `uZ`.

THE TIME DERIVATIVES OF THE DEPENDENT VARIABLES

The variable u`t` is the time derivative of the dependent variable u. You can also form mixed space-time derivatives as ux_i`t`, for example, `uxt`,

$$\frac{\partial^2 u}{\partial x \partial t}$$

Important

The `t` must be last in a mixed derivative. The second time derivatives can also be used, such as `utt` or `uxtt` (but not higher derivatives in time).

If the model contains a deformed mesh, there is, in addition to the usual time derivative `ut`, the *mesh time derivative* `uTIME`. This also holds for mixed space-time derivatives.

TANGENTIAL DERIVATIVE VARIABLES

On boundaries, edges, and points you also have access to *tangential derivative variables*. They have names such as uTx, uTy, and uTz. Using these variables, it is possible to create models with phenomena on boundaries, edges, or points as described with PDEs.

The tangential derivative variables represent the Cartesian components of the tangential projection of the gradient of shape function variables:

$$(\nabla u)_T = (I - \mathbf{n}\mathbf{n}^T) \cdot \nabla u$$

In this equation, $(\nabla u)_T$ is the tangential gradient, which consists of the tangential derivatives in each spatial direction, I is the unity tensor, $\mathbf{n}$ is the outward unit normal vector, and ∇u is the gradient of u.

LAGRANGE MULTIPLIER VARIABLES

If weak constraints are activated for boundary conditions that are constraints (Dirichlet boundary conditions), COMSOL Multiphysics adds variables for the Lagrange multipliers (one for each dependent variable) by adding `_lm` as a suffix to the dependent variable name. For example, for a dependent variable u, the corresponding Lagrange variable is `u_lm`. The Lagrange multipliers are available on boundaries, and you can also evaluate them on edges (in 3D) and points (in 2D and 3D).

VARIABLE INHERITANCE

On boundaries, edges, and points, gradients and second derivatives of the shape functions are available by *inheritance*; that is, the average of the values of the variables from the adjacent domains are computed. This process can progress for several levels.

For example, `ux` is the average on a boundary from the adjacent domains, then the average on an edge from the adjacent boundaries, and finally, the average at the points from the adjacent edges.

If possible, avoid using variable inheritance for gradients and second derivatives in a model. Instead, use the tangential derivative variables for equation-based modeling on boundaries.

For computations of integrals of reaction forces and fluxes, use The Reacf Operator.

For high accuracy reaction forces and fluxes in other circumstances, use weak constraints and Lagrange multipliers on boundaries instead of directly accessing the gradient through inheritance (see Calculating Accurate Fluxes).

Note

When you plot or evaluate—on a boundary, for example—the value of a variable that is discontinuous across that boundary (a thin resistive layer, for example), the value is the average of the value on the “up” and “down” sides of the boundary. You can use the `up` and `down` operators to get the value on either side of the boundary (see The Up, Down, and Mean Operators).

Solver Variables

The following table lists variables that are available during the solution process only. They can be used in solver settings in the Study branches but are not available for use in, for example, results evaluation, and plots.

TABLE 3-8: SOLVER VARIABLES

VARIABLE	DESCRIPTION
`niterCMP`	This is a global variable. It evaluates to the iteration number for nonlinear iterations. It starts from one and increases with one for each fully coupled or segregated iteration. It is used by some physics interfaces to control damping mechanisms. Examples are pseudo-time stepping in fluid dynamics and the penalty factor in the augmented Lagrangian method for contact problems in structural mechanics.
`gmg_level`	This is a global variable. It evaluates to the geometric multigrid level. It is zero for the top level (the one solved for), one for the next coarser level, and so on. It is used by some physics interfaces to control artificial stabilization.
`timestep`	This is a global variable. It evaluates to the current time step used by the time-dependent solver. It is used by some physics interfaces to control artificial stabilization. You can use it, for example, to create a stop condition that stops the time stepping if the time step becomes smaller than some threshold value.

Entering Ranges and Vector-Valued Expressions

You can enter ranges and vector-valued expressions such as extra grid-line coordinates using the following formats:

- A space-separated or comma-separated list of values: `10, 15, 23, 29.7, 30`.
- A delimited space-separated list using curly braces; for example, as an argument to a function: `cos({0 pi/4 pi/2})`.
- Equally-spaced values using the `range` function as in `range(`*`start value`*`,`*`step_size`*`,`*`end value`*`)`. For example, `range(0,0.2,3)`

creates the values 0, 0.2, 0.4,..., 2.6, 2.8, 3.0. The step size is 1 if you provide only start and end values and skip the step value.

Note

`start_value` can be either smaller or larger than *`end_value`*. In the latter case, the step size must be negative. For example, `range(0,-5,-100)` creates the values 0, –5, –10, ..., –95, –100, while `range(0,5,-100)` is an empty set of values.

Combine these formats in a single expression to create an array of values that contain an arbitrary number of segments with differently spaced values mixed with other freely specified values.

EXAMPLES USING THE RANGE FUNCTION

- `range(a,(b-a)/(n-1),b)` gives a list of *n* equally-spaced increasing values in the range [`a`, `b`] if `b` >`a` or decreasing values in the range [`b`, `a`] if `a` >`b`.
- `10^range(-3,3)` gives the exponentially increasing sequence 10^{-3}, 10^{-2}, …, 10^{3}.
- `1^range(1,10)` gives a sequence of length 10 where all elements equal 1. Multiplying the vector `1^range(1,n)` by a constant value `a` gives a vector of *n* elements all equal to `a`.
- `0^range(1,5)` gives the sequence `0 0 0 0 0`.

USING RANGES TO GENERATE ARRAYS

A convenient way to generate vectors of values is to use the **Range** dialog box, which you open by clicking the **Range** button ( ) next to most of the fields that accept vectors of values.

In that dialog box, use the **Entry method** list to select **Step** to enter a step size or **Number of values** to specify the number of values in the array. Specify the start value for an array of values in the **Start** field. Enter the step size in the **Step** field or the number of values in the **Number of values** field, depending on the setting in the **Entry method** list. Specify the end value for the array of values in the **Stop** field. By default, the spacing of the values is linear, but you can select a function to apply to all values. To do so, choose one of the available arithmetic and trigonometric functions from the **Function to apply to all values** list. For example, select **exp10** to create an array of exponentially increasing values. The list includes the following functions:

- The default value **None**, which means linear spacing using the `range` function directly with the values specified.

- The exponential functions **exp10** (base-10 exponential function) and **exp** (base-*e* exponential function), which create exponentially-spaced values using the specified range of values as powers of 10 and of the mathematical constant *e*, respectively.
- The trigonometric functions **cos** (cosine) and **sin** (sine), which create sinusoidally varying values.
- The square root function **sqrt**, which creates a vector with values that are the square roots of the values specified.

Click **Replace** to replace the contents in the field with the values specified in the **Range** dialog box.

Click **Add** to add the range of values to the end of the existing values in the associated field. That way you can create more complex ranges.

MODELING FEATURES THAT SUPPORT RANGES AND VECTOR-VALUED EXPRESSIONS

The following modeling features support ranges and vector-valued expressions:

- Extra grid lines in the **Axis** node's settings window.
- Interval coordinates when using the **Interval** node's settings window for 1D geometries.
- The **Copy**, **Move**, and **Rotate** transform features for geometry modeling.
- The times for output from the time dependent solver and the list of parameter values in the settings windows for study step nodes for time dependent and stationary solvers and for parametric sweeps.
- The contour levels, the streamline start-point coordinates, and the coordinates in arrow plots. Whenever you specify a number of coordinates in settings windows for plots, COMSOL Multiphysics uses scalar expansion—if one component is the same for all coordinates, enter a single number in the corresponding text field. For example, to get 101 linearly spaced coordinates from $y = 6$ to $y = 7$ along $x = 3$, enter it as the single scalar `3` for x and then `range(6,0.01,7)` for y. Thus, you need not enter 101 similar values for x.
- Element distribution in the meshing settings.

Adding Global Variables to the Model Builder

1 Open or create a model file.

2. In the **Model Builder**, right-click **Global Definitions** and select **Variables**.

   The **Variables** page opens in the settings window and a **Variables** node is added to the model under **Global Definitions**.

Model

Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**

Adding Local Variables to Individual Models

1. Open or create a model file.
2. In the **Model Builder**, right-click **Definitions** and select **Variables**.

   The **Variables** page opens in the settings window and a **Variables** node is added to the model under **Model>Definitions**.

Model

- Acoustics of a Muffler: Model Library path **COMSOL_Multiphysics/Acoustics/automotive_muffler**
- Tubular Reactor: Model Library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**
- Fluid Valve: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

Assigning Geometric Scope to a Variable

Note

The **Global Definitions Variables** page does not have a **Geometric Scope** section.

1. Create a local variable for a model.

2 In the **Geometric Scope** section, select the scope in the **Geometric entity level** list. The options are **Entire Geometry**, **Domain**, **Boundary**, **Edge** (3D only), and **Point**.

If a level other than **Entire Geometry** is selected, a **Selection** list for geometric entities of the selected type (boundaries, for example) displays.

See Also

Selecting and Deselecting Geometric Entities

Adding Variable Definitions

1 Create a global or local **Variable**.

2 On the **Variables** page, in the **Variables** table, click the first cell in the **Name** column. Enter a variable name.

3 In the **Expression** column, enter an expression that defines the variable. Expressions can contain spatial coordinates and solution components that are also available within the same scope.

The **Unit** column displays the unit of the variable.

4 In the **Description** column, enter an optional description.

5 Add more variables as required to the rows in the table.

Click any cell to move or delete the corresponding row. Click the **Move Up** (⇧), **Move Down** (⇩), or **Delete** (☒) buttons or right-click the cell and select **Move Up**, **Move Down**, or **Delete**.

Editing Variable Definitions

Click on any cell and enter changes directly in the table; or click any cell in the parameter row to edit. The **Name**, **Expression**, and **Description** fields under the table autofill with the same values. Edit the data in these fields.

Saving Variable Definitions to a Text File

1 Enter variables into the table.

2 Click the **Save to File** button (💾).

3 Enter a **File Name**, including the extension .txt. Click **Save** (💾).

The information is saved in space-separated columns in the same order as displayed on screen.

Loading Variable Definitions from a Text File

1 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Variables**.

2 Save data from a spreadsheet program or other format by saving the file with a .txt extension. Data must be separated by tabs or colons. Rows that begin with `#` or `%` are treated as comments and do not appear in the table.

3 In the settings window, click the **Load from File** button (📂).

4 Navigate to the text (.txt) file to load and click **Open**.

The data loads to the table. If there is other data already in the table, it adds it after the last row. Move or edit as required.

Summary of Common Predefined Variables

The following table summarizes the most important predefined variables that are generally available in all COMSOL Multiphysics models. Some are only available in certain geometries or in time dependent models, for example.

TABLE 3-9: COMMON PREDEFINED VARIABLES

TYPE OF VARIABLES	VARIABLES
Time	`t`
Frequency	`freq`
Eigenvalues	`lambda`
Position	x, y, z, r, X, Y, Z, R
Edge/surface parameters	`s`, `s1`, `s2`
Edge/surface normals	`n`, `nx`, `ny`, `nz`, `nr`
Edge tangents	`tx`, `ty`, `tz`, `tr`
Surface tangents	`t1x`, `t1y`, `t1z`, `t2x`, `t2y`, `t2z`
Edge/surface upward normals	`un`, `unx`, `uny`, `unz`
Edge/surface downward normals	`dn`, `dnx`, `dny`, `dnz`
Numerical constants	`eps`, `i`, `j`, `pi`
Mesh information	`h`, `dom`, `meshtype`, `meshelement`, `dvol`, `qual`, `reldetjac`, `reldetjacmin`

Global and Local Functions

In this section:

- About Global and Local Functions
- Analytic Functions
- Elevation Functions
- External Functions (Global Only)
- Gaussian Pulse Functions
- Image Functions
- Interpolation Functions
- MATLAB Functions (Global Only)
- Piecewise Functions
- Ramp Functions
- Random Functions
- Rectangle Functions
- Step Functions
- Triangle Functions
- Waveform Functions
- Specifying Discontinuous Functions

About Global and Local Functions

There are three broad categories of functions—**Analytic**, **Interpolation**, and **Piecewise**—and a number of templates for common function types, such as step and ramp functions. See Table 3-10for a list. You can also create external function interfaces to include functions written in C and MATLAB. Functions can be global or local in scope, although two function types—external functions and MATLAB functions—can only be defined globally.

To add functions to the **Model Builder**, right-click **Global Definitions** or **Definitions** and select a **Function**.

Other Function Characteristics

Smoothing

Many functions type have smoothing as an optional choice. Smoothing makes a function more well-behaved and more suitable for use in finite element modeling. It replaces jumps with smooth transitions that eliminate discontinuities and can represent the inertia present in real applications.

Plotting

Click the **Plot** button () in the upper-right corner of the settings window to create plots of all functions of 1–3 variables. For analytic functions, first define a range for the arguments in the **Plot Parameters** section.

Click the **Create Plot** button () in the upper-right corner of the settings window to create a customized plot of the function under Results, including default plot groups and plots.

Adding Functions to Materials

Define **Analytic**, **Interpolation**, and **Piecewise** functions for materials. See Materials. Then follow the instructions as described in this section.

Units for Function Inputs and Outputs

By default, functions expect dimensionless inputs and provide dimensionless outputs. In the **Units** section in the settings window for the Analytic, Interpolation, and Piecewise function nodes, define the units for the function's inputs and output. In the **Arguments** field, type a single unit for all inputs, or specify space-separated or comma-separated units for each input (type `m, s`, for example, for two input arguments with the units meter and second, respectively). In the **Function** field, type the unit for the function's output. See Using Units for general information about available units.

AVAILABLE GLOBAL AND LOCAL FUNCTIONS

The following table includes all global and local functions that are available, including links to more detailed information about each function type:

TABLE 3-10: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Analytic Functions	x^2 sin(x)	An analytic function is defined by a symbolic expression. It extends variables with the ability to bind parameters during function calls. In other words, the actual variable names in an expression are not required when writing the function. In addition to the arguments that are defined, analytic functions also recognize global parameters.
Elevation Functions (DEM)		The elevation function reads elevation data from files with digital elevation models (DEM files using the United States Geological Survey geospatial file format).
External Functions (Global Only)	f(x)	An external function interfaces to other external functions written in the C language. Then use those functions as any other functions defined in COMSOL Multiphysics. For example, for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file).
Gaussian Pulse Functions		The Gaussian pulse function is the common bell-shaped curve. It has a shape that is similar to a Gaussian (normal) distribution.
Image Functions		The image function provides a single color channel value as a 2D function of the space coordinates for any image in the BMP, JPEG, PNG, or GIF format.
Interpolation Functions		An interpolation function is defined by a table or file containing the values of the function in discrete points. The interpolation data can be both structured, (defined on a grid) or unstructured (defined on a generic point cloud).
MATLAB Functions (Global Only)	f(x)	Use a MATLAB function feature to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL Multiphysics. MATLAB functions are evaluated in MATLAB.

TABLE 3-10: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Piecewise Functions	⋀	A piecewise function is created by splicing together several functions, each defined on one interval. Define the argument, extrapolation and smoothing methods, and the functions and their intervals. The piecewise function is a function of one variable with different definitions on different intervals, which must not overlap or have any holes between them.
Ramp Functions	⟋	A ramp function is a linear increase with a user-defined slope that begins at some specified time. The ramp function is a function of one variable (the time *t*, for example).
Random Functions	〰	A random function generates white noise with uniform or normal distribution and has one or more arguments to simulate white noise. Its distribution is either uniform or normal. To generate a random function of the space coordinates x, y, and z, for example, use this function with three input variables—it returns the same value each time it is called for a given set of input arguments. To do a Monte Carlo simulation, add one or more additional input arguments that vary using during a parametric sweep, for example.
Rectangle Functions	⎍	A rectangle function is 1 in an interval and 0 everywhere else. This function (also called top hat or boxcar) is useful for band-pass filtering; use it to select values within an interval. It can also simulate a signal that is turned on during an interval or a load that is active on a part of a boundary, for example. The rectangle function is a function of one variable (the time *t*, for example).
Step Functions	┌	A step function is a sharp transition from 0 to some other value (amplitude) at some location (a certain time, for example). Create a single step function with a certain amplitude from an initial level to a final level at a start location. The step function is a function of one variable (the time t, for example).

TABLE 3-10: GLOBAL AND LOCAL FUNCTION DESCRIPTIONS

LINK TO FUNCTION	ICON	DESCRIPTION AND USE
Triangle Functions	Λ	A triangle function is a linear increase and linear decline within an interval and 0 everywhere else. Use the triangle function for band-pass filtering; that is, use it to select values within an interval. The triangle function is a function of one variable (the time *t*, for example).
Waveform Functions	∿	A waveform function is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. The waveform function is a function of one variable (the time *t*, for example).

See Also

- Specifying Discontinuous Functions
- Sorting Nodes by Space Dimension and Type
- Operators, Functions, and Variables Reference
- Using Units
- Materials
- Global and Local Functions

Analytic Functions

An **Analytic** function (x^2 sin(x)) is defined by a symbolic expression. It extends variables with the ability to bind parameters during function calls. In other words, analytic functions do not require the actual variable names in an expression when writing the function.

Model

If you have the:

- AC/DC Module, see Geoelectrics: Model Library path **ACDC_Module/Other_Industrial_Applications/geoelectrics**
- RF Module, see Second Harmonic Generation of a Gaussian Beam: Model Library path **RF_Module/Optics_and_Photonics/second_harmonic_generation**

1 Open or create a model file.

2 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Analytic** (x^2 sin(x)).

3 Enter a **Function name**.

4 Under **Parameters**:

a In the **Expression** field, enter the mathematical expression that defines the functions, such as `a+b*cos(c)`.

b Enter **Arguments** to the analytic function as comma-separated entries. For example, `a`,`b`,`c`.

c From the **Derivatives** list, select **Automatic** or **Manual** to enable COMSOL to evaluate the derivatives of all functions. COMSOL Multiphysics must know the derivative of a function if a variable is used that depends on the solution in a function argument.

- Select **Automatic** to compute the derivatives symbolically.
- Select **Manual** to enter the derivatives with respect to one or more of the arguments to the function. For undefined derivatives, COMSOL uses 0 as the value of the derivative. In this example, enter `1,cos(c),d(a+b*cos(c),c),0` in the associated field. Notice the use of the `d` operator.

5 Under **Periodic Extension**:

a Select the **Make periodic** check box to make the function periodic and extend its definition within an interval to the whole real axis.

b Define the interval in the **Lower limit** and **Upper limit** fields.

6 Under **Units**, specify units for the function's input and output, if desired.

7 Under **Advanced**, select the **May produce complex output for real arguments** check box if the defined function works similarly to `sqrt`; that is, it sometimes returns complex values for a real-valued input.

8 Under **Plot parameters**, in the **Plot Parameters** table or the fields under it, enter the **Argument** name, a **Lower limit**, and an **Upper limit**.

Use the **Move Up** (⇧), **Move Down** (⇩), or **Delete** (☒) buttons to rearrange and delete rows in the table, or right-click in any cell and select these options.

Click the **Plot** button ( ) to generate a preview plot of the function and set ranges for all the arguments. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Elevation Functions

The **Elevation** function ( ) makes it possible to import geospatial elevation data from digital elevation models (on the DEM file format using the USGS standard from the

United States Geological Survey) and map the elevation data to a function of *x* and *y*. A DEM file contains elevation data for a portion of the Earth's surface. The resulting function behaves essentially like a grid-based interpolation function.

1 Open or create a model file.
2 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Elevation (DEM)** ( ).
3 Enter a **Function name**.
4 Under **File**, specify the filename in the text field or click **Browse** to select a DEM file with elevation data in the **Elevation Data** dialog box. When you have open a DEM file, the **File** section displays the coordinates for the southeast corner. Click **Import** to import the elevation data in the specified DEM file into the model; otherwise, COMSOL Multiphysics references the elevation data on your file system. When you have imported the elevation data, the **File** section, under **Data imported into model**, contains information about the filename and the location for the data. You can then **Export** the elevation data to a file to reference it from that file instead of keeping it in the model.
5 Under **Interpolation and Extrapolation**:
   a From the **Interpolation** list, select an interpolation method: **Nearest neighbor** or **Linear** (the default).
   b From the **Extrapolation** list, select an extrapolation method (for values outside of the range for the elevation data): **Constant** (the default), **Linear**, **Nearest function**, or **Specific value**. In the last case, type the value in the **Value outside range** field (default: 0).

Click the **Plot** button ( ) to generate a preview plot of the elevation function using a surface plot with height to represent the output from the elevation function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot. Click the **Create Surface** button ( ) to create a **Parametric Surface** node that represents the elevation function as a parametric surface in the **Geometry** branch for 3D models.

External Functions (Global Only)

Note

The **External** function is only available globally on the **Global Definitions** node. See also model.func() as described in the *COMSOL Java API Reference Guide* for details about the interface to external functions and an example.

An **External** function (f(x)) interfaces to other external functions written in the C language (using a native C function or, through a wrapper function, interfacing with source code written in, for example, Fortran). Then use those functions as any other functions defined in COMSOL Multiphysics. For example, use it for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file).

1. Open or create a model file.
2. In the **Model Builder**, right-click **Global Definitions** and select **Functions>External** (f(x)).
3. Under **Functions**, in the **Library** field, enter a **Library** (the complete network path) or **Browse** to locate a library to import.
4. For each row in the **Functions** table, enter a **Function** name (`myfun`, for example).
5. For each **Function** in the table, enter a space-separated or comma-separated list of the names of its input **Arguments** (`x y` for example).

   Use the **Move Up** (↑), **Move Down** (↓), or **Delete** (✖) buttons and the fields under the table to edit the table contents.
6. Under **Derivatives**, enter data in this section based on the following.
   - **a** In the **Derivatives** table, each row contains a partial derivative of a function with respect to one of its arguments.
   - **b** The entries in the **Function** column must occur in the function table, and the entries in the **Argument** column must occur among the arguments listed for that function in the function table.
   - **c** The **Partial derivative** column contains expressions for the partial derivatives. Partial derivatives that are not specified in the table default to 0.

   Use the **Move Up** (↑), **Move Down** (↓), or **Delete** (✖) buttons and the fields under the table to edit the table contents.
7. Under **Advanced**:
   - **a** Enter a value in the **Initialization data** field. The value is sent to the library when it is loaded.
   - **b** Select the **Thread safe** check box to declare that the function is a thread-safe pure function (that is, a function that always returns the same results using the same input argument values and that do not have any side effects or output). Selecting this check box can then improve performance.

8 Under **Plot Parameters**: In the table, enter the limits of each argument to the function in a separate row from the top down. Use the **Move Up** (⇧), **Move Down** (⇩), and **Delete** (✖) buttons to position the rows if needed.

⇧ ⇩ ✖ Click the **Plot** button (🖌) to generate a preview plot of the function. Click the **Create Plot** button (🖌) to create a plot group and a plot for the function plot.

Gaussian Pulse Functions

The **Gaussian Pulse** function (⋀) is the common bell-shaped curve. It has a shape that is similar to a Gaussian (normal) distribution.

Model

If you have the:

- RF Module, see Transient Modeling of a Coaxial Cable: Model Library path **RF_Module/Verification_Models/coaxial_cable_transient**
- Chemical Reaction Engineering Module, see Surface Reactions in a Biosensor: Model Library path **Chemical_Reaction_Engineering_Module/Surface_Reactions_and_Deposition_Processes/reacting_pillars**

1 Open or create a model file.

2 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Gaussian Pulse** (⋀).

3 Enter a **Function name**.

4 Under **Parameters**:

a Enter a **Location** value for the Gaussian pulse mean s_0.

b Enter a **Standard deviation** σ of the normal distribution.

The Gaussian pulse has the same characteristics as the normal distribution: it is a pulse with a shape that is similar to a normal or Gaussian distribution as a function:

$$y(s) = \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(s-s_0)^2}{2\sigma^2}}$$

In the equation above, s is the input variable, s_0 is the location (mean), and σ is the standard deviation. This function is a function of one variable (the time t, for example).

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Image Functions

The **Image** function ( ) makes it possible to import an image (in BMP, JPEG, PNG, or GIF format) and map the image's RGB data to a scalar (single channel) function output value. By default the function's output uses the mapping (R+G+B)/3.

1. Open or create a model file.
2. In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Image** ( ).
3. Enter a **Function name**.
4. Under **File**, specify the filename in the text field or click **Browse** to select an image file in the **Image** dialog box. Click **Import** to import the image in the specified image file into the model; otherwise, COMSOL Multiphysics references the image on your file system. When you have imported the image, the **File** section contains information about the image's filename and size. You can then **Export** the image to a file to reference it from that file instead of keeping it in the model.
5. Under **Coordinates** define the 2D coordinates and if required, flip the image. Select the **In place** check box to use the pixels in the image as the coordinates or specify the image coordinates using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields. Select the **Flip horizontally** and **Flip vertically** check box to flip the image if required.
6. Under **Color Scaling**:

   a. From the **Scaling** list, select **Automatic** (the default) or select **Manual** to enter another scalar expression for the function's value in the **Expression** field. The default is `(r+g+b)/3`, which is the automatic scaling, giving a scalar value that is the mean of the RGB values in each pixel.

   b. From the **Interpolation** list, select an interpolation method: **Nearest neighbor** or **Linear** (the default).

   c. From the **Extrapolation** list, select an extrapolation method (for values outside of the range for the image): **Constant** (the default), **Linear**, **Nearest function**, or **Specific value**. In the last case, type the value in the **Value outside range** field (default: 0).

7 Under **Clipping**, select **None** (the default) for no clipping (the default) or select **Manual** to enter the values for the box-shaped region where the image will be rendered in the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields.

8 Under **Units**, define the units for the input arguments, which are the *x* and *y* coordinates, in the **Arguments** field (default: m) and for the function's output in the **Function** field (default: no unit).

Click the **Plot** button ( ) to generate a preview plot of the image function using a surface plot with height to represent the output from the image function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Interpolation Functions

An **Interpolation** function ( ) is defined by a table or file containing the values of the function in discrete points. The interpolation data can be structured (defined on a grid) or unstructured (defined on a generic point cloud).

Model

If you have the:

- Acoustics Module, see Muffler with Perforates: Model Library path **Acoustics_Module/Industrial_Models/perforated_muffler**
- CFD Module, see Transonic Flow in a Sajben Diffuser: Model Library path **CFD_Module/High_Mach_Number_Flow/sajben_diffuser**
- Corrosion Module, see Cathodic Protection of Steel in Reinforced Concrete: Model Library path **Corrosion_Module/Verification_Models/cathodic_protection_in_concrete**
- Heat Module, see Temperature Field in a Cooling Flange: Model Library path **Heat_Transfer_Module/Tutorial_Models/cooling_flange**
- Microfluidics Module, see Knudsen's Minimum: Model Library path **Microfluidics_Module/Rarefied_Flow/knudsen_minimum**
- Pipe Flow Module, see Geothermal Heating from a Pond Loop: Model Library path **Pipe_Flow_Module/Heat_Transfer/geothermal_heating**
- Structural Mechanics Module, see Nonlinear Magnetostrictive Transducer: Model Library path **Structural_Mechanics_Module/Magnetostrictive_Devices/nonlinear_magnetostriction**

1 Open or create a model file.

2 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Interpolation** ( ).

3 Under **Parameters,** select **Table** or **File** from the **Data source** list.

- If **Table** is selected, enter a **Function name** and enter coordinates t and function values $f(t)$ into the table cells. Use the **Move Up** ( ), **Move Down** ( ), or **Delete** ( ) buttons to rearrange and delete rows in the table. A function of one variable can be defined in this way. For functions of two or more variables, such as space-dependent data in 2D and 3D, use a file with the function data.
- *Optional:* Save the parameters to a text file to reuse in other models. Click the **Save to File** button ( ) and enter a **File name**, including the extension .txt. Click to **Save** the text file. The information is saved in space-separated columns in the same order as displayed on screen.
- *Optional:* Load data from a spreadsheet program or other format by saving the file with a .txt extension. Data must be separated by tabs or colons. Click the **Load from File** button ( ) and navigate to the text (.txt) file to load and click **Open**. The data loads to the table. If there is other data already in the table, it adds it after the last row. Move or edit as required.
- *Optional* (for functions defined by a table only): If you want to defined the inverse function f^{-1} for the interpolation function f, select the **Define inverse function** check box and enter a name for the inverse function in the **Inverse function name** field (the default is the name of the interpolation function with a suffix **_inv**).
- If **File** is selected to import function data:

a Enter a **Filename** (the complete network path) or **Browse** to locate a file to import.

b From the **Data format** list select **Spreadsheet**, **Grid**, or **Sectionwise**. The spreadsheet format is the default format, and that format is the easiest to use for functions defined on an unstructured grid or for general tabulated function values with one or more arguments.

- A *spreadsheet* file contains unstructured mesh and function data for space-dependent functions or general input variables and function values for functions of one or more variables; these files can be exported.

```
%Header (optional)
```

```
    Columns containing x, y (optional), and z (optional), or any 
other arguments, followed
    by function data columns.
```

Include function names in the header. In that case, the input columns must be labeled with `x`, `y`, and `z`, respectively, depending on input dimension. For example, a file with the following content creates two 2D functions named `myfun1` and `myfun2`:

```
% x y myfun1 myfun2
0 0 0.12 0.34
0 1 0.52 1.50
1 0 0.67 0.91
```

If the file does not include any header to indicate the function dimension, the software assumes that it is identical to the largest geometry dimension present in the model. A file with four columns, for example, is interpreted as one 3D function in a 3D model, two 2D functions in a 2D model, and three 1D functions in a 1D model.

- A *sectionwise* file has coordinates and function values; a CSV (comma-separated value) file exported from a spreadsheet program is in this format.

```
%Coordinates
One to three columns containing x, y (optional), and 
z (optional)
%Elements
Triangulation where each row contains the row indices of the 
points in the Coordinates section that make up one element 
(triangular in 2D, tetrahedral in 3D)
%Data (funname)
Column of data values for each point
```

It is possible to include more than one function in the file as long as a `%Data` header separates them one from the other.

- A *grid* file

```
% Grid
x grid points separated by spaces
y grid points separated by spaces (optional)
z grid points separated by spaces (optional)
% Data
Data values separated by spaces
```

Each row contains values for different x grid points for fixed values of y and z. The rows first increase the y grid value and then the z grid value. The grid points can also represent another independent variable that the data values depend on.

For example, the “grid points” can be temperature values and the data values the thermal conductivity at these temperatures.

It is possible to include more than one function in the file as long as a `%Data` header separates them one from the other.

Important

It is important to use a comment line starting with `%` to separate the grid points or other interpolation points and the data values that are associated with these coordinates or interpolation points.

c If **Spreadsheet** is selected, enter a **Number of arguments** (1–3). For all file types, enter information about the functions into the table. Add a **Function** name and its **Position in file**. The first function in the file has position 1, the following has position 2, and so on. For spreadsheet data, the first columns contain the arguments; the following columns can contain one or more function, and the positions entered are the relative position for each function’s data column.

Use the **Move Up** (↑), **Move Down** (↓), or **Delete** (✕) buttons to rearrange and delete rows in the table. See also the example below.

For the common case where the data source contains function values that are functions of the space coordinates, select the **Use space coordinates as arguments** check box. Then the function can be called without arguments when used in the model; the space coordinates are added as function arguments automatically.

Note

The Interpolation functions support 1, 2, or 3 arguments. You cannot define functions with more than three (3) arguments because the algorithm creates a mesh for the point cloud, which is not possible in four dimensions or higher.

4 Under **Interpolation and Extrapolation**:

The interpolation setting controls how the program evaluates the function between the discrete points where it is defined by the table or file, and the behavior of the function outside the domain where it is defined by the table or file.

a Select one of the following methods from the **Interpolation** list:

- For functions of one variable select **Nearest neighbor**, **Linear**, **Piecewise cubic**, or **Cubic spline**.

  Piecewise-cubic interpolation uses a piecewise-cubic Hermite polynomial with continuous first derivatives. It preserves the shape of the data and respects monotonicity. The **Cubic-spline** method also performs interpolation with a piecewise cubic polynomial. Here, even second derivatives are continuous; however, the method does not necessarily respect monotonicity.

- For functions of more than one variable, select **Nearest neighbor** or **Linear**. The other options are not supported.

b Select one of the following methods from the **Extrapolation** list to specify how to treat arguments outside the grid or mesh of points.

- **Constant**. Uses the value from the closest point inside the grid (for structured interpolation) or the value from the closest mesh element (for unstructured interpolation). The interpolation function evaluates the polynomial from the closest grid point at the actual point where a value is requested.
- **Linear**. The function is linear outside the grid with a continuous function value and continuous first derivative at the boundary of the grid. **Piecewise cubic** or **Cubic spline** must be selected from the **Interpolation** list in Step 1.
- **Nearest function**. Evaluates the polynomial from the closest grid point at the actual point where a value is requested.
- **Specific value**. Uses a single value, such as zero or NaN (Not a Number), everywhere outside the grid or mesh. Enter the value in the **Values outside range** field.

Note

Unstructured interpolation supports using a constant or a specific value only.

5 Under **Units**, specify units for the function's input and output, if desired.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot. A plot group for an interpolation function contains three connected line graphs: a line graph of the piecewise function and two line graphs using dashed red lines, Left Extrapolation and Right Extrapolation, that show how the selected extrapolation extends the interpolation function on both sides. In addition, a point graph shows the interpolation points (interpolation nodes).

EXAMPLE OF IMPORTING A FILE DATA SOURCE INTO A PARAMETER TABLE

The file named `temp.txt` contains temperature measurements in nine points in the plane:

```
10 3 310
20 3 309
30 3 314
10 6 302
20 6 307
30 6 311
10 9 307
20 9 308
20 9 314
```

The data columns contain *x*-coordinates, *y*-coordinates, and temperature values, respectively. To use this file as an interpolation function called `tempfun`, perform the following steps.

1 Select **File** from the **Data source** list.

2 Enter a **Filename** (the complete network path) or **Browse** to locate a file to import.

3 From the **Data format** list select **Spreadsheet**.

4 Enter a **Number of arguments**. In this example, enter `2`.

5 Enter the **Function** name `tempfun`.

6 Enter its **Position in file** as `1`. The first function in the file has position 1, the following has position 2, and so on.

The position in file for a function is the column after the space coordinates (or other function arguments) where it is defined. In this example with two arguments (space coordinates), the third column is Position 1 in the file.

7 If desired, adjust the interpolation and extrapolation settings in the **Interpolation and Extrapolation** section.

Use the function `tempfun` with `x` and `y` as input arguments in a 2D model to get the interpolated value for the temperature at any position. If the **Use space coordinates as arguments** check box is selected, use `tempfun` without explicitly adding the input arguments.

MATLAB Functions (Global Only)

Important

A COMSOL LiveLink for MATLAB license is required. Also confirm that MATLAB is installed on the same computer as COMSOL Multiphysics.

Use **MATLAB** functions node (f(x)) from COMSOL Multiphysics to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL Multiphysics. MATLAB functions are evaluated in MATLAB.

Note

See the *LiveLink for MATLAB User's Guide* for more information about using MATLAB functions.

Piecewise Functions

A **Piecewise** function ( ) is created by splicing together several functions, each defined on one interval. Define the argument, extrapolation and smoothing methods, and the functions and their intervals. The piecewise function is a function of one variable with different definitions on different intervals, which must not overlap or have any holes between them.

Model

If you have the Heat Transfer Module, see Radiative Heat Transfer in a Utility Boiler: Model Library path **Heat_Transfer_Module/ Process_and_Manufacturing/boiler**

1. Open or create a model file.
2. In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Piecewise** ( ).
3. Enter a **Function name**.

4 Under **Parameters**:

a Enter a piecewise function name in the **Argument** field.

b Select one of the following methods from the **Extrapolation** list to control what happens when the function is evaluated in a point that does not belong to any interval.

- **Constant**. Uses the function value from the endpoint of the closest interval. Uses the value from the start point of the first and the end point of the last interval on the corresponding sides.
- **None**. Evaluation fails outside of the intervals where it is defined. Trying to evaluate the function generates an error and evaluates to NaN (Not-a-Number).
- **Nearest function**. Evaluate the function from the closest interval. The function at the first or last interval (depending of the side) is used to evaluate at the actual point where a value is requested.
- **Specific value**. Enter a value in the **Values outside range** field. If selecting the specific-number method, assign a single value (usually zero or NaN) to all points outside the intervals.

c Select one of the following from the **Smoothing** list: **No smoothing**, **Continuous function**, **Continuous first derivative**, or **Continuous second derivative**.

The functions in contiguous intervals may not evaluate to the same value where the intervals meet. Apply smoothing to make the piecewise function more well-behaved numerically. Functions from neighboring intervals are then blended close to where the intervals meet.

- If **Continuous function**, **Continuous first derivative**, or **Continuous second derivative** is selected, enter a value in the **Relative size of transition zone** field. Relative size meaning in relation to the size of the intervals on both sides of the border.

5 Under **Parameters**, for each cell in the **Intervals** table:

a Enter **Start** and **End** interval limits. The intervals must not overlap, and there cannot be any holes between intervals.

> *Important*
>
> The intervals need to be contiguous.

b Enter an expression defining the **Function**.

6 Under **Units**, specify units for the function’s input and output, if desired.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot. A plot group for a piecewise function contains three connected line graphs: a line graph of the piecewise function and two line graphs using dashed red lines, Left Extrapolation and Right Extrapolation, that show how the selected extrapolation extends the piecewise function on both sides.

Ramp Functions

A **Ramp** function ( ) is a linear increase with a user-defined slope that begins at some specified time. The ramp function is a function of one variable (the time t, for example).

1 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Ramp** ( ).

2 Enter a **Function name**.

3 Under **Parameters**:

   a Enter a **Location** value s_0 for the start of the ramp. The function evaluates to 0 for values less than its start location and increases linearly for values greater than the location.

   b Enter a **Slope** k of the ramp.

   c To ensure that the value never exceeds a certain point, select the **Cutoff** check box and enter a value.

   For an input variable s, a start location s_0, and a slope k, the ramp function’s value is 0 for $s<s_0$ and $k(s-s_0)$ for $s\geq s_0$.

4 Under **Smoothing**:

   a To turn on the smoothing function at the start or the cutoff, select the **Smooth at start** or **Smooth at cutoff** check boxes.

   b Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. Smoothed ramp functions have continuous first and second derivatives.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Random Functions

A **Random** function ( ) generates white noise with uniform or normal distribution and has one or more arguments to simulate white noise. Its distribution is either uniform or normal.

To generate a random function of the space coordinates *x*, *y*, and *z*, for example, use this function with three input variables—it returns the same value each time it is called for a given set of input arguments. To do a Monte Carlo simulation, add one or more additional input arguments that vary using during a parametric sweep, for example.

1. In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Random** ( ).
2. Enter a **Function name**.
3. Under **Parameters**:
   - **a** Enter a **Number of arguments** to the random function.
   - **b** Select **Uniform** or **Normal** from the **Distribution** list.
   - If **Uniform** is selected, enter a **Mean** and **Range**. The range is the difference between the largest and smallest values that the function can return.
   - If **Normal** is selected, enter a **Mean** and **Standard deviation**.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Rectangle Functions

A **Rectangle** function ( ) is 1 in an interval and 0 everywhere else. This function (also called top hat or boxcar) is useful for band-pass filtering; use it to select values within an interval. It can also simulate a signal that is turned on during an interval or a load that is active on a part of a boundary, for example. The rectangle function is a function of one variable (the time *t*, for example).

Model

- Fluid Valve: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**
- If you have the Batteries & Fuel Cells Module, see Soluble Lead-Acid Redox Flow Battery: Model Library path **Batteries_and_Fuel_Cells_Module/Batteries/pb_flow_battery**

1 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Rectangle** ( ).

2 Enter a **Function name**.

3 Under **Limits**, enter a **Lower limit** and **Upper limit** to specify the interval for the rectangle function. For example, if the input argument is time, enter a start and end time. This function evaluates to 1 for values within an interval. Outside the interval it evaluates to 0.

4 Under **Smoothing**:

   a To turn on the smoothing function select the **Size of transition zone** check box.

   b Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. Smoothed rectangle functions have continuous first and second derivatives.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Step Functions

A **Step** function ( ) is a sharp transition from `0` to some other value (amplitude) at some location (a certain time, for example). Create a single step function with a certain amplitude from an initial level to a final level at a start location. The step function is a function of one variable (the time t, for example).

Model	Fluid Valve: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

1 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Step** ( ).

2 Enter a **Function name**.

3 Under **Parameters**:

  a Enter a **Location** (s_0) of the step. The value of the step function is the initial level for input values that are smaller than the location of the step.

  b In the **From** field, enter a start level (L_{start}).

  c In the **To** field, enter a final level (L_{end}).

  For an input variable s, a start location s_0, and initial level L_{start} and a final level L_{end}, the step function's value is L_{start} for $s < s_0$ and L_{end} for $s \geq s_0$. The amplitude of the step is $L_{\text{end}} - L_{\text{start}}$.

4 Under **Smoothing**:

  a To turn on the smoothing function select the **Size of transition zone** check box.

  b Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. Smoothed step functions have two continuous derivatives.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Triangle Functions

A **Triangl**e function (⋀) is a linear increase and linear decline within an interval and 0 everywhere else. Use the triangle function for band-pass filtering; that is, use it to select values within an interval. The triangle function is a function of one variable (the time t, for example).

1 In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Triangle** (⋀).

2 Enter a **Function name**.

3 Under **Limits**, enter a **Lower limit** and **Upper limit** to specify the interval for the triangle function. For example, if the input argument is time, enter a start and end time. In the midpoint of the interval, this function evaluates to 1, and moving toward the interval boundaries it falls off to 0. Outside the interval it evaluates to 0.

4 Under **Smoothing**:

  a To turn on the smoothing function select the **Size of transition zone** check box.

  b Enter a value in the **Size of transition zone** field to control the amount of smoothing. Use smoothing to improve the behavior of the model by avoiding discontinuities that are difficult to handle numerically. Smoothed triangle functions have continuous first and second derivatives.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Waveform Functions

A **Waveform** function ( ) is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. The waveform function is a function of one variable (the time t, for example).

Model	If you have the Batteries & Fuel Cells Module, see 1D Lithium Ion Battery Model for Thermal Models: Model Library path **Batteries_and_Fuel_Cells_Module/Batteries/ li_battery_1d_for_thermal_models**

1. In the **Model Builder**, right-click **Global Definitions** or **Definitions** and select **Functions>Waveform** ( ).
2. Enter a **Function name**.
3. Under **Parameters**, select a waveform from the **Type** list: **Sawtooth**, **Sine**, **Square**, or **Triangle**.
   a. If **Sine** is selected, enter a **Frequency**, **Phase**, and **Amplitude** value. Smoothing is not available for this waveform.
   b. For **Sawtooth**, **Square**, or **Triangle** waveforms, the **Smoothing** check box is selected by default. Enter a value in the **Transition zone** field. Smoothed waveform functions have continuous first and second derivatives.

      If required, turn off smoothing by clearing the **Smoothing** check box.
4. For **Sawtooth**, **Square**, or **Triangle**, enter a **Frequency**, **Phase**, and **Amplitude** value.

Click the **Plot** button ( ) to generate a preview plot of the function. Click the **Create Plot** button ( ) to create a plot group and a plot for the function plot.

Specifying Discontinuous Functions

To specify a discontinuous function, such as a step in space or time, use logical functions that evaluate to 1 if true and 0 otherwise. For instance, the function

```
sin(2*pi*t)*(t<10)
```

defines a sine wave that exists for times until 10 seconds and afterward takes on the value of 0.

If a coefficient or a material property contains a step function or some other discontinuity, convergence problems can arise. For time-dependent problems, the time-stepping algorithm can run into problems. For stationary problems, mesh-resolution issues can arise such as overshooting and undershooting of the solution due to infinite flux problems.

To avoid problems with a discontinuity, replace it with a smoothed step function that emulates steps. Doing so serves two purposes:

- Numerical reliability and convergence are improved.
- What is thought of as a step function is, in reality, a smoothed continuous function because of inertia.

SMOOTHING OF DISCONTINUOUS FUNCTIONS

The easiest way to create a smooth step is to use the predefined Step function that is found under **Global Definitions>Functions** or, for each model, under **Definitions>Functions**. It includes smoothing by default. The Piecewise, Ramp, Rectangle, and Triangle functions also include smoothing (active by default for Rectangle and Triangle functions).

The smoothed Step and Rectangle functions are defined by piecewise 5th-degree polynomials. The smoothed Ramp function is defined by a piecewise 4th-degree polynomial. The smoothed Triangle function is defined by a piecewise 3rd-degree polynomial. None of these functions have any overshoot or undershoot.

ADDITIONAL SMOOTHED FUNCTIONS

In addition, the following smoothed functions are available:

- `flsmhs`, a smoothed step function, or *Heaviside function*, with a continuous first derivative and overshoot on both sides of the step. The overshoot ensures that the integral from 0 to infinity is correct. `y=flsmhs(x,scale)` approximates the logical expression `y = (x>0)` by smoothing the transition within the interval $-\text{scale} < x < \text{scale}$. `fldsmhs` is the derivative of the smoothed Heaviside function.
- `flsmsign`, a smoothed sign function with a continuous first derivative. `y = flsmsign(x,scale)` approximates the function `y = sign(x)` by smoothing the transition within the interval $-\text{scale} < x < \text{scale}$. `fldsmsign` is the derivative of the smoothed sign function.

- `flc1hs`, a smoothed Heaviside function with a continuous first derivative without overshoot. Its syntax is similar to the functions just described.
- `flc2hs`, a smoothed Heaviside function with a continuous second derivative without overshoot. Its syntax is similar to the functions just described.

These functions can be useful as a complement and extension to the predefined Step function.

In the interval `-scale<x<scale`, the functions `flsmhs` and `flsmsign` are defined by a 7th-degree polynomial chosen so that the 2nd derivative is continuous. Moreover, the moments of order 0, 1, and 2 agree with those for the Heaviside function and the sign function, respectively. This implies that the functions have small overshoots.

Now consider an example. Use `flc1hs` to model the heat capacity C_p of a phase-changing material. Assume that a crystalline material has a heat capacity of 1 kJ/kg. Its melting point at the present pressure is 273.15 K. The liquid phase has a heat capacity of 2 kJ/kg. Create a parameter scale set 0.1 and then an **Analytic** node where an analytic function HeatCapacity is defined using the following expression with an argument `T`: `1+flc1hs(T-273,scale)`; then define a plot range of 272.5–273.5 K under **Plot Parameters** and click the **Plot** button in the **Analytic** settings window to plot C_p around the melting point.

Model Couplings

In this section:

About Model Couplings and Coupling Operators

Model couplings establish couplings between different parts of a model or between different models. A model coupling is defined by a *coupling operator*, taking an expression as its argument. When the operator is evaluated at a point in the destination, the value is computed by evaluating the argument on the source. Use a coupling operator to compute several quantities that use the same source and destination. All types of model couplings have a source and a destination.

Note

Scalar coupling operators, such as the Integration coupling operator, have a global destination that does not need to be defined; the values are available globally.

The source is a subset of a single model, while the destination can, depending on the type of coupling, be a subset of one or several models or a value that can be evaluated everywhere. The destination value is computed by evaluating the operator's argument at one or several points in the source. Both the source and the destination are purely geometrical objects, but the source is limited to one geometry whereas the destination is usually global.

Note

With extrusion and projection couplings, the final evaluation fails if the mapping fails, and therefore it is the destination that must be well defined.

To add model couplings to the **Model Builder**, right-click **Definitions** and select a **Model Coupling** from Table 3-11.

ABOUT COUPLING OPERATORS

Coupling operators are useful for modeling coupled problems and are a generalization of ordinary expressions. They are defined by first selecting the source, where the argument of the operator is evaluated, and, in some cases, a destination. An evaluation expression is not required to define coupling operators; instead these operators are used in different contexts, passing the expression to evaluate as a input argument.

There are three categories of coupling operators as listed in Table 3-11:

- *Extrusion*. These operators—General Extrusion, Linear Extrusion, Boundary Similarity, and Identity Mapping—connect a source and a destination and take an expression as an argument. When it is evaluated at a point in the destination, its value is computed by evaluating the argument at a corresponding point in the source.
- *Projection*. These coupling operators—General Projection and Linear Projection—evaluate a series of line or curve integrals on the source domain, where the line or curve positions depend on the positions of the evaluation points in the destination domain.
- *Scalar*. These operators—Integration, Average, Maximum, and Minimum—define a scalar value such as an integration, the average over a set of geometric entities, and the maximum or minimum value of an expression and have a "global destination" (that is, they are available everywhere in the model).

Coupling operators can:

- Make the value and the exact Jacobian of an expression available nonlocally.

- Take information from a boundary, for example, and make it available on other parts of a model (a domain, for example).
- Be used just for results evaluation and visualization purposes.
- Define nonlocal couplings including mesh transformations, integrals over domains and boundaries, and projections.

NONLOCAL COUPLINGS AND THE SPARSITY OF THE JACOBIAN

The Jacobian for problems formulated using the finite element method is usually large but sparse. This is because the solution at each mesh node can depend at most on the degrees of freedom at the neighboring mesh elements. However, by introducing nonlocal model couplings using coupling operators, nonlocal dependencies are created that fill up the rows and columns of the affected source and destination nodes. These additional elements might make the Jacobian matrix only slightly less sparse, with marginal effects on solution speed; it can also make it a great deal less sparse, in which case memory use and CPU time involved in solving the problem increases considerably. For this reason, take particular care when introducing nonlocal couplings.

TABLE 3-11: COUPLING OPERATOR DESCRIPTIONS

LINK TO SECTION	ICON	DESCRIPTION
EXTRUSION MODEL COUPLINGS See About Extrusion Model Coupling and Operators		
Defining a General Extrusion Model Coupling		A General Extrusion model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a general extrusion coupling to define a nonlinear mapping.
Defining a Linear Extrusion Model Coupling		A Linear Extrusion model coupling maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a linear extrusion when the correspondence between evaluation points in the source and destination is linear (and in some to nonlinear cases).
Defining a Boundary Similarity Model Coupling		A Boundary Similarity model coupling maps an expression defined on a part of a boundary to another part of a boundary with the same shape. Available in 2D and 3D, it is defined by a similarity mapping from the destination to the source. Also add Edge Map, One Point Map, or Two Point Map subnodes.

TABLE 3-11: COUPLING OPERATOR DESCRIPTIONS

LINK TO SECTION	ICON	DESCRIPTION
Defining an Identity Mapping Model Coupling		An Identity Mapping model coupling maps between geometric entities which overlap, possibly when viewed in different frames. When it is evaluated at a specific set of coordinates in the destination frame, its argument is evaluated with the same coordinates in the source frame.
PROJECTION MODEL COUPLINGS See About Projection Model Couplings and Operators		
Defining a General Projection Model Coupling		Use a General Projection operator to define integration along curves.
Defining a Linear Projection Model Coupling		Use a Linear Projection model coupling when the argument is to be integrated along a line, and the line depends linearly on the evaluation point.
SCALAR MODEL COUPLINGS See About Scalar Model Couplings and Operators		
Defining an Integration Model Coupling	∫du	An Integration model coupling integrates an expression over selected geometric entities. This is evaluated by integrating the argument over the source domain or summing the argument over the node points in the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.
Defining an Average Model Coupling		An Average model coupling computes the average of an expression over selected geometric entities. It can be evaluated anywhere in any model, and the value does not depend on the evaluation point. It is similar to the Integration operator; the difference being that the integral is divided by the volume, area, arc length, or number of points in the source, depending on the dimension.
Defining Maximum and Minimum Model Couplings		These operators compute the maximum or minimum of an expression over selected geometric entities and give the maximum and minimum values of the arguments over the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

See Also

- Model Couplings
- Sorting Nodes by Space Dimension and Type
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

About Extrusion Model Coupling and Operators

An extrusion coupling operator maps values from the source domain to the destination domain. When the domains are of the same space dimension, it is typically a point-wise mapping. When the destination domain has higher dimension than the source domain, the mapping is done by extruding point-wise values to the higher dimensions. Define the map between the source and destination as one of these operators: general extrusion, linear extrusion, boundary similarity, or identity mapping.

ABOUT GENERAL EXTRUSION COUPLING OPERATORS

A **General Extrusion** coupling operator ( ) maps an expression defined on a source domain to an expression that can be evaluated on any domain where the destination map expressions are valid. Compared to the *linear extrusion*, these operators define a more general, possibly nonlinear, relation between source and destination domains. Specifically, when the destination has more space dimensions than the source, the operator performs extrusion of values. See Defining a General Extrusion Model Coupling.

ABOUT LINEAR EXTRUSION COUPLING OPERATORS

A **Linear Extrusion** coupling operator ( ) maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use this to define a linear mapping of this kind. Linear extrusion can be used when the correspondence between evaluation points in the source and destination is linear, and in some nonlinear cases. Otherwise, use a general extrusion coupling.

These operators define a *linear extrusion* that maps between geometric parts of the same dimension. The parts can exist in geometries of different space dimensions. For example, couple from edges in 2D to those in 3D; or couple 2D domains to 3D faces. In these cases geometries of different space dimensions are needed for the source and destination. Define the linear extrusion by specifying points in both the source and destination. See Defining a Linear Extrusion Model Coupling.

ABOUT BOUNDARY SIMILARITY COUPLING OPERATORS

The **Boundary Similarity** coupling operator ( ) is slightly different for 2D and 3D.

In 3D, the destination map is a similarity that maps a destination boundary, onto a set of source boundaries. The mesh is always viewed in the mesh frame. See Defining a Boundary Similarity Model Coupling.

By default, the algorithm automatically chooses a map when symmetries make several maps possible. To control this choice in 3D, add an **Edge Map**, **One Point Map**, or **Two Point Map** subnode ( ). See Defining One-Point, Two-Point, or Edge Maps (3D).

- *Edge Map:* Specify that a certain destination edge should be mapped onto a certain source edge. Their relative direction is given by the property direction. The edges must be adjacent to the given boundary.
- *One-Point Map:* Specify that a certain destination vertex should be mapped onto a certain source vertex.
- *Two-Point Map:* Specify that two destination vertices should be mapped onto two source vertices.

In 2D, it works the same except the destination map is a similarity that maps a destination edge onto a set of source edges and there are no subnodes to add.

ABOUT IDENTITY MAPPING COUPLING OPERATORS

An **Identity Mapping** coupling operator maps between geometric entities which overlap, possibly when viewed in different frames. When it is evaluated at a specific set of coordinates in the destination frame, its argument is evaluated with the same coordinates in the source frame.

ABOUT SOURCE AND DESTINATION MAPPINGS

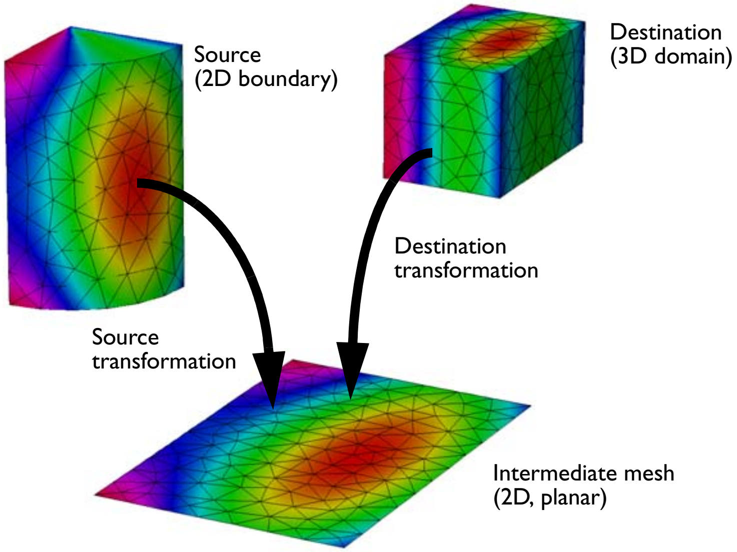


Figure 3-1: General extrusion mappings.

The definition of any extrusion model coupling involves two mesh maps, which are important to understand. The *source map* is a one-to-one mapping that maps the mesh of the physical source of dimension srcedim to an *intermediate mesh* of the same dimension embedded in a space of dimension idim ≥ srcedim. The *destination map* is a mapping from the destination of dimension dstedim, where the operator can be evaluated, to the same space that contains the intermediate mesh.

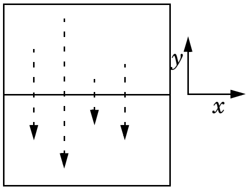
When the value of the coupling operator is requested somewhere in the destination, the software transforms the destination points using the destination map. It compares the resulting coordinates to the elements in the intermediate mesh to find corresponding locations in the physical source. This means that the source map must be inverted but not the destination map. The latter can in fact be non invertible, which is, for example, the case when dstedim > idim, leading to an extrusion.

To avoid the need to solve a nonlinear system of equations for every destination point, the software assumes that the source map is linear on each element of the intermediate mesh. In practice, the map is often trivial and leaves the coordinates unchanged, but it can also rescale, stretch, bend, or reflect the mesh.

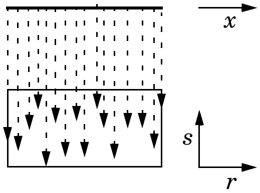
EXAMPLES OF EXTRUSION MODEL COUPLING

Note

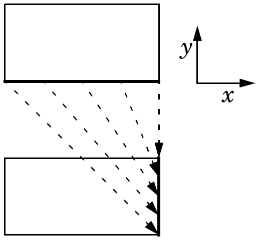
All the graphics in these examples use **General Extrusion** model coupling.



One application of a **General Extrusion** coupling is to mirror the solution on the x-axis. This can be useful for analysis. Both source and destination are two-dimensional, as well as the intermediate mesh (srcedim = idim = dstedim). The source map to enter is `x, y`, and the destination map is `x, -y`. This can also be done with **Linear Extrusion**.



Another **General Extrusion** example is to extrude the solution in the 1D geometry to a 2D domain along the s-axis. The source map is `x`, and the destination map is `r`, so here srcedim = idim = 1, dstedim = 2.

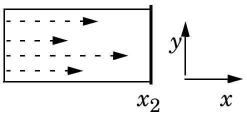


Another example maps values on the lower boundary of a rectangle that extends from $x = -1$ to $x = 1$ and from $y = 0$ to $y = 1$, to the right boundary on the same rectangle. The source map is `(x+1)/2` and the destination map is `y`. Both maps have a single component since srcedim = idim = dstedim. This map is also linear and can be done with **General** or **Linear Extrusion**, or with **Boundary Similarity**.

Extrusion Model Coupling—Example 2

Consider the case of a single rectangular domain where the source term in Poisson's equation comes from the inward flux over the right boundary for the corresponding y coordinate.

$$\begin{aligned} -\Delta u &= \frac{\partial}{\partial n}u(x_2, y) && \text{on } \Omega \\ u &= xy && \text{on } \partial\Omega \end{aligned}$$



The figure to the left illustrates the extrusion process. The values of the influx on the boundary become available throughout the domain by extrusion along the *y*-axis. The source map is y, and the destination map is y.

About Projection Model Couplings and Operators

The two types of *projection coupling operators*—**General Projection** and **Linear Projection** ( )—evaluate a series of line integrals on the source domain, where the line positions depend on the positions of the evaluation points in the destination domain. In this way it is possible to compute the integral of an expression over one space variable for a range of different points along the other space axis, giving a result that varies over the latter space variable. For example, obtain the average along the *y* direction of a variable *u* defined on some two-dimensional domain in the *xy*-plane by calculating the integral

$$\bar{u}(x) = \int u(x, y)\, dy$$

COMSOL Multiphysics uses a method whereby it first applies a one-to-one map to the mesh of the source domain. The last space dimension in the transformed mesh is the one integrated over, so the lines used to integrate are vertical in the transformed source mesh. The software takes the placement of the vertical lines in the transformed source mesh from the positions of the transformed destination evaluation points. It then carries out the integrals in the source domain over curves that correspond to the vertical lines in the transformed source mesh.

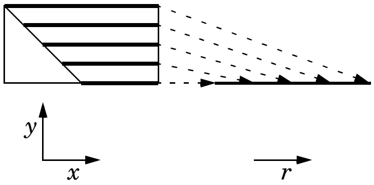
Define the map between source and destination in two ways: as a *linear projection* or as a *general projection*.

EXAMPLES OF PROJECTION COUPLINGS

Note

All the graphics in these examples use **General Projection** model coupling. These can also be done using **Linear Projection**.

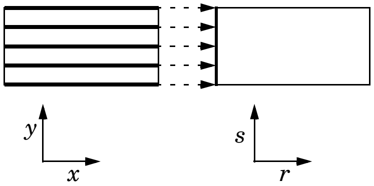
Projection Coupling Operators—Example 1



For each point r, the coupling operator returns the integral

$$v(r) = \int_{\substack{y = r/2 \\ (x, y) \in S_2}} u(x, y)dx$$

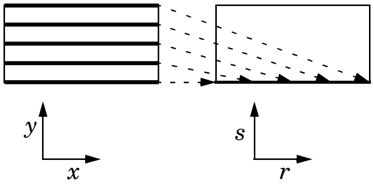
The source map is `y`, `x`, and the destination map is `r/2`.



For each point $(0, s)$, the coupling operator returns the integral

$$v(0, s) = \int_{\substack{y = s \\ (x, y) \in S_2}} u(x, y)dx$$

The source map is `y`, `x`, and the destination map is `s`.



For each point $(r, 0)$, the coupling operator returns the integral

$$v(r, 0) = \int_{\substack{y = r/2 \\ (x, y) \in S_2}} u(x, y)dx$$

The source map is `y`, `x`, and the destination map is `r/2`.

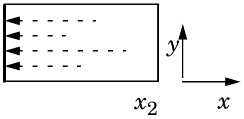
Note

The integration can also sweep nonrectangular domains. The integrals include only the source domains; they exclude other domains and the external area.

Projection Coupling Operators—Example 2

Consider the case of a single rectangular domain with Poisson's equation. Integrate the solution squared along lines parallel to the x-axis and make the result available for analysis on the left boundary.

$$\begin{aligned} -\Delta u &= 1 \quad \text{on } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$



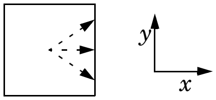
The figure illustrates the projection process. Project the integral of the solution squared on the boundary. The source map is `y`, `x` and the destination map is `y`. If the projection operator is called `genproj1`, the desired result is obtained by evaluating `genproj1(u^2)`.

About Scalar Model Couplings and Operators

ABOUT INTEGRATION MODEL COUPLINGS AND OPERATORS

An integration coupling operator is the value of an integral of an expression over a set of geometric entities (domains, for example). Thus, the integration coupling operator is always a scalar quantity. The operator can compute an expression at a vertex (point) or the integral of an expression over one or several domains, boundaries, or edges. Integration coupling operators have global destination, so they can be evaluated anywhere in the model. Because it is an operator, define one integration operator (`intop1`, for example) for a part of the geometry (a boundary, for example) and then use that several times in the model to compute integrals over that boundary for different integrands. For example, `intop1(T)` is the integral of the temperature T over the boundary, and `intop1(1)` is simply the length (2D) or area (3D) of the boundary. Also, using the `dest` operator it is possible to create convolution integrals.

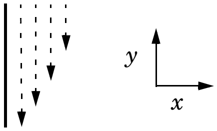
Examples of the Use of Integration Coupling Operators



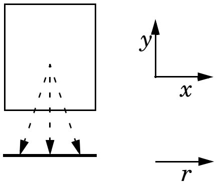
Consider Poisson's equation on a rectangular domain. The integral of the solution squared serves as the influx in a Neumann boundary condition on the right boundary. There is a Dirichlet boundary condition on the left boundary, and the top and bottom boundaries have zero influx.

For example, define an integration coupling operator called `intop1`, with the rectangular domain as source. The influx for the Neumann boundary condition is obtained as `intop1(u^2)`.

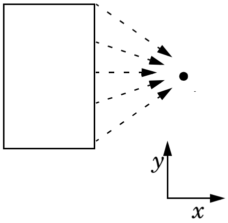
$$\begin{array}{ll}
-\Delta u = 1 & \text{on } \Omega \\
u = x & \text{on } \partial\Omega_1 \\
\dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_{2,3} \\
\dfrac{\partial u}{\partial n} = -\displaystyle\int_{\Omega} u^2 d\Omega & \text{on } \partial\Omega_4
\end{array}$$



A second example is when a scalar value from a vertex is used everywhere on a boundary to which the vertex belongs. In structural mechanics use this type of coupling to formulate displacement constraints along a boundary in terms of the displacements of the end point. In electromagnetics the same technique can implement *floating contacts.*



Another example is to use the integral over a domain in a 2D geometry along a domain in another 1D geometry. This approach is helpful for process-industry models where two processes interact.



Integration coupling operators can implement *integral constraints.* First define a coupling operator at some vertex in such a way that it represents the value of the integral to be constrained. Then use a point constraint to set the coupling operator, and thereby the integral, to the desired value.

Integration coupling operators (as well as the average, maximum, and minimum coupling operators are also useful for evaluating, for example, integrated quantities. To evaluate, for example, the total current across a boundary in a 2D Electric Currents model, define an integration coupling operators such as `intop1` with a source on the boundary where the current flows. Then the value of `intop1(ec.normJ*ec.d)`, where `normJ` is the current density norm (SI unit: A/m^2) and `d` is the thickness of the 2D geometry (SI unit: m), is the total current flowing across that boundary (SI unit: A).

ABOUT AVERAGE MODEL COUPLINGS

Computes the average of an expression over selected geometric entities. It can be evaluated anywhere in any model, and the value does not depend on the evaluation point. It is similar to the Integration operator; the difference being that the integral is divided by the volume, area, arc length, or number of points in the source, depending on the dimension.

ABOUT MAXIMUM AND MINIMUM MODEL COUPLINGS

These operators compute the maximum or minimum of an expression over selected geometric entities and give the maximum and minimum values of the arguments over the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point. Two arguments can be given, and the returned value is the value of the second argument evaluated in the max/min of the first argument. This is useful for evaluating, for example, the location of the maximum or minimum. In a 2D model where the temperature T is solved for, use the following syntax for the maximum operator `maxop1` in a **Global Evaluation** node, for example, to get the *x*- and *y*-coordinate for the maximum of the temperature: `maxop1(T,x)`and `maxop1(T,y)`.

Note

When a **Max/Min Volume**, **Max/Min Surface**, or **Max/Min Line** plot is used, the maximum and minimum values, along with the coordinates for the corresponding locations, appear in a table (underneath the plot with the default COMSOL Desktop layout).

Defining a General Extrusion Model Coupling

A **General Extrusion** model coupling ( ) maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a General Extrusion coupling to define a nonlinear mapping.

Model

If you have the:

- Acoustics Module, see Flow Duct: Model Library path **Acoustics_Module/Industrial_Models/flow_duct**
- Chemical Reaction Engineering Module, see Packed Bed Reactor: Model Library path **Chemical_Reaction_Engineering_Module/Packed_Bed_Reactors/packed_bed_reactor**
- Geomechanics Module, see Concrete Beam With Reinforcement Bars: Model Library path **Geomechanics_Module/Tutorial_Models/concrete_beam**
- RF Module, see Radar Cross Section: Model Library path **RF_Module/Tutorial_Models/radar_cross_section**
- Subsurface Flow Module, see Aquifer Characterization: Model Library path **Subsurface_Flow_Module/Fluid_Flow/aquifer_characterization**

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>General Extrusion** ( ).

2 Enter an **Operator name** or use the default.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities in the **Graphics** window.
- Select **All domains**, for example, to add all applicable geometry entities to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required on the settings window.

5 Under **Destination Map**, specify the general extrusion destination map by entering expression in the **x-expression**, **y-expression**, and **z-expression** fields.

This maps each point in the destination to a point in the intermediate mesh, where the argument of the extrusion operator is evaluated. A general extrusion operator can be evaluated at any point where the destination map expressions are defined.

6 Under **Source**:

Note

The number of destination map expressions is the same as the space dimension of the intermediate mesh. For example, if the intermediate mesh is in two-dimensional space, there is no **z-expression** field.

a Select a **Source frame** to use on the source domain. In most cases the **Source** section's default settings can be used.

b Optionally, select the **Use source map** check box and enter expressions in the **x-expression**, **y-expression**, and **z-expression** fields for the source map from the source to the intermediate mesh. The number of source map expressions is the same as the number of destination map expressions. With the default source map expressions, the intermediate mesh can be considered identical to the source.

The dimensionality idim of the intermediate space is determined by the number of non-empty source and destination map expressions, which must be the same, and must also satisfy srcedim ≤ idim ≤ srcsdim, where srcedim is the dimension of the source selection, and srcsdim is the dimension of the source geometry.

7 Under **Advanced**:

a Select an option from the **Mesh search method** list to specify what should happen if an evaluation point in the destination is mapped to a point outside the source.

- If **Use tolerance** is selected (the default setting) the result depends on the other field definitions in this section.
- If **Closest point** is selected, the closest point in the source selection is used.

b Enter a value in the **Extrapolation tolerance** field. If the mapped point is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be in the source. Otherwise, the mapping fails.

c Select the **Use NaN when mapping fails** check box to evaluate the operator to NaN (Not-a-Number) when the mapping fails. Otherwise an error occurs.

Defining a Linear Extrusion Model Coupling

A **Linear Extrusion** model coupling ( ) maps an expression defined on a source domain to an expression that can be evaluated on the destination domain. Use a linear extrusion when the correspondence between evaluation points in the source and destination is linear (and in some to nonlinear cases).

Model

If you have the:

- Acoustics Module, see Porous Absorber: Model Library path **Acoustics_Module/Industrial_Models/porous_absorber**
- CFD Module, see Turbulent Flow Over a Backward Facing Step: Model Library path **CFD_Module/Single-Phase_Benchmarks/ turbulent_backstep**
- Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: Model Library path **Heat_Transfer_Module/Verification_Models/ turbulent_backstep**
- Structural Mechanics Module, see Vibrations of an Impeller: Model Library path **Structural_Mechanics_Module/Tutorial_Models/impeller**

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Linear Extrusion** ( ).

2 Enter an **Operator name** or use the default.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities in the **Graphics** window.
- Select **All domains**, for example, to add all applicable geometry entities to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.

5 Under **Source**:

a Select a **Source frame** to use on the source domain. In most cases the **Source** section default settings can be used.

b Optionally, select the **Use source map** check box and enter expressions in the **x-expression**, **y-expression**, and **z-expression** fields for the source map from the source to the intermediate mesh. With the default source map expressions, the intermediate mesh can be considered identical to the source.

6 Under **Source Vertices**, as required, define a single point (vertex) for **Source vertex 1**, **Source vertex 2**, **Source vertex 3**, and **Source vertex 4**.

Note

The selection of **Source Vertices** and **Destination Vertices** together define the linear mapping from the destination to the source.

- The number of source vertices entered is be the same as the number of destination vertices. This number must be at least one, and not more than 1 + min(srcsdim,dstsdim) where srcsdim and dstsdim are the dimensions of the source and destination geometries, respectively.
- If not all source vertices selections are used, the empty selections must be last.
- Click the **Activate Selection** button (⏻) to define the source vertices. Use the **Add to Selection** (➕), and **Clear Selection** (🧹) buttons as required.

Select and add one single vertex (point) for each source vertex in the **Graphics** window or from the **Selection List**.

7 Under **Destination** define the mapping from destination to source.

a Select an option from the **Destination geometry** list if there is more than one geometry in the model.

- A linear extrusion operator can only be evaluated on the destination geometry and the destination vertices must be chosen in the destination geometry.

b Select an option from the **Destination frame** to evaluate the destination vertex coordinates.

8 Under **Destination Vertices**, as required, select a single destination vertex for each of **Destination vertex 1**, **Destination vertex 2**, **Destination vertex 3**, and **Destination vertex 4**. The number of destination vertices should be the same as the number of source vertices. If not all destination vertex selections are used, the empty selections must be last. Click the **Activate Selection** button (⏻) to define the source vertices. Select and add one single vertex (point) for each destination vertex in the **Graphics** window or from the **Selection List**.

9 Under **Advanced**:

a Select an option from the **Mesh search method** list to specify what should happen if an evaluation point in the destination is mapped to a point outside the source.

- If **Use tolerance** is selected (the default) the result depends on the other field definitions in this section.
- If **Closest point** is selected, the closest point in the source selection is used.

b Enter a value in the **Extrapolation tolerance**. If the mapped point is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be in the source. Otherwise, the mapping fails.

c Select the **Use NaN when mapping fails** check box to evaluate the operator to NaN (Not-a-Number) if the mapping fails. Otherwise an error occurs.

Defining a Boundary Similarity Model Coupling

A **Boundary Similarity** model coupling ( ) maps an expression defined on a part of a boundary to another part of a boundary with the same shape. Available in 2D and 3D, it is defined by a similarity mapping from the destination to the source.

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Boundary Similarity** ( ).

2 Enter an **Operator name** or use the default.

3 Under **Source Boundaries**, select **Manual** or **All boundaries** from the **Selection** list to define the source selection of a boundary similarity operator.

- If **Manual** is selected, select geometric entities in the **Graphics** window.
- Select **All boundaries** to add all boundaries in the geometry to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** (🧹) buttons as required.

4 Under **Destination Boundary**, select a single boundary as the destination. There can only be one destination boundary, which you select in the **Graphics** window or from the **Selection List**.

Click the **Activate Selection** button (⏻) to define the destination boundary. Use the **Add to Selection** (+), and **Clear Selection** (🧹) buttons as required.

5 Under **Advanced**:

2D

For 2D models, select a relative **Direction** of the source and destination edges—**Automatic orientation** (the default), **Same orientation**, or **Opposite orientation**. This specifies the relative direction of the source and boundary edges and specifies which one of the two possible similarity mappings between the source and destination edge to use.

3D

For 3D models, a similar direction functionality is provided by the subnodes **One Point Map**, **Two-Point Map**, and **Edge Map** to exactly specify the similarity mapping between the source and destination when more than one possibility exists.

a Select the **Use source map** check box to have a nonlinear correspondence between the source and destination. The source map is specified by entering expressions in the **x-expression**, **y-expression**, and **z-expression** fields.

b Select an option from the **Mesh search method** list to specify what should happen if an evaluation point in the destination is mapped to a point outside the source.

- If **Use tolerance** is selected (the default) the result depends on the other field definitions in this section.
- If **Closest point** is selected, the closest point in the source selection is used.

c Enter a value in the **Extrapolation tolerance** field. If the mapped point is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be in the source. Otherwise, the mapping fails.

d Select the **Use NaN when mapping fails** check box to evaluate the operator to NaN (Not-a-Number) if the mapping fails. Otherwise an error occurs.

Defining One-Point, Two-Point, or Edge Maps (3D)

Important

Only one map feature is allowed per boundary similarity coupling and only one source or destination point per field is allowed.

In the **Model Builder**, right-click the **Boundary similarity** node and select **One Point Map**, **Two-Point Map**, or **Edge Map** ( ). Then follow these instructions:

ONE-POINT MAPS

1 Under **Points**, click the **Activate Selection** button (⏻) to select a single **Point on source** adjacent to the source selection. Click in the **Graphics** window and select the point, or use the **Selection List**.

2 Click the **Activate Selection** button (⏻) to select a single **Point on destination** adjacent to the destination selection. Click in the **Graphics** window and select the point, which is mapped to the selected source point by the similarity mapping from destination to source.

Use the **Add to Selection** (➕) and **Clear Selection** (🧹) buttons as required.

TWO-POINT MAPS

1 Under **Source Points**, click the **Activate Selection** button (⏻) to select a single **First point on source**. Click in the **Graphics** window and select the point, or use the **Selection List**.

2 Click the **Activate Selection** button (⏻) to select a single **Second point on source** adjacent to the source selection. Click in the **Graphics** window and select the point, or use the **Selection List**.

3 Under **Destination Points**, click the **Activate Selection** button (⏻) to select a single **First point on destination**. Click in the **Graphics** window and select the point, or use the **Selection List**.

4 Click the **Activate Selection** button (⏻) to select a single **Second point on destination** adjacent to the destination selection. Click in the **Graphics** window and select the point, or use the **Selection List**. The destination points are mapped to the corresponding points on the source by the similarity mapping.

Use the **Add to Selection** (➕) and **Clear Selection** (🧹) buttons as required.

EDGE MAPS

1 Under **Edges**, click the **Activate Selection** button (⏻) to select a single **Source edge**. Click in the **Graphics** window and select the edge, or use the **Selection List**.

2 Click the **Activate Selection** button (⏻) to select a single **Destination edge**. Click in the **Graphics** window and select the edge, or use the **Selection List**. The destination edge is mapped to the source edge by the similarity mapping from destination to source.

Use the **Add to Selection** (➕) and **Clear Selection** (🧹) buttons as required.

3 Under **Advanced**, select a relative **Direction** of the source and destination edges—**Automatic**, **Same**, or **Opposite orientation**. This specifies the relative direction of the

source and boundary edges and specifies which one of the two possible similarity mappings between the source and destination edge to use.

Defining an Identity Mapping Model Coupling

An **Identity Mapping** model coupling ( ) maps between geometric entities that overlap, possibly when viewed in different frames. When it is evaluated at a specific set of coordinates in the destination frame, its argument is evaluated with the same coordinates in the source frame.

Model	If you have the CFD Module, see Solar Panel in Periodic Flow: Model Library path **CFD_Module/Single-Phase_Tutorials/solar_panel**

1. In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Identity Mapping** ( ).
2. Enter an **Operator name** or use the default.
3. Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window.
4. Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.
   - If **Manual** is selected, select geometric entities in the **Graphics** window.
   - Select **All domains**, for example, to add all applicable geometry entities to the **Selection** list.

   Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.
5. Under **Frames** select a **Source frame** to use on the source geometric entity and a **Destination frame** to use on the destination geometric entity. In most cases the default **Spatial frame** can be used.

6 Under **Advanced**:

a Select an option from the **Mesh search method** list to specify what should happen if an evaluation point in the destination is mapped to a point outside the source.

- If **Use tolerance** is selected (the default) the result depends on the other field definitions in this section.
- If **Closest point** is selected, the closest point in the source selection is used.

b Enter a value in the **Extrapolation tolerance**. If the mapped point is within a distance of the extrapolation tolerance times the mesh element size, the point is considered to be in the source. Otherwise, the mapping fails.

c Select the **Use NaN when mapping fails** check box to evaluate the operator to NaN (Not-a-Number) if the mapping fails. Otherwise an error occurs.

Defining a General Projection Model Coupling

A projection coupling operator evaluates an expression defined on a source domain by integration along lines or curves depending on the evaluation point in the destination domain. Use a **General Projection** model coupling ( ) to define integration along curves.

Note

It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.

ABOUT THE GENERAL SOURCE AND DESTINATION MAP VERTICES

The **General Projection** operator is defined by mapping the source to an abstract intermediate space of dimension $\mathrm{srcedim}$, and the destination to the subspace of dimension $\mathrm{srcedim}-1$ obtained by setting the last coordinate to 0. Here $\mathrm{srcedim}$ is the dimension of the source selection. To every point in the destination, there corresponds a vertical line in the intermediate space, obtained by allowing the last coordinate to vary while the remaining coordinates are given by the destination map. The set of points in the source selection which are mapped onto this line by the source map, is a line or curve, and the projection operator is evaluated by integrating along this line or curve.

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>General Projection** ( ).

2 Enter an **Operator name** or use the default.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities in the **Graphics** window.
- Select **All domains**, for example, to add all applicable geometric entities to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** (🧹) buttons as required on the settings window.

5 Under **Source Map**, specify the general projection source map by entering expressions in the **x-expression**, **y-expression**, and **z-expression** fields.

Note

The dimension of the intermediate space equals the dimension srcedim of the source selection. If the selection has lower dimension than the source geometry, specify only the first srcedim expressions.

Use expressions containing space coordinates in the source geometry when defining the map. Remember that the map must be approximately linear within each mesh element.

6 Select a **Source frame** from the list.

7 Under **Destination Map**, enter an **x-expression** and, depending on dimensions, **y-expression** for each coordinate except the last in the intermediate space.

The destination map has one field less than the source map. When defining the map it is permissible to use expressions containing space coordinates in the destination geometry. The destination mapping can be highly nonlinear or non invertible.

- If the selection has lower dimension than the source geometry, specify only the first srcedim−1 expressions. A general projection operator can be evaluated at any point where its destination map is defined.
- If the source selection has dimension 1, no destination map needs to be specified, and consequently this section is not shown if the source geometry is 1D. In this case, it is probably better to use an **Integration coupling**.

8 Under **Advanced**, enter an **Integration order** of the numerical integration method.

Defining a Linear Projection Model Coupling

Use a **Linear Projection** model coupling ( ) when the argument is to be integrated along a line, and the line depends linearly on the evaluation point.

The linear projection maps between source domains and destination domains of the nearest lower dimension. The domains can exist in geometries of different space dimensions. For example, couple from domains in 2D to edges in 3D; or couple 3D domains to 2D domains. Define the linear projection by specifying points in both the source and destination domains.

Note

It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.

ABOUT THE LINEAR SOURCE AND DESTINATION MAP VERTICES

The **Source Vertices** and **Destination Vertices** together define a linear mapping from the destination to the source. An evaluation point in the destination geometry is first orthogonally projected onto the linear space spanned by the destination vertices (unless they span the entire space). The projected point is then mapped to the source geometry by a linear mapping taking each destination vertex to the corresponding source vertex. Let L be the line through this point which is parallel to a line through the first and the last source vertex.

If the source selection lies in the linear space spanned by the source vertices, the Linear Projection operator is evaluated by integrating along the line L. In general the operator is evaluated by integrating along the line or curve in the source selection which is mapped to L under orthogonal projection onto the linear space spanned by the source vertices.

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Linear Projection** ( ).
2 Enter an **Operator name** or use the default.
3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities in the **Graphics** window.
- Select **All domains**, for example, to add all applicable geometry entities to the **Selection** list.

Use the **Add to Selection** (➕), **Remove from Selection** (➖), and **Clear Selection** (🧹) buttons as required on the settings window.

5 Under **Source**, select a **Source frame** from the list to evaluate the coordinates of the source vertices in the selected frame.

Now specify the linear projection by giving a set of points in the source geometry and in the destination geometry. The order of the vertices is significant. COMSOL Multiphysics constructs a linear projection from the source domain to the destination domain using the subspaces spanned by the vertices. Denote the map rank by n, denote the source vertices by $x_0, x_1,..., x_n$, and denote the destination vertices $x'_0, x'_1,\ldots, x'_n$. After padding the source and destination vertices' vectors with zeros as necessary, the software solves the following matrix equation for a *transformation matrix* T and a *translation vector* V:

$$
\begin{aligned}
x'_0 &= Tx_0 + V \\
x'_1 - x'_0 &= T(x_1 - x_0) \\
&\ldots \\
x'_n - x'_0 &= T(x_n - x_0)
\end{aligned}
$$

For the projection model coupling there must be one more vertex in the source geometry than in the destination geometry.

6 Under **Source Vertices**, as required, select a single source vertex for each of **Source vertex 1**, **Source vertex 2**, **Source vertex 3**, and **Source vertex 4**.

Note

The selection of **Source Vertices** and **Destination Vertices** together define the linear mapping from the destination to the source.

- Select srcedim + 1 source vertices where srcedim is the dimension of the source selection. Depending on the dimension of the source selection, it may be that some of the last source vertex selections should be left empty.
- If not all source vertices selections are used, the empty selections must be last.
- Click the **Activate Selection** button (⏻) to define the source vertices. Use the **Add to Selection** (✚) and **Clear Selection** (🧹) buttons as required.

Select and add one single vertex (point) for each source vertex in the **Graphics** window or from the **Selection List**.

7 Under **Destination**, select an option from the **Destination geometry** list if there is more than one geometry in the model.

- A linear projection operator can only be evaluated on the destination geometry and the destination vertices must be chosen in the destination geometry.
- The destination vertex coordinates are evaluated in the selected **Destination frame**.

8 Under **Destination Vertices**, as required, select a single destination vertex for each of **Destination vertex 1**, **Destination vertex 2**, and **Destination vertex 3**. The number of destination vertices should be one less than the number of source vertices. If not all destination vertex selections are used, the empty selections must be last.

- Select srcedim destination vertices where srcedim is the dimension of the source selection. Depending on the dimension of the source selection, it may be that some of the last destination vertex selections should be left empty.
- Click the **Activate Selection** button (⏻) to define the source vertices. Use the **Add to Selection** (✚) and **Clear Selection** (🧹) buttons as required.

Select and add one single vertex (point) for each destination vertex in the **Graphics** window or from the **Selection List**.

9 Under **Advanced** enter an **Integration order** of the numerical integration method.

Defining an Integration Model Coupling

An **Integration** model coupling (∫du) integrates an expression over selected geometric entities. This is evaluated by integrating the argument over the source domain or summing the argument over the node points in the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

Model

- Acoustics of a Muffler: Model Library path: **COMSOL_Multiphysics/ Acoustics/automotive_muffler**
- Fluid Valve: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/ fluid_valve**

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Integration** (∫du).

2 Enter an **Operator name** or use the default. This is the name that is used to access the operator in the model, so use a name that describes the integration operator.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window. The source selection defines the part of the geometry that the integration operator integrates over.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities on the **Graphics** window.
- Select **All domains**, for example, to add all the applicable geometric entities to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required on the settings window.

5 Under **Advanced**, from the **Method** list select **Integration** or **Summation over nodes**. In most cases use integration. Summation over nodes is useful, for example, for calculating reaction forces. If **Integration** is selected, enter a value in the **Integration order** field. Also, when working with multiple frames, select a **Frame** from the list for the volume element to be used in the integration. For axisymmetric geometries, select the **Compute integral in revolved geometry** check box to perform the integration in 3D (for a 2D axisymmetric model) or in 2D (for a 1D axisymmetric model).

Defining an Average Model Coupling

An **Average** model coupling ( ) computes the average of an expression over selected geometric entities. It can be evaluated anywhere in any model, and the value does not depend on the evaluation point. It is similar to the Integration operator; the difference being that the integral is divided by the volume, area, arc length, or number of points in the source, depending on the dimension.

Model

Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**

---

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Average** ( ).

2 Enter an **Operator name** or use the default. This is the name that is used to access the operator in the model, so use a name that describes the average operator.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window. The source selection defines the part of the geometry that the average operator computes the average for.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

   - If **Manual** is selected, select geometric entities on the **Graphics** window.
   - Select **All domains**, for example, to add all applicable geometric entities to the **Selection** list.

   Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required on the settings window.

5 Under **Advanced**:

   a Enter an **Integration order**.

   b When working with multiple frames, select a **Frame** from the list for the volume element to be used in the average.

The **Maximum** ( ) and **Minimum** ( ) coupling operators compute the maximum or minimum of an expression over selected geometric entities and gives the maximum or minimum value of the expression in the argument over the source domain. The operator can be evaluated anywhere in any model, and the value does not depend on the evaluation point.

Model

Deformation of a Feeder Clamp: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp**

Defining Maximum and Minimum Model Couplings

1 In the **Model Builder**, right-click **Definitions** and select **Model Couplings>Maximum** ( ) or **Minimum** ( ).

2 Enter an **Operator name** or use the default (`maxop1`, for example). This is the name that is used to access the operator in the model, so use a name that describes the maximum or minimum operator.

3 Under **Source Selection**, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list turns on the button with the same name on the **Graphics** window. The source selection defines the part of the geometry that the maximum or minimum operator computes the maximum or minimum within.

4 Select **Manual** or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list.

- If **Manual** is selected, select geometric entities on the **Graphics** window.
- Select **All domains**, for example, to add all applicable geometric entities to the **Selection** list.

Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required on the settings window.

5 Under **Advanced**, the **Point type** controls the choice of evaluation points—the result is more accurate with more points but more points also means a slower evaluation.

a From the **Point type** list, select **Node points** (the default), **Integration points**, or **Lagrange points** to compute the maximum or minimum by evaluating the argument expression at a finite set of points in the source and taking the maximum or minimum of these values.

b If **Integration points** is selected, enter an **Integration order**. The default is `4`.

c If **Lagrange points** is selected, enter a **Lagrange order**. The default is `2`.

Coordinate Systems

In this section:

About Coordinate Systems

COMSOL Multiphysics uses a global Cartesian coordinate system by default to specify material properties, loads and constraints in all physics interfaces and on all geometric entity levels (points, edges, boundaries, and domains). In boundary conditions and fluid domains, the global system is generally interpreted as having fixed axis directions in space, that is, it is a *spatial frame* system. When specifying properties of solid materials, the global system axes are instead fixed in the material. In other words, it is a *material frame* system in that context.

Not only the global coordinate system, but also systems defined as a rotation relative to the global system, are context-dependent in this way. Such systems are collectively referred to as *relative coordinate systems*, to distinguish them from *absolute coordinate systems*.

2D

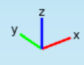
3D

For 2D and 3D models, the **Boundary System** node is automatically added under **Definitions**.

To add another coordinate system to the **Model Builder**, right-click **Definitions**, and select a **Coordinate Systems** from Table 3-12.

The spatial Cartesian coordinate system coordinates default to the following names in 2D and 3D (in 2D axisymmetric geometries, COMSOL Multiphysics uses cylindrical coordinates):

GEOMETRY	DEFAULT NAME OF SPACE COORDINATES
2D	*x y*
3D	*x y z*
Axial symmetry 2D	*r* φ *z*

In 3D, an image displays in the lower-left corner of the **Graphics** window  to indicate the orientation of this coordinate system.

User-defined coordinate systems can be used on all geometric entity levels to simplify the modeling process. In some of the physics interfaces, these coordinate systems can be used to define orthotropic and anisotropic material properties that are not aligned with the global Cartesian coordinate system, for example. See Table 3-12 for an overview of the available coordinate systems. Note in particular that some coordinate systems specify absolute directions in space, while others specify a rotation relative to the default global system, as indicated by the Type column in the table.

TABLE 3-12: COORDINATE SYSTEM DESCRIPTIONS

COORDINATE SYSTEM	ICON	TYPE	DESCRIPTION
About Base Vector Coordinate Systems		relative	1D, 2D, and 3D. Define this using a set of base vectors to form a coordinate system, which are declared as orthonormal.
About Boundary Coordinate Systems		absolute	2D and 3D. A local base vector system on 2D boundaries (t, n) and on 3D boundaries (t1, t2, n). Use it to apply loads that apply in a normal or tangential direction on a boundary that is not aligned with the global Cartesian coordinate system. This coordinate system is always available.
About Cylindrical Coordinate Systems		absolute	2D and 3D. Use this where rotational symmetry about the axis is required. Not applicable in geometries with 2D axial symmetry, where a cylindrical coordinate system is the default coordinate system.
About Mapped Coordinate Systems		absolute	1D, 2D, and 3D. This can deal with translated and rotated coordinate systems. Use this to create a system that defines a mapping from the frame coordinate system.
About Rotated Coordinate Systems		relative	2D and 3D. Use this to define rotation about the out-of-plane direction in 2D and Euler angles in 3D.
About Spherical Coordinate Systems		absolute	3D only. Use this when a field or property using spherical coordinates is to be specified.
About Scaling Coordinate Systems		absolute	For physics interfaces that support infinite elements or perfectly matched layers only. Use this coordinate system, which is similar to a mapped coordinate system, to arbitrarily deform the domain.

See Also

- Coordinate Systems
- Sorting Nodes by Space Dimension and Type
- Spatial Coordinate Variables

About Base Vector Coordinate Systems

Define a **Base Vector System** ( ) using a set of base vectors to form a coordinate system. The system does not necessarily need to be orthonormal, but when it is, declaring it orthonormal and linear enables simplifications which improve performance.

Model

If you have the:

- Acoustics Module, see SAW Gas Sensor: Model Library path **Acoustics_Module/Industrial_Models/saw_gas_sensor**
- MEMS Module, see SAW Gas Sensor: Model Library path **MEMS_Module/Sensors/saw_gas_sensor**
- Structural Mechanics Module, see Piezoelectric Shear-Actuated Beam: Model Library path **Structural_Mechanics_Module/Piezoelectric_Effects/shear_bender**

A vector $\mathbf{F}$ is represented by its contravariant components $[F_1, F_2, F_3]^T$ in the base of the new base vector system defined by the base vectors $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ on the form $\mathbf{F} = F_1\mathbf{u}_1 + F_2\mathbf{u}_2 + F_3\mathbf{u}_3$. Expressing the base vectors as components in another system, for example the global spatial system $[\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z]$, gives the transformation matrix between bases:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}^{-1} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \left\{ \begin{matrix} |\mathbf{u}_i| = 1 \\ \mathbf{u}_i \cdot \mathbf{u}_j = \delta_{ij} \end{matrix} \right\} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}^{T} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

where the last equality holds when the base vector system is orthonormal.

Note that you specify the base vectors as components in the default global system, which is context-dependent. The base vector system is therefore a relative coordinate system whose interpretation depends on the interpretation of the global system in the current context.

Defining a Base Vector System

1. In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Base Vector System** ( ).
2. In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.
3. Under **Settings**, in the **Coordinate names** table, the default names are entered—`x1`, `x2`, and `x3`. In planar 2D models, `x1` and `x2` are typically the in-plane coordinates, and `x3` is the out-of-plane coordinate. Note that these coordinate names are only used as indices for vector and tensor variable names and cannot be evaluated as variables.
4. Under **Base vectors**, define the base vectors in terms of the global Cartesian coordinates (typically `x`, `y`, and `z`); one base vector on each row (two for 2D and three for 3D).

1D

For 1D models, select which basis vector is parallel to the 1D geometry. Select an option from the **In-plane index** list. The default is **1**.

2D

2D Axi

For 2D models, select which basis vector to compute as the cross product of the two in-plane vectors specified. Select an option from the **Out-of-plane index** list. The defaults are **3** for a plane 2D model and **2** for an axisymmetric 2D model. For example, to map the first vector, `x1`, to the direction defined by $y = x$ in 2D, enter `1` in the fields under **x** and **y** on the **x1** row.

5. Under **Simplifications**, you can set properties for the base vector systems which help simplifying the coordinate transformations. Select the **Assume orthonormal** check box if the coordinate system is orthonormal.

About Boundary Coordinate Systems

2D

3D

For 2D and 3D models, a **Boundary System** node is automatically added under **Definitions**.

A **Boundary System** ( ) is a local base vector system on 2D boundaries ($\mathbf{t}$, $\mathbf{n}$) and on 3D boundaries ($\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$). Use it to apply loads and other boundary conditions in a normal or tangential direction on a boundary that is not aligned with the global Cartesian coordinate system.

Model

Many Model Library examples use this coordinate system. For one example, see Electric Sensor: Model Library path **COMSOL_Multiphysics/ Electromagnetics/electric_sensor**

Common applications for this coordinate system include specifying pressure or normal displacement on a surface.

To specify the boundary coordinate system, specify the direction of the normal and a direction that is projected onto the boundary, normalized, and used as the first tangent vector.

The normal direction is in most cases the outward-pointing normal vector. The general definition of the normal (for shells, for example), is the direction of the normal vector $\mathbf{n}$, which can be plotted using the variables for its components (typically `nx`, `ny`, and `nz`). See Normal Variables.

2D Boundary Coordinate System

2D

In 2D the local coordinate system is defined by ($\mathbf{t}_1$, $\mathbf{n}$, $\mathbf{t}_0$), representing the tangential and normal direction of the boundary. This coordinate system is always right-oriented. For exterior boundaries the normal is by default directed out from the domain. The user can reverse the normal direction.

The second tangent direction ($\mathbf{t}_0$) is the cross product between normal vector ($\mathbf{n}$) and the first tangent direction ($\mathbf{t}_1$). This method always gives a right-oriented orthonormal system, unless the tangent direction is parallel to the normal.

3D Boundary Coordinate System

3D

In 3D the local coordinate system is defined by ($\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{n}$), representing two tangential directions ($\mathbf{t}_1$ and $\mathbf{t}_2$) and one normal direction ($\mathbf{n}$).This coordinate system is always right-oriented but not always orthogonal. For exterior boundaries the normal is by default directed out from the domain. The user can reverse the normal direction.

Common applications for this coordinate system include specifying pressure or normal displacement on a surface.

The second tangent direction ($\mathbf{t}_2$) is the cross product between the specified normal vector ($\mathbf{n}$) and the first tangent vector ($\mathbf{t}_1$). This method always gives a right-oriented orthonormal system, unless the tangent direction is parallel to the normal.

Defining a Boundary System

2D

3D

For 2D and 3D models, a **Boundary System** node is automatically added under **Definitions**.

1 In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Boundary System** ( ).

2 In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.

3 Under **Settings**, select a **Frame type**—**Geometry coordinate system**, **Reference configuration**, or **Deformed configuration** (the default).

4 In the **Coordinate names** table, the default names are entered—`t1`, `t2`, and `n` (for 3D models) or `t1`, `n`, and `to` (for 2D models). Click the table cells to edit the names.

5 To reverse the direction of the normal for the boundary system, select the **Reverse normal direction** check box.

6 Select an option from the **Create first tangential direction from** list: **Global Cartesian** or **Manual**.

7 If **Global Cartesian** is selected, select **1**, **2**, or **3** (that is, x, y, or z) from the **Axis** list.

8 If **Manual** is selected, default values are displayed for the local tangent variables `t1x`, `t1y`, and `t1z` (3D) or `t1x` and `t1y` (2D). Enter other values as required to define a tangent direction by specifying directions for a local tangent plane in the **x**, **y**, and **z** fields.

About Cylindrical Coordinate Systems

A **Cylindrical System** ( ) can be used in 2D and 3D where rotational symmetry about the axis is required.

Note

The cylindrical coordinate system is not applicable in geometries with 2D axial symmetry.

Model

If you have the AC/DC Module and Particle Tracing Module, see Magnetic Lens: Model Library path **ACDC_Module/Particle_Tracing/magnetic_lens**

The local coordinate system is defined by (r, φ , a) where r represents the radial distance from the longitudinal axis, φ is the azimuthal angle, and a is the distance from the origin along the longitudinal axis. In 2D models, only the origin can be specified, whereas in 3D models, the longitudinal axis direction, $\mathbf{a}$, and the radial base vector, $\mathbf{e}_r$ (φ = 0), can be specified as well. These direction vectors are automatically normalized.

The definitions of the cylindrical coordinates in terms of the global Cartesian coordinates $\mathbf{r} = \mathbf{r}(x, y, z)$ are

$$\begin{bmatrix} r \\ \varphi \\ a \end{bmatrix} = \begin{bmatrix} |\mathbf{r} - (\mathbf{r}_0 + \mathbf{a}(\mathbf{a} \cdot (\mathbf{r} - \mathbf{r}_0)))| \\ \operatorname{atan}\dfrac{(\mathbf{a} \times \mathbf{e}_r) \cdot (\mathbf{r} - \mathbf{r}_0)}{\mathbf{e}_r \cdot (\mathbf{r} - \mathbf{r}_0)} \\ \mathbf{a} \cdot (\mathbf{r} - \mathbf{r}_0) \end{bmatrix}$$

Defining a Cylindrical System

1. In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Cylindrical System** ( ).
2. In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.
3. Under **Settings,** select a **Frame type**—**Mesh (x, y, z)**, **Material (x, y, z)**, **Spatial (x, y, z)** (the default), or **Geometry (x, y, z)**. Select the frame type that matches the frame used by the physics interface where the cylindrical coordinate system is used.

Note

If the frame type does not match the one used by the physics interface, the coordinate system does not appear in the physics interface's **Coordinate system** lists.

4. In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `phi`, and `a`. In planar 2D models, `r` and `phi` are in-plane polar coordinates, and `a` is the out-of-plane coordinate.
5. Specify the location of the **Origin** of the cylindrical coordinate system in the global Cartesian system. The default is an origin coinciding with the one from the global system.
6. For 3D, enter the **Longitudinal axis direction**. The default is the z direction in the global system.
7. In 3D, specify the **Direction of axis** φ **=0**, where φ is the azimuthal angle. The default direction is the x direction in the global system.

About Mapped Coordinate Systems

Use a **Mapped System** ( ) to create a system that defines a mapping from the frame coordinate system.

This method can deal with translated and rotated coordinate systems:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1(x, y, z) \\ u_2(x, y, z) \\ u_3(x, y, z) \end{bmatrix}$$

Defining a Mapped System

1 In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Mapped System** ( ).

2 In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.

3 Under **Settings,** select a **Frame type**—**Mesh (x,y,z)**, **Material (x,y,z)**, **Spatial(x,y,z)** (the default), or **Geometry frame coordinate**. Select the frame type that matches the frame used by the physics interface where the cylindrical coordinate system is used.

Note

If the frame type does not match the one used by the physics interface, the coordinate system does not appear in the physics interface's **Coordinate system** lists.

4 In the **Coordinate names** table, the default names are entered—`x1`, `x2`, and `x3`. In planar 2D models, `x1` and `x2` are typically the in-plane coordinates, and `x3` is the out-of-plane coordinate.

5 Under **Coordinate mapping**, the **Coordinate** column displays the **Coordinate names** with the **Expression** column displaying the associated mapped coordinate.

6 Under **Simplifications**, select the **Assume orthonormal** check box if the defined coordinate system is orthonormal.

About Rotated Coordinate Systems

Use a **Rotated System** () to define rotation about the out-of-plane axis in 2D and Euler angles in 3D.

Model

If you have the MEMS Module, see Gecko Foot: Model Library path **MEMS_Module/Actuators/gecko_foot**.

On the **Rotated System** page, define the rotation relative to the global Cartesian coordinate system. In 3D models, specify the local coordinate system (x_1, y_1, z_1) using three consecutive Euler angles (rotation angles) α, β, and γ. See Figure 3-2.

In 2D models, describe the rotated coordinate system by the rotation angle about the out-of-plane vector. In both cases the origin of the coordinate system can be defined.

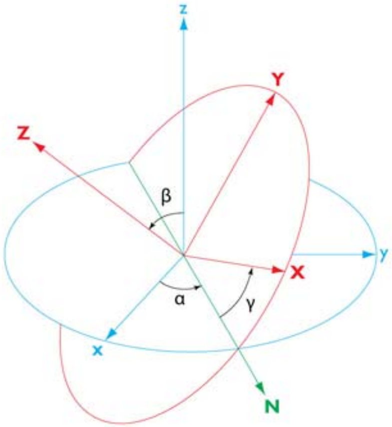


Figure 3-2: 3D Euler angles in a rotated coordinate system.

The transformation matrix for the 3D case is then

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \sin\beta\sin\alpha \\ \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\beta\cos\gamma & -\sin\beta\cos\alpha \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{bmatrix}^T \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Defining a Rotated System

1. In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Rotated System** ( ).
2. In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.
3. Under **Settings**, in the **Coordinate names** table, the default names are entered—`x1`, `x2`, and `x3`. In planar 2D models, `x1` and `x2` are typically the in-plane coordinates, and `x3` is the out-of-plane coordinate.
4. Specify an out-of-plane axis or Euler angles:

2D	For 2D models, select an out-of-plane axis from the **Out-of-plane axis** list (first, second, or third coordinate direction into or out-of screen), and then if necessary adjust the base vectors in the table under **Base vectors**. Enter the **Rotation about out-of-plane axis** (in radians). The default is 0.

3D	For 3D models, enter the **Euler angles (Z-X-Z)** (in radians) in the α, β, and γ fields (see the graphics in the settings window for definitions of these angles). The default values are 0 for all angles.

5. In the **Origin** edit field, specify the location of the origin of the rotated coordinate system. Define it as a vector with two (for 2D) or three (for 3D) components.

   The default is the origin for the global Cartesian coordinate system. Using another origin translates the coordinates in the rotated system by that distance from the global Cartesian origin.

About Spherical Coordinate Systems

Use a **Spherical System** ( ) to define a spherical coordinate system in 3D by its origin, zenith axis, and azimuth axis.

The coordinates of a local spherical coordinate system are (r, θ, φ) where r represents the radial distance from the origin, θ is the inclination, and φ is the azimuthal angle. Specify—in terms of the global Cartesian coordinates x, y, and z—the position of the origin, the axis $\theta = 0$ (the zenith axis, $\mathbf{Z}$), and the axis $\theta = \pi/2$, $\varphi = 0$ (the azimuth axis, $\mathbf{A}$). The direction vectors are automatically normalized.

This is a mapped normalized system using the following transform in global coordinates

$$r = |\mathbf{r} - \mathbf{r}_0|$$

$$\theta = \mathrm{acos}\left(\frac{\mathbf{Z} \cdot (\mathbf{r} - \mathbf{r}_0)}{|\mathbf{r} - \mathbf{r}_0|}\right)$$

$$\varphi = \mathrm{atan2}(\mathbf{r}_\perp \cdot (\mathbf{Z} \times \mathbf{A}), \mathbf{r}_\perp \cdot \mathbf{A})$$

where $\mathbf{r}_0$ is the position of the origin, $\mathbf{Z}$ is a unit vector along the axis $\theta = 0$, and the component of $\mathbf{r} - \mathbf{r}_0$ in the plane $\theta = \pi/2$ is

$$\mathbf{r}_\perp = (\mathbf{r} - \mathbf{r}_0 - \mathbf{Z}(\mathbf{Z} \cdot (\mathbf{r} - \mathbf{r}_0)))$$

Defining a Spherical System

1 In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Spherical System** ( ).

2 In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.

3 Under **Settings,** select a **Frame type**—**Mesh (x, y, z)**, **Material (x, y, z)**, **Spatial (x, y, z)** (the default), or **Geometry (x, y, z)**. Select the frame type that matches the frame used by the physics interface where the cylindrical coordinate system is used.

Note

If the frame type does not match the one used by the physics interface, the coordinate system does not appear in the physics interface's **Coordinate system** lists.

4 In the **Coordinate names** table, the default **Coordinate names** are entered—`r`, `theta`, and `phi`.

5 Enter the location of the **Origin** in the global Cartesian coordinate system. The default is an origin coinciding with that of the global system.

6 Enter the **Direction of axis** θ**=0** (the *zenith axis*). The default axis direction is the *z* direction in the global Cartesian system.

7 Define the **Direction of axis** θ**=**π**/2,** φ **=0** (the *azimuth axis*). The default direction is the *x* direction in the global Cartesian system.

About Scaling Coordinate Systems

Use a **Scaling System** ( ) to create a system that maps the geometry, as represented by the independent coordinates of an underlying frame, onto a virtual geometry represented by virtual scaling system coordinates. Physics interfaces that support infinite elements or perfectly matched layers accept the scaling system coordinates as being the physical domain, in which the underlying frame coordinates are seen as a parameterization. Therefore, using a scaling coordinate system you can arbitrarily deform the domain, essentially in the same way as when using Deformed Geometry with a Prescribed Deformation feature.

Note

The Scaling System is only available for physics interfaces that support infinite elements or perfectly matched layers (PMLs).

See Also

About Infinite Element Domains and Perfectly Matched Layers

The scaling coordinate system is defined as a map from real frame coordinates to virtual scaling system coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1(x, y, z) \\ x_2(x, y, z) \\ x_3(x, y, z) \end{bmatrix}$$

The selected (setting invisible if there is only one frame) frame coordinates are seen as a parameterization of the “true geometry” in which the physics is solved. What you specify in the **Coordinate mapping** table is therefore “true position” for each point in the mesh, expressed in the frame coordinates. When applied to a domain with a compatible material model in a physics interface, the equations in that domain are first reformulated in terms of the virtual x_1, x_2, and x_3 coordinates but then automatically mapped back to the frame coordinates. This leads to explicit transformation expressions appearing in the equations.

Defining a Scaling System

1. In the **Model Builder**, right-click **Definitions** and select **Coordinate Systems>Scaling System** ( ).
2. In the **Coordinate System Identifier** section, enter an **Identifier** or use the default identifier.
3. Under **Coordinate mapping**, the **Coordinate** column displays the virtual coordinate names with the **Expression** column displaying the map from underlying frame coordinates to virtual coordinates. The default expressions are the spatial coordinates x, y, and z, which means no scaling.

About Infinite Element Domains and Perfectly Matched Layers

SIMULATION OF INFINITE DOMAINS

Simulation of unbounded, infinite, domains is a challenge encountered in many types of physics. Normally, any physics interface simulates a process within a bounded domain represented by the geometry drawn in, or imported into, COMSOL Multiphysics. But in many cases the domain is delimited only by artificial boundaries inserted to limit the extent of the model to a manageable *region of interest*. You may not be interested in the details of the solution far away from any sources or material inhomogeneities, but the solution inside the region of interest must not be affected by the presence of the artificial boundaries. You simply want it to behave as if the domain was really of infinite extent.

Artificial truncation of the domain can be handled in a number of different ways. Some physics interfaces include special boundary conditions to absorb outgoing propagating waves without spurious reflections, so called low-reflecting boundary conditions. Others allow impedance boundary conditions which can be tuned to account for a finite impedance between the model boundary and a reference at infinity. Such boundary conditions are often efficient and useful, but generally lack some generality and sometimes accuracy.

Another way to accomplish the same desired effect is to apply a coordinate scaling to a layer of virtual domains surrounding the physical region of interest. For stationary and transient problems, these virtual domains can be stretched out towards infinity, giving rise to *infinite elements*. To absorb outgoing waves in a frequency-domain problem, you must instead stretch the virtual domains into the complex plane, creating *perfectly matched layers* (PMLs).

Because of their common background as coordinate stretching, infinite elements and PMLs in COMSOL Multiphysics share a number of important features.They share part of the user interface and many modeling principles can be translated directly from one to the other. In the description below, infinite elements and PMLs are therefore sometimes referred to collectively as *scaling systems*.

STANDARD GEOMETRY CONFIGURATIONS

Automatic scaling systems are available in COMSOL Multiphysics for three distinct geometrical configurations: Cartesian, Cylindrical and Spherical. Which ones you can use depends on the space dimension of the model.

Plane 2D Models

The available scaling types in plane 2D models are Cartesian and Cylindrical. Cartesian domains are stretched in one or two directions depending on whether they are attached to an edge or to a corner of the physical region of interest.

Important

It is important that separate, normally quadratic, domains are drawn at the corners.

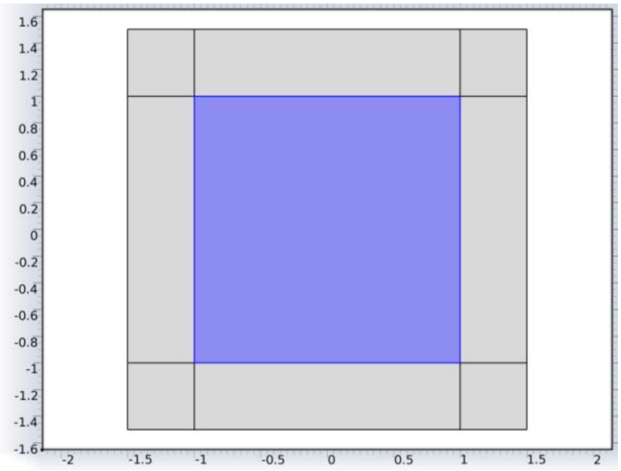


Figure 3-3: Typical Cartesian scaling configuration. Note the distinction between edge and corner domains.

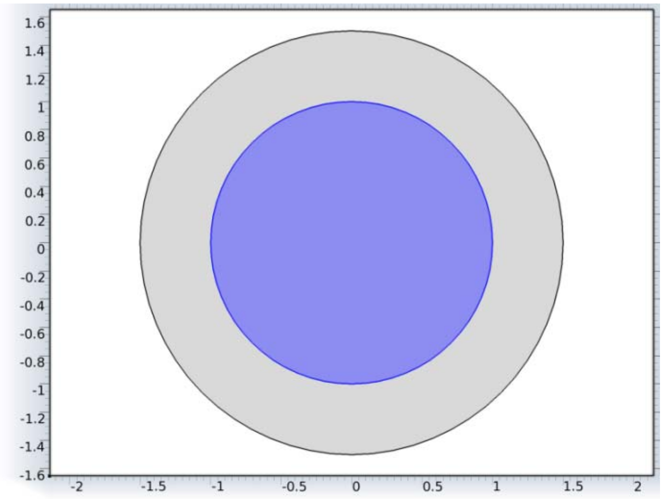


Figure 3-4: Example of cylindrical scaling configuration in plane 2D. You must specify the center point of the model when different from the origin of the coordinate system.

Axisymmetric 2D Models

The available scaling types in 2D axisymmetric models are cylindrical and spherical. The axisymmetric cylindrical configuration from the practical point of view behaved identically to the plane 2D Cartesian option. Similarly, the axisymmetric spherical scaling is very similar to plane 2D cylindrical scaling, except that it is always centred on the axis.

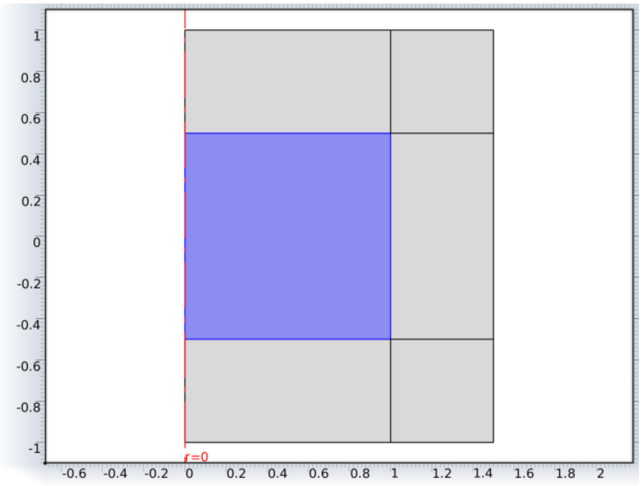


Figure 3-5: Axisymmetric cylindrical scaling uses domains of three distinct types: with radial stretching, with axial stretching, and with both radial and axial stretching. The latter are the corner zones, which must be drawn as distinct domains.

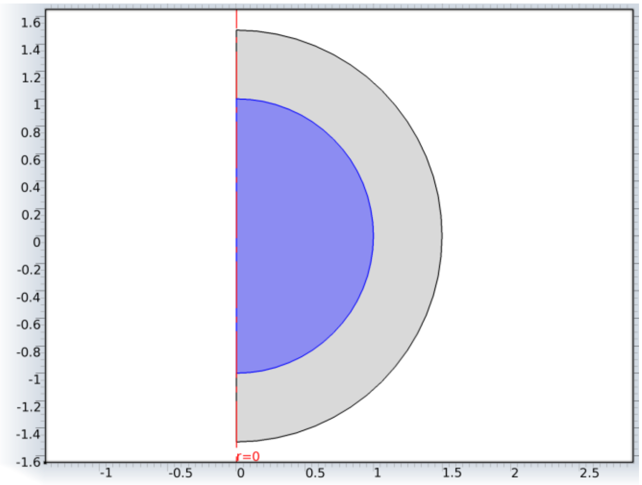


Figure 3-6: Axisymmetric spherical scaling assumes radial stretching in an annulus centered at a point on the axis. If the centre point is not the origin of the coordinate system, you must specify its axial position.

3D Models

The available scaling types in 3D are Cartesian and spherical. The Cartesian scaling domains are of three different types. Depending whether they are attached to a surface, an edge or a point in the physical domain, they are stretched in one, two or three directions respectively. Spherically scaled domains are always stretched only in the spherical domain's radial direction.

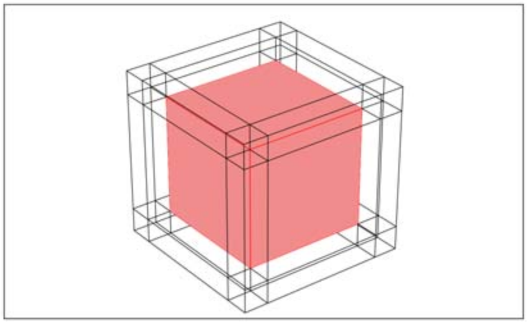


Figure 3-7: Typical Cartesian scaling system configuration. Note the distinction between face, edge and corner domains.

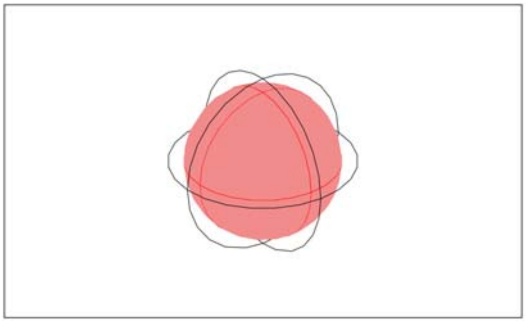


Figure 3-8: Typical spherical scaling system configuration.

NOTE ON AVAILABILITY

Infinite elements (∞) and perfectly matched layers are available only for some physics interfaces and when COMSOL Multiphysics is used together with certain add-on modules. If you have not added to your model any physics interface which is compatible with infinite elements or perfectly matched layers under the available licenses, you cannot add any of these features to the model.

Further, after turning a domain into a perfectly matched layer or infinite element, that domain is not allowed in the active selection of physics interfaces and individual features which are not compatible with these special domain types. This means that the scaled domains are either not be selectable at all, or display as **Not applicable** in the selection list.

NOTE ON COMPATIBILITY WITH PREVIOUS IMPLEMENTATIONS

In older versions of COMSOL, both infinite elements and PMLs are in two different places in the user interface: in some cases, the features are added from the **Definitions** context menu and in some cases from the individual physics interface context menu. This description concerns only the functionality found on the **Definitions** context menu. The infinite element and PML features still present in the physics interfaces are documented together with their interface. But those will eventually be phased out in favor of the implementation under **Definitions**.

Identity and Contact Pairs

In this section:

- About Pairs
- Identity Pairs
- Contact Pairs

About Pairs

Pairs are available for assemblies (that is, geometries created by not forming a union of all geometry objects as the final step), where there is a need to connect boundaries between parts. By default, pairs are created automatically when forming an assembly. There are two types of pairs:

- An *Identity pair* is a pair that by default make the fields across two connected boundaries (one from each connecting part) continuous. This is equivalent to the continuity that is obtained by default on interior boundaries in a geometry created by forming a union. Some physics interfaces provide special boundary conditions for identity pairs to model "slit conditions" such as resistive layers.

  Use an identity pair to specify two selections of boundaries that overlap but belong to different parts of an assembly. Then assign a boundary condition to connect the physics in the two parts in a physics interface. Identity pairs connect overlapping boundaries in different connecting parts of an assembly.
- A *Contact pair* is a pair that define boundaries where the parts may come into contact but cannot penetrate each other under deformation for modeling of structural contact and multiphysics contact.

  Use a contact pair to specify two selections of boundaries that cannot penetrate each other under deformation. The contact pairs define boundaries for parts that can come into contact.

Note

Contact pair modeling requires the Structural Mechanics Module or MEMS Module.

See Also

- Identity Pairs
- Contact Pairs
- Sorting Nodes by Space Dimension and Type
- Specifying Boundary Conditions for Identity Pairs
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

Identity Pairs

Use an **Identity Pair** node ( ) to specify two selections of boundaries that overlap but belong to different parts of an assembly. Then assign a boundary condition to connect the physics in the two parts in a physics interface. Identity pairs connect overlapping boundaries in different connecting parts of an assembly.

Model

Thin-Layer Diffusion: Model Library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**

1 In the **Model Builder**, right-click **Definitions>Pairs** and select **Identity Boundary Pair** ( ) (the most common), **Identity Edge Pair** (3D only) ( ), or **Identity Point Pair** ( ). An identity edge pair, for example, can be useful for connecting two edges in a shell model.

2 In the **General** section, enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair. Also, for boundary pairs, it is possible to change the type of pair from an identity pair to a contact pair from the **Pair type** list. For automatically generated pairs, select the **Manual control of selections** check box to change the selection of source and destination boundaries (edges, points).

The operator mapping an expression *E* on the source side to the destination side is denoted `src2dst_`*`pn`*`(`*`E`*`)`, where *pn* is the pair name. The *variable* `src2dst_`*`pn`* (defined on the destination) is `1` where there is a corresponding source point, and `0` otherwise. The corresponding operator and variable for use on the source side are denoted `dst2src_`*`pn`*.

3 Under **Source Boundaries** (**Source Edges** or **Source Points**):

  a Click the **Activate Selection** button (⏻) to define the source boundaries.

  b Select **Manual** or **All Boundaries** for the boundaries on the source side.

  If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section.

  Use the **Add to Selection** (➕), **Remove from Selection** (➖), and **Clear Selection** (🧹) buttons as required.

4 If required, click the **Swap Source and Destination** button (⇕) to swap the source boundaries and the destination boundaries.

5 Under **Destination Boundaries**:

  a Click the **Activate Selection** button (⏻) to define the source boundaries.

  b Select **Manual** or **All Boundaries** for the boundaries on the destination side.

  These boundaries should overlap the source boundaries. The condition that connects the physics on the destination and source boundaries is specified in the physics interface. For example, it can be a constraint that constrains a dependent variable (temperature, for example) on the destination side to be equal to a dependent variable on the source side.

  If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section.

  Use the **Add to Selection** (➕), **Remove from Selection** (➖), and **Clear Selection** (🧹) buttons as required.

6 If there are several frames in the model, the **Frame** section is available. Select the **Source frame** and the **Destination frame**. Source and destination points are connected if their coordinates in their respective frames are equal.

7 If required, click the **Swap Source and Destination** button (⇕) to swap the source boundaries and the destination boundaries.

Contact Pairs

Use a **Contact Pair** node (⋈) to specify two selections of boundaries that cannot penetrate each other under deformation. The contact pairs defined boundaries for parts that may come into contact (boundaries that cannot penetrate each other under deformation). For more information about contact modeling and guidelines for selecting source and destination boundaries for contact pairs, see the Structural Mechanics Module or MEMS Module documentation.

Note

Contact pair modeling requires the MEMS Module or Structural Mechanics Module.

Model

- If you have the MEMS Module, see Microgripper: Model Library path **MEMS_Module/Piezoelectric_Devices/microgripper**
- If you have the Nonlinear Structural Materials Module, see Snap Hook: Model Library path **Nonlinear_Structural_Materials_Module/Plasticity/snap_hook**
- If you have the Structural Mechanics Module, see Cylinder Roller Contact: Model Library path **Structural_Mechanics_Module/Verification_Models/cylinder_roller_contact**

1 In the **Model Builder**, right-click **Definitions>Pairs**. Select **Contact Pair** ( ).

2 In the **General** section, enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair. Also change the type of pair from an identity pair to a contact pair from the **Pair type** list. For automatically generated pairs, select the **Manual control of selections** check box to change the selection of source and destination boundaries.

The operator mapping an expression E on the source side to the destination side is denoted `src2dst_`*`pn`*`(`*`E`*`)`, where *_pn* is the pair name. Similarly, for a contact pair there is an operator `src2dst_`*`pn`*`_mph` that is suited for use in multiphysics coupling. The variable `geomgap_dst_`*`pn`* is the geometric gap between the source and the destination, seen from the destination side (following the normal of the destination boundary).

The corresponding operators and variables for use on the source side are denoted `dst2src_`*`pn`*, `dst2src_`*`pn`*`_mph`, `geomgap_src_`*`pn`*.

3 Under **Source Boundaries**:

   a Click the **Activate Selection** button (⏻) to define the source boundaries.

   b Select **Manual** or **All Boundaries** for the boundaries on the source side.

   If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section.

   Use the **Add to Selection** (➕), **Remove from Selection** (➖), and **Clear Selection** (🧹) buttons as required.

4 If required, click the **Swap Source and Destination** button (⇕) to swap the source boundaries and the destination boundaries.

5 Under **Destination Boundaries**:

   a Click the **Activate Selection** button (⏻) to define the source boundaries.

   b Select **Manual** or **All Boundaries** for the boundaries on the destination side.

If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section.

The contact algorithm constrains the destination boundaries so that they do not penetrate the source boundaries.

6 If required, click the **Swap Source and Destination** button (⇕) to swap the source boundaries and the destination boundaries.

7 Under **Advanced**:

   a The **Search method** defaults to **Fast**—the algorithm only keeps track of source and destination points that have a distance less than a certain *search distance*. Select **Direct** for a slower, but more robust, search.

   b Select the **Manual control of search distance** check box to tune the search distance (SI unit: m). By default, the search distance is taken as 0.01 times the diagonal of the geometry’s bounding box.

   c If the **Manual control of search distance** check box is selected, enter a different value in the **Distance** field.

Probes

In this section:

- About Probes
- Domain Probes, Boundary Probes, and Edge Probes
- Defining a Domain Point Probe and Adding a Point Probe Expression
- Defining a Boundary Point Probe and Adding a Point Probe Expression
- Global Variable Probes

About Probes

Probes ( ) monitor the development of a scalar-valued quantity (real or complex-valued number) from a time-dependent, frequency-domain, or parametric simulation by two different results presentations: tabulated data and 1D graph plots. It is meaningful both to probe while solving as a monitor and diagnostic tool, and to probe after the computation is finished for results analysis. On top of this functionality, also a probe variable with entire model scope is defined. This variable can be used as any other variable, for example in equations, boundary conditions or in a Stop condition.

Plot while solving is a technique used to briefly interrupt the simulation and launch some predefined plot commands and then continue with the simulation. Both normal plots and graphs can be plotted for probes during the simulation.

There are these types of probes (see Table 3-13 for the icon by space dimension):

- *Domain probes*, *boundary probes*, and *edge probes* make it possible to probe the average, minimum, maximum, or integral of a field quantity over a domain, on a boundary, or along an edge.
- *Domain point probes* and *boundary point probes* provide the value of some field quantity at a point in the domain or on a boundary. Any point within the domain or on the boundary can be defined.
- Use *Global variable probes* ( ) for probing the value of any global variable.

The probes automatically create a **Probe Table** node for displaying numerical results in the **Results** window and an associated plot group with a **Probe Table Plot** node that plots the probe data as a line graph in a separate **Probe Plot** window. For further processing, the probes also add data sets such as **Domain Point Probe** data sets (▣), which contain the probe data. For further control, specify the table and plot window each probe uses.

When the simulation has finished, click the **Update Results** button (↻) in the probe settings windows to change the settings for a probe and update the results information. Then right-click the **Definitions** node and select **Update Probes**.

TABLE 3-13: PROBE TYPES AND ICONS BY SPACE DIMENSION

ICON	PROBE TYPE	MODEL SPACE DIMENSION
	Domain	3D
	Domain	2D
	Domain	1D
	Boundary	3D
	Boundary	2D
	Boundary	1D
	Edge	3D
	Domain Point	3D
	Domain Point	2D
	Boundary Point	3D
	Global Variable	1D, 2D, and 3D

See Also

- Domain Probes, Boundary Probes, and Edge Probes
- Defining a Domain Point Probe and Adding a Point Probe Expression
- Defining a Boundary Point Probe and Adding a Point Probe Expression
- Global Variable Probes
- Sorting Nodes by Space Dimension and Type
- Getting Results While Solving
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

Domain Probes, Boundary Probes, and Edge Probes

Use a **Domain Probe** ( ), **Boundary Probe** ( ), or **Edge Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency-domain, or parametric solution).

Model

If you have the Batteries & Fuel Cells Module:

- For a boundary probe example, see Edge Effects in a Spirally Wound Li-Ion Battery: Model Library path **Batteries_and_Fuel_Cells_Module/ Batteries/li_battery_spiral_2d**
- For a domain probe example, see Mass Transport Analysis of a High Temperature PEM Fuel Cell: Model Library path **Batteries_and_Fuel_Cells_Module/PEMFC/ht_pem**

If you have the Nonlinear Structural Materials Module:

- For a boundary probe example, see Snap Hook: Model Library path **Nonlinear_Structural_Materials_Module/Plasticity/snap_hook**

1 In the **Model Builder**, right-click **Definitions** and select **Probes>Domain Probe**, **Boundary Probe**, or **Edge Probe (3D)**.

The **Domain**, **Boundary**, or **Edge Probe** page opens in the settings window and a node is added in the **Model Builder**.

2 Under **Probe Settings**, select an option from the **Type** list—**Average**, **Maximum**, **Minimum**, or **Integral**.

3 If needed, enter or edit a name for the **Probe variable**.

4 Under **Source Selection**, from the **Selection** list, select **Manual** or **All domains**, **All boundaries**, or **All edges**. The default is to use all domains, boundaries, or edges. If **Manual** is selected, select the applicable geometric entity in the **Graphics** window. Use the **Add to Selection** (✚), **Remove from Selection** (▬), and **Clear Selection** ( ) buttons as required in the settings window.

5 Under **Expression**:

   a Click the **Replace Expression** ( ) or **Insert Expression** (✚) buttons to select predefined expressions based on the physics of the model. Or enter a user-defined **Expression**.

   b Optionally select a compatible unit from the **Unit** list or select the **Description** check box to enter a description of the data set (or edit the default).

6 Optionally, adjust the settings in the **Integration Settings** section:

   a From the **Method** list, select **Integration** (the default) or **Summation**. Only reaction forces use the summation method.

   b Enter an integer value in the **Integration order** field.

   c When working with multiple frames, select a **Frame** from the list for the volume element to be used in the integration.

7 Optionally, adjust the table and plot window to use for outputting the probe data in the **Table and Window Settings** section.

   a From the **Output table** list, select **Default**, **New table**, or any existing probe table. By default, COMSOL Multiphysics uses a probe table that is created automatically.

   b From the **Plot window** list, select **Default**, **New window**, or any existing plot window. By default, COMSOL Multiphysics uses a probe table plot that is created automatically.

Defining a Domain Point Probe and Adding a Point Probe Expression

Use a **Domain Point Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex-valued number) from a dynamic simulation (a time-dependent, frequency-domain, or parametric simulation). A **Point Probe Expression** ( ) can be added as a subnode to **Domain Point Probes** and **Boundary Point Probes**. A default **Point Probe Expression** subnode is always available.

Model

- Process Control Using a PID Controller: Model Library path **COMSOL_Multiphysics/Multidisciplinary_Models/pid_control**
- If you have the RF Module, see Second Harmonic Generation of a Gaussian Beam: Model Library path **RF_Module/Optics_and_Photonics/second_harmonic_generation**

1 In the **Model Builder**, right-click **Definitions** and select **Probes>Domain Point Probe**.

2 Under **Point Selection** select an option from the **Line entry method** list (3D only)—**Point and surface normal** (the default), **Point and direction**, **Two points**, or **None**. If **None** is selected, or in 2D, go to Step c to define coordinates for the probe location directly.

   a If **Point and surface normal** is selected, click at a position on the surface of the geometry. The direction becomes the inward surface normal as defined by the geometry, which for an exterior boundary means that the probe location can be anywhere from the start position to the end of the geometry in the normal direction, as indicated by the value in the **Depth along line** field. If **Point and direction** is selected, the direction becomes that of a ray directed away from the point in the current camera view (that is, the direction depends on the view). If **Two points** is selected, from the **Point being modified** list, first select **First point** and click on the geometry to define the first point (starting point). Then select **Second point** and click to define the second point (ending point) for the line. Select **None** to enter coordinates directly.

   b Enter a **Depth along line** or use the slider to select a value between 0 and 1 that determines the probe location along the line anywhere from the stating point (0) to the ending point (1).

   c Enter **Coordinates**. For 2D enter **x** and **y** coordinates; for 3D enter **x**, **y**, and **z** coordinates. In the 3D case, the **Line entry method** switches to **None**.

   d Select the **Snap to closest boundary** check box to snap the selected points to the grid.

3 Define an expression to probe at the point in the **Point Probe Expression** subnode.

Note

A **Point Probe Expression** subnode is automatically added under **Domain Point Probe**. To add more such nodes, right-click **Domain Point Probe** and select **Point Probe Expression**.

In the **Model Builder**, under **Domain Point Probe**, click **Point Probe Expression**.

4 Edit or enter a name for the **Probe variable**.

5 Under **Expression**:

a Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**.

b Optionally select a compatible unit from the **Unit** list or select the **Description** check box to enter a description of the data set (or edit the default).

6 Optionally, adjust the table and plot window to use for outputting the probe data in the **Table and Window Settings** section.

a From the **Output table** list, select **Default**, **New table**, or any existing probe table. By default, COMSOL Multiphysics uses a probe table that is created automatically.

b From the **Plot window** list, select **Default**, **New window**, or any existing plot window. By default, COMSOL Multiphysics uses a probe table plot that is created automatically.

Defining a Boundary Point Probe and Adding a Point Probe Expression

Use a **Boundary Point Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex valued number) from a dynamic simulation (time-dependent, frequency, parametric). A **Point Probe Expression** ( ) can be added as a subnode to **Domain Point Probes** and **Boundary Point Probes**. A default **Point Probe Expression** subnode is always available.

1 In the **Model Builder**, right-click **Definitions** and select **Probes>Boundary Point Probe**.

2 In the **Graphics** window, select a single boundary to add to the **Boundary Selection** section's **Selection**. Use the **Add to Selection** ( ) and **Clear Selection** ( ) buttons

as required. To add a boundary, first select it in the **Graphics** window or from the **Selection List**.

3. Under **Point Selection** enter the **Coordinates**. The position of the point is indicated by a red dot on the selected surface in the **Graphics** window. Click on the surface to move the point, or enter **x**, **y**, and **z** coordinates. If the point is not on the boundary, the probe location becomes the closest point on the boundary, with coordinates indicated by **On surface** under the fields.
4. Define an expression for the **Point Probe Expression**.

Note

A **Point Probe Expression** node is automatically added under **Boundary Point Probe**. To add more nodes, right-click **Boundary Point Probe** and select **Point Probe Expression**.

In the **Model Builder**, under **Boundary Point Probe**, click **Point Probe Expression**.

5. Edit or enter a name for the **Probe variable**.
6. Under **Expression**:
   - **a** Click the **Replace Expression** (▶ ▾) or **Insert Expression** (✚ ▾) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**.
   - **b** Optionally select a compatible unit from the **Unit** list or select the **Description** check box to enter a description of the data set (or edit the default).
7. Optionally, adjust the table and plot window to use for outputting the probe data in the **Table and Window Settings** section.
   - **a** From the **Output table** list, select **Default**, **New table**, or any existing probe table. By default, COMSOL Multiphysics uses a probe table that is created automatically.
   - **b** From the **Plot window** list, select **Default**, **New window**, or any existing plot window. By default, COMSOL Multiphysics uses a probe table plot that is created automatically.

Global Variable Probes

Use a **Global Variable Probe** ( ) to monitor the development of a scalar-valued quantity (real or complex-valued number) from a dynamic simulation (time-dependent, frequency, parametric).

Model

If you have the:

- Plasma Module, see Harmonic Content of the Power Deposition into a Dual Frequency Capacitively Coupled Plasma: Model Library path **Plasma_Module/Capacitively_Coupled_Plasmas/harmonic_content**
- RF Module, see Microwave Filter on PCB: Model Library path **RF_Module/RF_and_Microwave_Engineering/ pcb_microwave_filter_with_stress**
- MEMS Module, see Transient Response of a Biased Resonator—2D: Model Library path **MEMS_Module/Actuators/ biased_resonator_2d_transient**

1 In the **Model Builder**, right-click **Definitions** and select **Probes>Global Variable Probe**.

The **Global Variable Probe** page opens in the settings window and a node is added in the **Model Builder**.

2 Edit or enter a name for the **Probe variable**.

3 Under **Expression**:

a Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**.

b Optionally select a compatible unit from the **Unit** list or select the **Description** check box to enter a description of the data set (or edit the default).

4 Optionally, adjust the table and plot window to use for outputting the probe data in the **Table and Window Settings** section.

a From the **Output table** list, select **Default**, **New table**, or any existing probe table. By default, COMSOL Multiphysics uses a probe table that is created automatically.

b From the **Plot window** list, select **Default**, **New window**, or any existing plot window. By default, COMSOL Multiphysics uses a probe table plot that is created automatically.

4

Visualization and Selection Tools

COMSOL Multiphysics provides a number of tools to visualize and control how you view models and select parts of the model geometry in the Graphics window and the settings windows.

In this chapter:

Working with Geometry

In this section:

About Using Geometry in COMSOL Multiphysics

COMSOL Multiphysics has many ways to zoom, rotate, and move around the **Graphics** window in order to adjust the view of a model during its creation and analysis. There are also several ways to select each part of the geometry, the geometric entities, and create reusable selections and views.

For example:

- Highlight, select, or hide any part of the geometry using buttons, mouse clicks, keyboard shortcuts, or combinations of actions. See About Geometric Entities and About Selecting Geometric Entities.
- Create User-Defined Selections to reuse throughout the model.
- View an object where the edges of a part are represented by solid lines *(wireframe rendering*). See User-Defined Views.
- Learn about Capturing and Copying Screenshots of the Graphics window.

COMSOL Multiphysics also provides many visual tools to help you create a model. For example:

- Different colors, thicker edges (with OpenGL and Software rendering only, not DirectX—see Editing Preferences Settings), or larger points to highlight the different geometric entities selected.
- A Selection List window to list all the specific geometric entities in the model and to locate and select complex geometric entities.
- Logical selection of overlapping objects in the Graphics window.
- A variety of Show/Hide and Select buttons in the Graphics toolbar.
- Highlighting of geometry objects in the Graphics window when selecting the corresponding geometry feature node in the Model Builder.
- The option to create predefined Views of the geometry.
- The ability to choose different view combinations of Names, Identifiers, Tags, and Types in the Model Builder (see Building a COMSOL Model).

About Geometric Entities

Conceptually, a geometry is a collection of bounded *geometric entities*. The entities are *connected manifolds*, that is, volumes, surfaces, curves, or points. Table 4-1 summarizes the technical terms used for these entities.

Geometric entities of the maximum dimension are called *domains*, while of the next highest dimension they are called *boundaries*. The boundaries are sometimes referred to as *faces* in 3D and *edges* in 2D. The *vertices* are also called *points*.

The following rules apply to domains:

- The (interiors of the) domains are disjointed. However, this is only strictly true if the finalization method is to form a union. When it is to form an assembly, domains may overlap (though that is normally considered a modeling error).
- Every geometric entity is bounded by entities of smaller dimension. In particular, a domain (in 3D, 2D, or 1D) is bounded by boundaries, edges (in 3D), and vertices (in 3D and 2D). A boundary (in 3D or 2D) is bounded by edges (in 3D) and vertices. An edge is bounded by vertices.

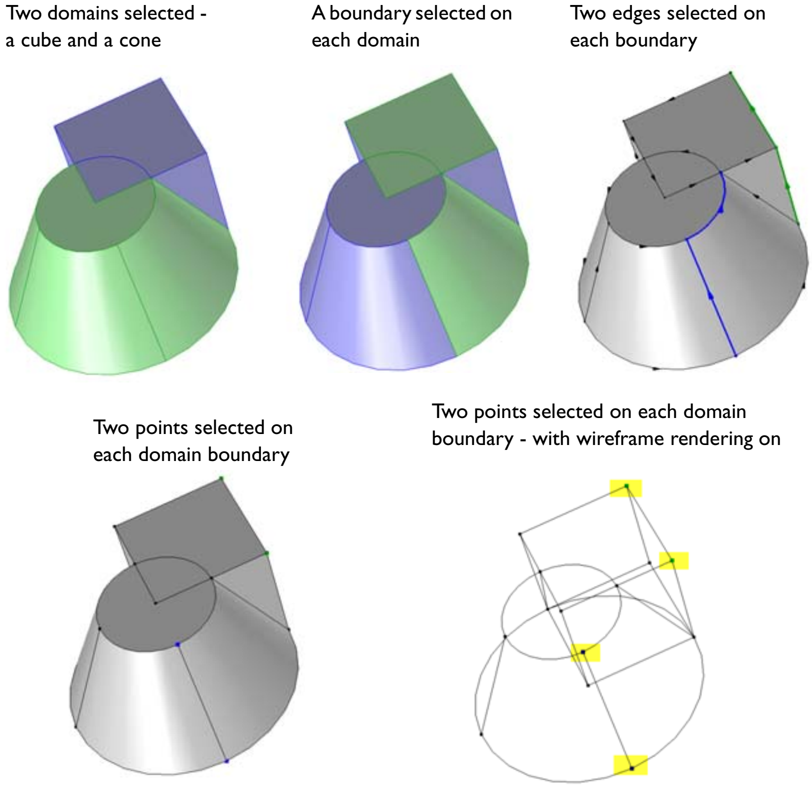
COMSOL Multiphysics geometric entities include *domains*, *boundaries*, *edges* (3D only), and *points*. For example, a 3D cube consists of one domain with six boundaries. The six boundaries have 12 edges and the edges connect at eight points (see

Figure 4-1). This enables visualization of a cube by displaying one or more of these four types. For instance, generate a wireframe plot by rendering only the cube's edges.

TABLE 4-1: NAMES OF GEOMETRIC ENTITIES IN DIFFERENT SPACE DIMENSIONS

ENTITY DIMENSION	NAME IN 3D	NAME IN 2D	NAME IN 1D	NAME IN 0D
3D	domain			
2D	boundary	domain		
1D	edge	boundary	domain	
0D	vertex	vertex	boundary	domain

Geometry objects are *adjacent* if they connect directly to each other. Hence all boundaries, edges, and points on the cube are adjacent to the domain. An edge on the cube is adjacent to two boundaries and two points.



Two domains selected - a cube and a cone

A boundary selected on each domain

Two edges selected on each boundary

Two points selected on each domain boundary

Two points selected on each domain boundary - with wireframe rendering on

Figure 4-1: A 3D geometry comprises domains, boundaries, edges, and points.

About Selecting Geometric Entities

Throughout COMSOL there are many selection lists all based on the same principle—select a *domain*, *boundary*, *edge*, or *point* and use the same methods to add or remove these geometric entities. For example, physics interface windows for defining equations and material properties, boundary conditions, sources, and other parts of the model's physics, or the **Variables** node's (a=) variable definitions for variables that are not defined in the entire model.

There are toolbar buttons, mouse click options, page settings, and keyboard shortcuts available to help move, select, and highlight geometric entities. Often there is more than one way to do the same thing. The power of COMSOL is that every level of geometry can be treated individually. See Table 4-4 for a list of different ways to complete the same task.

Table 4-2 lists the buttons that display on every settings window that has a geometric entity selection list as displayed in Figure 4-4.

TABLE 4-2: GEOMETRIC ENTITY SELECTION LIST BUTTONS

BUTTON	NAME	DESCRIPTION
	Create Selection	Use this button to create Selection nodes under model Definitions to represent various parts of the geometry and simplify the process of assigning materials, model equations, boundary conditions, and other model features. See Creating User-Defined Selection Nodes. See also Object Selection in the COMSOL Multiphysics Reference Guide for information about making object selections in the Geometry branch; that way you can make a selection that only includes a few of the geometric entities from one or more geometry objects.
	Copy Selection	Use this button to copy the selection from the list in the settings window to the clipboard.
	Paste Selection	Use this button if you have a list of geometric entities in a file or document that you want to type in and then paste into a selection list. If you copy a selection from a document to the clipboard, you can paste the selection directly using Ctrl+V. An example is a list of geometric entity numbers described as a step in a COMSOL Multiphysics modeling instruction. See Pasting Selections.
	Zoom Selected	Use this button to zoom into the selected geometric entity.

TABLE 4-2: GEOMETRIC ENTITY SELECTION LIST BUTTONS

BUTTON	NAME	DESCRIPTION
➕	Add to Selection	Use this button to add a geometric entity to the Selection list. See Table 4-4.
➖	Remove from Selection	Use this button to remove a geometric entity from the Selection list. See Table 4-4.
🧹	Clear Selection	Use this button to clear a selection.

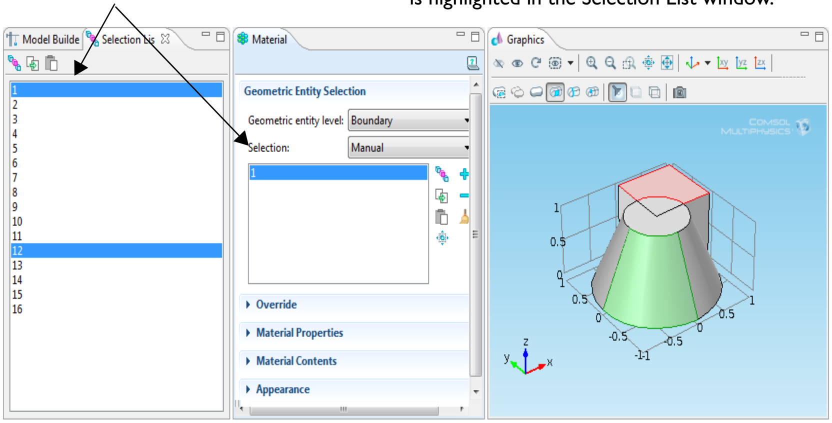
VISUALIZATION AIDS TO HELP WITH SELECTION

Geometric Entities Highlighted in Color in the Graphics Window

COMSOL Multiphysics highlights geometric entities at different stages of selection. Selected lines also have a thicker line width and selected points are larger. In Figure 4-1, each geometric entity is highlighted in red, blue, green, or with no highlight to indicate its status. It cycles between red (left-click to select the geometric entity), blue (right-click to add it to the selection), green (left-click to show that it is added), and no highlight (right-click to cancel the selection).

The green geometric entity (labeled 1) has been added to the Material selection list.

The red geometric entity (labeled 12) has NOT been added to the Material selection list, but it is highlighted in the Selection List window.



Geometric Entity Selection

Geometric entity level: Boundary

Selection: Manual

Figure 4-2: Selecting geometric levels - highlighted to indicate status.

Red The geometric entity is highlighted but not added to the model selection list. Left-click once with the mouse. If you right-click it locks all red geometric entities. See Figure 4-2 for an example.

Blue The geometric entity has been added to the model selection list. The selection is highlighted in blue after you right-click once. See Figure 4-1 for an example.

Green After the geometric entity is added to the selection list, when you click anywhere (directly in the Graphics window, from the settings window, or in the Selection window) it is highlighted green, which corresponds to the name of the geometric entity in the Selection section in the settings window. See Figure 4-1 and Figure 4-2 for examples.

No highlight If the geometric entity is not highlighted it means it is not selected.

About Overlapping Objects

When you click a 3D geometry comprised of several objects, geometric entities of the same type might overlap at the point where you click. Overlapping objects are selected starting with the closest geometric entity and ending with the overlapping entity the furthest away.

For example, if you click a circle boundary that overlaps with a sphere boundary as well as a square boundary, the first click selects the circle boundary (the point of origin), then the second click selects the next closest boundary on the sphere, and the last click selects the boundary on the square. Then the cycle starts again with the point of origin, the circle boundary.

The Graphics Toolbar Buttons

Some of the toolbar buttons available on the **Graphics** window are different based on the space dimension of the model. The buttons also correspond to domain, boundary, edge, and point level nodes that display under the physics interface, which are also based on the model space dimension.

See Also

Physics Nodes by Space Dimension

TABLE 4-3: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

BUTTON	NAME
1D and 1D Axisymmetric Models	
—▪	Select Boundaries

TABLE 4-3: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

BUTTON	NAME
	Select Box
	Select Domains
	Select Objects
2D and 2D Axisymmetric Models	
	Select Boundaries
	Select Box
	Select Domains
	Select Objects
	Select Points
	Align with Work Plane*
	Work Plane Clipping*
*These buttons are available when using a Work Plane to define a 2D object in 3D. See Drawing on the 2D Work Plane in 3D.	
3D Models	
	Select Boundaries
	Select Box
	Select Domains
	Select Edges

TABLE 4-3: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

BUTTON	NAME
	Select Objects
	Select Points
xy	Go to XY View
yz	Go to YZ View
zx	Go to ZX View
	Scene Light
	Transparency
	Wireframe Rendering
Available on all Toolbars	
	Hide Selected
	Show Selected
	Reset Hiding
	View All
	View Hidden Only
	View Unhidden
	Zoom In
	Zoom Out

TABLE 4-3: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

BUTTON	NAME
	Zoom Selected
	Zoom Extents
	Go to Default View
	Image Snapshot

Using the Selection List Window

Use the **Selection List** window to make it easier to choose objects, for example, while working with complex geometries and when you need to easily locate a geometric entity that is not easily viewed. The Selection List is particularly useful when you know the geometric entity number to select; for example, when you are following Model Library instructions to build a model (in that case you can also copy and paste the selections directly from the instructions).

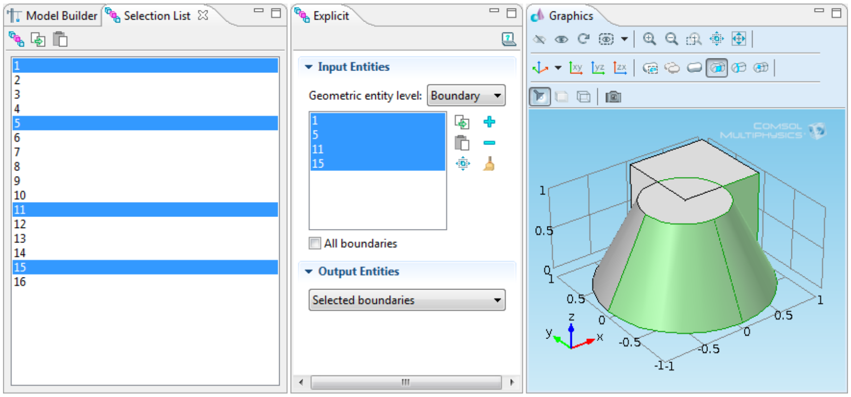
To open this window from the main menu, select **View>Selection List** ( ). COMSOL determines the geometric entities listed based on where in the model you are working.

The **Selection List** window displays all geometric entities of a certain type (boundaries, for example). This is different from the selection lists in the settings window, which contains a list of the selected entities only. Click on any item to see it highlighted in the Graphics window and select items as described in Selecting and Deselecting Geometric Entities. For example, use the Selection List in these situations:

- When in the Model Builder under the **Geometry** node ( )—the geometry objects are displayed in the Selection List, for example, `blk1` and `cone1`. The display is a tree structure with the geometry features as parent nodes and the objects that it generates as child nodes. Selecting a top-level node in the tree automatically selects all child nodes.
- When in the Model Builder under the **Mesh** node ( ), the list includes also information on which entities are meshed. If the Geometry has Mesh Construction

entities, the list also specifies if a construction entity has been removed, see Mesh Control Entities.

- When working in windows with Selection or Geometric Scope sections (a **Selection** window under a **Definitions** node (≡) for example), or anywhere you assign materials, physics, boundary conditions, and other model settings. The Selection List displays the specific *geometric entity level* selected (domain, boundary, edge, or point). See Figure 4-3.
- When working in the Geometry node with *Chamfer* or *Fillet* features and you want to locate specific points. Specify the selection level by clicking the **Select Points** buttons in the Graphics toolbar and add the points to the **Vertices to fillet** or **chamfer**.



Model Builder | Selection List

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Explicit

Input Entities

Geometric entity level: Boundary

1
5
11
15

□ All boundaries

Output Entities

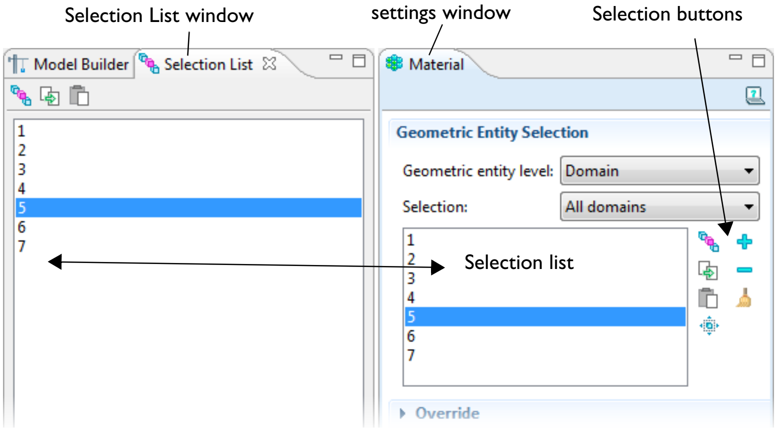
Selected boundaries

Figure 4-3: Using a combination of the Selection List and Selection window to choose boundaries on a 3D model.

Click the **Create Selection** button ( ) to create an explicit named selection that includes the selected entities from the **Selection List** window. Click the **Copy Selection** button ( ) to copy the selection to the clipboard (from which you can paste it into other selection lists). Click the **Paste Selection** button ( ) to paste a selection from the clipboard (or another selection that you type) in the **Selection** field in the **Paste Selection** dialog box that opens.

Figure 4-4 shows the difference between a **Selection List** window ( ) and the selection lists that are found in the settings window for many nodes. Normally the list of selected entities is shorter in the settings window, which only shows the selected entities (boundaries in this case). See also Figure 4-3 above and Figure 4-6 in the

User-Defined Selections section for more examples and comparisons of the different Selection features.



Geometric entity level: Domain

Selection: All domains

Figure 4-4: Selection List window compared to the Selection list in the settings window.

AN EXAMPLE OF SELECTING BOUNDARIES WITH THE SELECTION LIST

This example demonstrates the selection of boundaries. When there is the possibility of overlapping geometric entities, it is recommended that you use the **Selection List** window to ensure the correct part of the geometry is selected.

1. From the main menu, select **View>Model Library**.
2. Navigate to the **COMSOL Multiphysics>Structural Mechanics>mast_diagonal_mounting** model file. Double-click to open it.
3. In the **Model Builder**, click the **Model** node.
4. In the Graphics toolbar, click the **Select Boundaries** button ( ).

5 From the main menu, select **View>Selection List**.

The **Selection List** window lists all the boundaries contained in all the domains.

There are many ways to select the boundaries:

- Click on any boundary contained in the **Selection List** window.
- Ctrl+click to select more than one boundary from the list at a time. The boundaries are highlighted at the same time in the **Graphics** window.
- Rotate the geometry as required and click on it to highlight boundary numbers in the **Selection List** window.
- In the **Selection List**, shift-click to select contiguous items.
- In the **Graphics** window, shift-click to add an object to the selection. If objects are overlapping, you may need to click more than twice to select all objects.

6 For a selection in the **Selection List** window, right-click to confirm the selection (which adds it to a currently selected boundary condition, for example). Also click the **Create Selection** button ( ) to create an explicit named selection that includes the selected entities from the **Selection List** window. Click the **Copy Selection** button ( ) to copy the selection to the clipboard (from which you can paste it into other selection lists).

Selecting and Deselecting Geometric Entities

Note	The same methods are used when working with 2D geometric entities except there are no edges.

Add and remove 3D geometric entities (domains, boundaries, edges, or points) to selection lists in different ways, including buttons on the Graphics toolbar, using the **Selection List** window, clicking directly on the geometry, or clicking buttons in the settings window. In the **Graphics** window the geometric entities are color highlighted as you make the selections and you can lock items as you select them.

At various stages of selecting geometric entities, it is also useful to lock a selection to prevent it from being removed by accident—for example, if you are making multiple selections among many overlapping objects. When an object is locked (that is to say, added to a selection list) it is highlighted in blue.

If you start by setting the **Geometric entity level** to **Boundary**, and then select **All Boundaries**, the **Selection** list displays all these boundaries. If you make any changes to this list, for example, remove a boundary, the **Selection** list reverts to **Manual**. This option is useful, however, for locating and determining which items to add or remove from the selection and is the same as those geometric entities that display in the **Selection List** window.

See Table 4-4 for the many different ways to select geometric entities using toolbar buttons, mouse click options, page settings, and keyboard shortcuts.

TABLE 4-4: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Select any level of geometry:	In the **Graphics** window, click the geometric entity. In the settings window, select from a selection list. Open the **Selection List** window and click entity names.
Select all parts of the geometry:	From the main menu select **Edit>Select All** or press Ctrl+A. This highlights and selects all entities but does NOT confirm the selection or lock it if the model changes. To lock a selection, on some pages in the **Setting** window, the option to select **All domains**, **All boundaries**, **All edges**, or **All points** adds all geometric entities of that type to the list. The selected items are highlighted in the **Graphics** window and all entities are kept selected even if the geometry changes. Open the **Selection List** window, shift+click to select all the entity names. You can also click the main geometry feature to select all entities that the feature includes.
Deselect all parts of the geometry not added to a selection list:	From the main menu select **Edit>Deselect All**. In the **Graphics** window, click outside of the geometry or press Ctrl+D to deselect all selections whereas locked selections remain intact.
Select adjacent geometric entities (3D):	In the **Graphics** window, click a geometric entity, for example, an edge. Then click multiple times to cycle through all the edges that are next to an edge. Hold the Ctrl key down while clicking to confirm selection of all the adjacent geometric entities at the same time.

TABLE 4-4: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Select, move, and rotate at the same time:	Multiple mouse actions can be done at the same time in 3D. For example, use the mouse to rotate or move the object left and right to locate the geometric entity to add to a selection list, then right-click and left-click to add to the selection or click the **Select Box** button (3D , 2D , and 1D)
Lock parts of the geometry during selection:	On some settings windows, click the **Select all** (**domains**, **boundaries**, **edges**, or **points**) check box. See Creating User-Defined Selection Nodes for an example.
Add items to a selection list:	For user-defined selections this action must be completed on the selection page. See Creating User-Defined Selection Nodes for information. In the **Graphics** window, left-click (red highlight) then right-click (blue highlight) a geometric entity. Or select one or more geometric entities and click the **Add to Selection** button (+) in the settings window. In the **Selection List** window, select the entity names to add, and click the **Add to Selection** button (+) in the settings window. You can also paste selections from a file. See Pasting Selections for information.
Select multiple parts of the geometry:	In the **Graphics** window, click the **Select Box** button (3D , 2D , and 1D) then click and hold the left mouse button to draw a square over the geometry. It is like a "rubberband" selecting all entities enclosed by this operation. Shift-click adds additional geometric entities to the **Selection list**. In the **Selection List** window, select the entity names to add using the Shift or Ctrl keys. Click the **Add to Selection** button (+) in the settings window to add the geometric entity to the **Selection list**.
Use an -OR operation with the current selection 2D	*2D only:* In the **Graphics** window, press Ctrl+click on the geometric entity you want to select. Then click the **Select Box** button ( ) (and click and hold the left mouse button to draw a square over the geometry. Presently selected objects are deselected when inside the **Select Box** and vice versa.

TABLE 4-4: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Use an -AND operation with the current selection 3D	*3D only:* In the **Graphics** window, press Ctrl+click on the geometric entity you want to select. Then click the **Select Box** button ( ) and click and hold the left mouse button to draw a square over the geometry. Only the presently selected objects inside the rubber-band box stay selected. This makes it possible to select a detail from several selected objects.
Select only objects:	In the **Graphics** window, click the **Select Objects** button (3D , 2D , and 1D). To select more than one at a time, press Ctrl+click or Shift+click.
Select only domains:	In the **Graphics** window, click the **Select Domains** button (3D , 2D , and 1D). To select more than one at a time, press Ctrl+click or Shift+click. In the settings window, select **Domain** from a **Selection list** and then click in the **Graphics** window. Only domains are highlighted.
Select only boundaries:	In the **Graphics** window, click the **Select Boundaries** button (3D , 2D , and 1D). To select more than one at a time, press Ctrl+click or Shift+click. In the settings window, select **Boundary** from a **Selection list** and then click in the **Graphics** window. Only boundaries are highlighted.
Select only edges:	In the **Graphics** window, click the **Select Edges** button ( ). To select more than one at a time, press Ctrl+click or Shift+click. In the settings window, select **Edge** from a **Selection list** and then click in the **Graphics** window. Only edges are highlighted.
Select only points:	In the **Graphics** window, click the **Select Points** button (3D and 2D . To select more than one at a time, press Ctrl+click or Shift+click. In the settings window, select **Point** from a **Selection list** and then click in the **Graphics** window. Only points are highlighted.

TABLE 4-4: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Remove a geometric entity from a selection list:	For user-defined selections this action must be completed on the selection page. See Creating User-Defined Selection Nodes for information. In the **Graphics** window, highlight or select the geometric entity and click the **Remove from Selection** button (▬) in the settings window. In the **Graphics** window, left-click to highlight and select the geometric entity to remove. Any blue geometric entity turns green. To remove that geometric entity, right-click the mouse button. In the **Selection List** window, select the entity names to remove using the Shift or Ctrl keys. Click the **Remove from Selection** button (▬) in the settings window to remove the geometric entity from the **Selection list**.
Clear all items from a selection list:	For user-defined selections this action must be completed on the selection page. See Creating User-Defined Selection Nodes for information. Click the **Clear Selection** button (🧹) in the settings window.

Zooming In and Out in the Graphics Window

TASK	ACTION
Zoom in and out:	Click the **Zoom In** button (⊕) to zoom in. Click the **Zoom Out** button (⊖) to zoom out. *3D only*: Click the middle mouse button and drag it forward and backward to zoom in and out of the object. The zoom is centered where the first click is made in the **Graphics** window.
Zoom into a general area of the geometry:	Click the **Zoom Box** button (⬚) then click and drag to highlight a section of the geometry to zoom into.
Zoom into a geometric entity:	Click any domain, boundary, edge, or point. Click the **Zoom Selected** button (⊡) to display to the extent of the window. This button is also available in connection with the selection lists for domains, boundaries, edges, and points.
Zoom out to the view the complete geometry:	Click the **Zoom Extents** button (✥) to zoom out and fit the complete geometry into the window.

Changing Views in the Graphics Window

TASK	ACTION
Change the view to the xy-, yz-, or zx-plane (3D models only):	Click the **Go to View XY** (xy), **YZ** (yz), or **ZX** (zx) buttons. The first click selects the plane view with a positive normal direction. A second click on the same button switches to a negative normal direction.
Change the view to the default:	Click the **Go to Default View** button (↳). This is available for all space dimensions.
Display a user-defined view:	After creating a **View** under the **Definitions** node, click the down arrow next to the **Go to View** button (↳) and select a user-defined view from the list.

TASK	ACTION AND RESULT	OPERATION ORDER
Rotate the geometry about the axes:	Left-click and hold down the mouse button while dragging it in any direction. This rotates the scene around the axes parallel to the screen X- and Y-axes with origin in the scene rotation point.	left-click
Move the visible frame on the image plane in any direction, (like using a camera shift lens):	Right-click and hold down the mouse button while dragging it in any direction.	right-click
Rotate about the X- and Y axes in the image plane (tilt and pan the camera):	Press Ctrl and left-click. While holding down both buttons, drag the mouse in any direction. This places the rotation coordinate system in the camera and rotates around the axes parallel to the screen X- and Y-axes.	Ctrl+left-click
Move the camera in the plane parallel to the image plane:	Press Ctrl and right-click. While holding down both buttons, drag the mouse in any direction.	Ctrl+right-click
Rotate the camera around the axis:	Press Ctrl+Alt, then left-click. While holding down all buttons, drag the mouse in any direction.	Ctrl+Alt+ left-click
Move the scene in the plane orthogonal to the direction the camera looks at:	Right-click then press Ctrl. While holding down both buttons, drag the mouse in any direction.	right-click+Ctrl
Move the scene in the plane orthogonal to the axis between the camera and the scene rotation point:	Right-click the mouse and press Alt. While holding down both buttons, drag the mouse in any direction.	right-click+Alt
Move the camera into and away from the object (dolly in/out):	Click the middle mouse button and then press Ctrl. While holding down both buttons, drag the mouse in any direction.	middle-click+Ctrl

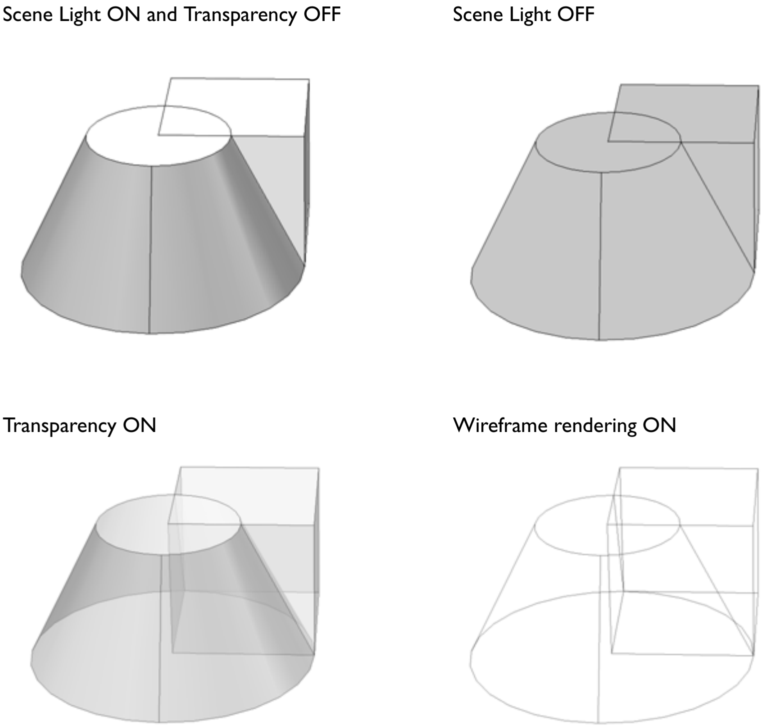
TASK	ACTION AND RESULT	OPERATION ORDER
Rotate the camera about its axis between the camera and the scene rotation point (roll direction):	Press Alt, then left-click. While holding down both buttons, drag the mouse in any direction.	Alt+left-click
Move the camera along the axis between the camera and the scene rotation point:	Press Alt, then middle-click. While holding down both buttons, drag the mouse in any direction.	Alt+middle-click

Lighting, Transparency, and Wireframe Rendering

3D

The following features are available for the 3D View.

TASK	ACTION
Turn Scene Light ON or OFF:	In the **Graphics** window or any other plot window, click the **Scene Light** button ( ). See Figure 4-5. When creating a **View**, this action toggles the **Scene light** check box on the **View** page.
Turn Transparency ON or OFF:	In the **Graphics** window or any other plot window, click the **Transparency** button ( ). See Figure 4-5. When creating a **View**, this action toggles the **Transparency** check box on the **View** page.
Turn Wireframe Rendering ON or OFF:	In the **Graphics** window, click the **Wireframe Rendering** button ( ). See Figure 4-5. When creating a **View**, this action toggles the **Wireframe Rendering** check box on the **View** page. See also Editing Preferences Settings to set the level of graphic detail to Wireframe and speed up the rendering of complex models or to improve visual appearance.



Scene Light ON and Transparency OFF

Scene Light OFF

Transparency ON

Wireframe rendering ON

Figure 4-5: Scene light, transparency, and wireframe rendering examples applied to a geometry.

Hiding and Showing Geometric Entities

Note

Selection methods vary based on the model's space dimension.

Selecting an item in any **Selection list** highlights the corresponding geometric entities or objects in the **Graphics** window for selection or deselection. Only the geometric entities you can see in the **Graphics** window are available for selection; that is to say hidden objects cannot be selected.

TASK	ACTION
Hide selected geometric entities:	In the **Graphics** window, select any geometric entity (domain, boundary, edge, or point) and click the **Hide Selected** button ( ). When creating a **View**, right-click the **View** node and select **Hide Geometry Objects**. Select a **Geometric entity level** from the list to hide. When creating a **View**, right-click the **View** node and select **Hide Geometry Entities**. Select a **Geometric entity level** from the list to hide.
Show selected geometric entities:	In the **Graphics** window, select any geometric entity (domain, boundary, edge, or point) and click the **Show Selected** button ( ).
Reset all hidden objects to the default:	In the **Graphics** window, click the **Reset Hiding** button ( ) to reset all hidden domains, boundaries, edges, or points to the default. This removes any **Hide Geometry Objects** or **Hide Geometric Entities** subnode added to a **View** node.
View hidden geometric entities:	In the **Graphics** window, click the **View Hidden Only** button ( ) to display only hidden domains, boundaries, edges, or points.
View unhidden geometric entities:	In the **Graphics** window, click the **View Unhidden** button ( ) to display any domains, boundaries, edges, or points not hidden.
View all geometric entities:	In the **Graphics** window, click the **View All** button ( ) to display all hidden and unhidden domains, boundaries, edges, or points.

Note

The **View Hidden Only** button ( ), **View Unhidden** button ( ), or **View All** button ( ) that appears on the Graphics window toolbar indicates the view you get when you click the button, whereas the bullet in the associated menu containing these three options shows the current view.

User-Defined Selections

In this section:

About the Selections Node

Create selection nodes under model **Definitions** to represent various parts of the geometry and simplify the process of assigning materials, model equations, boundary conditions, and other model features.

Each user-defined Selection can be reused during the model design and renamed with descriptive titles—for example, “Tube,” “Wall,” or “Fluid.” Changes to the Selection (for example, by adding or removing a boundary) updates all features in the model that use that particular Selection.

There are different types of selections—**Explicit Selections**, different types of **Boolean Selections**, selections of adjacent geometric entities (**Adjacent Selections**), and selections by enclosing part of the geometry by a bounding box or ball (**Ball and Box Selections**)—to add these, right-click a **Definitions** node and select from the **Selections** options as listed in Table 4-5.

In the selection settings windows (except explicit selections) click the **Zoom Selected** ( ) button to zoom in on the selected geometric entities.

TABLE 4-5: SELECTION TYPE DESCRIPTIONS

SELECTION TYPE	ICON	SEE	DESCRIPTION
Explicit		Defining an Explicit Selection Node	Use an Explicit node to create the selection using the normal selection tools for individual geometric entities (boundaries, for example) on the geometric entity level chosen.
Union		Defining a Boolean Selection Node	Use the Union node to create the selection as the union (addition) of two or more selections.
Intersection			Use the Intersection node to create the selection as the intersection of two or more selections.
Difference			Use the Difference node to create the selection as the difference between a set of one or more selections and another set of one or more selections.
Complement			Use the Complement node to create the selection as the complement (inverse) of one or more selections.
Adjacent		Defining an Adjacent Selection Node	Use the Adjacent node to create the selection as the adjacent geometric entities (boundaries, for example) to one of more selections.
Ball		Defining a Box or Ball Selection Node	Use the Ball node to create the selection by enclosing part of the geometry by a bounding ball (sphere) to select geometric entities that are partially or completely inside the ball.
Box			Use the Box node to create the selection by enclosing part of the geometry by a bounding box to select geometric entities that are partially or completely inside the box.

See Also

- Object Selection in the *COMSOL Multiphysics Reference Guide*
- User-Defined Selections
- Sorting Nodes by Space Dimension and Type
- Creating Selections From Geometric Primitives and Features
- Pasting Selections
- About Selecting Geometric Entities
- Selecting and Deselecting Geometric Entities

Opening an Example Model

Figure 4-6 uses a Model Library model example where several user-defined selections have already been created.

TO OPEN THIS MODEL

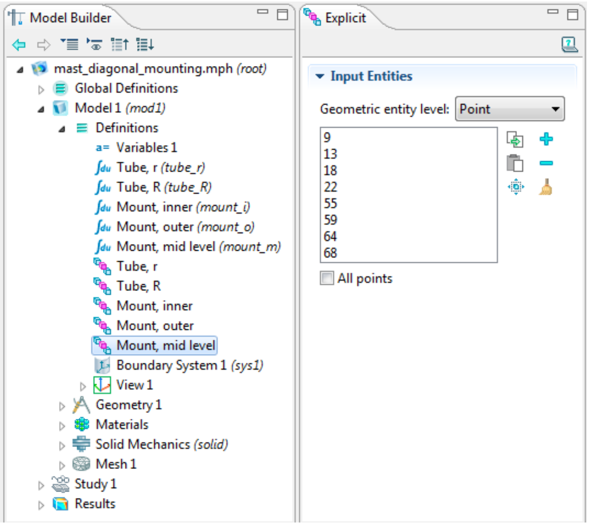
1 From the main menu, select **View>Model Library**.

2 Navigate to the **COMSOL Multiphysics>Structural Mechanics>mast_diagonal_mounting** model file. Double-click to open it.

3 Click the **Definitions** node under **Model 1**. Several nodes display in the **Model Builder**.

4 Click the **Fixed face** and **Force faces** nodes to see how each are defined.

5 Click the **Boundary Load** node under **Solid Mechanics**. The **Boundary Load** page opens in the settings window. Under **Boundaries**, **Force faces** is the **Selection**. **Fixed face** is also available to be selected from the **Selection** list.



Geometric entity level: Point

9
13
18
22
55
59
64
68

□ All points

Figure 4-6: An example of an Explicit selection window for the Mount, mod level node. The Selection nodes under Definitions are renamed by the user.

Creating User-Defined Selection Nodes

There are several ways to create Selection nodes using the **Create Selection** (▨) button.

- Defining an Explicit Selection Node
- Defining a Boolean Selection Node
- Defining an Adjacent Selection Node
- Defining a Box or Ball Selection Node
- Duplicating a Selection Node from the Definitions Node Level
- Creating a Selection Node from the Selection List Window
- Creating a Selection Node from a Settings Window
- Creating Selections From Geometric Primitives and Features

Figure 4-6 is an example of working with the Selection nodes under Definitions. In the settings windows for all selection nodes except **Explicit** selection nodes click the **Zoom Selected** ( ) button to zoom in on the selected geometric entities.

See Also

About Selecting Geometric Entities, Using the Selection List Window, and, Selecting and Deselecting Geometric Entities for other examples using the Selection List feature.

Defining an Explicit Selection Node

Use an **Explicit** node ( ) to create the selection using the normal selection tools for individual geometric entities (boundaries, for example) on the geometric entity level that chosen.

Model

- Acoustics of a Muffler: Model Library path: **COMSOL_Multiphysics/Acoustics/automotive_muffler**
- Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**
- See also Figure 4-6 for a model example

1 In the **Model Builder**, right-click any **Definitions** node ( ) under a **Model**, and choose **Selections>Explicit** ( ).

2 Right-click or press F2 to **Rename** the **Explicit** node as required.

3 Under **Input Entities** add geometric entities.

   a Select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the geometric entities to add or remove from the selection list.

   b Select and add geometric entities of the chosen type in the **Graphics** window or other selection methods, or select the **All domains**, **All boundaries**, **All edges**, or **All points** check box to add all of the geometric entities of that type to the list. The selected items are highlighted in the **Graphics** window. Selecting the check box for all geometric entities locks all entities as selected even if the geometry changes.

2D

For 2D models, there are no edges, and the related selection options are not available.

1D

For 1D models, there are no edges or points, and the related selection options are not available.

Important

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output type, for example, from domains to boundaries.

4 Define the geometric objects that the selection contains. The options available and defaults depend on the selection in the **Geometric entity level** list.

- If **Domain** is the input the default output is the **Selected domains**. Select other options as required. Select **Adjacent boundaries**, **Adjacent edges**, or **Adjacent points** to use the boundaries, edges, or points next to the selected domains as the selection output (available options depend on the space dimension of the model). This makes it possible to, for example, make a selection of all boundaries around a domain by first selecting the domain. Depending on the selection output, choose to include **Exterior boundaries** (the default) or **Interior boundaries**; **Exterior edges** (the default) or **Interior edges**; or **Exterior points** (the default) or **Interior points**. Click to select or clear the check boxes as needed.
- If **Boundary** is the input, the default output is the **Selected boundaries**. Select other options as required. Select **Adjacent edges** or **Adjacent points** to use the edges or points next to the selected boundaries as the selection output (available options depend on the space dimension of the model). Depending on the selection output, choose to include **Exterior edges** (the default) or **Interior edges** for edges, or **Exterior points** (the default) or **Interior points** for points. Click to select or clear the check boxes as needed.
- If **Edge** is the input, the default output is **Selected edges**.

  Select **Adjacent points** to include the points next to the selected edges as the selection output either as **Exterior points** (the default) or **Interior points**. Click to select or clear the check boxes as needed.
- The **Point** output is the same as the selection input (that is, the selected points).

Defining a Boolean Selection Node

Boolean selections are useful to combine two or more selections (union), create a selection of overlapping geometric entities (intersection), create selection of entities that are in one selection but not in another (difference), and to create a selection as the complement (inverse) of one or more selections (all boundaries except the inflow boundaries, for example). The complement can be convenient if, for example, a selection is needed that consists of all boundaries except one or a few. Then it is easier to create a selection that is the complement (just the one boundary, for example) and use a **Complement** selection node to create its complement.

The Boolean selection nodes—**Union** ( ), **Intersection** ( ), **Difference** ( ), and **Complement** ( )—contain similar sections.

1 In the **Model Builder**, right-click any **Definitions** node ( ) under a **Model**, and choose **Selections>Union** ( ), **Selections>Intersection** ( ), **Selections>Difference** ( ), or **Selections>Complement** ( ).

2 Right-click or press F2 to **Rename** the **Union** node, for example, as required.

- Select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add.

2D

For 2D models, there are no edges, and the related selection options are not available.

1D

For 1D models, there are no edges or points, and the related selection options are not available.

3 In the **Selections to add** list, for example, add the selections to form the union or other Boolean operations. Click the **Add** button (+) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Add the selections to add (union), intersect, or invert (complement). For the **Difference** selection node add selections to add and selections to subtract. Also delete selections from the lists using the **Delete** button ( ) and move them using the **Move Up** (⇧) and **Move Down** (⇩) buttons.

Important

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output type, for example, from domains to boundaries.

Defining an Adjacent Selection Node

The **Adjacent** ( ) selection type outputs selections adjacent to specified geometric entities or selections. For example, select all domains adjacent to some boundaries or all boundaries adjacent to some domains.

The adjacent geometric entities can be of any type (domains, boundaries, edges, or points) regardless of the geometric entity level for the input selections.

1. In the **Model Builder**, right-click any **Definitions** node ( ) under a **Model**, and choose **Selections>Adjacent** ( ).
2. Right-click or press F2 to **Rename** the **Adjacent** node as required.
3. Under **Input Entities**, select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add.
4. In the **Input selections** list, add the selections for which you want to select adjacent geometric entities. Click the **Add** button ( ) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Add the selections to use as the source for the selection of adjacent geometric entities. Also delete selections from the lists using the **Delete** button ( ) and move them using the **Move Up** ( ) and **Move Down** ( ) buttons.

2D

For 2D models, there are no edges, and the related selection options are not available.

1D

For 1D models, there are no edges or points, and the related selection options are not available.

Important

If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the output type, for example, from domains to boundaries.

5 Under **Output Entities**, define the adjacent geometric objects that the selection contains:

- Select **Adjacent domains** to create a selection that contains all domains adjacent to the input selections.
- Select **Adjacent boundaries** to create a selection that contains all boundaries adjacent to the input selections. If the input selections contain domains, select the **Exterior boundaries** and **Interior boundaries** check boxes to control which types of adjacent boundaries the output selection contain. By default, the output selection contains adjacent exterior boundaries only.
- Select **Adjacent edges** to create a selection that contains all edges adjacent to the input selections. If the input selections contain domains or boundaries, select the **Exterior edges** and **Interior edges** check boxes to control which types of adjacent edges the output selection contain. By default, the output selection contains adjacent exterior edges only.
- Select **Adjacent points** to create a selection that contains all points adjacent to the input selections. If the input selections contain domains, boundaries, or edges, select the **Exterior points** and **Interior points** check boxes to control which types of adjacent points the output selection contain. By default, the output selection contains adjacent exterior points only.

Defining a Box or Ball Selection Node

Another way to selection geometric entities is to define an enclosing **Ball** ( ) or **Box** ( ) to select geometric entities that are completely or partially inside the ball or box.

1 In the **Model Builder**, right-click any **Definitions** node ( ) under a **Model**, and choose **Selections>Ball** ( ) or **Selections>Box** ( ).

2 Right-click or press F2 to **Rename** the **Box** or **Ball** node as required.

3 Select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add. To base the selection on other selections, select **From selections** under **Input Entities** and then use the buttons under the **Selections** list to add selections.

4 For these selections, define the dimension of the ball or box and select the condition for the geometric entities to be selected (from the **Include entity if** list under **Output Entities**):

- Select **Entity intersects ball** or **Entity intersects box** (the default) to select all geometric entities that intersect the enclosing ball or box; that is, the selection includes all entities that are partially or completely inside the ball or box.
- Select **Entity inside ball** or **Entity inside box** to select all geometric entities that are completely inside the enclosing ball or box; that is, the selection includes all entities that are completely inside the ball or box.
- Select **Some vertex inside box** or **Some vertex inside ball** to select all geometric entities where at least some vertex is inside the enclosing ball or box.
- Select **All vertices inside ball** or **All vertices inside box** to select all geometric entities where all vertices are inside the enclosing ball or box. This selection might differ slightly compared to when selecting **Entity inside ball** or **Entity inside box** if the geometric entity is outside the ball or box at some points between the vertices.

Note

The ball and box selections use the rendering mesh to determine which entities fit the selection condition. You can control the detail for the rendering in the **Preferences** dialog box (select **Graphics** and then use the **Detail** list under **Rendering**).

5 Define the position and dimension of the enclosing box or ball using the fields for the maximum and minimum coordinates for the box or the position and radius for the ball. The inputs are based on the length unit for the geometry.

Duplicating a Selection Node from the Definitions Node Level

In the **Model Builder**, under **Definitions** right-click any **Selection** node and select **Duplicate** ( ).

Creating a Selection Node from the Selection List Window

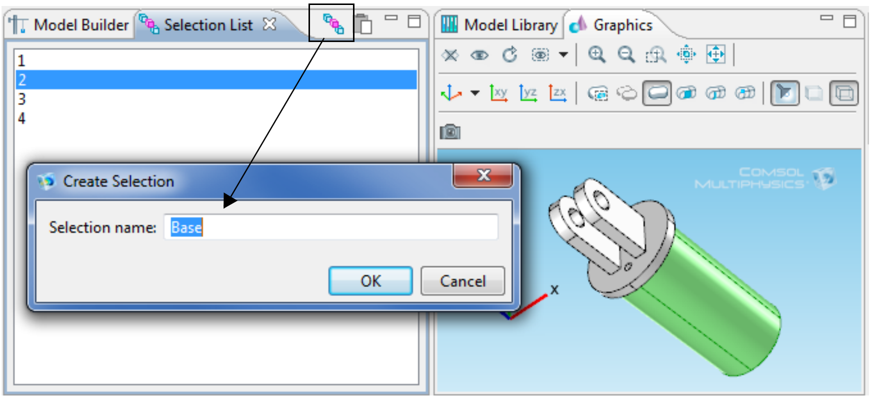
1 From the main menu, select **Options>Selection List** ( ).

COMSOL displays the geometric entities in this window based on where in the model you are working.

2 On the **Selection List** window, select the geometric entities to duplicate and add to the new **Selection** node added under **Definitions**.

3 Click the **Create Selection** button ( ) and enter a **Selection name** in the window that opens. See Figure 4-7.

4 In the **Model Builder** the new **Selection** node is added under **Definitions**.



Selection name: Base

Figure 4-7: Creating a selection from the Selection List window.

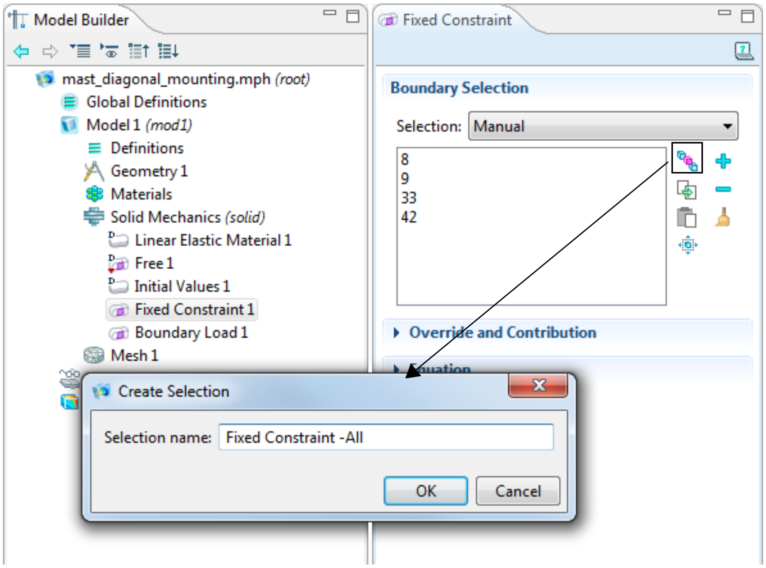
Creating a Selection Node from a Settings Window

1 At any time during model creation, click a feature node that has the option to add a geometric entity to a selection (for example, under the **Materials** node or the **Solid Mechanics** physics interface node).

2 In the settings window that opens, select an option from the **Selection** list, for example, **Manual** or **All boundaries**.

3 Click the **Create Selection** button ( ) and enter a **Selection name** in the window that opens. See Figure 4-8.

4 In the **Model Builder** the new **Selection** node is added under **Definitions**.



Selection: Manual

8
9
33
42

Selection name: Fixed Constraint -All

Figure 4-8: Creating a selection from the settings window for a Fixed Constraint feature.

Creating Selections From Geometric Primitives and Features

For all geometric primitives in 3D, 2D, and 1D geometries—for example, blocks, spheres, squares, polygons, and intervals—as well as for all geometry features—for example, union, intersection, fillet, array, and mirror—create selections for each type of geometric entity that the resulting geometry objects consist of. The following list shows the geometric entity types for geometry objects that are “solids” in each space dimension:

- 3D: domains, boundaries, edges, and points
- 2D: domains, boundaries, and points
- 1D: domains and points

To create these selections, click the geometry node in the **Model Builder** and then select the **Create selections** check box in the settings window’s **Selections of Resulting Entities** section.

The selections become available with the name of the geometry node (Block 1, for example) in all applicable **Selection** lists in the settings windows for the physics interface nodes, for example. There are no explicit Selection nodes for these selections.

Also create an explicit **Object Selection** node ( ) for selecting all or a few of the geometric entities at a specific geometric entity level for one or more of the nodes above the **Object Selection** node in the geometry sequence. This way it is possible to make a selection that only includes a few of the geometric entities from one or more geometry objects.

Using selections based on a geometry object makes it possible to track, for example, all boundaries in the final geometry that belong to that geometric object, even if its original boundaries are intersected by other geometry objects.

Pasting Selections

If you have a list of geometric entities (boundaries, for example) in a file or document you can copy it to the clipboard and then use Ctrl+V to directly paste that list into a selection list in a settings window. An example of this is a list of geometric entity numbers described as a step in a COMSOL Multiphysics modeling instruction.

You can also use the **Paste Selection** button ( ) if you have a list of geometric entities that you want to paste into a selection list. The button is available in the same places as the **Create Selection** button (see Figure 4-6, Figure 4-7, and Figure 4-8) You can type selection data directly into the **Paste Selection** window that opens when the button is clicked.

PASTING GEOMETRIC ENTITY INFORMATION INTO A SELECTION LIST

1 Prepare or copy the information to insert into the selection list. For example, copy a list of numbers from a text file or PDF file (highlight and press Ctrl+C).

2 On the window or page next to the selection list where you want to add a selection from file, press Ctrl+V to paste the selection directly, or click the **Paste Selection** button ( ).

3 In the **Paste Selection** window, paste (press Ctrl+V) or enter the list of geometric entities into the **Selection** field. Data in the list or entered in the field can include commas and spaces as separators (1, 3), ranges (10–34), and words (*and*). Click **OK** to paste the selection into the selection list.

User-Defined Views

In this section:

The View Nodes

By default a **View** node is added to all 2D ( ), 2D axisymmetric ( ), and 3D ( ) models.

2D

2D Axi

For 2D and 2D axisymmetric models, an **Axis** ( ) subnode is also added where you can set the axis coordinates and manual spacing.

The **View** nodes and subnodes have information about a view. For 2D and 2D axisymmetric models, it controls the settings to display or hide geometry labels and direction arrows, and lock the axis.

3D

For 3D models, a **Camera** subnode ( ) and three **Directional Light** nodes ( ), including default settings, are also added.

The **View** nodes and subnodes have information about a view. For 3D models the settings include (in addition to, for example, settings for displaying geometry labels and direction arrows) transparency, lighting sources (directional lights, headlights, spotlights, and point lights), lighting attributes (diffuse, specular, and ambient light), and camera settings.

Also right-click the **View** node to add **Hide Geometric Entities** and **Hide Geometry Object** nodes for any space dimension.

If a **View** is customized—for example, by renaming it *Transparency with spotlight*—it can be selected from the list of views ( ) on the **Graphics** toolbar. Create additional **View** nodes by right-clicking **Definitions** ( ) and selecting **View** and then experiment by switching views to find the best way to illustrate a model.

Note

To display the **Views** node under **Results** ( ), click the **Show** button ( ) and select **Advanced Results Options**. This is useful, for example, when 2D axisymmetric revolved plots or 2D cut plane plots for 3D models are created.

To reset the **View** node's settings and its subnodes to the default settings, right-click the **View** node and select **Reset to Default** ( ).

Editing the 2D View and Axis Nodes

2D

2D Axi

For 2D and 2D axisymmetric models, the **View** node ( ) controls the settings to display or hide geometry labels and lock the axis.

Also right-click the **View** node to add **Hide Geometric Entities** and **Hide Geometry Object** nodes.

1 To add additional **View** nodes, in the **Model Builder**, right-click **Definitions** ( ) and select **View** ( ) or click **View** to edit the default node.

2 Click to select the **Show geometry labels** to display the geometry short name labels in the **Graphics** window. Depending on what you are selecting, the **Graphics** window shows labels for the geometry objects or geometric entities (domain labels, boundary labels, edge labels, or point labels).

3 Click to select the **Show direction arrows** to display the direction arrows in the **Graphics** window.

4 Select the **Lock axis** check box to store the current axis limits so that the zoom tools can temporarily be used, for example, but by revisiting the **View** node restore the axis limits to the values in the view at the time the **Lock axis** check box is selected.

5 To define the **Axis**, in the **Model Builder**, under **Definitions>View**, click the **Axis** node ( ).

6 In the **Axis** section, enter **x minimum**, **x maximum**, **y minimum**, and **y maximum** coordinates for 2D models and **r minimum**, **r maximum**, **z minimum**, and **z maximum** coordinates for 2D axial symmetry.

7 Select the **Manual spacing** check box to enter **x spacing** and **y spacing** coordinates for 2D models and **r spacing** and **z spacing** coordinates for 2D axial symmetry, to control the grid spacing manually.

8 Under **Grid**, enter **Extra x** and **Extra y** values for 2D models and **Extra r** and **Extra z** for 2D axisymmetric models if extra grid points are required in addition to the ones generated by the settings for the grid spacing. Click the **Range** button ( ) as required.

Note

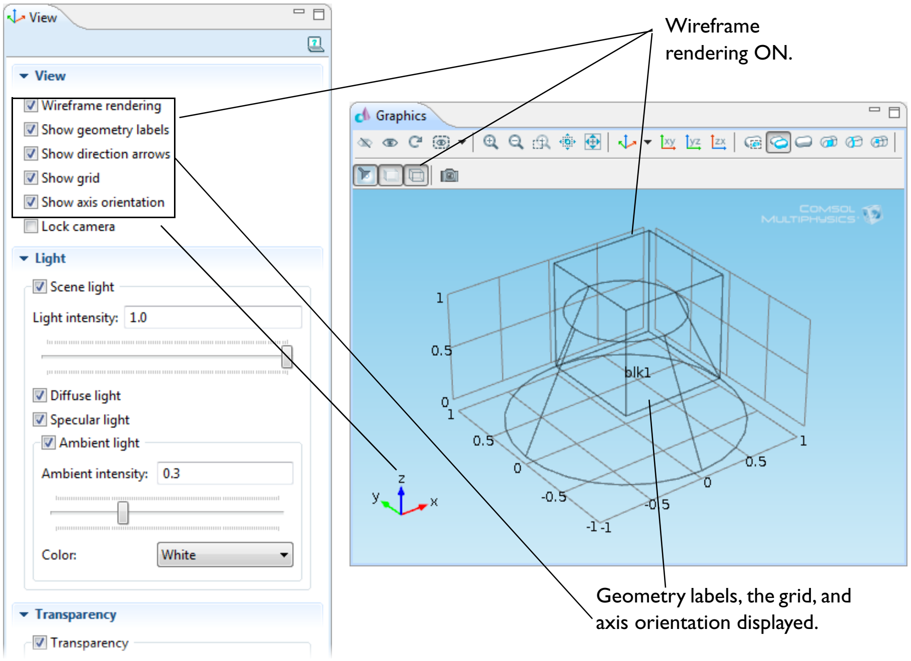
The default precision for the 2D grid axis labels is 4 digits. You can change the precision in the **Preferences** dialog box, using the **2D axis** field under **Precision** on the **Graphics** page.

About 3D Views

The **3D View** node ( ) has many options to add light sources and define the light attributes. Other functions include displaying or hiding geometry labels, transparency, wireframe rendering, a numbered grid, and axis orientation features in the **Graphics** window. See Figure 4-9. The rest of this section describes the available 3D View features.

See Also

- About the 3D View Light Sources and Attributes
- Editing the 3D View Camera Node
- Editing the 3D View Node
- Editing the 3D View Light Source Nodes
- Hiding Geometry Objects and Geometric Entities in a View



View

View

☑ Wireframe rendering
☑ Show geometry labels
☑ Show direction arrows
☑ Show grid
☑ Show axis orientation
□ Lock camera

Light

☑ Scene light

Light intensity: 1.0

☑ Diffuse light
☑ Specular light
☑ Ambient light

Ambient intensity: 0.3

Color: White

Transparency

☑ Transparency

Wireframe rendering ON.

Geometry labels, the grid, and axis orientation displayed.

Figure 4-9: An example of a 3D View settings window with all View option check boxes selected. The Graphics window shows what the check boxes represent.

Note

The default precision for the 3D grid axis labels is 3digits. You can change the precision in the **Preferences** dialog box, using the **3D grid** field under **Precision** on the **Graphics** page.

About the 3D View Light Sources and Attributes

Light sources are the **Directional Light**, **Point Light**, **Spotlight**, or **Headlight** nodes. *Light attributes* are the scene light components, which include diffuse, specular, and ambient light (see Figure 4-10). Combined, these attribute and source settings enable the software to render the 3D model to look realistic.

LIGHT ATTRIBUTES

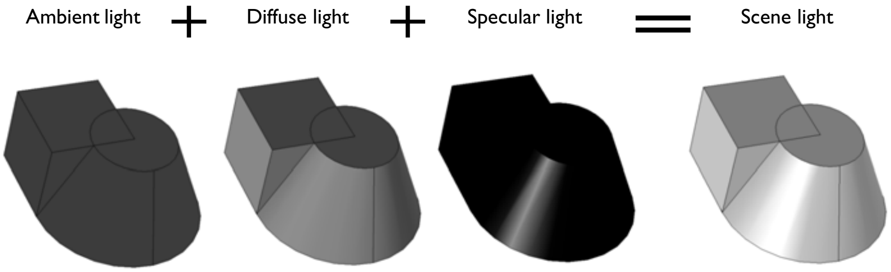


Figure 4-10: Scene light is a combination of ambient, diffuse, and specular light attributes. The default Scene light color is white and in this example the attributes display as different shades of black and gray. The Scene light is further enhanced using the various light source nodes.

Scene Light

For all geometry, scene light is applied by default and is a combination of the different light attributes (or types)—diffuse, specular, and ambient light. The light intensity and ambient intensity levels are also attributes of the scene light. The diffuse, specular, and ambient light attributes can be turned ON or OFF by selecting or clearing the corresponding check boxes (see Editing the 3D View Node). The intensity levels are adjusted either with a slider or by entering a number between 0 and 1.

Think of **Scene light** as being comprised of ambient light, the base amount of light, plus specular light to add depth to curves and diffuse light to soften the lighting and add contrast. See Figure 4-10 for examples of the attributes:

- **Ambient light** is the available light surrounding the geometry. By itself, ambient light makes a 3D object look like a 2D object. The addition of diffuse and specular light adds the contrast and depth needed to define 3D geometry.
- **Diffuse light** is directional and spreads out over the object, like a flashlight shining on a sphere. This generally adds contrast and depth of field to 3D objects.
- **Specular light** is directional and reflects off the surface of a sphere or curve in a geometry. It is based on the angle between the viewer and the light source.

Apply the different sources of light (**Directional Light**, **Point Light**, **Spotlight**, or **Headlight** nodes) to further enhance how the geometry displays.

LIGHT SOURCES

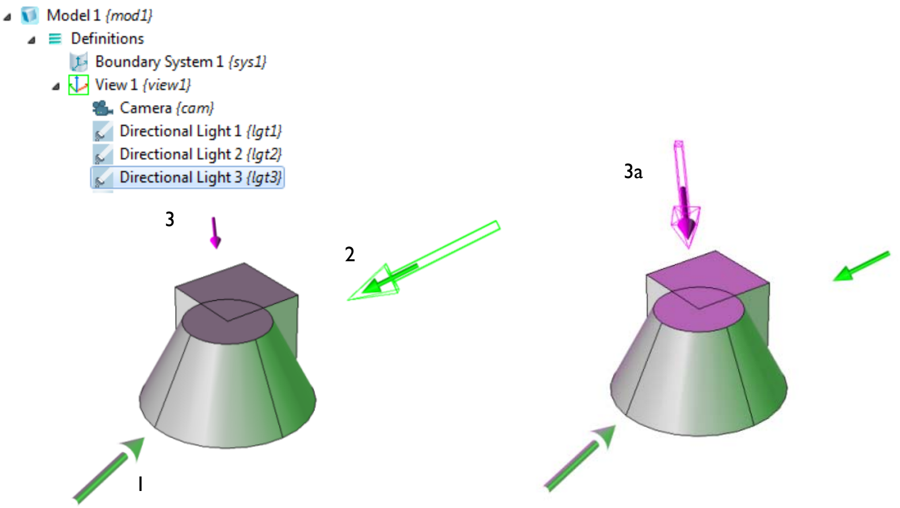
3D

For 3D models, also add these light source nodes—**Directional Light** ( ), **Point Light** ( ), **Spotlight** ( ), and **Headlight** ( )—to adjust how the color and shading displays in the Graphics window using different combinations of these settings. Each light source is described below. See also Editing the 3D View Light Source Nodes.

Each light source has a unique light marker displayed in the Graphics window, which can be set to different colors to determine the angle and intensity to display the light, for example. Figure 4-11 shows three directional light settings and markers displayed in the Graphics window. The light markers are placed at the user-defined x, y, and z coordinates and are used to adjust the light and specular intensities on the object. You can show or hide the markers and change the color. The color of the marker corresponds to the light hitting the object.

Directional Light

A *directional light* represents light that falls from a direction on all objects in the scene, like sunlight where all the light rays are parallel. Directional lights therefore have no position. Figure 4-11 is an example of three **Directional Light** ( ) nodes where the color is changed and the light intensity adjusted for Directional Light 3. The markers are labeled 1, 2, and 3 to correspond to the nodes. The **Directional Light 3** node's light intensity setting is changed from 0.24 to 1—the change in arrow size corresponds to the increase in light intensity. The wireframe around a light marker means that the corresponding node is selected in the **Model Builder**. Adjust the direction of the light, the light and specular intensity levels, and the color as required.



Model 1 *{mod1}*
Definitions
Boundary System 1 *{sys1}*
View 1 *{view1}*
Camera *{cam}*
Directional Light 1 *{lgt1}*
Directional Light 2 *{lgt2}*
Directional Light 3 *{lgt3}*

Figure 4-11: Examples of directional light markers and the location of each directional light. The markers and light can be color coded to see which areas need adjustment. The markers indicate which node is selected in the Model Builder (a wireframe around the arrow in 2 and 3a), and the level of light intensity applied (the size of the arrow; compare 3 and 3a).

Point Light

A *point light* is like a light bulb. The light has a position and emits the light equally in all directions. Point light have therefore no direction. Figure 4-12 is an example of two **Point Light** ( ) nodes added to the **View** and all other nodes disabled. The position of the light, the light and specular intensity levels, and the color can be adjusted as required.

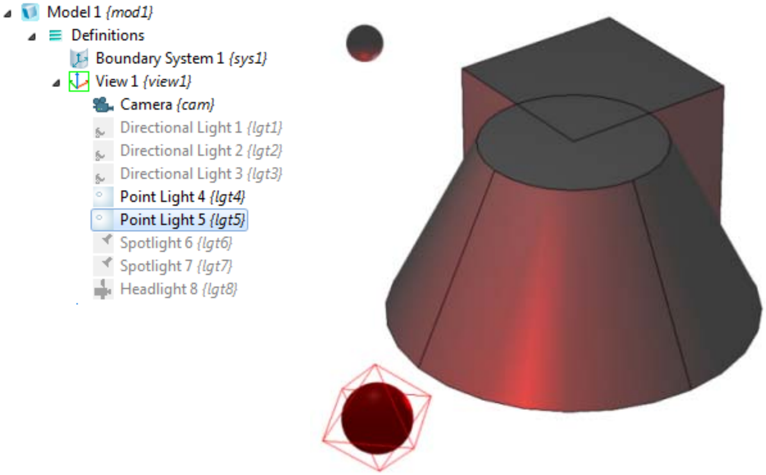


Figure 4-12: An example of two Point Light nodes added to a View. All other nodes are disabled to show only the Point Light effects to the geometry. Point Light 5 is selected in the Model Builder as indicated by the wireframe around the light marker.

Spotlight

A *spotlight* is like a flashlight and has both a position and a direction. Figure 4-13 is an example of two **Spotlight** ( ) nodes added to a **View** with all other nodes disabled. The position and direction of the light, the light and specular intensity levels, and the color can be adjusted as required. In addition, the spread angle can be adjusted as shown in the figure. The width of the light marker corresponds to the spread angle.

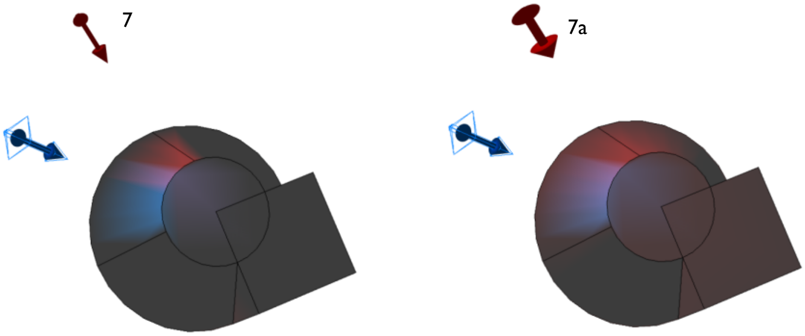


Figure 4-13: Two Spotlight nodes displayed with all other nodes disabled showing where the Spotlight is focused. When the spread angle is changed from 20 to 100 for Spotlight 7 (as indicated by the change in marker width between 7 and 7a), a corresponding change is made to the model.

Headlight

A *headlight* is a directional light, which points to the scene from the camera position. The **Headlight** ( ) is similar to the **Directional Light** with the only difference being that it is always locked to the camera's coordinate system and a direction pointing from the camera is used. Figure 4-14 is an example of one **Headlight** node added to a **View** with the **Directional Light** nodes enabled. The headlight's position and direction cannot be changed; it is based on the **Directional Light** node's (or nodes') *x*, *y*, and *z* coordinates. The light and specular intensity levels and the color can be adjusted as required.

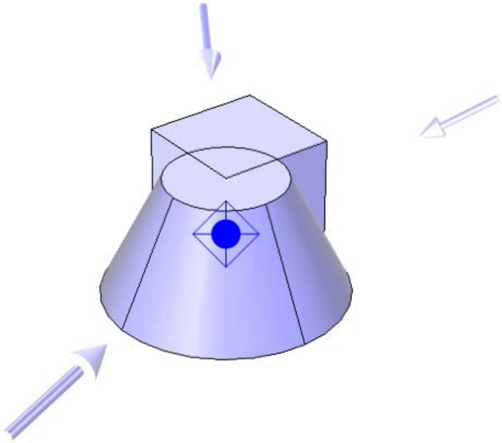


Figure 4-14: An example of a blue Headlight source with three Directional light sources set to white. If the geometry is rotated, you can adjust and view the effects of the Headlight source on the geometry based on the shades of blue and white.

Editing the 3D View Node

The 3D **View** node ( ) has the following sections (see Figure 4-9):

VIEW

- Select the **Wireframe rendering** check box to view the edges of the object as solid lines. The **Wireframe Rendering** button ( ) is turned on or off in the **Graphics** window at the same time.
- Select the **Show geometry labels** check box to display the geometric short name in the **Graphics** window.
- Select the **Show direction arrows** check box to display direction arrows on edges in the **Graphics** window. The direction arrows indicate the directions for which the edge parameterization values increase.
- By default the **Show grid** check box is selected and displays a numbered grid in the **Graphics** window around the object. Click to clear the check box to hide the grid.
- By default the **Show axis orientation** check box is selected and the axis orientation indicator for the global Cartesian coordinate directions is displayed in the lower-left

corner of the **Graphics** window. Click to clear the check box to hide the axis orientation indicator.

- Select the **Lock camera** check box to store the current camera settings so that the zoom tools can temporarily be used, for example, but by revisiting the **View** node restore the camera settings s to the values in the view at the time the **Lock camera** check box is selected.

LIGHT

The **Scene light** setting is a default that always displays and is based on the geometry. The **Scene light**, **Diffuse light**, **Specular light**, and **Ambient light** check boxes are selected by default. To hide and disable all light sources, click to clear the **Scene light** check box. The **Scene Light** button ( ) is turned on or off in the **Graphics** window at the same time.

- Click to clear the **Diffuse light**, **Specular light**, and **Ambient light** check boxes as required.
- Enter a value for the **Ambient intensity** or use the slider to select a level. Watch the changes in the **Graphics** window at the same time to help choose a level.
- Select a **Color** from the list—**Custom**, **Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White** (default), or **Yellow**. The color is only applied to ambient light. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.

TRANSPARENCY

Select the **Transparency** check box to turn on transparency. The **Transparency** button ( ) is activated in the **Graphics** window at the same time. Enter a value between 0 and 1 in the **Alpha** field for the alpha blending, where 0 means a fully transparent color and 1 means a fully opaque color, or use the slider to select a transparency level. Watch the changes in the **Graphics** window at the same time to help choose a level.

See Also

Editing the 3D View Camera Node

Editing the 3D View Camera Node

Use the **Camera** node ( ) to orient the camera view in 3D models.

1. In the **Model Builder**, under **Definitions** click to expand the **View** node where you want to define the camera position.
2. Click the **Camera** ( ) node.
3. In the **Camera** section, select **Perspective** (the default) or **Orthographic** (parallel) from the **Projection** list as required. The perspective projection shows distant objects as smaller to provide a realistic 3D projection, whereas the orthographic projection shows all objects using their actual size regardless of the distance. Also enter a **Zoom angle** or use the **Zoom** buttons on the **Graphics** toolbar. The **Preserve aspect ratio** check box is selected by default to make the increments equal in the *x*- and *y*-directions. Click to clear this check box to make the geometry fill the graphics window with unequal increments. This can be useful when working with slender geometries.
4. In the **Graphics** window, left-click and hold the mouse to orient the geometry on the axes. The corresponding coordinates are displayed in the settings window under the **Position**, **Target**, and **Up Vector** sections. Or enter **x, y**, and **z** coordinates in each section.
   - The **Position** is the location of the camera and the **Target** default is 1 length unit in front of the camera position. The **Up Vector** value assigns the orientation of the camera roll direction, usually in the *z* plane.
5. By default the values in the **Center of Rotation** section define the center of the geometry in the view. Explicitly control the center of rotation by entering a center location in the **x**, **y**, and **z** fields and then click the **Apply** button ( ) at the top of the settings window.
6. Right-click the mouse and shift the geometry left, right, up, or down as required. This shift operation moves the currently visible frame on the image plane. The corresponding dimensionless values that display in the settings window under **View Offset** are relative to the image width and height, respectively. Alternatively, enter **x** and **y** values for the view offset. The value in the **Orthographic scale** field defines the size in scene length of the viewing block along the longest side of the canvas. If the view's camera setting uses orthotropic projection, zoom in or out by increasing or decreasing the value of the orthographic scale. For a perspective projection this setting has no effect.

Editing the 3D View Light Source Nodes

For 3D models, add these light source nodes—**Directional Light** ( ), **Point Light** ( ), **Spotlight** ( ), or **Headlight** ( )—to adjust how the color and shading displays in the **Graphics** window using different combinations of these settings. Add a maximum of eight light sources (nodes) in any combination to each View. For example, add three Directional Light, two Point Light, two Spotlight, and one Headlight node combination.

POSITION

This setting is available for the **Point Light** ( ) and **Spotlight** ( ) features. Enter **x**, **y**, and **z** coordinates for the position of the light source.

DIRECTION

This setting is available for the **Directional Light** ( ) and **Spotlight** ( ) features.

The **x**, **y**, and **z** coordinates define the direction in which the light falls on the objects in the scene. For directional lights the arrow is a visualization of that direction with a placement that is calculated automatically depending of the scene's bounding box. The arrow points in the direction of the light.

Tip	Change the color to something other than white to observe where the light hits the object.

SETTINGS

Specify a number of settings for the light.

Spread Angle

This setting is available for the **Spotlight** ( ) feature. Enter a **Spread angle** (in degrees). Also watch the changes in the **Graphics** window at the same time to help choose an angle. When adjusting the spread angle, the arrow increases and decreases in width as the angle value increases and decreases. The default is 20 degrees. See Figure 4-13 for an example.

Light Intensity

This setting is available for all light sources. Enter a **Light Intensity** or use the slider to select values between 0 and 1 (default value: 0.5). Watch the changes in the **Graphics** window at the same time to help choose a level. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object.

Specular Intensity

This setting is available for all light sources. **Specular Intensity** is the intensity of the light that reflects off the surface of the geometry and is used to define spheres and curves. This setting is turned ON and OFF on the View page (the default is ON), and the levels are adjusted for each **Specular intensity** using the field or the slider to select values between 0 and 1 (default value: 1). Watch the changes in the **Graphics** window at the same time to help choose a level. See Figure 4-10 for an example of specular light.

Color

This setting is available for all light sources. Select a **Color**—**White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. See also Color for the **View** node's settings window to set the color of the scene light.

Lock to Camera Coordinate System

This setting is available for all light sources. By default, the **Lock to camera coordinate system** check box is selected to make the light rotate together with the camera. Click to clear as required; the light then follows the geometry.

See Also

To lock the camera settings, see Editing the 3D View Camera Node.

Show Light Marker

This setting is available for all light sources. By default, the **Show light marker** is selected. The light marker is associated to the type of light (directional light, point light, spotlight, or headlight) being applied to the object. See Figure 4-15 for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

A wireframe around a light marker indicates that its node is selected in the Model Builder. When adjusting the light intensity, the light marker changes in length as the corresponding intensity is changed on the object. When adjusting the spread angle for a spotlight, the arrow increases and decreases in width as the angle value increases and decreases.

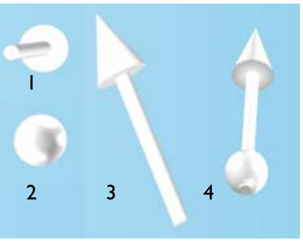


Figure 4-15: The light markers for each type of light source (1) Headlight (2) Point light (3) Direction light, and (4) Spotlight.

Hiding Geometry Objects and Geometric Entities in a View

Hide a set of geometry objects or some of their geometric entities—in the created or imported geometry—or geometric entities—in the analyzed (finalized) geometry used for modeling—at a specific geometric entity level for any View.

Note

The **Hide Geometry Objects** feature hides geometry objects or parts of such objects and affects the the view in the Geometry branch only. The **Hide Geometric Entities** feature hides geometric entities (domains, boundaries, edges, or points) in the analyzed geometry and affects the view in the Materials, physics interfaces, Mesh, studies, and Results branches.

1 In the **Model Builder**, under **Definitions** click to expand the **View** node where you want to hide the geometry objects or entities. See also About Geometric Entities.

2 Right-click **View** and select:

- **Hide Geometry Objects** ( ) to hide geometry objects or some of their geometric entities in a **View** setting when viewing the geometry in the **Geometry** branch. For example, hide a geometry objects, or hide a boundary of a 3D geometry object to make it possible to view the inside of the object.
- **Hide Geometric Entities** ( ) to hide specific geometric entities (in the analyzed geometry) in a **View** setting when viewing the model for defining, for example, materials, physics, and meshes, and for results and plotting. For example, hide some domains that are not of interest in the model, or some boundaries only on a specific domain to get a clearer view of some of the model. You can include the

hidden objects (entities) in a plot by selecting the **Show hidden objects** check box in the corresponding plot group's settings window.

3 If **Hide Geometry Objects** is selected, in the **Selection** section select a **Geometric entity level** to hide—**Object** (the default), **Domain**, **Boundary**, **Edge**, or **Point**. Then select the geometry objects or some of their geometric entities that you want to hide. The list under the **Geometric entity level** list contains the selected objects. Use the **Add to Selection** ( ), **Remove from Selection** ( ), **Clear Selection** ( ), and **Zoom Selected** ( ) buttons as required.

If **Hide Geometric Entities** is selected:

a From the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge**, or **Point**. Selection from this list activates the selection button with the same name in the **Graphics** window's toolbar.

b Select **Manual** (the default) or **All domains**, **All boundaries**, **All edges**, or **All points** from the **Selection** list. The list under the **Selection** list contains the selected entities that the View hides when working with the materials, physics, meshes, studies, and results.

- If **Manual** is selected, go to the **Graphics** window and select the geometric entities that you want to hide.
- Select **All domains**, for example, to hide all applicable geometric entities (domains, boundaries, edges, or points) by adding them to the list of selected entities. Use the **Create Selection** ( ), **Copy Selection** ( ), **Paste Selection** ( ), **Zoom Selected** ( ), **Add to Selection** ( ), **Remove from Selection** ( ), and **Clear Selection** ( ) buttons as required.

Capturing and Copying Screenshots

To quickly capture a screenshot image of a plot, press Ctrl+C when the Graphics window or another plot window has focus. The screenshot image is then available on the clipboard so that you can paste it into, for example, a document. Also use the **Image Snapshot** button (📷) on the Graphics toolbar to capture an image snapshot of a plot. To do so, follow these steps:

1. In the **Graphics** window ( ) or any other plot window, click the **Image Snapshot** button (📷) to open the **Image Snapshot** dialog box.
2. Under **Image**, from the **Size** list select **Manual** (the default) to define the image size manually using the settings below, or select **Current** to use the current size of the **Graphics** window. Of the settings below, only the **Antialiasing** check box is then available.
3. Select a **Unit** to define the image size—**Millimeters (mm)**, **Inches (in)**, or **Pixels (px)** (the default).
4. Select the **Lock aspect ratio** check box to maintain the calculation of the width and height (if one or the other is changed).
5. Enter the **Width** and **Height** in the units selected for the image.
6. Enter the **Resolution** for the image in DPI (dots per inch). The default value is 96 DPI.
7. The **Antialiasing** check box is selected by default. Click to clear if required. Antialiasing minimizes distortion such as jagged edges in the image.
8. Under **Layout**, the **Title**, **Legend**, **Axes**, and **Logotype** check boxes (1D and 2D images) or the **Title**, **Legend**, **Grid**, **Axis orientation**, and **Logotype** check boxes (3D images) are selected by default to display the information on the screenshot if you select the **Include** check box. You can then also edit the selections for including these parts of the plot.
9. Enter a **Font** size in points (pt). The default value is 9 pt.
10. Select a **Background**—**Current** or **Color**. Current is the background color in the plot window on the COMSOL Desktop. If **Color** is chosen, click the **Color** button to select a custom color from the color palette that opens.
11. Under **Output**, select the target: **Clipboard** (the default) copies the image to the clipboard. **File** saves the image to a file.

12 If **File** is selected:

a Select a file **Format**—**BMP**, **JPEG**, or **PNG** (the default).

b Enter a file path in the **Filename** field, or click **Browse** to specify the name and location of the file.

13 Click **OK** to generate the image snapshot.

See Also	Printing from the COMSOL Desktop

5

Geometry Modeling and CAD Tools

The CAD tools in COMSOL Multiphysics provide many possibilities to create geometries using solid modeling and boundary modeling. This chapter covers geometry modeling in 1D, 2D, and 3D with examples of solid modeling, boundary modeling, Boolean operators, and other CAD tools in COMSOL Multiphysics. In addition, it shows how to use the tools for exploring geometric properties, such as volumes and surfaces. There is also information about using external CAD data. For details about the different geometry primitives and operations, see Geometry Modeling Features in the *COMSOL Multiphysics Reference Guide*.

In this chapter:

Creating a Geometry for Successful Analysis

Several techniques can ensure that a geometry results in a good mesh and gives reasonable solution times for the finite element analysis. They include the use of symmetry and eliminating small details, gaps, holes, and singularities.

USING SYMMETRIES

Using symmetry is one of the most effective ways to reduce the size of a finite element model. For axially symmetric geometries, a 2D axisymmetric model is sufficient. You can easily visualize the results in a full 3D geometry using a Revolution 2D data set. Other common cases of symmetry are sector symmetry and symmetry and antisymmetry planes, which can reduce the size of a 3D model.

MAKING THE GEOMETRY MATCH THE BOUNDARY CONDITIONS

In many solutions, the modeling domain is unbounded or too large for successful meshing and analysis. For these cases a suitable boundary condition can replace the exterior of the domain.

❗ *Important*

It is important for the geometry to be large enough to validate the boundary conditions.

For outflows in fluid-flow models, for example, the boundary should be perpendicular to the fully developed flow. Inspections and modifications of the solved model might be necessary to verify the validity of the boundary condition. Also, for some applications, infinite elements or perfectly matched layers (PMLs) are available for modeling diffusion or wave propagation in unbounded domains.

AVOIDING EXCESSIVELY SMALL DETAILS, HOLES, AND GAPS

Many geometries, especially those designed using a CAD system without finite element analysis, contain small holes, details, and gaps. These small features can make the domain unbounded and must be removed before analysis. Small details and holes can lead to large meshes and finite element models or even failure during mesh generation. Make sure the snapping feature is activated to avoid small gaps and mismatches between the geometry objects.

The CAD Import Module contains tools for automatic and interactive repair and defeaturing of 3D CAD data.

For a 2D or a 3D model you can also remove small details and prepare the geometry for efficient meshing by using *virtual geometry operations*.

AVOIDING SINGULARITIES AND DEGENERACIES IN THE GEOMETRY

A singularity in a geometry is a sharp corner or angle that can create problems during meshing and analysis. In reality, a sharp reentrant corner leads to infinite stress values in a stress analysis of a perfectly elastic material. To avoid a singularity, round sharp corners using fillets. The stress value for a sharp corner is finite in the finite element analysis, but refinement of the mesh increases the stresses in the corner without limit.

A degeneracy in the geometry can occur during solid modeling. For example, fillet areas that taper to a point and the apex of a cone can become degenerate points. These degeneracies might cause problems for the mesh generator and problems during the analysis of the model. A common degeneracy in the geometry occurs when a 3D solid is created (for example, a cylinder) by rotation about an axis that touches the rotation area. In such cases, it is better to create the solid object by extruding a cross section or to use geometric 3D primitives.

ASSOCIATIVE GEOMETRY

Associative geometry is a concept for the automatic updating of applied physical properties, such as boundary conditions and equation coefficients, under geometric transformations. Thus, once you have defined the physical properties of a finite element model and return to the Geometry branch to modify the geometric model, COMSOL Multiphysics updates the physical properties according to the geometry modifications.

The associative geometry functionality utilizes geometry-mapping information between the groups of geometric entities (points, edges, boundaries, and domains) in the *finalized geometry* and the corresponding groups in the geometric model.

This geometry mapping is not always without ambiguities. Thus, COMSOL Multiphysics makes some heuristic decisions when mapping the physical properties between the finalized geometry (the object on which the physical properties are imposed) and the geometric model. In some cases the resulting updated physical properties might not be the ones that are expected.

Working with the Geometry Sequence

Once you have added a model, a Geometry node, representing the *geometry sequence* of the model, appears in the window. Initially, a geometry sequence only contains a **Form Union** node (see Finalizing the Geometry for more information on this node). You create the model's geometry by adding nodes to the sequence and building them. The following 2D example illustrates how to work with the geometry sequence. The example includes the following sections:

- Adding a Model Geometry
- Adding and Building Nodes
- Editing Nodes
- The Current Node
- Errors and Warnings

Adding a Model Geometry

1 Start a new model by clicking the **New** button (□) in the main toolbar.

2 In the **Model Wizard** window, select **2D** as space dimension and click **Finish** (🏁).

A new Model node appears in **Model Builder** containing a geometry sequence node **Geometry 1**.

Adding and Building Nodes

1 Add a **Square** node to the sequence by right-clicking **Geometry 1** and selecting **Square** (□).

2 Click the **Build Selected** button (▦) in the settings window to build the node and to create a unit square.

3 Add a Circle node by right-clicking **Geometry 1** and selecting **Circle** (○).

4 Set **Radius** to 0.25 and in the **Size and Shape** section and **y** to 0.5 in the **Position** section.

5 Add a Difference node by right-clicking **Geometry 1** and selecting **Difference** (⧉) in the **Boolean Operations** menu.

Note

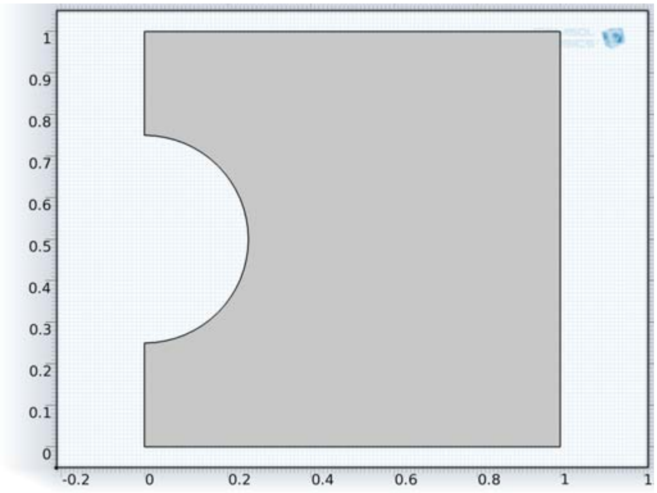
When you add the Difference node, the software builds the Circle node (and all its preceding nodes) before showing the settings window for the Difference node.

6 Select the object **sq1** and right-click to add it to the **Objects to add** list.

7 Click the **Activate Selection** button (⏻) in the **Objects to subtract** section and add **c1** to its list.

8 Click **Build Selected** (▦) to build the node and to subtract the circle from the square.

9 Click the **Zoom Extents** button (✥) in the Graphics toolbar to view the resulting object.

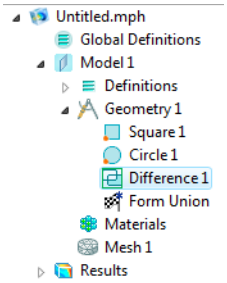


Also notice that when selecting a node that represents a geometry object in the Model Builder highlights that geometry object in the Graphics window.

Note

If you select multiple nodes in the Geometry branch, the settings window and the highlighting in the Graphics window represent the geometry object for the topmost of the selected nodes only.

The **Model Builder** shows the three added nodes.

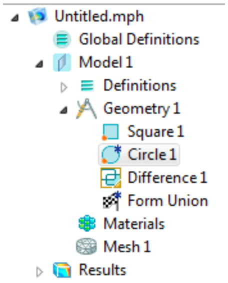


Editing Nodes

To edit a node, select it in **Model Builder**, and make changes in the settings window that appears. Nodes that you have edited display with an asterisk (*) at the upper-right corner of their icons in the **Model Builder** window. Feature nodes that depend on edited nodes display with a yellow triangle at the lower-right corner of the feature node's icon to indicate that they need to be rebuilt.

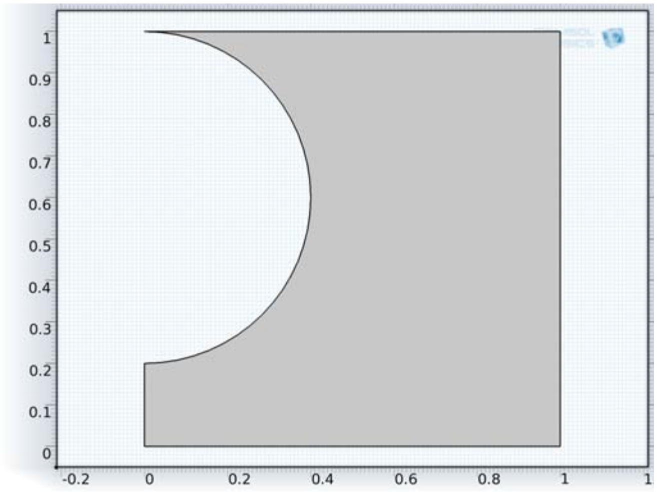
1 Select **Circle 1** in **Model Builder** and set **Radius** to 0.4 in the **Size and Shape** section and **y** to 0.6 in the **Position** section of the settings window.

An asterisk and a yellow triangle appears in **Model Builder** in the icons of **Circle 1** and **Difference 1**, respectively.



2 Right-click **Geometry 1** and select **Build All** to build all nodes in the sequence (except **Form Union**) using the new settings.

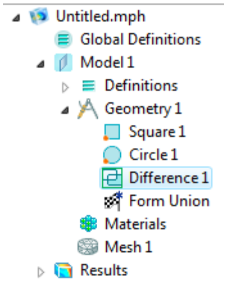
The following geometry appears in the **Graphics** window.



The icons of **Circle 1** and **Difference 1** in **Model Builder** now appear without the asterisk and the yellow triangle, respectively.

The Current Node

When you add a geometry node, it is inserted in the sequence after the *current node*. To indicate the current node, it appears with a quadratic frame around its icon. When you have added a node, it automatically becomes the current node. If you select a node and build it, this node becomes current. At this stage **Difference 1** is the current node in this example. In **Model Builder** you can see a quadratic frame around its icon.



The frame is green to show that the current node is built. If the current node needs to be rebuilt the frame becomes yellow.

ADDING A NODE AT AN ARBITRARY POSITION

Add a Rectangle node between **Circle 1** and **Difference 1**.

3 Right-click **Difference 1** in **Model Builder** and select **Build Preceding** ( ).

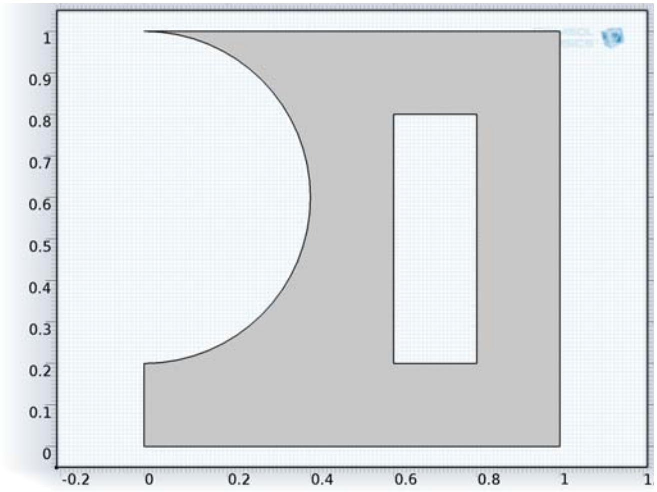
4 Right-click **Geometry 1** and select **Rectangle** ( ).

5 Set **Width** to 0.2 and **Height** to 0.6 in the **Size** section and set **x** to 0.6 and **y** to 0.2 in the **Position** section.

6 Click the **Build Selected** button ( ) in the settings window to build the node and to create the rectangle.

7 Select **Difference 1** in **Model Builder**, click the **Activate Selection** button ( ) in the **Objects to subtract** section, add **r1** to its list, and click **Build Selected** ( ).

The following geometry appears in the **Graphics** window.

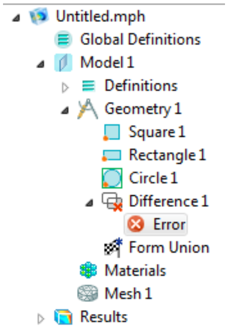


Errors and Warnings

If an error occurs when you build a node, the build stops. The node with the problem then gets an **Error** subnode ( ) that contains the error message. Also, the feature node's icon appears with a red cross in the lower-left corner.

1 Select **Difference 1** in **Model Builder**, click the **Activate Selection** button ( ) in the **Objects to subtract** section, click the **Clear Selection** button ( ), and click **Build Selected** ( ).

Note that an Error node appears in **Model Builder**.



Untitled.mph
Global Definitions
Model 1
Definitions
Geometry 1
Square 1
Rectangle 1
Circle 1
Difference 1
Error
Form Union
Materials
Mesh 1
Results

After a successful build of a node, a warning message can sometimes appear as a **Warning** subnode (⚠). If a warning message exists, the node's icon appears with a yellow triangle in the lower-left corner.

The COMSOL Multiphysics Geometry and CAD Environment

In this section:

See Also

- About Sequencing in COMSOL Multiphysics
- Opening a Context Menu to Add Nodes
- Moving Nodes in the Model Tree
- Copying, Pasting, and Duplicating Nodes
- Expanding and Collapsing All Nodes
- Geometry Modeling Features in the *COMSOL Multiphysics Reference Guide*
- Geometry in the *COMSOL Java API Reference Guide*

Overview of Geometry Modeling Concepts

In COMSOL Multiphysics use *solid modeling* or *boundary modeling* to create objects in 1D, 2D, and 3D. They can be combined in the same geometry (*hybrid modeling*).

During solid modeling, a geometry is formed as a combination of solid objects using *Boolean operations* like union, intersection, and difference. Objects formed by combining a collection of existing solids using Boolean operations are known as *composite solid objects*. Boundary modeling is the process of defining a solid in terms of its boundaries, for example using lines to create a solid hexagonal domain in 2D. You can combine such a solid with *geometric primitives*—common solid modeling shapes like blocks, cones, spheres, rectangles, and circles, which are directly available in COMSOL Multiphysics.

In 3D, you can form 3D solid objects by defining 2D solids in *work planes* and then *extruding* and *revolving* these into 3D solids. It is also possible to *embed* 2D objects into the 3D geometry.

See Also

Details about how to define the geometric primitives and CAD operations are found in Geometry Modeling Features in the *COMSOL Multiphysics Reference Guide*.

You can also overlay additional nonsolid objects on top of solid objects to control the distribution of the mesh and to improve analysis capabilities. For example, you can add a curve object to a geometry to control the element size in the vicinity of this curve, or add a point to guarantee a mesh vertex in a specific location or to create a time-dependent or parametric-value graph at that location in the geometry.

The settings for the nodes making up a geometry sequence can be changed at any time and the whole sequence can be re-run. It is also possible to parameterize the geometry using one or more parameters that define properties of a geometric primitive, for example. COMSOL Multiphysics then takes the parameterization into account as part of the geometry sequence for each step in a parametric sweep. You can also insert geometry sequences from other models into your current sequence.

You can import 2D geometries from DXF files and 3D geometries from STL and VRML files.

Note

See Import in the *COMSOL Multiphysics Reference Guide* for details of how to import these CAD file formats.

The CAD Import Module provides an interface for the import of CAD files in Parasolid, SAT (ACIS), Inventor, Pro/E, SolidWorks, STEP, and IGES formats. In addition, the CATIA V5 Import Module provides an interface for CATIA V5 files.

The products LiveLink for Inventor, LiveLink for Pro/ENGINEER, and LiveLink for SolidWorks, offer bidirectional links to these respective products. Using these, run parametric geometry sweeps driven from the COMSOL environment but operating directly on the geometries in the respective CAD package environments.

The following sections provide an overview of available geometric primitives and geometry operations and features.

Geometric Primitives

See Also

- Creating Selections From Geometric Primitives and Features
- The Geometry Toolbar Buttons

In the *COMSOL Multiphysics Reference Guide:*

- Features for Geometric Primitives
- Insert Sequence from File

You can combine and operate on all geometric primitives using Boolean operations and other geometry operations described in this section and, in more detail, in the *COMSOL Multiphysics Reference Guide*. The geometric primitives provide building blocks of basic geometric shapes for creating geometries in 1D, 2D, and 3D.

If you want to refer to the domains, boundaries, edges, or points (geometric entities) in the model, COMSOL Multiphysics can create selections for all geometric entities that a geometric primitive consists of. It is also possible to create selections in the geometry feature nodes for the resulting geometry objects from such operations.

For some geometric primitives (blocks and spheres, for example) you can also add layers for modeling of sandwich structures of for creating, for example, layers of layers of concentric spheres).:

Note

Right-click the **Geometry** node to add the geometric primitives in Table 5-6 to the geometry sequence. In 3D, some options are found under the **More Primitives** submenu.

TABLE 5-1: 1D, 2D, AND 3D GEOMETRY PRIMITIVES

ICON	NAME	DIMENSION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Bézier Polygon	2D, 3D	Bézier Polygon
	Block	3D	Block
	Circle	2D	Circle
	Cone	3D	Cone
	Cylinder	3D	Cylinder
	Eccentric Cone	3D	Eccentric Cone
	Ellipse	2D	Ellipse
	Ellipsoid	3D	Ellipsoid
	Helix	3D	Helix
	Hexahedron	3D	Hexahedron

TABLE 5-1: 1D, 2D, AND 3D GEOMETRY PRIMITIVES

ICON	NAME	DIMENSION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Interpolation Curve	2D, 3D	Interpolation Curve
	Interval	1D	Interval
f(s)	Parametric Curve	2D, 3D	Parametric Curve
	Parametric Surface	3D	Parametric Surface
	Point	1D, 2D, 3D	Point
	Polygon	2D, 3D	Polygon
	Pyramid	3D	Pyramid
	Rectangle	2D	Rectangle
	Sphere	3D	Sphere
	Square	2D	Square
	Tetrahedron	3D	Tetrahedron
	Torus	3D	Torus

General Geometry Features

For 2D geometry objects, you can use geometry features such as fillets and tangents to construct the geometry. There are also general geometry features in all space dimensions for splitting geometry objects and to delete geometry objects or geometric entities (domains, boundaries, edges, and points). The following table provides an overview of available general geometry features:

TABLE 5-2: GENERAL GEOMETRY FEATURES

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Chamfer	Create a chamfer at a corner of a 2D geometry object.	Chamfer
	Fillet	Create a fillet at a corner of a 2D geometry object.	Fillet

TABLE 5-2: GENERAL GEOMETRY FEATURES

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Tangent	Create a tangent from an edge to another edge or point in a 2D geometry.	Tangent
	Split	Split one or several objects into their entities.	Split
	Delete Entities	Delete geometric entities from the objects they belong to or to delete entire geometry objects,	Delete Entities
	Edit Object	Edit the properties of a 2D object.	Edit Object

Boolean Operations

Use Boolean operations to create new geometry objects by forming unions, set differences, and intersections—and combinations of those operations—of existing geometry objects. You have access to the following Boolean operations in all space dimensions:

TABLE 5-3: BOOLEAN OPERATIONS

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Union	Create the union of the selected objects.	Union
	Intersection	Create the intersection of the selected objects.	Intersection
	Difference	Create a set difference by taking the selected object with the largest volume (area, length) and subtract the others.	Difference
	Compose	Compose a geometry object by specify Boolean operations using a set formula.	Compose

Geometry Object Transforms

You can use the transforms to create arrays of geometry objects and to move, rotate, mirror, and scale geometry objects. All the transforms are available in all space dimensions, except Rotate, which is not applicable for 1D geometries.

TABLE 5-4: GEOMETRY OBJECT TRANSFORMS

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Array	Create an array of geometry objects.	Array
	Copy	Copy a geometry object.	Copy
	Mirror	Mirror a geometry object.	Mirror
	Move	Move a geometry object.	Move
	Rotate	Rotate a geometry object.	Rotate
	Scale	Scale a geometry object.	Scale

Geometry Object Conversions

The geometry conversion operations make it possible to, for example, convert a 3D solid to a surface (boundary) object for modeling a shell or other thin structure. The following conversion operations are available:

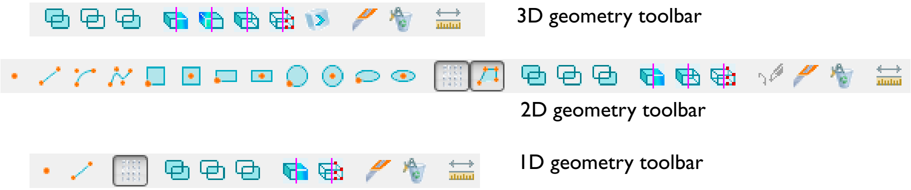
TABLE 5-5: GEOMETRY OBJECT CONVERSION OPERATIONS

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Convert to Solid	Convert a geometry object to a solid.	Convert to Solid
	Convert to Surface	Convert a geometry object to a surface (3D only).	Convert to Surface
	Convert to Curve	Convert a geometry object to a curve (2D, 3D).	Convert to Curve

TABLE 5-5: GEOMETRY OBJECT CONVERSION OPERATIONS

ICON	NAME	DESCRIPTION	LINK TO THE COMSOL MULTIPHYSICS REFERENCE GUIDE
	Convert to Point	Convert a geometry object to a point.	Convert to Point
	Convert to COMSOL	Available with the CAD Import Module. Converts an object to a COMSOL kernel representation (3D).	See the *CAD Import Module User's Guide*.

The Geometry Toolbar Buttons



See Also

In the *COMSOL Multiphysics Reference Guide:*

- The Geometry Toolbar
- Geometry Features
- Features for Geometric Primitives
- Features for Geometric Operations
- Features for Virtual Operations
- Insert Sequence from File

Figure 5-1: The 3D, 2D, and 1D geometry toolbars.

TABLE 5-6: 1D, 2D, AND 3D GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
●	Draw Point	1D, 2D	To draw a point, first click the Draw Point button, then click in the Graphics window.
⟋	Draw Interval	1D	To draw an interval in 1D, first click the Draw Interval button, then click the start and end points in the Graphics window.
⟋	Draw Line	2D	To draw a polygon consisting of line segments or Bézier curves in 2D, first click one of the buttons Draw Line, Draw Quadratic, or Draw Cubic. Then click the control points of the segments in the Graphics window. Click one point for each linear segment, two points for each quadratic segment, and three points for each cubic segment. If you want to switch segment type, click one of the buttons and then click some more control points. Close the polygon by right-clicking anywhere in the Graphics window. Then, a solid Bézier polygon appears, and a corresponding Bézier Polygon feature node appears in the geometry sequence. If you want to modify the polygon (for instance, change from solid to curve) you can edit the Bézier Polygon node by clicking it to display its settings window.
⌒	Draw Quadratic	2D	
∿	Draw Cubic	2D	
□	Draw Square	2D	To draw a square, click Draw Square or Draw Square (center). Then, click one corner (or the center) of the square in the Graphics window. Drag the mouse to the desired position of a corner. When you release the mouse button, a solid square appears, and a Square feature node appears in the geometry sequence.
⊡	Draw Square (Center)	2D	
▭	Draw Rectangle	2D	To draw a rectangle, click Draw Rectangle or Draw Rectangle (center). Then, click one corner (or the center) of the rectangle in the Graphics window. Drag the mouse to the desired position of a corner. When you release the mouse button, a solid rectangle appears, and a Rectangle feature node appears in the geometry sequence.
▭	Draw Rectangle (Center)	2D	

TABLE 5-6: 1D, 2D, AND 3D GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
	Draw Circle	2D	To draw a circle, click Draw Circle (center) or Draw Circle. Then, click the circle's center (or one corner of the circle's bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the circle's bounding box. When you release the mouse button, a solid circle appears, and a Circle feature node appears in the geometry sequence.
	Draw Circle (Center)	2D	
	Draw Ellipse	2D	To draw an ellipse, click Draw Ellipse (center) or Draw Ellipse. Then, click the ellipse's center (or one corner of the ellipse's bounding box) in the Graphics window. Drag the mouse to the desired position of a corner of the ellipse's bounding box. When you release the mouse button, a solid ellipse appears, and an Ellipse feature node appears in the geometry sequence.
	Draw Ellipse (Center)	2D	
	Snap Coordinates	1D, 2D	By default, the mouse pointer snaps to the grid points and geometry vertices (for example, the corners of a rectangle). To disable snapping, click the the Snap Coordinates button.
	Draw Solid	2D	The Draw Solid button is used to toggle between drawing solid (2D) objects or outlines (curves) and can be used in combination with all the draw buttons on the toolbar (for example, to draw squares, circles, and rectangles). It is selected by default to draw solid objects. When used to draw a polygon in combination with the Draw Line, Draw Quadratic, and Draw Cubic operations, the solid object (or outline) displays once the object is closed or the points joined.
	Union	1D, 2D, 3D	Select the objects to compose in the Graphics window and click the Union button to create a union of the selected objects.
	Intersection	1D, 2D, 3D	Select the solid objects to compose in the Graphics window and click the Intersection button to create an intersection of the selected objects.

TABLE 5-6: 1D, 2D, AND 3D GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
	Difference	1D, 2D, 3D	Select the solid objects to compose in the Graphics window and click the Difference button to take the selected object with the largest volume (area, length) and subtract the others.
	Convert to Solid	1D, 2D, 3D	Select the objects to convert in the Graphics window and click Convert to Solid.
	Convert to Surface	3D	Select the objects to convert in the Graphics window and click Convert to Surface.
	Convert to Curve	2D, 3D	Select the objects to convert in the Graphics window and click Convert to Curve.
	Convert to Point	1D, 2D, 3D	Select the objects to convert in the Graphics window and click Convert to Point.
	Convert to COMSOL	3D	Available with the CAD Import Module. Select the objects to convert in the Graphics window, and then click the Convert to COMSOL button.
	Edit Object	2D	To edit a 2D object, select the object and click the Edit Object button.
	Split	1D, 2D, 3D	To split one or several objects into their entities, select them in the Graphics window, and then click the Split button.
	Delete	1D, 2D, 3D	To delete geometric entities (domains, boundaries, edges, or points) from the objects they belong to or to delete entire geometry objects, select the entities or objects and click the Delete button. If you delete objects corresponding to primitive features their nodes disappear from the sequence. If you delete other objects or if you delete geometric entities a Delete Entities node appears in the sequence.
	Measure	1D, 2D, 3D	To measure the volume, area, or length of a selected domain, face, or edge, respectively, click the Measure button. The result appears in the Messages window. Using this button it is also possible to view the coordinates of a vertex, the distance between two vertices, or the number of entities and the geometry representation (only if you have license for the CAD Import Module) of an object.

Selecting the Space Dimension for the Model Geometry

Most of the problems solved with COMSOL Multiphysics are three-dimensional (3D) in the real world. In many cases, it is sufficient to solve a two-dimensional (2D) or one-dimensional (1D) problem that is close, or equivalent, to the real problem.

Tip

It is good practice to start a modeling project by building one or several 2D (or 1D) models before going to a 3D model.

This is because 2D models are easier to modify and solve much faster. Thus, modeling mistakes are much easier to find when working in 2D. Once the 2D model is verified, you are in a much better position to build a 3D model.

Note

Not all physics interfaces are available in all space dimensions. See the documentation for the physics interfaces in COMSOL Multiphysics and its modules for the supported space dimensions.

1D PROBLEMS

1D

The following is a guide through some of the common approximations made for 1D problems. Remember that modeling in 1D usually represents some 2D or 3D geometry under the assumption that nothing changes in the other dimensions.

Cartesian Coordinates

In a 1D model you view a single straight line thar represents the only space dimension where there is spatial variation. 1D models can represent structural elements such as beams and bars with a given cross section using special elements.

Axial Symmetry (Cylindrical Coordinates)

In an axially symmetric 1D model you view a straight line that represents the radial direction in an axially symmetric geometry.

2D Problems

2D

The following is a guide through some of the common approximations made for 2D problems. Remember that modeling in 2D usually represents some 3D geometry under the assumption that nothing changes in the third dimension.

Cartesian Coordinate Systems

In this case you view a cross section in the *xy*-plane of the actual 3D geometry. The geometry is mathematically extended to infinity in both directions along the *z*-axis, assuming no variation along that axis. All the total flows in and out of boundaries are per unit length along the *z*-axis. A simplified way of looking at this is to assume that the geometry is extruded one unit length from the cross section along the *z*-axis. The total flow out of each boundary is then from the face created by the extruded boundary (a boundary in 2D is a line).

There are usually two approaches that lead to a 2D cross-section view of a problem:

- When it is known that there is no variation of the solution in one particular dimension
- When there is a problem where the influence of the finite extension in the third dimension can be neglected.

In some applications there are special 2D assumptions, such as the *plane strain* and *plane stress* conditions for 2D stress analysis in solid mechanics.

In addition to the unit-depth assumption, some physics interfaces (for solid mechanics and heat transfer, for example) provide the thickness as a user-defined property in 2D models. In the case of heat transfer, the thickness is used when including out-of-plane heat transfer in the model.

Axial Symmetry (Cylindrical Coordinates)

1D Axi

2D Axi

If the 3D geometry can be constructed by revolving a cross section about an axis, and no variations in any variable occur when going around the axis of revolution, an axisymmetric physics interface can be used.

The spatial coordinates are called *r* and *z*, where *r* is the radius. The flow at the boundaries is given per unit length along the third dimension. Because this dimension is a revolution, you have to multiply all flows with αr, where α is the revolution angle (for example, 2π for a full turn).

3D PROBLEMS

3D

This section discusses issues that should be addressed before starting to implement a 3D model.

Although COMSOL Multiphysics fully supports arbitrary 3D geometries, it is important to simplify the problem. This is because 3D problems easily get large and require more computer power, memory, and time to solve. The extra time spent on simplifying a problem is probably well spent when solving it.

Is it possible to solve the problem in 2D? Given that the necessary approximations are small, the solution is more accurate in 2D because a much denser mesh can be used. See 2D Problems if this is applicable.

Are there symmetries in the geometry and model? Many problems have planes where the solution on either side of the plane looks the same. A good way to check this is to flip the geometry around the plane, for example, by turning it upside down around the horizontal plane. You can then remove the geometry below the plane if you do not see any differences between the two cases regarding geometry, materials, and sources. Boundaries created by the cross section between the geometry and this plane need a symmetry boundary condition, which is available in all 3D physics interfaces.

Do you know the dependence in one direction so it can be replaced by an analytical function? You can use this approach either to convert 3D to 2D or to convert a layer to a boundary condition (see Converting Geometry Objects).

About Selecting the Space Dimension and Coordinate System

COMSOL Multiphysics uses a global Cartesian or cylindrical (axisymmetric) coordinate system. The geometry dimension and coordinate system is selected when starting a new model in the Model Wizard. Variable names for the spatial coordinates are *x*, *y*, and *z* for Cartesian coordinates and *r*, φ, and *z* for cylindrical coordinates.

These coordinate variables (together with the variable *t* for the time in time-dependent models) make up the *independent variables* in COMSOL Multiphysics models.

See Also	Coordinate Systems

THE COORDINATE SYSTEMS AND THE SPACE DIMENSION

The labels assigned to the coordinate system variables vary according to the space dimension:

- Models that are opened using the space dimensions 1D, 2D, and 3D use the Cartesian coordinate independent variable labels *x* (1D), *y* (1D and 2D), and *z* (3D).
- In 2D axisymmetric geometries, the *x*-axis represents the *r* label, which is the radial coordinate, while the *y*-axis represents the *z* label, the height coordinate.
- In 1D axisymmetric geometries, the default radial coordinate is labelled *r*, and represented by the *x*-axis.

Note	The names of the fields for positions, displacements, and other coordinate data in the settings windows for geometry objects and transform operations adapt to the space dimension and coordinate system in the model: For example, for a Cartesian 2D geometry the position is specified using the **x** and **y** fields, whereas for a 2D axisymmetric geometry the positions are specified using the **r** and **z** fields. For 2D work plane geometries the positions are specified using the **xw** and **yw** fields.

For axisymmetric cases the geometry model must fall in the positive half plane (that is to say, $r \geq 0$).

See Also	• About Cylindrical Coordinate Systems • Defining a Cylindrical System

Selecting the Coordinate System and Space Dimension

In the **Model Wizard**, select **3D**, **2D axisymmetric**, **2D**, **1D axisymmetric**, or **1D** from the **Select Space Dimension** list. You can do this before starting a new model or by right-clicking the root node in the **Model Builder** and selecting **Add Model** for creating models with multiple geometries. The **Model** node's icon indicates the space dimension as in Table 5-7.

TABLE 5-7: SPACE DIMENSION ICONS IN THE MODEL BUILDER

ICON	SPACE DIMENSION
	3D
	2D axisymmetric
	2D
	1D axisymmetric
—	1D
•	0D (space-independent models for chemical reactions or other ODEs/DAEs)

Note

Not all physics interfaces are available for all coordinate systems and space dimensions.

The nodes under a physics interface in the Model Builder and Plot Groups and Plots use different icons and colors to help visualize a model.

See Also

- Physics Nodes by Space Dimension
- Color Coding for Plot Groups and Plot Types

Creating Composite Geometry Objects

You can form a *composite geometry object* by combining objects with Boolean operations: **Union**, **Intersection**, **Difference**, or **Compose**. Also form complex geometries using Boolean formulas that include multiple objects and operations.

1 In the **Model Builder**, click the **Geometry** node ( ), select objects in the **Graphics** window. Then click the **Union** ( ), **Intersection** ( ), or **Difference** ( ) buttons.

- OR -

In the **Model Builder**, right-click **Geometry** and select **Boolean Operations** then select **Union** ( ), **Intersection** ( ), **Difference** ( ), or **Compose** ( ).

2 For **Union**, **Intersection**, or **Compose**, in the **Input objects** section, add geometry from the **Graphics** window.

For **Difference**, click the **Activate Selection** button ( ) in the **Objects to add** or **Objects to subtract** sections. Then add or subtract geometry in the **Graphics** window. See Working with Geometry for information about how to do this.

3 Select the **Keep input objects** check box to keep the initial objects.

4 Select the **Keep interior boundaries** check box as required, depending on if you want to divide the interior or not. The default setting is to keep the interior boundaries.

5 Edit the **Relative repair tolerance** default if the geometry has larger defects, or if you need to resolve extremely small details.

During the creation of a composite object, COMSOL automatically repairs the generated geometry object by removing small edges and faces. Edit the default if required. The value is relative to the overall dimensions of the geometry. For example, if the dimensions are in meters, the default repair tolerance of 10^{-6} makes the geometry repair heal gaps that are smaller than a micrometer (10^{-6} m).

6 For **Compose** only, enter a **Set formula** that defines the Boolean operation. Set formulas are only available for solid objects. The symbols `+`, `-`, and `*` represent *set union*, *set difference*, and *set intersection*, respectively. Use these set formulas to form a composite object by combining the selected solid objects according to the set formula. For example, entering `(cyl1+sph1)*blk1` first forms the union of the cylinder, `cyl1`, and the sphere, `sph1`, and then forms the intersection of that union with the block, `blk1`.

7 Click the **Build Selected** button ( ).

Adding Affine Transformations to Geometry Objects

Mirroring, moving, rotating, and scaling are *affine transformations* applied to geometry objects.

1 Define an object.

2. In the **Model Builder**, right-click the **Geometry** node ( ) and select an option from the **Transforms** menu:
   - **Mirror**—Specify the reflection axis or reflection plane.
   - **Move**—Specify displacements in all space directions.
   - **Rotate**—Specify the rotation angle with a rotation axis or center of rotation.
   - **Scale**—Specify center of scaling and scaling factors in all space directions.
3. Under **Input**, select the **Input objects** in the **Graphics** window. The objects display in the **Input objects** list. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.
4. Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.

Mirror

1. 2D and 3D: Under **Point on Line of Reflection**, enter a point to be fixed during reflection by entering **x**, **y**, and **z** coordinates as required based on the model dimensions.
2. 2D and 3D: Under **Normal vector to Line of Reflection**, enter a vector in the direction to reflect by entering **x**, **y**, and **z** coordinates as required based on the model dimensions.
3. 1D: Under **Point of Reflection**, enter the coordinate of the point of reflection in the **x** field.

Move

Under **Displacement**, enter the **x**, **y**, and **z** coordinates as required by the model dimension. To create several copies, enter a comma-separated list of displacements in the fields.

Rotate

1. Under **Rotation Angle**, enter the rotational angle in the **Rotation** field.
2. 2D: Under **Center of Rotation**, enter **x** and **y** coordinates.

   3D: Under **Point on Axis of Rotation**, enter the **x**, **y**, and **z** coordinates.
3. 3D: Under **Axis of Rotation**, select an **Axis type**—**Cartesian** or **Spherical**.
4. If **Cartesian** is selected, enter a direction vector in the **x**, **y**, and **z** fields.

   If **Spherical** is selected, enter the angles **theta** (polar, zenith) and **phi** (azimuth).

Scale

1. Under **Scale Factor**, the **Scaling** default is **Isotropic**. Enter a **Factor**.

   If **Anisotropic** is selected as the **Scaling**, enter the **x**, **y**, and **z** coordinates as required by the model dimension.

2. Under **Center of Scaling**, enter the **x**, **y**, and **z** center point coordinates as required by the model dimension.

Creating an Array of Identical Geometry Objects

Create a rectangular or linear array of duplicates of geometry objects.

1. In the **Model Builder**, right-click Geometry and select **Transforms>Array** ( ).
2. Under **Input**, select the **Input objects** to duplicate in the **Graphics** window. The objects display in the **Input objects** list. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.
3. Under **Size**, the **Array type** defaults to **Rectangular** (2D) or **Three-dimensional** (3D). Enter the number of duplicates in each coordinate direction in the **x size**, **y size**, and (3D only) **z size** fields.

   If **Linear** is selected as the **Array type**, enter the number of duplicates in the **Size** field.

4. Under **Displacement**, enter the **x**, **y**, and (3D only) **z** coordinates as required by the model dimension.

Copying and Pasting Geometry Objects

There are two methods to copy and paste geometry objects.

Tip

Duplicate is a convenient way to copy and paste in one step. In other words, it combines the **Copy** and **Paste** functions.

METHOD 1

When using the copy/paste functionality, the copied feature initially contains the same data as the copied node, but there is no future connection between the two nodes. For example, if the original node is changed, it has no effect on the second node that was copied. In this case, you would use **Transforms>Copy** to keep the nodes linked.

With this method, a copy of the geometry (a rectangle, circle, sphere, or ellipse, for example) is inserted into the same geometry sequence, or another geometry sequence in the same model, and is added after the current feature of the selected geometry sequence. The copy feature can also be used for **Work Plane** geometry sequences.

Important	The copied object must be pasted into a model with the same space dimension. For example, a sphere can only be pasted into a 3D model.

1 In the **Model Builder**, under the **Geometry** node, right-click the geometry object to copy (for example **Rectangle**, **Circle**, **Sphere**, or **Ellipse**) and select **Copy** ( ).
2 Right-click **Geometry** and select **Paste** (**Rectangle**, **Circle**, **Sphere**, or **Ellipse**, for example) ( ). A copy of the node is added to the end of the geometry sequence.
3 Click the **Build Selected** button ( ).

Note	It is also possible to copy/paste and duplicate nodes corresponding to operation features, such as the Union node.

METHOD 2

Make a displaced copy of a geometry objects. This method creates a node in the model tree that contains a reference to other objects in the geometry sequence that are copied and keeps the objects linked (unlike a simple copy and paste function). If a copied node is modified, any feature corresponding to the input object also includes the changes the next time the sequence is built.

1 In the **Model Builder**, right-click **Geometry** and select **Transforms>Copy** ( ).
2 Under **Input**, select the **Input objects** to duplicate in the **Graphics** window. The objects display in the **Input objects** list. Use the **Add to Selection** ( ), **Remove from Selection** ( ), and **Clear Selection** ( ) buttons as required.
3 **Keep input objects** is selected by default. Click to clear the check box if required.

4 Under **Displacement**, enter the **x**, **y**, and (3D only) **z** coordinates as required by the model dimension.

See Also

- Copying, Pasting, and Duplicating Nodes
- Copy in the *COMSOL Multiphysics Reference Guide*

Converting Geometry Objects

Transform one or more geometry objects into a different type of object by *converting* it, for example, converting from a solid to a curve. The object is then called a composite object, which then becomes a solid, face, curve, or point object depending on the target type chosen.

Converting an object is useful for closed domains in 2D and 3D—create solid domains by converting the surrounding curve object or face object to a solid object. This is necessary to apply physics and material properties to the domain.

Available conversions are:

2D

- Convert a solid object into a curve or point object.
- Convert a curve object defining at least one closed domain into a solid object.
- Convert a curve object into a point object (2D and 3D).

3D

- Convert a solid object into a surface, curve, or point object.
- Convert a surface object defining at least one closed domain into a solid object.
- Convert a surface object into a curve, or point object.
- Convert a curve object into a point object (2D and 3D).

1. In the **Model Builder**, click the **Geometry** node, select an object in the **Graphics** window. Then click the **Convert to Solid** ( ), **Convert to Surface** (3D) ( ), **Convert to Curve** ( ), or **Convert to Point** ( ) buttons.

   - OR -

   In the **Model Builder**, right-click **Geometry** and select an option from the **Conversions** menu—**Convert to Solid**, **Convert to Surface** (3D only), **Convert to Curve**, or **Convert to Point**.
2. Under **Input**, select the **Input objects** to convert in the **Graphics** window. The objects display in the **Input objects** list. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.
3. Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.
4. Edit the **Relative repair tolerance** default if you experience problems with the convert operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. Geometric entities that have a distance less than the absolute repair tolerance are merged.
5. Click the **Build Selected** button ( ).

Editing 2D Geometry Objects

Use the **Edit Object** ( ) feature to adjust the edges and vertices for a 2D geometry object or to add or delete edges and vertices in the object.

See Also

Edit Object in the *COMSOL Multiphysics Reference Guide*

1. In the **Model Builder**, right-click a 2D **Geometry** and select **Edit Object** ( ).
2. Under **Select Object**, select the **Input object** to edit in the **Graphics** window. When selected, a copy is made of the input object, and subsequent changes operate on this copy. Changes made in the **Input object** after the copy has been made has no effect on the **Edit Object** feature. For this reason, it is not possible to change the **Input object** after it has been selected.

   Use the **Add to Selection** (+), **Remove from Selection** (−), **Clear Selection** ( ), and **Zoom Selected** ( ) buttons as required.

3 Under **Edit Edges**, click the **Activate Edges** button (⏻). In the **Graphics** window, select and add the **Edge** number to edit (or use the **View>Selection** list method). The parameters for the current edge are displayed in the table for **x (m)**, **y (m)**, and **Weights**, and under the table for the **Degree** list and the **Start vertex** and **End vertex** fields.

Tip

When the **Edit Object** feature is the current feature node, you can visualize the edited object by observing the edge and vertex numbers displayed in the **Graphics** window next to the edges or vertices.

- If required, click **New** to create a new edge. The new edge is linear, not connected to any other edges or vertices and has both the start and the end coordinates set to 0.
- If required, click **Delete** to delete the current edge. Deleting an edge also deletes its adjacent vertices, if these vertices are not connected to other edges.

4 Click in the table cells to edit the **x (m)**, **y (m)**, and **Weights** and modify the control points of the edge. If the **x** or **y** value for the first or last control point is modified, any adjacent edges and vertices are automatically updated with the same value.

Note

The **m** in parentheses indicates the current geometry length unit; in this case the default unit: meter.

5 Select the **Degree**—**Linear**, **Quadratic**, or **Cubic**—to change the degree of the edge. When increasing the degree, the control points are automatically recalculated so that the shape of the edge does not change. When decreasing the degree, the control points are recalculated so as to approximate the old shape of the edge.

6 Under **Start vertex**, click the **Activate Edges** button (⏻) to move the start point of the edge to a different vertex. In the **Graphics** window, select and add the **Edge** number. Under **End vertex**, click the **Activate Vertices** button (⏻) to move the end point of the edge to a different vertex. In the **Graphics** window, select and add the **Vertices** number.

- If required, click **Disconnect** to disconnect the start or end vertex of an edge from the rest of the object. A new vertex is created with coordinates matching the start or end point of the edge. This vertex can be moved without affecting the other edges that were previously connected to this edge.

7 Under **Edit Vertices**, click the **Activate Vertices** button ( ). In the **Graphics** window, select and add the **Vertex** number (node number) to edit (or use the **View>Selection** list method). The parameters for the current vertex are displayed in the **x** and **y** fields under **Coordinates**, and the vertex is highlighted in the **Graphics** window.

Tip

When the **Edit Object** feature is the current feature node, you can visualize the edited object by observing the edge and vertex numbers displayed in the **Graphics** window next to the edges or vertices.

- If required, click **New** to create a new vertex. The coordinates for the new vertex are set to 0.
- If required, click **Delete** to delete the current vertex. Only isolated vertices can be deleted. For other vertices, the **Delete** button is disabled.
- Click **Snap to Closest** to delete the current vertex. Any edges connected to the deleted vertex are modified so that the start or end point is moved to the closest remaining vertex.

8 Under **Coordinates** edit the **x** and **y** fields as required. Any adjacent edges are automatically updated with the same values.

9 Under **Selections of Resulting Entities**, select the **Create selections** check box to create predefined selections for all geometric entities (all for some domains, boundaries, edges, and points) that the resulting geometric object consists of.

Note

These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the **Model Builder**.

The **Edit Object** function can also be started using the **Edit Object** button ( ) in the Geometry toolbar, which is visible when the geometry node is selected in the **Model Builder**.

1 In the **Model Builder** click the **Geometry** node.

2 Select a single geometry object in the **Graphics** window.

3 Click the **Edit Object** button ( ). The **Graphics** window displays handles for vertices and edge control points in the selected object.

4. Use the mouse to drag vertices and control points to new locations. The image is updated in the **Graphics** window in real time to show the effect on the object being edited.
5. Right-click to exit the object editing mode and save the changes. A new **Edit Object** node is added in the **Model Builder**. Alternatively, if the object being edited is an **Edit Object** feature, the changes are incorporated in the existing feature and no new node is added in the **Model Builder**.
6. Click the left mouse button outside the object being edited to cancel the editing operation.

Another way to edit a geometry object is to hold down the Alt key and left-click on the object in the **Graphics** window. This is equivalent to selecting the object and then using the **Edit Object** toolbar button.

Splitting Geometry Objects

The split operation splits an object into its entities:

- A solid splits into solids corresponding to its domains.
- A surface object splits into surface objects corresponding to its faces.
- A curve object splits into curve objects corresponding to its edges.
- A point object splits into point objects corresponding to its vertices.
- A general (mixed) object splits into solids (corresponding to the domains), surface objects (corresponding to faces not adjacent to a domain), curve objects (corresponding to edges not adjacent to a face or domain), and point objects (corresponding to vertices not adjacent to an edge, face, or domain).

1. In the **Model Builder**, click the **Geometry** node, select objects in the **Graphics** window. Then click the **Split** button ( ).

   -OR-

   In the **Model Builder**, right-click **Geometry** and select **Split**.
2. Under **Input**, select the **Input objects** to split in the **Graphics** window. The objects display in the **Input objects** list. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.
3. Select the **Keep input objects** check box to use the selected geometry objects for further geometric operations.
4. Click the **Build Selected** button ( ).

Deleting Objects, Entities, and Sequences

DELETING OBJECTS AND ENTITIES

To delete objects or entities, click the Geometry node in the **Model Builder**, select objects or entities in the **Graphics** window, and click the **Delete** button (✕). Alternatively, use a **Delete Entities** feature according to the following.

1 In the **Model Builder**, right-click **Geometry** and select **Delete Entities** ( ).

2 Under **Input**, select a **Geometric entity level** to delete—**Object**, **Domain**, **Boundary**, **Edge**, or **Point**. Select the geometry in the **Graphics** window. Use the **Add to Selection** (+), **Remove from Selection** (−), and **Clear Selection** ( ) buttons as required.

3 Click the **Build Selected** button ( ).

If you use the **Delete** button (✕) to delete objects, the software deletes the selected objects that correspond to primitive features by deleting their nodes from the sequence. If you delete objects that do not correspond to primitive features or if you delete geometric entities a **Delete Entities** node appears in the sequence.

Tip

If you want to delete all objects created by a feature, it is better to right-click the feature, and select **Delete** (✕) or **Disable** (⊘).

DELETING A GEOMETRY SEQUENCE

Note

Undo is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

To delete a geometry sequence, right-click the **Geometry** node in the **Model Builder** and select **Delete Sequence** ( ). Click **Yes** or **No** to confirm the deletion.

Creating 3D Geometries From 2D Work Planes and 3D Faces

In addition to creating 3D geometries directly using 3D geometric primitives, it is also possible to form 3D geometries based on 2D sections (2D geometries) created in work planes or faces in the existing 3D geometry. A *work plane* is a 2D plane oriented anywhere in the 3D space. Quick options make it easy to create a work plane that is parallel to any of the main Cartesian planes or to a face in an existing 3D geometry.

There are several methods that can be used to create 3D solid objects from the 2D sections or faces:

- Extrude
- Revolve
- Sweep (a general sweeping operation along a curve)

In addition, you can use a 2D section as an "embedded" surface in the 3D geometry.

See Creating a 3D Geometry Model for an example of how to create a 3D geometry using work planes.

Drawing on the 2D Work Plane in 3D

See Also	Work Plane in the *COMSOL Multiphysics Reference Guide*

When using the **Work Plane** ( ) node to define 2D objects in 3D (for example, to extrude into a 3D object), one of the features available with the 3D projection settings enables you to draw on the work plane in 3D. These instructions are an example of this feature. When this check box is selected, two additional buttons are made available in the **Graphics** window—the **Align with Work Plane** button ( ) and the **Work Plane Clipping** button ( ). The standard 2D model geometry draw toolbar (Figure 5-1) is also available for use..

Important	Some computer graphic cards may not be able to run the work plane rendering. In these cases, the work plane is rendered as in previous versions of COMSOL (it displays as a blue plane). It is possible to go to the 3D work plane but it is not possible to draw on the plane.

Tip

If you prefer to draw on the work plane in 3D, you can change the default to always select the draw on work plane in 3D option by selecting the **Draw on work plane in 3D** check box in the **Geometry** section of the **Preferences** dialog box.

1. Create a model and add a geometry (for example, a **Sphere**).
2. Right-click to add a **Work Plane** node under the main **Geometry** node.
3. On the **Work Plane** settings window under **Work Plane>3D projection**, select the **Draw on work plane in 3D** check box. See Figure 5-2 and Figure 5-3 to see what happens when the check box is selected.
4. Under **Work Plane**, click the **Plane Geometry** node. The geometry displays in the **Graphics** window. See Figure 5-3 for an example.
   - Click the **Align with Work Plane** button ( ) to rotate and move the camera to see the work plane from the top down.
   - Click to toggle the **Work Plane Clipping** button ( ) ON and OFF. When ON, use it to cut away all geometries above the work plane and make it easier to draw when objects are overlapping within the work plane. The clipping is not done when looking at the work plane from the side.

Note

Drawing on the work plane works just as drawing in 2D. The 3D work plane adapts its size to the drawn geometry. It is also possible to select and transform objects.

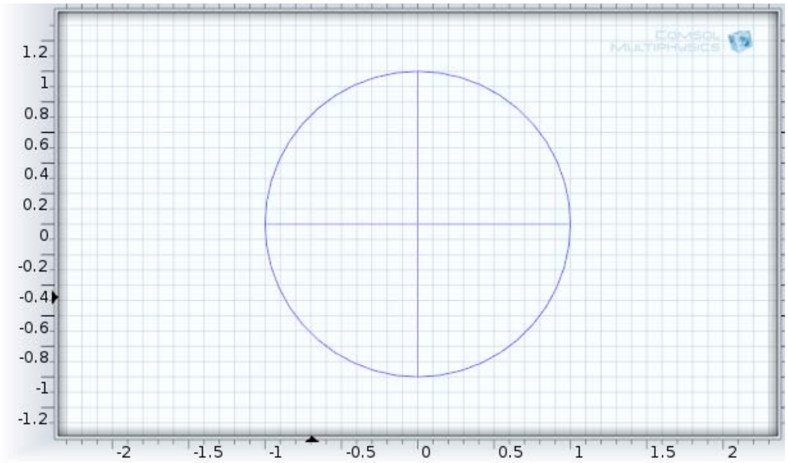


Figure 5-2: The Graphics window displaying the Work Plane Geometry without the Draw on work plane in 3D check box selected.

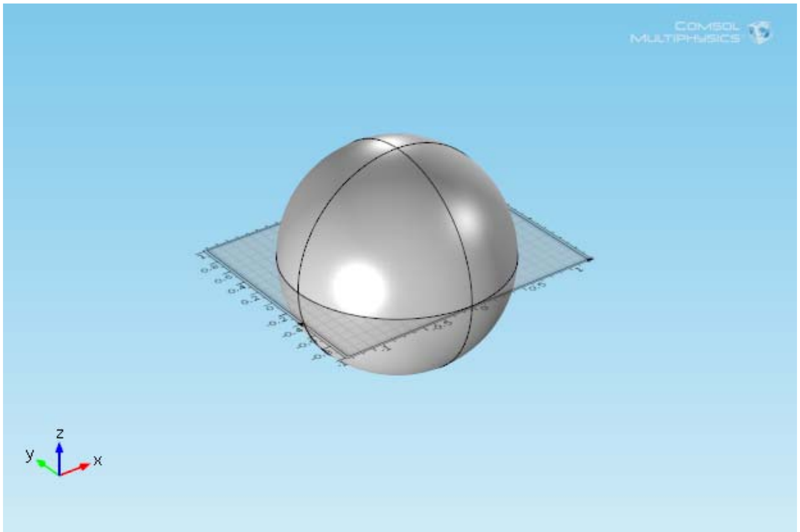


Figure 5-3: The Graphics window displaying the Work Plane Geometry with the Draw on work plane in 3D check box selected.

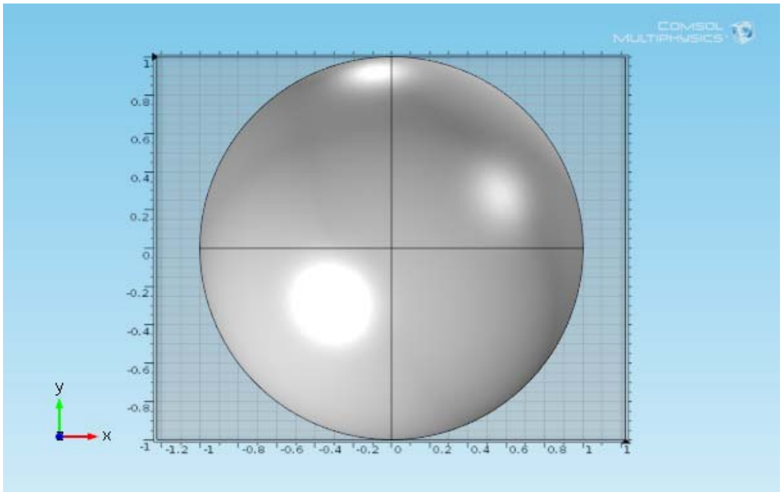


Figure 5-4: Click the Align with Work Plane button to display the geometry from the top down.

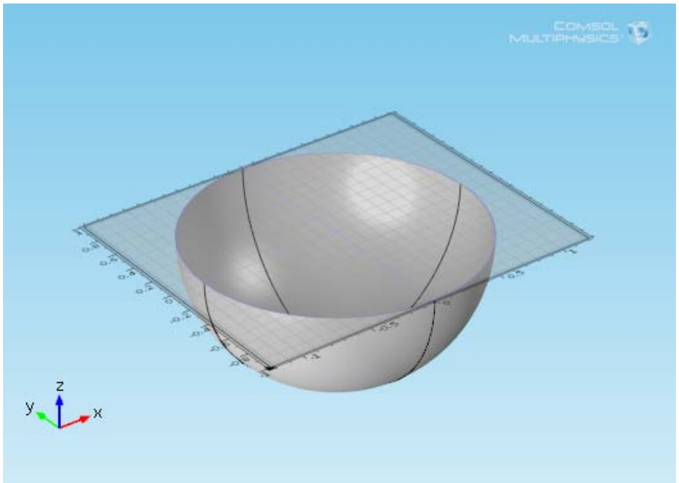


Figure 5-5: Click to turn on the Work Plane Clipping button and cut away all geometries above the work plane.

Finalizing the Geometry

To finalize the geometry, COMSOL Multiphysics evaluates the geometry sequence from the top down. The final node in the geometry sequence, the **Finalize** node, determines how to form the finalized geometry.

Note

In 2D and 3D, it is possible to add virtual operation nodes after the **Finalize** node.

There are two finalization methods, which also determine the name of the node: **Form Union** or **Form Assembly** (🏁). There are some differences and aspects to consider when choosing the finalization method:

- The default method is to form a *union*. The software then forms a union from all participating geometry objects that the geometry sequence contains or creates. The union is divided into domains separated by boundaries according to the participating geometry objects. You can mesh the entire geometry and model the physics by assigning material properties, boundary conditions, and other data for the model. It is also possible but often not necessary to specify boundary conditions (including "slit conditions") on interior boundaries between domains in the geometry. By default, COMSOL Multiphysics ensures continuity in the physics fields across interior boundaries.
- The alternative method is to form an *assembly*. The software then treats the finalized geometry as a collection of parts, where each geometry object represents a part. This means that you must use pairs to connect boundaries where a field is continuous, but it also makes is possible to use special pair conditions for applications such as contact modeling. By default, pairs are created automatically when forming an assembly. An assembly can also be useful for meshing each part independently in, for example, thin geometries with high aspect ratios. Another case where you need to use an assembly is when the geometry is too complex for forming a union, which might be the case when importing an assembly geometry from CAD data. Finally, using pairs gives you access to special boundary conditions that can model "slit conditions" such as resistive layers, but most of these are also available on interior boundaries in geometries formed as a union.

Removing Interior Boundaries

To remove interior boundaries, clear the **Keep interior boundaries** check box in the **Finalize** node or in a Boolean operations such as **Union** or **Compose**. Removing interior boundaries is good practice if the interior boundary is an effect of the geometry modeling and does not represent a border between different materials or between domains with different properties. When you remove the interior boundaries, the resulting geometry consists of fewer domains and puts fewer constraints on the mesh generation.

Note

It is sometimes useful to keep interior boundaries for controlling the mesh. In such cases, use virtual operations such as Ignore Edges and Ignore Faces but keep the input boundaries for mesh control. The interior boundaries are then not part of the geometry for defining physics and materials, but they are still present during meshing to, for example, define areas where you want to use a finer mesh. See Mesh Control Entities for more information.

Creating a 1D Geometry Model

This section describes how to create a 1D geometry model, which is essentially a line that can consist of one or several segments.

1 Double-click the COMSOL icon to launch COMSOL.

   The **Model Wizard** opens.

2 On the **Select Space Dimension** page, click the **1D** button.

3 Click the **Finish** button (🏁).

4 In the **Model Builder**, right-click **Geometry** and select **Interval** ( ).

5 Select a **Number of intervals**—**One** or **Many**. Select **Many** to get an object consisting of a sequence of connected intervals.

   - If **Many** is selected, in the **Points** field, enter a comma-separated list of coordinates.
   - If **One** is selected, enter the interval endpoint coordinates in the **Left endpoint** (default= 0) and **Right endpoint** (default= 1) fields.

6 Click the **Build Selected** button ( ).

   Add points to the geometry to divide the domain into two domains.

7 In the **Model Builder**, right-click **Geometry** and select **Point** (●).

8 Under **Point**, enter the **x** coordinate.

9 Click the **Build All** button ( ).

Creating a 2D Geometry Model

This section describes how to build a 2D cross section of a heat sink and introduces 2D geometry operations in COMSOL. At this time, you do not model the physics that describe the operation of the heat sink.

Assume that you want to estimate the maximum amount of heat dissipated by a heat sink placed around a resistor for high-power applications. The heat sink consists of an extruded aluminum profile as in Figure 5-6. If the effects at the ends of the elongated heat sink are neglected, the model can be simplified and a decent estimate obtained of the heat dissipated by creating a 2D cross section.

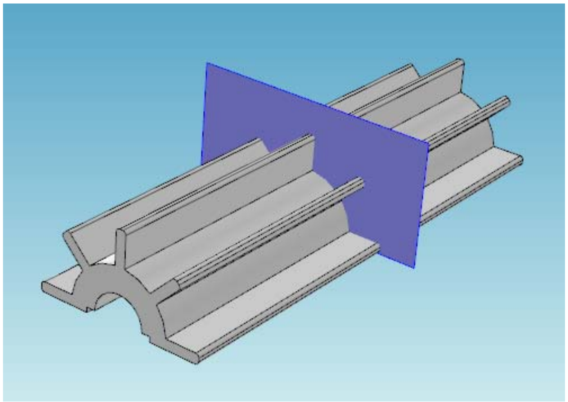


Figure 5-6: Example of a 3D heat sink model with cross section.

CREATING A BASIC 2D GEOMETRY MODEL

1 Double-click the COMSOL icon to launch COMSOL.

The **Model Wizard** opens.

2 On the **Select Space Dimension** page, click the **2D** button.

3 Click the **Finish** button (🏁).

Tip

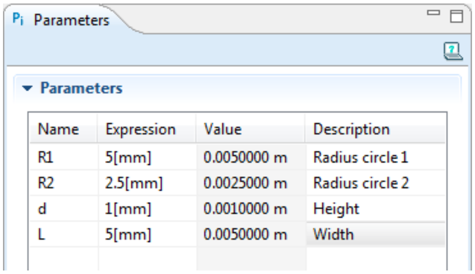
This model geometry can be used in a future example in the Free Meshing and Structured Meshes sections of this guide. It is recommended you save the model for this purpose.

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

The following steps explain how to create two circles to form the core of the heat sink in Figure 5-6. To investigate different dimensions of the heat sink, parameterize the geometry. Start by defining the radius of the outer arc of the heat sink, the radius of the inner arc, and the thickness and the length of the heat sink flanges.

1 In the **Model Builder**, right-click **Global Definitions** ( ) and select **Parameters** (Pi).

2 In the **Parameters** table, enter, or copy and paste, the **Name**, **Expression**, and **Description**. The **Value** column automatically displays the **Expression** value.



Parameters

Parameters

Name	Expression	Value	Description
R1	5[mm]	0.0050000 m	Radius circle 1
R2	2.5[mm]	0.0025000 m	Radius circle 2
d	1[mm]	0.0010000 m	Height
L	5[mm]	0.0050000 m	Width

ADDING TWO CIRCLES WITH PREDEFINED PARAMETERS

1 In the **Model Builder**, under **Model 1**, right-click **Geometry 1** and select **Circle** ( ).

2 Under **Size and Shape**, enter `R1` in the **Radius** field.

3 Click the **Build Selected** button ( ).

A circle (`c1`) with radius `R1` displays in the **Graphics** window.

4 Right-click **Geometry 1** and select **Circle** ( ).

5 Under **Size and Shape**, enter `R2` in the **Radius** field.

6 Click the **Build Selected** button ( ).

A circle with radius `R2` displays in the **Graphics** window. Click the **Zoom Extents** button ( ) to see both circles.

SUBTRACTING THE SMALLER CIRCLE

1 In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations> Difference** ( ).

2 Under **Difference**, the **Activate Selection** button ( ) is selected by default. It activates the **Objects to add** list for choosing objects.

3 In the **Graphics** window, select the object **c1** (the larger circle) by left- and then right-clicking it.

c1 is highlighted in red, then blue and added to the **Objects to add** list.

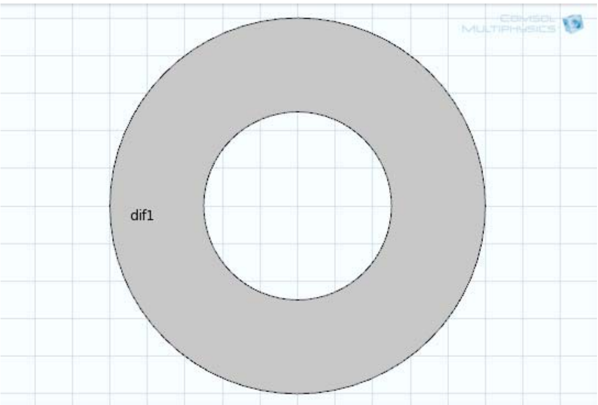
4 Click the **Activate Selection** button (⏻) to the right of the **Objects to subtract** list. This activates this section.

5 Select the object c2 (the smaller circle) by left- and then right-clicking it.

c2 is highlighted in red, then blue and added to the **Objects to subtract** list.

6 Click the **Build Selected** button ( ).

The object **dif1** is created by subtracting the smaller circle from the larger circle.



INTERSECTING WITH RECTANGLE

1 In the **Model Builder**, right-click **Geometry 1** and select **Rectangle** ( ).

2 Under **Size**:

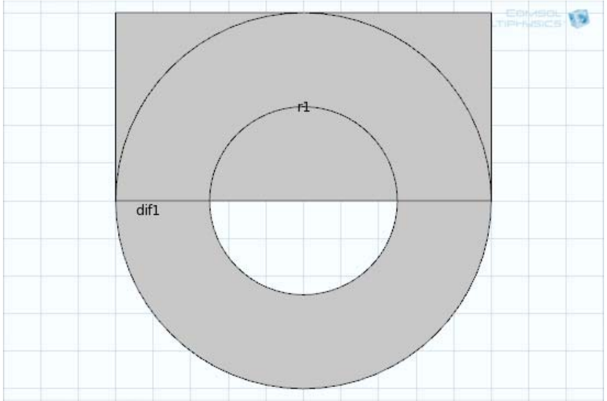
a In the **Width** field enter 2*R1.

b In the **Height** field, enter R1.

3 Under **Position**, enter -R1 in the **x** field.

4 Click the **Build Selected** button ( ).

The interaction operation creates the object r1 (not related to the circle radius), which coincides with the intersecting area of the two input objects.

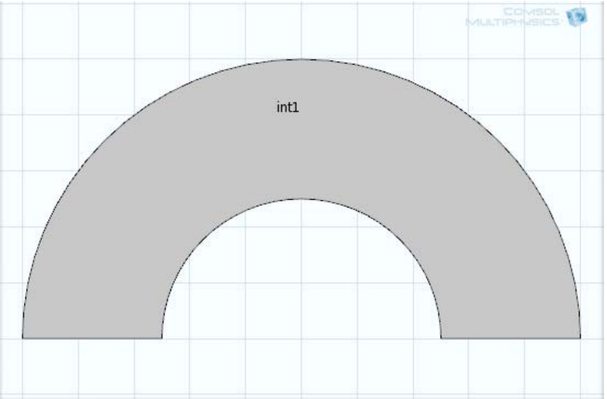


5 In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations> Intersection** ( ).

6 Select both objects, `dif1` (the combined circle) and `r1` (the rectangle), by left-clicking and then right-clicking them.

Each object is highlighted in red, then blue and added to the **Input Objects** list.

7 Click the **Build Selected** button ( ) to create the object `int1`.



ADDING A RECTANGLE TO CREATE A FLANGE

1 In the **Model Builder**, right-click **Geometry 1** and select **Rectangle** ( ).

2 Under **Size**:

a In the **Width** field, enter `L`.

b In the **Height** field, enter `d`.

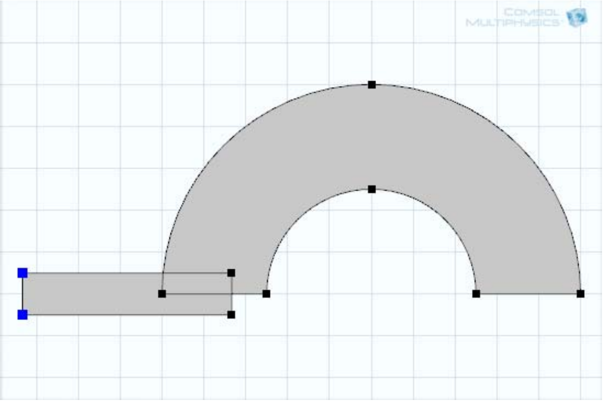
3 Under **Position**:

a In the **x** field enter `-(2/3*R1+L)`.

b In the **y** field enter `-d/2`.

4 Click the **Build Selected** button ( ).

The object `r2` (not related to the circle radius) is created. Next, round the sharp edges of the flange by using fillets. Click the **Zoom Extents** button ( ).



ADDING A FILLET TO ROUND THE FLANGE EDGES

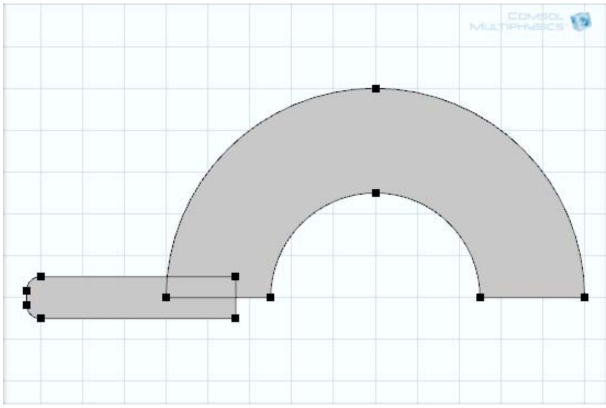
1 In the **Model Builder**, right-click **Geometry 1** and select **Fillet** ( ).

2 Select **Vertices 1** and **4** (the left-hand corners, as highlighted in the previous figure) on object `r2` (the small rectangle).

3 Click the **Add to Selection** button (+) to add these points to the **Vertices to fillet** section.

4 On the **Fillet** page, under **Radius**, enter `d/3` in the **Radius** field.

5 Click the **Build Selected** button ( ) to create object `fil1`.



ADDING ROTATE OPERATIONS TO CREATE FIVE FLANGES

Rotate the flange 45 degrees and keep the original input object to create five flanges on top of the heat sink.

Adding Rotate 1 to Create Object Rot1

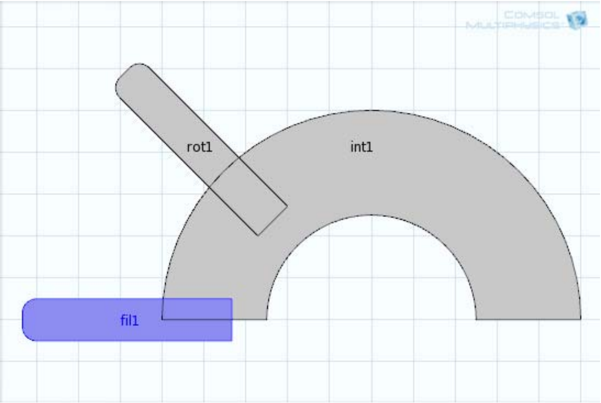
1 In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ( ).

2 Select object `fil1` (the filleted rectangle) and add it to the **Input Objects** list.

3 On the **Rotate** page, under **Input**, select the **Keep input objects** check box.

4 Under **Rotation Angle**, enter `-45` in the **Rotation** field.

5 Click the **Build Selected** button ( ) to create object `rot1`. Click the **Zoom Extents** button ( ).



Adding Three More Rotations to the Model

1 In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ( ).

2 Select the object `rot1` (the resulting rotated filleted rectangle) to add it to the **Input Objects** list.

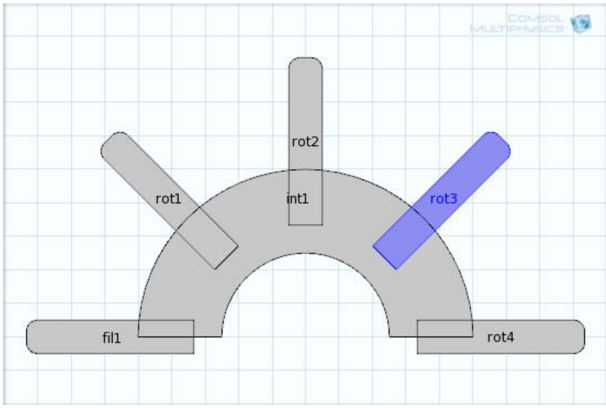
3 On the **Rotate** page, under **Input**, select the **Keep input objects** check box.

4 Under the **Rotation Angle** section, enter `-45` in the **Rotation** field.

5 Click the **Build Selected** button ( ) to create object `rot2`.

6 Repeat the above steps to create object **rot3** and **rot4**. Use object **rot2** to create **rot3** and object **rot3** to create **rot4**.

7 When done, click the **Zoom Extents** button ( ) to view the completed object.



REMOVING INTERIOR BOUNDARIES IN UNION OPERATIONS

1 In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations>Union** ( ).

2 Select the objects `int1`, `fil1`, `rot1`, `rot2`, `rot3`, and `rot4`.

3 Under **Union**, click to clear the **Keep interior boundaries** check box to remove the interior boundaries in the union operation. This is good practice if these boundaries do not defined separate parts using different materials, for example.

4 Click the **Build All** button ( ). Click the **Zoom Extents** button ( ). The final geometry is shown in Figure 5-7.

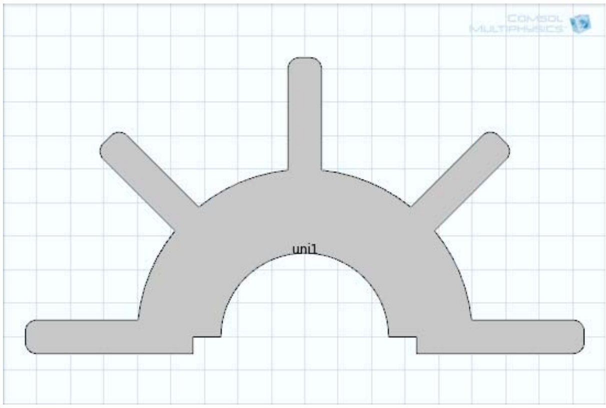
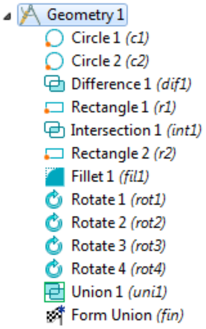


Figure 5-7: Final 2D object created in the Model Builder.

VIEWING THE GEOMETRY SEQUENCE

Figure 5-8 shows the geometry sequence used to create Figure 5-7. All primitive objects and the fillet operation are parameterized through the radius of the inner and outer heat sink arcs, the length and thickness of the flanges, and the radius of the fillets. You can change the parameter values in the **Parameters** table or for any object to create alternative heat sink geometries. The sequence still remains and when you click the **Build All** button (▦) a new geometry is created.



- Geometry 1
  - Circle 1 *(c1)*
  - Circle 2 *(c2)*
  - Difference 1 *(dif1)*
  - Rectangle 1 *(r1)*
  - Intersection 1 *(int1)*
  - Rectangle 2 *(r2)*
  - Fillet 1 *(fil1)*
  - Rotate 1 *(rot1)*
  - Rotate 2 *(rot2)*
  - Rotate 3 *(rot3)*
  - Rotate 4 *(rot4)*
  - Union 1 *(uni1)*
  - Form Union *(fin)*

Figure 5-8: An example of a 2D geometry sequence.

RE-RUNNING THE GEOMETRY SEQUENCE WITH DIFFERENT PARAMETERS

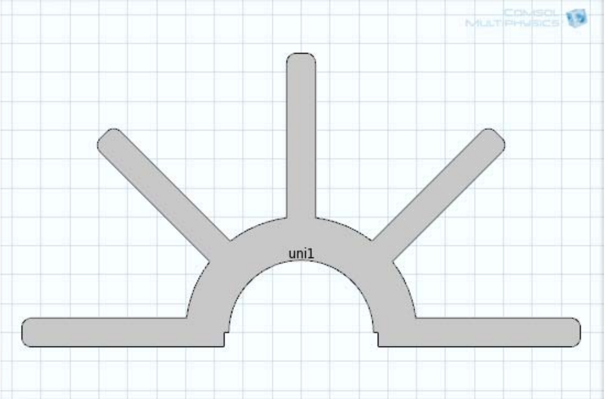
1 In the **Model Builder**, under **Global Definitions**, click **Parameters** (Pi).

2 Under **Parameters**, enter the following settings in the table. Replace the previous data:.

NAME	EXPRESSION	VALUE	DESCRIPTION
R1	`4e-3[m]`	0.0040 m	Radius Circle 1
R2	`2.5e-3[m]`	0.0025 m	Radius Circle 2
d	`1e-3[m]`	0.0010 m	Height
L	`7e-3[m]`	0.0070 m	Width

3 In the **Model Builder**, click **Geometry 1**.

4 Click the **Build All** button (▦). Click the **Zoom Extents** button (⊕) to view the geometry as defined by the new parameters.



Creating a 3D Geometry Model

Figure 5-9 shows the geometry of a heat sink used for cooling microprocessors. Creating 3D Geometries Using the Model Builder describes the steps to create this geometry and introduces 3D drawing tools and techniques.

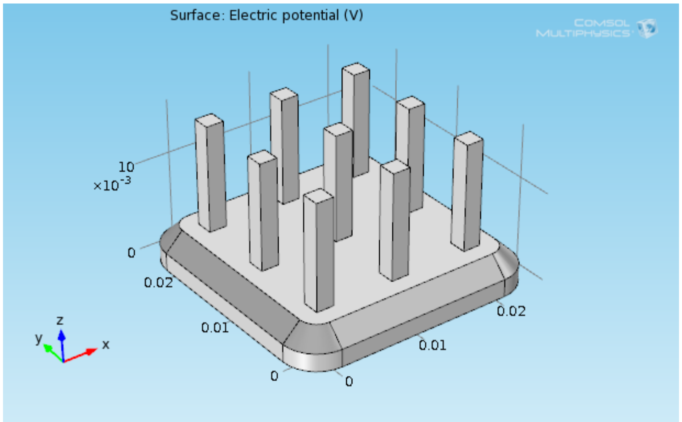


Figure 5-9: Example of a 3D heat sink model.

Creating 3D Geometries Using the Model Builder

CREATING A BASIC 3D GEOMETRY MODEL

1 Double-click the COMSOL icon to launch COMSOL.

The **Model Wizard** opens.

2 On the **Select Space Dimension** page, click the **3D** button.

3 Click the **Finish** button (🏁).

Tip

This model geometry can be used in a future exercise in the Free Meshing and Structured Meshes sections of this guide. It is recommended you save the model for this purpose.

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

1 In the **Model Builder**, right-click **Global Definitions** and select **Parameters** (Pi).

2 In the **Parameters** table, enter these settings:

NAME	EXPRESSION	VALUE	DESCRIPTION
`L1`	`1.5e-2`	0.015	Pillar thickness (in the heat sink)
`L2`	`3e-3`	0.0030	Pillar Length (in the heat sink)

USING WORK PLANES TO CREATE A BÉZIER POLYGON

Use work planes to create 2D geometries, which you then extrude or revolve to create 3D objects.

Creating a Bézier Polygon

1 In the **Model Builder**, under **Model 1**, right-click **Geometry 1** and select **Work Plane** ( ).

2 Under **Work Plane**, select **xz-plane** from the **Plane** list.

3 Under the **Work Plane 1** node, right-click **Plane Geometry** and select **Bézier Polygon** ( ).

4 On the **Bézier Polygon** settings window, under **Polygon Segments**, click **Add Linear**.

Segment 1 (linear) displays in the **Added segments** list.

5 Under **Control points**:

a In row **1**, enter `-2e-3` in the **xw** field.

b In row **2**, enter `-4e-3` in the **xw** field.

6 Click **Add Linear** to add **Segment 2 (linear)** to the **Added segments** list. Some of the **Control points** are automatically filled in with values; the control points from the previous line are already filled in as the starting points for the next line.

7 Under **Control points**, in row **2**, enter `2e-3` in the **yw** field.

8 Click **Add Linear** to add **Segment 3 (linear)** to the **Added segments** list.

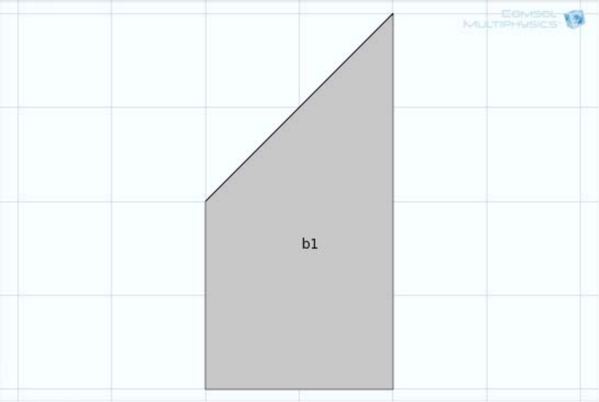
a In row **2**, enter `-2e-3` in the **xw** field.

b In row **2**, enter `4e-3` in the **yw** field.

9 Click **Add Linear** to add **Segment 4 (linear)** to the **Added segments** list.

10 Under **Control points**, in row **2**, enter `0` in the **yw** field.

11 Click **Close Curve** then click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).



REVOLVING A 2D OBJECT TO CREATE A 3D OBJECT

1 In the **Model Builder**, right-click **Work Plane1** and select **Revolve** ( ).

The **Revolve** page opens in the settings window and the 2D Bézier Polygon displays in the **Graphics** window.

2 On the **Revolve** page, under **Revolution Angles**, enter `90` in the **End angle** field.

Note

The **Revolution Axis** corresponds to the position of the y-axis in the work plane's 2D coordinate system.

3 Under **General**, clear the **Unite with input objects** check box. **Work Plane 1** is required for the next steps.

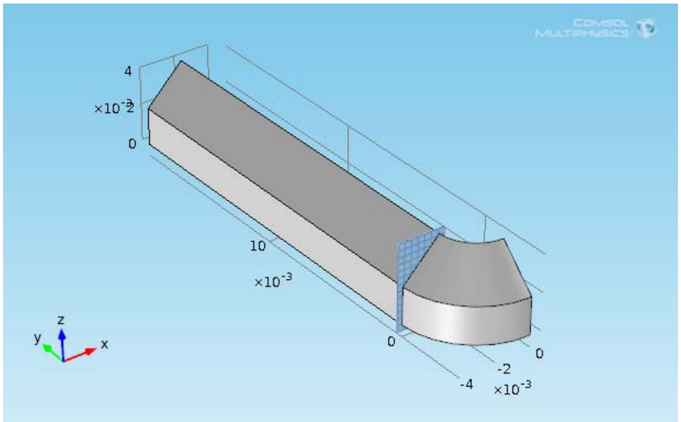
4 Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ) to view the object **rev1**.

ADDING AN EXTRUSION AND UNION

1 In the **Model Builder**, right-click **Work Plane1** and select **Extrude** ( ).

2 Under **Distances from Plane**, enter `-2e-2` in the **Distances** row.

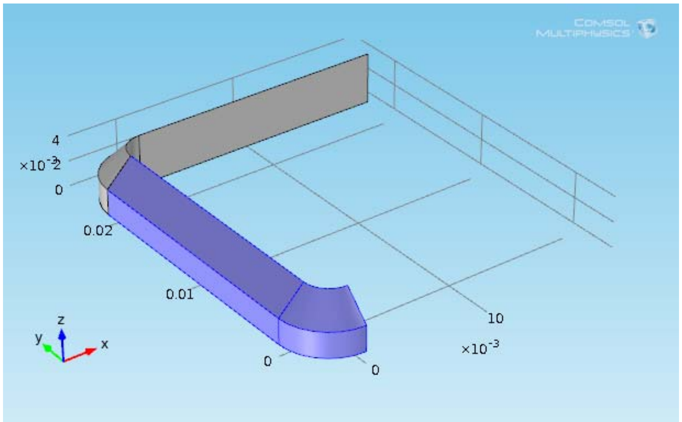
3 Click the **Build Selected** button ( ) and the **Zoom Extents** button ( )to view the object **ext1**.



4. In the **Model Builder**, under **Model 1**, right-click **Geometry 1** and select **Boolean Operations>Union** ( ).
5. Select the objects **rev1** and **ext1** and add them to the **Input objects** section.
6. On the **Union** page, under **Union**, click to clear the **Keep interior boundaries** check box to remove the interior boundary between the corner section and the edge section of the heat sink.
7. Click the **Build Selected** button ( ). Objects **rev1** and **ext1** are combined to create object **uni1**.

ADDING A ROTATION TO THE 3D OBJECT

1. In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ( ).
2. Select the object **uni1** and add it to the **Input objects** section under **Input**.
3. Select the **Keep input objects** check box to leave the input object intact as a rotation of the object is created.
4. Under **Rotation Angle**, enter `-90` in the **Rotation** field.
5. Under **Point on Axis of Rotation**:
   a. In the **x** field, enter `1e-2`.
   b. In the **y** field, enter `1e-2`.
6. Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ) to view the object **rot1**.



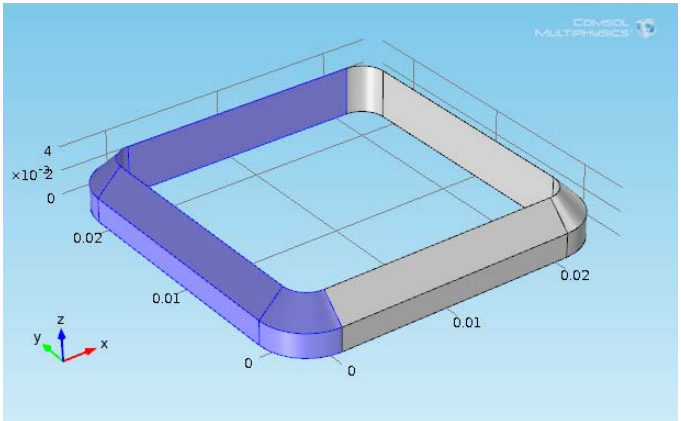
CREATING UNION 2

1. In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations> Union** ( ).
2. Select the objects **uni1** and **rot1** and add them to the **Input objects** section on the **Union** page.
3. Under **Union**, click to clear the **Keep interior boundaries** check box.
4. Click the **Build Selected** button ( ) to create object **uni2**.

ADDING A SECOND ROTATION

1. In the **Model Builder**, right-click **Geometry1** and select **Transforms>Rotate** ( ).
2. Select the object **uni2** and add it to the **Input objects** section under **Input**.
3. Select the **Keep input objects** check box.
4. Under **Rotation Angle**, enter `-180` in the **Rotation** field.
5. Under **Point on Axis of Rotation**:
   a. In the **x** field, enter `1e-2`.
   b. In the **y** field, enter `1e-2`.

6 Click the **Build Selected** button ( ).



CREATING UNION 3

1 Right-click **Geometry 1** and select **Boolean Operations>Union**.

The **Union** page opens in the settings window.

2 Select the objects **uni2** and **rot2** and add them to the **Input objects** section under **Union**.

3 Click to clear the **Keep interior boundaries** check box.

4 Click the **Build Selected** button ( ) to create object **uni3**.

CREATING WORK PLANE 2 AND ADDING A SQUARE

1 In the **Model Builder**, right-click **Geometry 1** and select **Work Plane**.

The **Work Plane** page opens in the settings window and a **Work Plane 2** node is added in the **Model Builder**.

2 Click the **Build Selected** button ( ).

3 To the right of the settings window, click the **Show Work Plane** button ( ). Use the projection of the 3D geometry on the xy-plane as a guide for creating the middle section of the heat sink base.

4 In the **Model Builder**, under **Work Plane 2**, right-click **Plane Geometry** and select **Square** ( ).

The **Square** page opens in the settings window.

5 Under **Size**, enter `2.4e-2` in the **Side length** field.

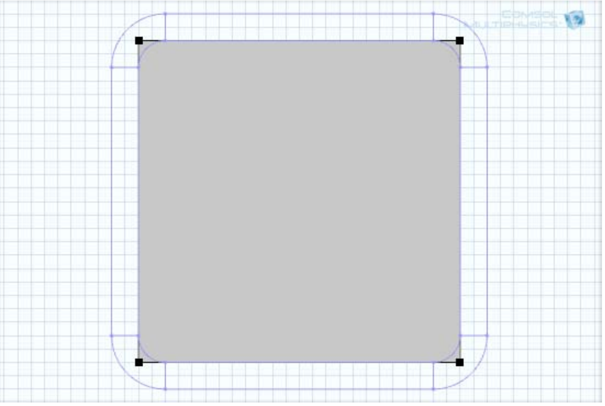
6 Under **Position**:

   a Select **Center** from the **Base** list.

   b In the **xw** field, enter `1e-2`.

   c In the **yw** field, enter `1e-2`.

7 Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).



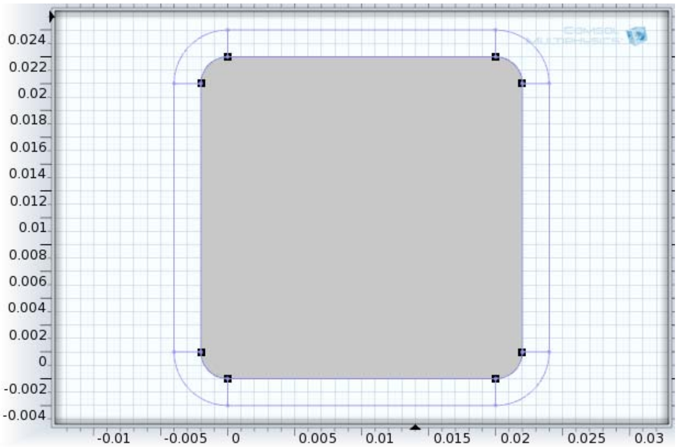
TRIMMING THE SQUARE TO FIT USING THE FILLET OPERATION

1 In the **Model Builder**, under **Work Plane 2**, right-click **Plane Geometry** and select **Fillet** ( ).

2 Add points 1, 2, 3, and 4 on the object **sq1** to the **Vertices to fillet** section under **Points**.

3 Under **Radius**, enter `2e-3` in the **Radius** field.

4 Click the **Build Selected** button.( )



ADDING EXTRUDE 2 AND COMBINING OBJECTS TO COMPLETE THE BASE

1 In the **Model Builder**, right-click **Geometry 1** and select **Extrude** ( ).

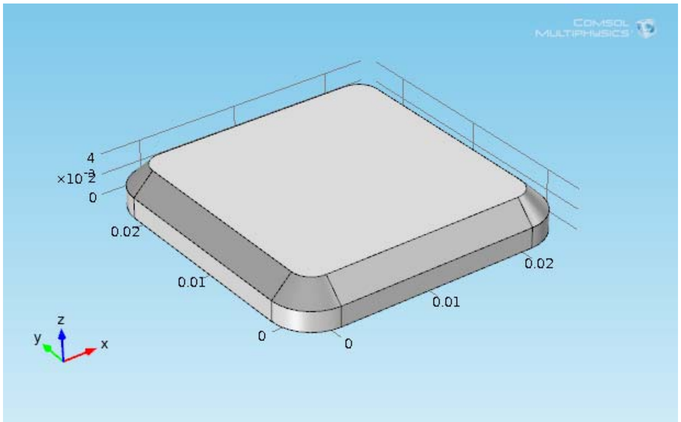
2 Under **Distances from Plane**, enter `4e-3` in the **Distances** row.

3 Click the **Build Selected** button ( ).

4 In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations> Union** ( ).

5 Select the objects **uni3** and **ext2** to add to the **Input objects** section under **Union**.

6 Click the **Build Selected** button ( ) to create object **uni4**. This completes the base of the heat sink.



DRAWING THE UPPER PART OF THE HEAT SINK

Creating a Work Plane and a Square

1 In the **Model Builder**, right-click **Geometry 1** and select **Work Plane** ( ).

The **Work Plane** page opens in the settings window and a **Work Plane 3** node is added in the **Model Builder**.

2 Under **Work Plane**, enter `4e-3` in the **z-coordinate** field.

3 From the **3D projection** list, **Entire 3D geometry** is selected by default, and this visualizes the projected edges of the heat sink's base in the work plane.

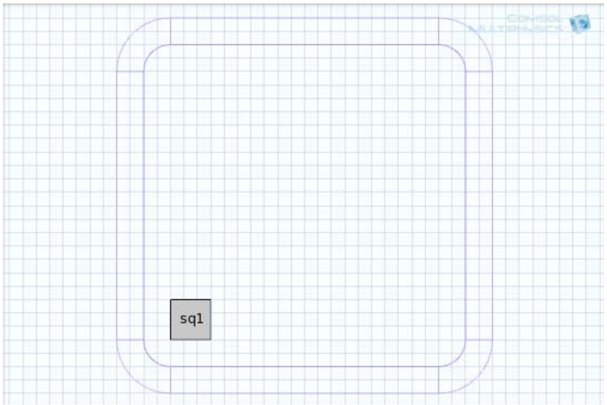
4 Click the **Build Selected** button ( ).

5 To the right of the settings window, click the **Show Work Plane** button ( ).

6 In the **Model Builder**, under **Work Plane 3**, right-click **Plane Geometry** and select **Square** ( ).

7 Under **Size**, enter `L2` in the **Side length** field.

8 Click the **Build Selected** button ( ) to create square **sq1** with side length L2.



ADDING ARRAYS

1 In the **Model Builder**, under **Work Plane 3**, right-click **Plane Geometry** and select **Transforms>Array** ( ).

2 Add the object **sq1** to the **Input objects** section under **Input**.

3 Under **Size**:

   a In the **xw size** field, enter `3`.

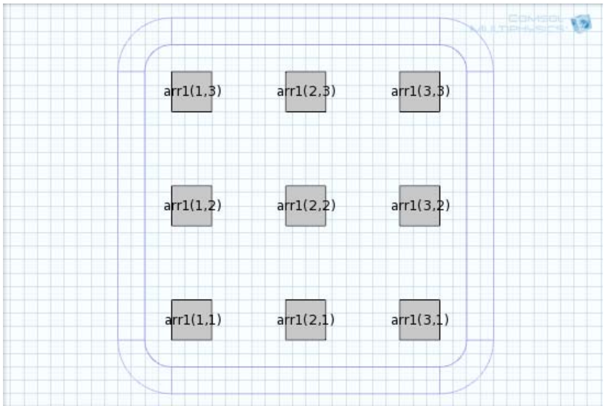
   b In the **yw size** field, enter `3`.

4 Under **Displacement**:

   a In the **xw** field, enter `1e-2-L2/2`.

   b In the **yw** field, enter `1e-2-L2/2`.

5 Click the **Build Selected** button ( ).



Adding Extrude 3 and Combining Objects (Union)

1 In the **Model Builder**, right-click **Geometry 1** and select **Extrude** ( ).

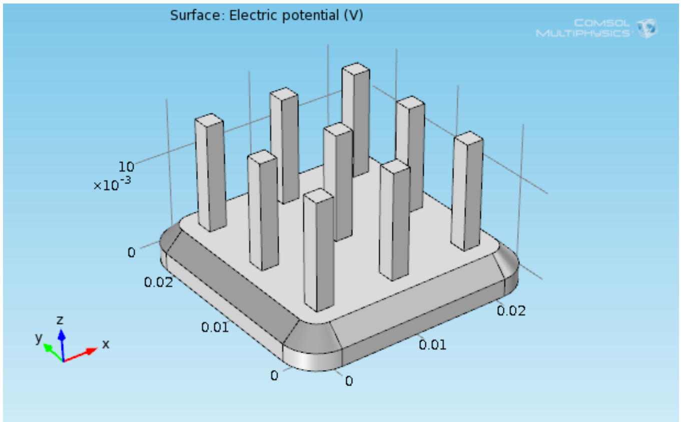
2 Under **Distances from Plane**, enter `L1` in the **Distances** row.

3 Click the **Build Selected** button ( ) and the **Zoom Extents** button ( ).

4 Right-click **Geometry 1** again and select **Boolean Operations>Union** ( ).

5 Add all the objects (`uni4` and all the `ext3` just built) to the **Input objects** list under **Union**.

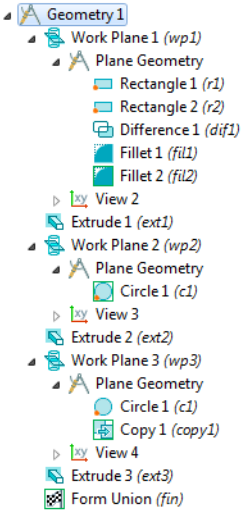
6 Click the **Build All** button ( ) to complete the heat sink geometry.



THE GEOMETRY SEQUENCE

Figure 5-10 shows the list of the geometry in the **Model Builder** used to create Figure 5-9. You can edit any node for each of the drawing operations.

In this case, the upper part of the heat sink is parameterized through the thickness and height of the heat sink pillars. You can edit the parameter values defined previously to change the heat sink geometry.



- Geometry 1
  - Work Plane 1 *(wp1)*
    - Plane Geometry
      - Rectangle 1 *(r1)*
      - Rectangle 2 *(r2)*
      - Difference 1 *(dif1)*
      - Fillet 1 *(fil1)*
      - Fillet 2 *(fil2)*
    - View 2
  - Extrude 1 *(ext1)*
  - Work Plane 2 *(wp2)*
    - Plane Geometry
      - Circle 1 *(c1)*
    - View 3
  - Extrude 2 *(ext2)*
  - Work Plane 3 *(wp3)*
    - Plane Geometry
      - Circle 1 *(c1)*
      - Copy 1 *(copy1)*
    - View 4
  - Extrude 3 *(ext3)*
  - Form Union *(fin)*

Figure 5-10: Sample 3D Model Builder Geometry Sequence. From the Busbar model in the COMSOL Multiphysics Model Library.

RE-RUNNING THE GEOMETRY SEQUENCE WITH DIFFERENT PARAMETERS

1 In the **Model Builder**, under **Global Definitions**, click **Parameters** ( ).

2 Under **Parameters** enter the following settings in the table. Replace the previous data:

NAME	EXPRESSION	VALUE	DESCRIPTION
L1	1.2e-2	0.012	Pillar thickness (in the heat sink)
L2	2e-3	0.0020	Pillar Length (in the heat sink)

3 In the **Model Builder**, click **Geometry 1**.

4 Click the **Build All** button ( ) and the **Zoom Extents** button ( ) to view the geometry as defined by the new parameters.

Using Virtual Geometry Operations

This section contains two examples that demonstrate the use of the virtual operations that are available for 2D and 3D geometries. The first example shows how to use the Ignore Vertices operation (or the Form Composite Edges operation) to remove a very short edge and how to use the Ignore Edges operation (or the Form Composite Faces operation) to prepare the geometry for swept meshing. The second example illustrates how you can use the Collapse Edges operation (or the Merge Vertices operation) to prepare the geometry for efficient meshing.

In this section:

- Virtual Geometry Operations
- Importing a Virtual Geometry
- Example 1: Forming Composite Edge and Face by Ignoring Vertex and Edges
- Example 2: Merging Vertices by Collapsing Edge

Virtual Geometry Operations

For a 2D or a 3D geometry you can add *virtual geometry operations* after the **Finalize** node to, for example, remove small details from the geometry and to prepare it for efficient meshing and analysis. The geometry sequence before the **Finalize** node defines the “real” (original) geometry. The geometry resulting from a virtual operation is referred to as a *virtual geometry*. These virtual operations are available from the **Geometry>Virtual Operations** menu:

- Ignore Vertices
- Ignore Edges
- Ignore Faces (3D only)
- Form Composite Edges
- Form Composite Faces (3D only)
- Form Composite Domains

- Collapse Edges
- Merge Vertices

Note

For details about these operations, see Features for Virtual Operations in the *COMSOL Multiphysics Reference Guide*. For examples on how to use the virtual operations, see Using Virtual Geometry Operations.

Another set of virtual operations is available for mesh control; that is for controlling the position and distribution of mesh elements without being part of the geometry used for analysis. You can use the following mesh control features:

- Mesh Control Vertices
- Mesh Control Edges
- Mesh Control Faces (3D only)
- Mesh Control Domains

Note

For details about the mesh control features, see Features for Mesh Control in the *COMSOL Multiphysics Reference Guide*. For examples on how to use the mesh control features, see Mesh Control Entities.

Importing a Virtual Geometry

It is possible to import a virtual geometry to a sequence using the Import node. The source can be either a COMSOL Multiphysics file or a sequence. If you import from a file or a sequence containing a virtual geometry the Import node imports the “real” (original) geometry that was used to define the virtual geometry.

Example 1: Forming Composite Edge and Face by Ignoring Vertex and Edges

MODEL WIZARD

1 Go to the **Model Wizard**.

2 Click **Finish** ( ).

GEOMETRY 1

Import 1

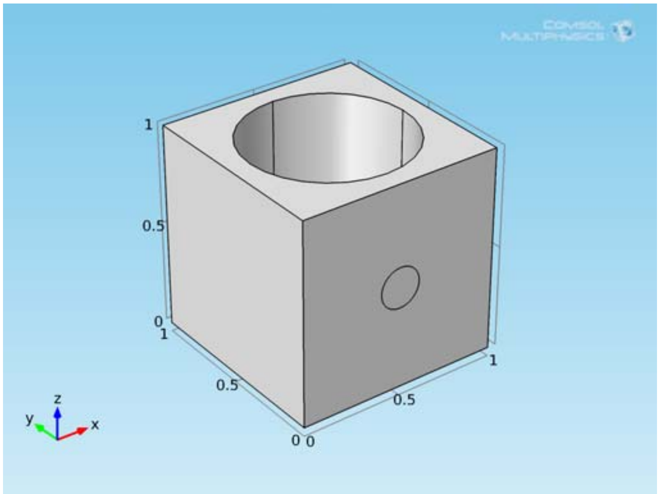
1 In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Import** ( ).

2 Go to the settings window for Import. Under the **Import** section, click **Browse**.

3 In the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click the file `virtualgeom_demo_1.mphbin`.

4 Click **Import**.

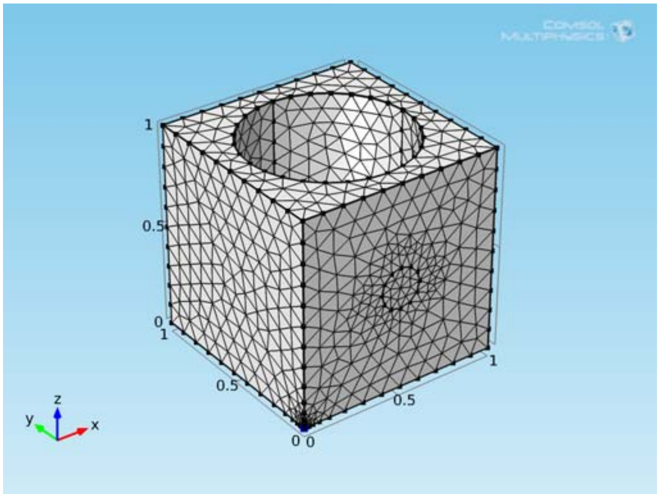
The imported geometry displays in the **Graphics** window.



MESH 1

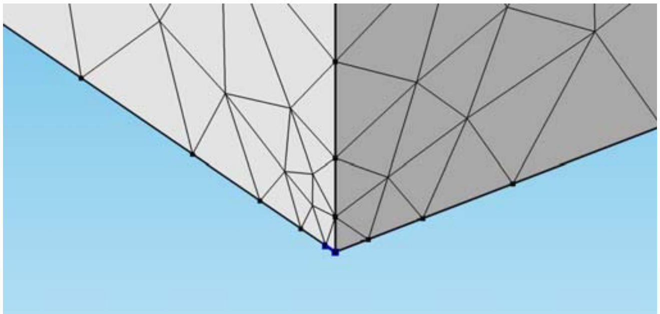
In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

The resulting mesh displays in the **Graphics** window.



Note

A **Warning** node is added under **Mesh 1** indicating that there is a very short edge in the geometry. Use the **Zoom Selected** button ( ) and the **Zoom Out** button ( ) on the **Graphics** toolbar to locate this edge in the **Graphics** window.



Eliminate the short edge by ignoring the vertex between this edge and its adjacent longer edge.

GEOMETRY 1

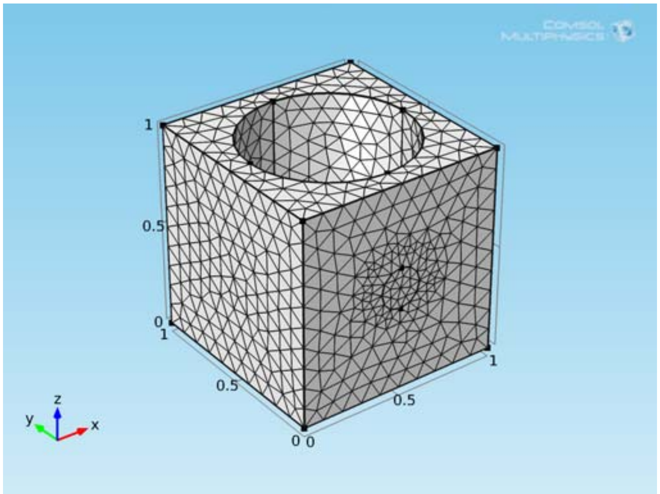
Ignore Vertices 1

1. In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Virtual Operations>Ignore Vertices** ( ).
2. Select Point 3.
3. Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

The mesh displays in the **Graphics** window.



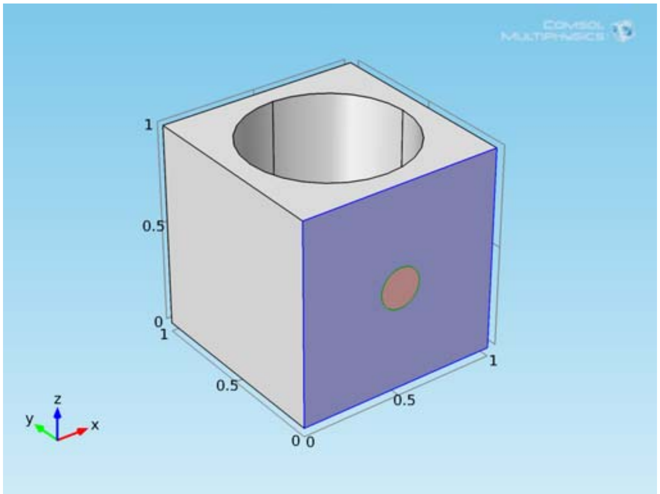
The geometry’s domain is well suited for swept meshing.

Swept 1

1. Right-click **Model 1>Mesh 1** and choose **Swept** ( ).
2. Go to the settings window for Swept. In the upper-right corner of the **Source Faces** section, click **Activate Selection** ( ).
3. Select Boundary 3.
4. In the upper-right corner of the **Destination Faces** section, click **Activate Selection** ( ).
5. Select Boundary 4.

Size

1. In the **Model Builder**, click the **Size** node ( ).
2. Go to the settings window for Size. Locate the **Element Size** section. From the **Predefined** list, choose **Finer**.
3. Click the **Build All** button ( ).
4. Click **OK** to close the close the **COMSOL Error** window. COMSOL fails to create a swept mesh for this geometry due to the circular imprint on one of the linking faces of the sweep.



Use the **Ignore Edges** operation to remove this imprint.

GEOMETRY 1

Ignore Edges 1

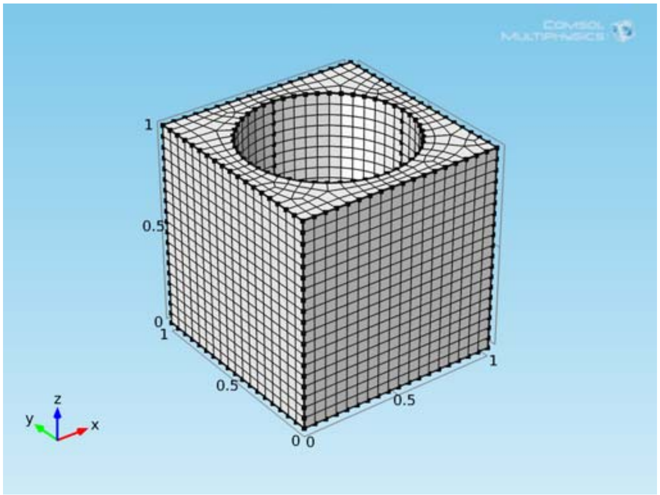
1. In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Virtual Operations>Ignore Edges** ( ).
2. Select Edges 14 and 15.

   Make sure to select the **Ignore adjacent vertices** check box such that the vertices of the imprint disappear as well.
3. Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

The swept mesh displays in the **Graphics** window.



Tip

The same virtual geometry can be achieved by using the **Form Composite Edges** and the **Form Composite Faces** operations.

GEOMETRY 1

Disable

1 In the **Model Builder**, right-click **Model 1>Geometry 1>Ignore Edges 1** and choose **Disable** ( ).

2 Right-click **Ignore Vertices 1** and choose **Disable** ( ).

Form Composite Edges 1

1 Right-click **Geometry 1** and choose **Virtual Operations>Form Composite Edges** ( ).

2 Select Edges 2 and 6.

3 Click the **Build Selected** button ( ).

Form Composite Faces 1

1 In the **Model Builder**, right-click **Geometry 1** and choose **Virtual Operations>Form Composite Faces** ( ).

2 Select Boundaries 2 and 8.

3 Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

Example 2: Merging Vertices by Collapsing Edge

MODEL WIZARD

1 Go to the **Model Wizard**.

2 Click **Finish** ( ).

GEOMETRY 1

Import 1

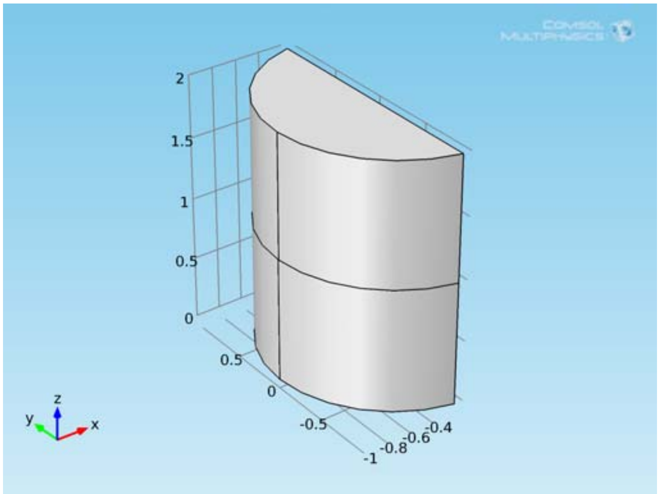
1 In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Import** ( ).

2 Go to the settings window for Import. Locate the **Import** section. Click **Browse**.

3 In the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click the file `virtualgeom_demo_2.mphbin`.

4 Click **Import**.

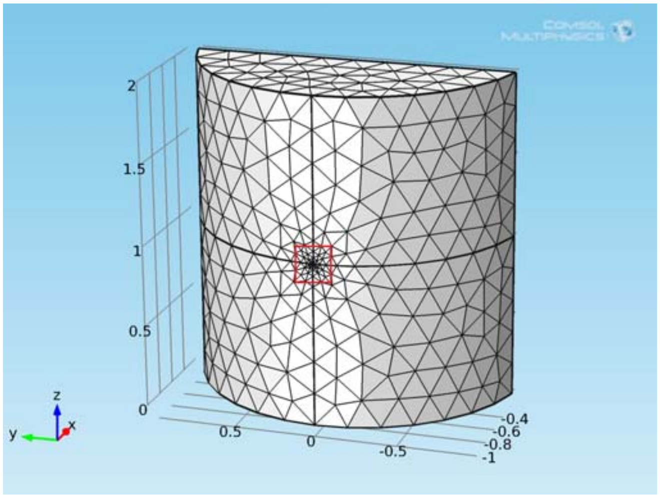
The imported geometry displays in the **Graphics** window.



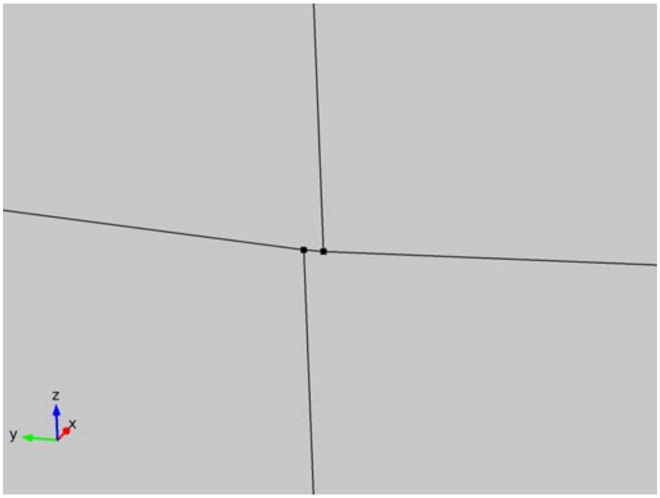
GEOMETRY 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

The resulting mesh displays in the **Graphics** window.



The figure shows that the mesh is very fine in the region marked by the red box. If you select **Model 1>Geometry 1** in the **Model Builder** and zoom into this region you can find the reason for the fine mesh. There is a very short edge in the conjunction of the four curved faces of the geometry.



Remove this short edge by collapsing it into a vertex.

Collapse Edges 1

1 In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Virtual Operations>Collapse Edges** ( ).

2 Select Edge 4.

3 Click the **Build Selected** button ( ).

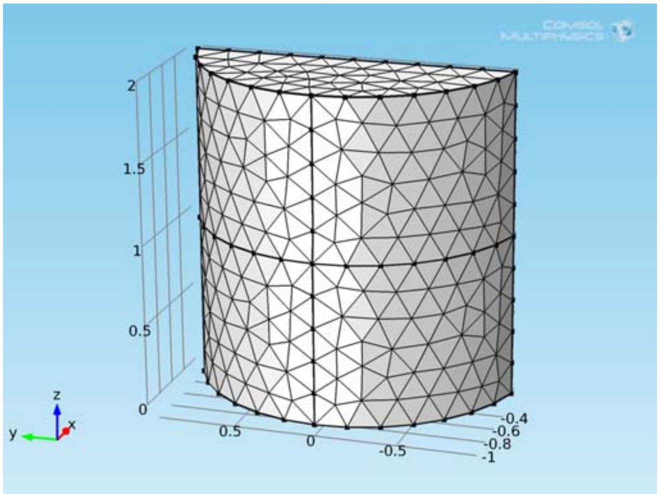
Tip

It is also possible to remove this short edge by an **Ignore Edges** operation forming a composite face of the two adjacent faces. However, then the small distance between the vertices of the ignored edge would still be present.

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

The mesh displays in the **Graphics** window.



Tip

The same virtual geometry can be achieved by using the **Merge Vertices** operation.

GEOMETRY 1

Right-click **Collapse Edges 1** and choose **Disable** ( ).

Merge Vertices 1

1. In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Virtual Operations>Merge Vertices** ( ).
2. Select Point 2.
3. Go to the settings window for Merge Vertices. Locate the **Vertex to Remove** section. Click **Activate Selection** (⏻).
4. Select Point 3.
5. Click the **Build Selected** button ( ).

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ( ).

6

Materials

This chapter describes the material databases included with COMSOL Multiphysics and the add-on modules.

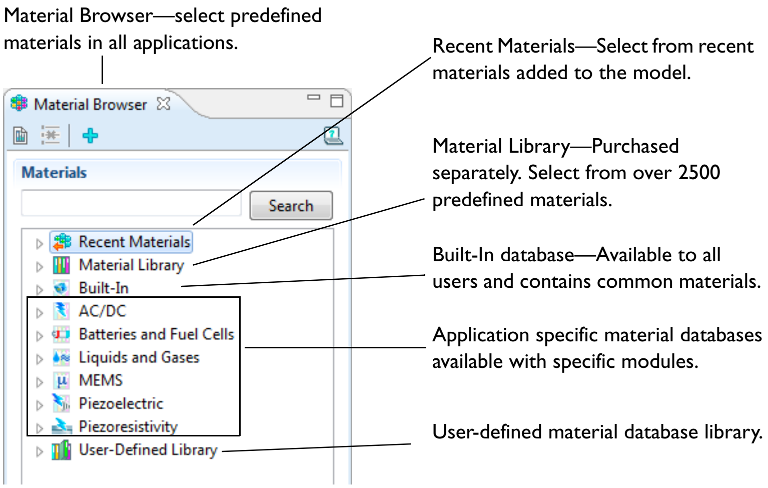
In this chapter:

Materials Overview

In this section:

- About the Material Databases
- About Using Materials in COMSOL
- Opening the Material Browser
- Using Material Properties

About the Material Databases


Material Browser—select predefined materials in all applications.
Recent Materials—Select from recent materials added to the model.
Material Library—Purchased separately. Select from over 2500 predefined materials.
Built-In database—Available to all users and contains common materials.
Application specific material databases available with specific modules.
User-defined material database library.
Material Browser
Materials
Search
Recent Materials
Material Library
Built-In
AC/DC
Batteries and Fuel Cells
Liquids and Gases
MEMS
Piezoelectric
Piezoresistivity
User-Defined Library

All COMSOL modules have predefined material data available to build models. The most extensive material data is contained in the separately purchased Material Library, but all modules contain commonly used or module-specific materials. For example, the *Built-In* database is available to all users but the *MEMS* database is included with the MEMS Module and Structural Mechanics Module. Also create custom materials and material libraries by researching and entering material properties.

All the material databases (including the Material Library) are accessed from the **Material Browser**. These databases are briefly described below.

Recent Materials

From the **Recent Materials** folder ( ), select from a list of recently used materials, with the most recent at the top. This folder is available after the first time a material is added to a model.

Material Library

An optional add-on database, the **Material Library** ( ), contains data for over 2500 materials and 20,000 property functions.

Built-In

Included with COMSOL Multiphysics, the **Built-In** database ( ) contains common solid materials with electrical, structural, and thermal properties.

See Also

Predefined Built-In Materials for all COMSOL Modules

AC/DC

Included in the AC/DC Module, the **AC/DC** database ( ) has electric properties for some magnetic and conductive materials.

Batteries and Fuel Cells

Included in the Batteries & Fuel Cells Module, the **Batteries and Fuel Cells** database ( ) includes properties for electrolytes and electrode reactions for certain battery chemistries.

Liquids and Gases

Included in the Acoustics Module, CFD Module, Chemical Reaction Engineering Module, Heat Transfer Module, MEMS Module, Pipe Flow Module, and Subsurface Flow Module, the **Liquids and Gases** database ( ) includes transport properties and surface tension data for liquid/gas and liquid/liquid interfaces.

MEMS

Included in the MEMS Module and Structural Mechanics Module, the **MEMS** database (µ) has properties for MEMS materials—metals, semiconductors, insulators, and polymers.

PIEZOELECTRIC

Included in the Acoustics Module, MEMS Module, and Structural Mechanics Module, the **Piezoelectric** database ( ) has properties for piezoelectric materials.

PIEZORESISTIVITY

Included in the MEMS Module, the **Piezoresistivity** database ( ) has properties for piezoresistive materials, including p-Silicon and n-Silicon materials.

USER-DEFINED LIBRARY

The **User-Defined Library** folder ( ) is where user-defined materials databases (libraries) are created. When any new database is created, this also displays in the **Material Browser**.

Important

The materials databases shipped with COMSOL Multiphysics are read-only. This includes the Material Library and any materials shipped with the optional modules.

See Also

Creating Your Own User-Defined Libraries

About Using Materials in COMSOL

USING THE MATERIALS IN THE PHYSICS SETTINGS

The physics set-up in a model is determined by a combination of settings in the **Materials** and physics interface nodes. When the first material is added to a model, COMSOL automatically assigns that material to all domains in the geometry. Different geometric entities can have different materials. The following example uses the *heat_sink.mph* model file contained in the Heat Transfer Module and CFD Module Model Libraries.

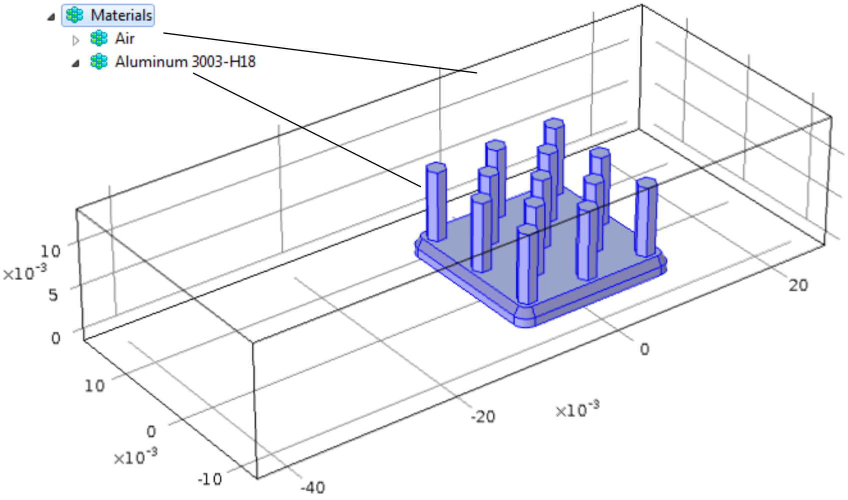


Figure 6-1: Assigning materials to a heat sink model. Air is assigned as the material to the box surrounding the heat sink, and aluminum to the heat sink itself.

If a geometry consists of a heat sink in a container, **Air** can be assigned as the material in the container surrounding the heat sink and **Aluminum** as the heat sink material itself (see Figure 6-1). The **Conjugate Heat Transfer** interface, selected during model set-up, uses a **Fluid** model to simulate nonisothermal flow, with heat transfer by convection and conduction, in the box surrounding the heat sink, and a **Heat Transfer in Solids** model to simulate heat conduction in the heat sink. The **Heat Transfer in Solids 1** settings use the material properties associated to the **Aluminum 3003-H18** materials node, and the **Fluid 1** settings define the flow using the **Air** material properties. The other nodes under **Conjugate Heat Transfer** define the initial and boundary conditions.

All physics interface properties automatically use the correct **Materials'** properties when the default **From material** setting is used. This means that one node can be used to define the physics across several domains with different materials; COMSOL then uses the material properties from the different materials to define the physics in each domain. If material properties are missing, the **Material Contents** section on the

Materials page displays a stop icon (🛑) to warn about the missing properties and a warning icon (⚠) if the property exists but its value is undefined.

See Also

The Material Page

There are also some physics interface features where you can explicitly select a material from which material properties will be retrieved. The default setting is then always to use the **Domain material** on each domain (that is, the materials defined on the same domains as the physics interface). In addition to the **Domain Material**, you can select any other material that is present in the model, regardless of its selection. The selected material's properties will then be applied to all domains in the feature's selection.

EVALUATING AND PLOTTING MATERIAL PROPERTIES

You can access the material properties for evaluation and plotting like other variables in a model using the variable naming conventions and scoping mechanisms:

- To access a material property throughout the model (across several materials) and not just in a specific material, use the special material container `root.material`. For example, `root.material.rho` is the density ρ as defined by the materials in each domain in the geometry. For plotting, you can type the expression `material.rho` to create a plot that shows the density of all materials.

Note

If you use a temperature-dependent material, each material contribution asks for a special model input. For example, `rho(T)` in a material `mat1` asks for `root.mat1.def.T`, and you need to define this variable (`T`) manually—if the temperature is not available as a dependent variable—to make the density variable work.

- To access a material property from a specific material, you need to know the tags for the material and the property group. Typically, for the first material (Material 1) the tag is `mat1` and most properties reside in the default **Basic** property group with the tag `def`. The variable names appear in the **Variable** column in the table under **Output properties** in the settings window for the property group; for example, `Cp` for the heat capacity at constant pressure. The syntax for referencing the heat capacity at constant pressure in Material 1 is then `mat1.def.Cp`. Some properties are anisotropic tensors, and each of the components can be accessed, such as `mat1.def.k11`, `mat1.def.k12`, and so on, for the thermal conductivity. For material properties that are functions, call these with input arguments such as `mat1.def.rho(pA,T)` where `pA` and `T` are numerical values or variables representing the absolute pressure and the temperature, respectively. The functions can be plotted directly from the function nodes' settings window by first specifying suitable ranges for the input arguments.

- Many physics interfaces also define variables for the material properties that they use. For example, `solid.rho` is the density in the Solid Mechanics interface and is equal to the density in a material when it is used in the domains where the Solid Mechanics interface is active. If you define the density in the Solid Mechanics interface using another value, `solid.rho` represents that value and not the density of the material. If you use the density from the material everywhere in the model, `solid.rho` and `material.rho` are identical.

Opening the Material Browser

1 Open or create a model file.

2 From the **View** menu choose **Material Browser** or right-click the **Materials** node and choose **Open Material Browser**.

3 Under **Material Selection**, search or browse for materials.

   - Enter a **Search** term to find a specific material by name, UNS number (Material Library materials only), or DIN number (Material Library materials only). If the search is successful, a list of filtered databases containing that material displays under **Material Selection**.

Tip

To clear the search field and browse, delete the search term and click **Search** to reload all the databases.

   - Click to open each database and browse for a specific material by class (for example, in the Material Library) or physics module (for example, MEMS Materials).

Important

Always review the material properties to confirm they are applicable for the model. For example, **Air** provides temperature-dependent properties that are valid at pressures around 1 atm.

4 When the material is located, right-click to **Add Material to Model**.

A node with the material name is added to the **Model Builder** and the **Material** page opens.

Using Material Properties

See Also	For detailed instructions, see Adding Predefined Materials and Material Properties Reference.

About the Material Browser

Use the nodes under **Materials** to add predefined or user-defined materials, specify specific properties, model inputs, functions, values and expressions, or create a custom material library.

MATERIAL OVERVIEW

This section provides an overview of the materials in the model and where they are used.

The **Material** column lists the current materials in the model with their names and, within curly braces, their tags.

The **Selection** column lists the geometric entities selected for the material (the domains, boundaries, or edges where the material is defined).

If a material property in a physics interface takes its value from a material and no material is defined for the same geometric selection, a stop sign (🛑) displays in the leftmost column, and the **Material** column contains **Entities needing a material**. The **Selection** column contains the geometric entities where a material definition is missing.

In this section:

- The Material Browser Window
- The Material Page
- Errors Relating to the Material Nodes
- Property Groups

See Also

- Materials
- About Selecting Geometric Entities
- Global and Local Functions

The Material Browser Window

Note

When using the **Material Browser**, the words *window* and *page* are interchangeable. For simplicity, the instructions refer only to the **Material Browser**.

The *Material Browser* window ( ) contains a number of databases with a broad collection of physical, elastic, electromagnetic, fluid, piezoelectric, and thermal properties of materials (Figure 6-2). Use the Material Browser to find predefined materials and add them to the Model Builder, or create a custom material library.

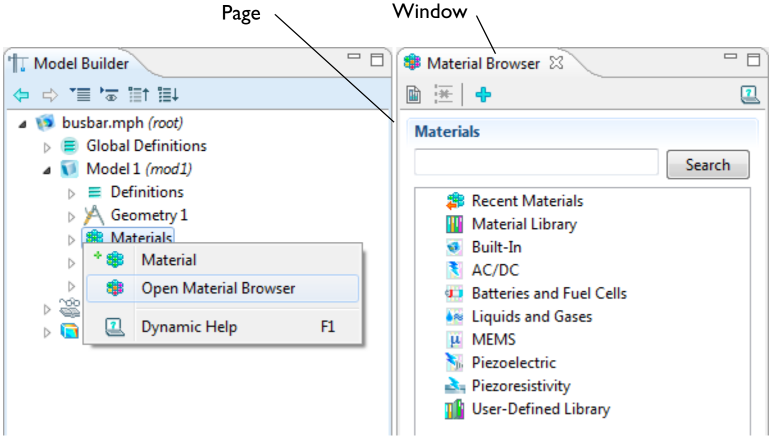


Figure 6-2: Opening the Material Browser page in the Material Browser window.

MATERIALS

Browse all the available material databases or search in the **Materials** section. **Search** a specific material by name, UNS number, or DIN number. Also browse for a specific material by class (for example, in the Material Library) or physics module (for example, MEMS). When the material is located, right-click to **Add Material to Model**.

INFORMATION

When browsing the material databases, in particular the **Model Library**, some materials include additional information—UNS number, DIN number, and/or Composition.

PROPERTIES

While browsing the databases, predefined **Properties** are listed in a table together with their **Expression**, **Unit**, and **Property groups** to which the material property belongs. If **Property groups** is empty, the material propery is a **Basic** property.

FUNCTIONS

While browsing some materials, predefined **Functions** are listed in a table together with their **Type**, **Expression**, and **Property groups** where the function is used. If **Property groups** is empty, the function is defined in the **Basic** property group.

INPUT

While browsing some materials, predefined **Input** are listed in a table together with their **Unit** and **Property groups** which require the input to evaluate its properties.

See Also

- Materials

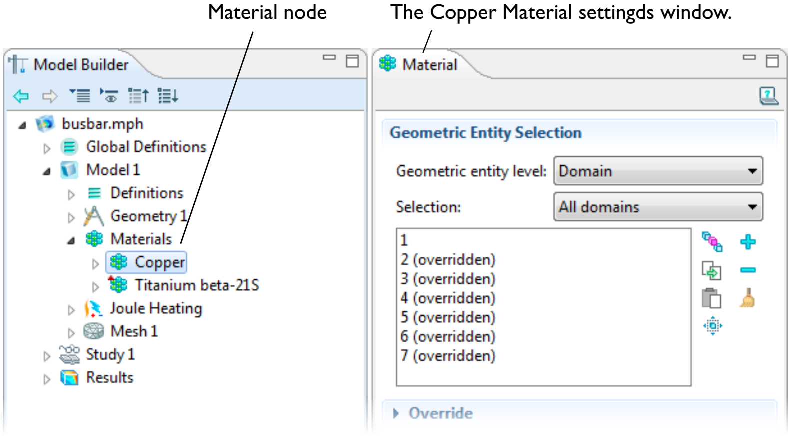
When you have the Material Library:

- Material Library Quick Start
- About the Material Library Folders and Properties

The Material Page

The **Material** page ( ) summarizes the predefined or user-defined material properties. This is where you can add or change material properties to fit your model and assign the material to geometric entities (domains, boundaries, edges or points).

After adding a material, click the material node (for example, **Air**) in the **Model Builder**. The **Material** page opens in the settings window (Figure 6-3).


Material node

The Copper Material settingds window.

Geometric entity level: Domain

Selection: All domains

1
2 (overridden)
3 (overridden)
4 (overridden)
5 (overridden)
6 (overridden)
7 (overridden)

Figure 6-3: Click the Copper node to open the Material settings window for the node.

GEOMETRIC ENTITY SELECTION

Assign the material to a specific **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point**—on the geometry in the **Graphics** window (the geometry in the model).

Note

If the model contains physics on different geometric entity levels, such as solid mechanics in domains coupled to beams on edges, and the physics use the same material, you need to add two Material nodes with the same material, one defined in the domains, and the other defined on the edges.

MATERIAL PROPERTIES

Add additional material properties to the model if they are not already included in the default settings.

A Note about Adding Basic Material Properties

Material properties can be added to the **Basic** group or to any **User-Defined Property Group** from two locations—the **Material** and **Property Group** pages.

- When material properties are added from the **Basic** node’s or a user user-defined group node’s **Property Group** page, they are listed under **Quantities>Output Properties**.
- When basic material properties are added from the **Material** page, they are listed under **Material Properties>Basic Properties** and will always be added to the **Basic** group.

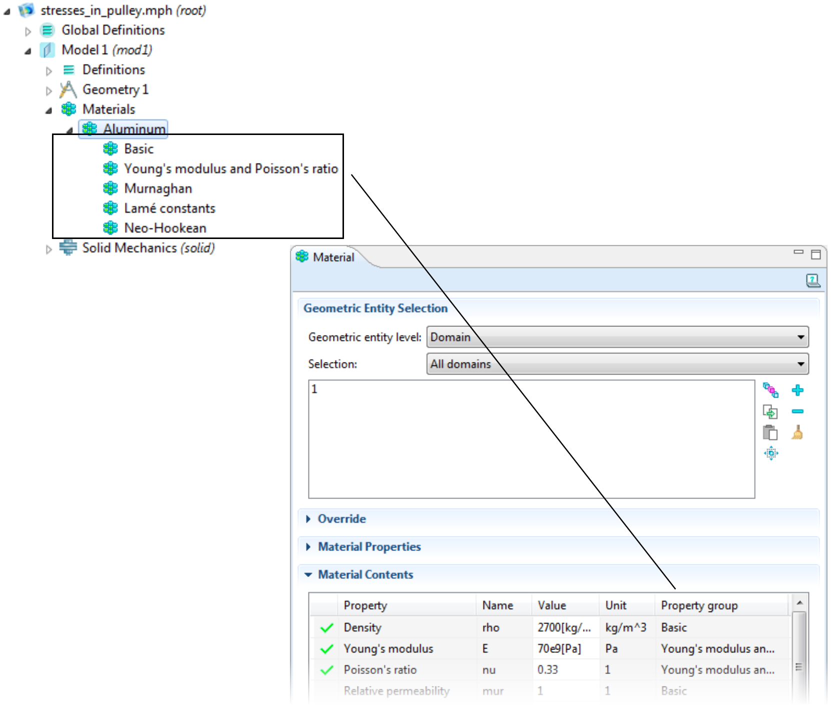
MATERIAL CONTENTS

This section lists all the material properties that are defined for the material or required by the physics interfaces in the model. The table lists the **Property**, **Name**, **Value**, **Unit**, and the **Property group** that the material property belongs to. The **Property group** corresponds to the nodes in the **Model Builder** with the same name (see Figure 6-4). If required, edit the values or expression for the property’s **Value**. A stop sign displays in the left column to indicate when a required property is missing a **Value**.

The left column provides visual cues about the status of each property:

- A stop sign (🛑) indicates that an entry in the **Value** column is required. It means that the material property is required by a physics interface but undefined. When you enter a value in the **Value** column, the material property is added to the correct property group.
- A warning sign (⚠) indicates that the material property has been added to the material but is still undefined. An entry is only required if the material property will be used in the model.

- A green check mark (✓) indicates that the property has a **Value** and is currently being used in the physics of the model.
- Properties with no indication in the left column are defined but not currently used by any physics interface in the model.



- stresses_in_pulley.mph *(root)*
  - Global Definitions
  - Model 1 *(mod1)*
    - Definitions
    - Geometry 1
    - Materials
      - Aluminum
        - Basic
        - Young's modulus and Poisson's ratio
        - Murnaghan
        - Lamé constants
        - Neo-Hookean
    - Solid Mechanics *(solid)*

Material

Geometric Entity Selection

Geometric entity level: Domain

Selection: All domains

1

Override

Material Properties

Material Contents

	Property	Name	Value	Unit	Property group
✓	Density	rho	2700[kg/...	kg/m^3	Basic
✓	Young's modulus	E	70e9[Pa]	Pa	Young's modulus an...
✓	Poisson's ratio	nu	0.33	1	Young's modulus an...
	Relative permeability	mur	1	1	Basic

Figure 6-4: An example of Property Groups on the Material page.

APPEARANCE

Control or change the default appearance of a material in the Graphics window when working in the Materials or Physics interface sections.

2D

For 2D models, only a change of color is possible.

3D

For 3D models, the material appearance is with both color and texture.

The default **Family** appearance is taken from the material selected, which has a suitable color (2D and 3D models), and texture (3D models) applied. Select another option from the **Family** list if required—**Air**, **Aluminum**, **Brick**, **Concrete**, **Copper**, **Gold**, **Iron**, **Lead**, **Magnesium**, **Plastic**, **Steel**, **Titanium**, or **Water**. Select **Custom** to make further adjustments. The default custom settings are inherited from the material selected last from the **Family** list.

Note

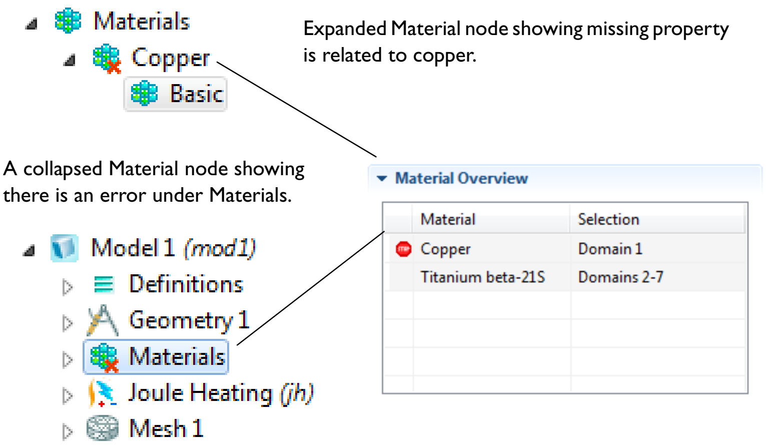
The appearance includes texture and reflectance only when the preference settings for the visualization are optimized for quality. When optimized for performance, the appearance includes color only. Control this using the **Optimize for** list in the **Graphics** section in the **Preferences** dialog box.

See Also

- Errors Relating to the Material Nodes
- Materials
- Customizing the Appearance of a Material in the Graphics Window

Errors Relating to the Material Nodes

The Material nodes in the **Model Builder** also indicate when there is a material error. For example, if one or more **Output properties** are deleted, but assigned to a part of the geometry, then, depending on the location in the Model Tree, the icon indicates where the error is located and lists the details on the **Materials** page. See Figure 6-5.



Expanded Material node showing missing property is related to copper.

A collapsed Material node showing there is an error under Materials.

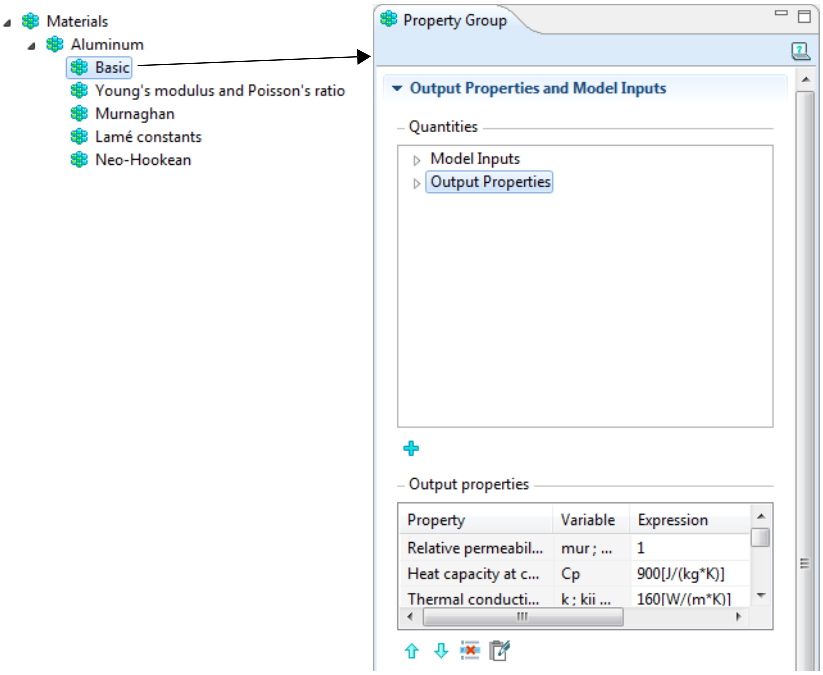
	Material	Selection
●	Copper	Domain 1
	Titanium beta-21S	Domains 2-7

Figure 6-5: An example of a Material node error. When the Materials node is expanded, the icon shows which node is missing information. When the Materials node is collapsed, the error indication also displays at the top level. Details about the missing information also display on the Materials page.

Property Groups

The **Property Group** page is where output properties and model inputs are added, local parameters defined, and expressions for material properties entered in a specific property group such as **Basic** (Figure 6-6). The property groups are subnodes to a material node.

The **Property Group** page is in the settings window and opens when the node is clicked (for example, **Basic**) under the material name in the **Model Builder**.



Property	Variable	Expression
Relative permeabil...	mur ; ...	1
Heat capacity at c...	Cp	900[J/(kg*K)]
Thermal conducti...	k ; kii ...	160[W/(m*K)]

Figure 6-6: An example of a Basic Property Group page.

OUTPUT PROPERTIES AND MODEL INPUTS

View predefined material properties in the **Output properties** table. Under **Quantities** add additional material properties to the **Output properties** list or add model inputs to the **Model inputs** list. The model inputs are physical quantities such as temperature that are used as inputs in the expressions that define the output properties. For example, adding **Temperature** as a model input with the variable name `T` makes it possible to use an expression such as `300[J/(kg*K)]*T[1/K]` that works regardless of the name of the actual dependent variable for temperature in the model that uses the temperature-dependent material. Without the model input, the expression above only works with a temperature variable called `T`.

If required, edit the expressions in the **Output properties** subsection's **Expression** column; edit directly in the table or by clicking the **Edit** button (📝), which opens the **Edit** dialog box. For an anisotropic material property (tensor), select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** when entering the data in the **Edit** dialog box. In the

Expression column, use a syntax with curly braces such as `{k11, k21, k31, k12, k22, k32, k13, k23, k33}` to enter anisotropic material properties.

A Note about Adding Basic Material Properties

Material properties can be added to models from two pages—**Material** and **Property Group**. In both cases, the properties belong to the **Basic** property group.

- When this information is accessed from the **Basic Property Group** page, it is listed under **Quantities>Output Properties** and the **Variable** is listed in the table.
- When this information is accessed from the **Material** page, it is listed under **Material Properties>Basic Properties** and the **Name** is listed in the tables.

LOCAL PARAMETERS

Enter a user defined **Parameter** and its corresponding **Expression** and organize the table as required.

About Automatic Adding of Property Groups to a Material

Material property groups are automatically added to the material node in the **Model Builder**. Also add additional predefined property groups or create a **User defined property group**. The available properties are grouped together differently according to the physical context. For example, the node in Figure 6-4 has five default property groups.

Each of the property groups has a **Property Group** page. When a **Model Builder** node is clicked (for example, **Basic**), the **Property Group** page displays specific information about that property group. The property groups are summarized in a **Material Contents** table on the **Material** page.

Adding Predefined Materials

In this section:

- Working in the Material Browser
- Working on the Material Page
- Customizing the Appearance of a Material in the Graphics Window
- Working on the Property Group Page
- Adding a User-Defined Property Group

Note

The first material added to a model is assigned to the entire geometry.

Working in the Material Browser

Note

When using the Material Browser, the words *window* and *page* are interchangeable. For simplicity, the instructions refer only to the Material Browser.

OPENING THE MATERIAL BROWSER

1 Open or create a model file.

2 From the **Options** menu select **Material Browser** or right-click the **Materials** node and select **Open Material Browser**.

The **Material Browser** opens by default to the right of the settings window.

ADDING A PREDEFINED MATERIAL TO A MODEL

1 Open the **Material Browser**.

2 Under **Materials**, search or browse for materials.

- Enter a **Search** term to find a specific material by name, UNS number (Material Library materials only), or DIN number (Material Library materials only). If the search is successful, a list of filtered databases containing that material displays under **Materials**.

- Click to open each database and browse for a specific material by class (for example, in the Material Library) or physics module (for example, MEMS).

3 When the material is located, right-click to **Add Material to Model**.

A node with the material name is added to the **Model Builder** and the **Material** page opens.

Tip

To clear the search field and browse, delete the search term and click **Search** to reload all the databases.

Important

Always review the material properties to confirm they are applicable for the model. For example, **Air** provides temperature-dependent properties that are valid at pressures around 1 atm.

Working on the Material Page

VIEWING A SUMMARY OF THE MATERIAL PROPERTIES

1 Add a material and geometry to the model.

2 In the **Model Builder**, click the material node to open the **Material** page in the settings window.

The **Material** page summarizes all the predefined data and has these sections:

- **Geometric Scope**: select the object **Domain**, **Boundary**, **Edge** (3D only), or **Point** to assign material to the geometry domain, boundary, edge or point.
- **Material Properties**: select and define specific material parameters to add to a model.
- **Material Contents**: lists all the default material property data. The table contains the **Property**, **Name**, **Value**, **Unit**, and the **Property group**. The **Property group** corresponds to the nodes in the **Model Builder** with the same name. If required, edit a default property **Value**.

ASSIGNING GEOMETRIC SCOPE TO A MATERIAL

See Also	Working with Geometry

1. Add a material and geometry to the model.
2. On the **Material** page, from the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Selection from this list activates the button with the same name in the **Graphics** window.
3. Select **Manual** or **All Domains**, **All Boundaries**, **All Edges**, or **All Points** from the **Selection** list.

ADDING PREDEFINED MATERIAL PROPERTIES

1. Add a material to the model.
2. In the **Model Builder**, click the material node to open the **Material** page.

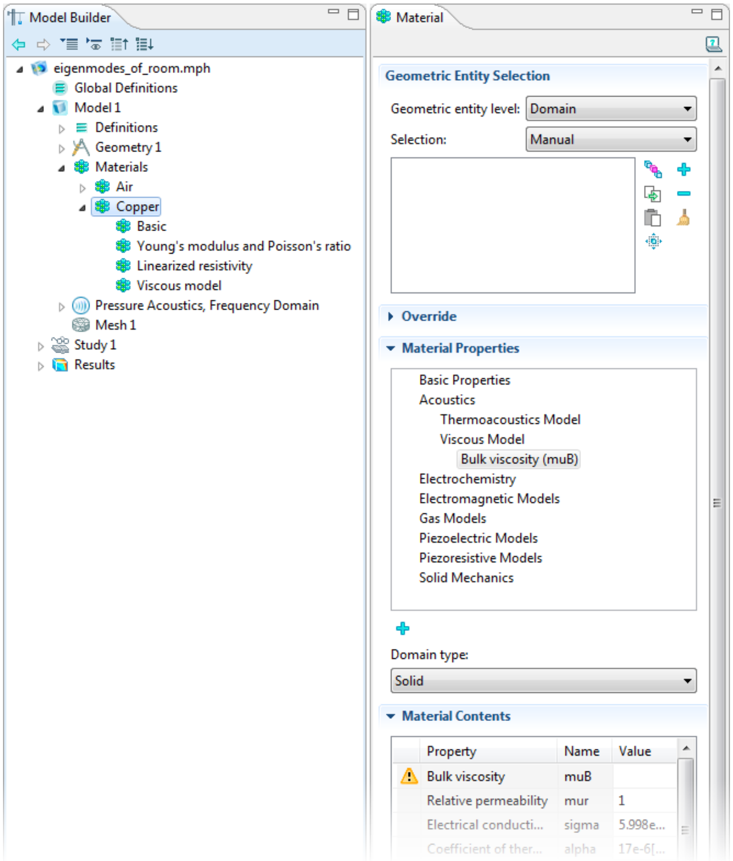
Important	Review the default properties listed in the **Material Contents** table before adding new material properties.

3. Browse the **Material Properties** section to locate the property to add to the model. For example, under **Acoustics>Viscous Model** select **Bulk viscosity (muB)** and right-click to **Add to Material** or click the **Add** button (+).

   See Figure 6-7. In this example, a **Viscous model** node is added to the **Model Builder**, and its associated properties are added to the **Material Contents** table. Any properties requiring **Values** are highlighted by warning signs (⚠) or stop signs (🛑) in the left-hand column.
4. Scroll through the **Material Contents** table to locate properties with a warning sign (⚠) in the column that indicates that it is missing data. Enter a **Value** into the

table next to the applicable property. In the example, the **Bulk viscosity** property requires data. The left column provides visual cues about the status of each property:

- A stop sign (⛔) indicates that an entry in the **Value** column is required. It means that the material property is required by a physics interface but undefined.
- A warning sign (⚠) indicates that the material parameter is undefined. An entry is only required if the material property is used in the model.
- A green check mark (✓) indicates that the property has a **Value** and is currently being used in the physics of the model.
- Properties with no indication in the left column are defined but not currently used by any physics interface in the model.



Geometric entity level: Domain

Selection: Manual

Domain type: Solid

	Property	Name	Value
⚠	Bulk viscosity	muB	
	Relative permeability	mur	1
	Electrical conducti...	sigma	5.998e...
	Coefficient of ther...	alpha	17e-6[...

Figure 6-7: Adding a material property to a model. Some properties may require data.

Tip

To delete a property group, right-click the property group node (in the **Model Builder**) and select **Delete** ( ). The **Basic** property group cannot be deleted.

Customizing the Appearance of a Material in the Graphics Window

Control or change the default appearance of a material in the Graphics window when working in the Materials or Physics interface sections.

2D

For 2D models, only a change of color is possible.

3D

For 3D models, the material appearance is with color and using the lighting model.

To display both color and texture for a customized 3D model material appearance, a Graphics Quality setting is required. From the main menu, select **Options>Preferences>Graphics** and then select **Quality** from the **Optimize for** list. Click **OK** and restart COMSOL Multiphysics for the changes to take effect.

See Also

Editing Preferences Settings

The default **Family** appearance is taken from the material selected, which has a suitable color (2D and 3D models), and texture (3D models) applied. Select another option from the **Family** list if required—**Air**, **Aluminum**, **Brick**, **Concrete**, **Copper**, **Gold**, **Iron**, **Lead**, **Magnesium**, **Plastic**, **Steel**, **Titanium**, or **Water**. Select **Custom** to make further adjustments. The default custom settings are inherited from the material selected last from the **Family** list.

Note

See About the Output Materials Properties, Table 6-7 for the defaults.

Specular Color, Diffuse Color, and Ambient Color

For each of these properties, click the **Color** button to assign a **Custom** specular color or select a standard color from the list—**Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White**, or **Yellow**.

The combination of **Specular color**, **Diffuse color**, and **Ambient color** give a 3D object its overall color.

- **Specular color** is the color of the light of a specular reflection (specular reflection is the type of reflection that is characteristic of light reflected from a shiny surface).
- **Diffuse color** represents the true color of an object; it is perceived as the color of the object itself rather than a reflection of the light. Gets darker as the surface points away from the light (shading). As with Ambient color, if there is a texture, this is multiplied by the colors in the texture, otherwise it is as if it has a white texture.
- **Ambient color** is the color of all the light that surrounds an object; it is the color seen when an object is in low light. This color is what the object reflects when illuminated by ambient light rather than direct light. Creates the effect of having light hit the object equally from all directions. As with Diffuse color, if there is a texture, this is multiplied by the colors in the texture, otherwise it is as if it has a white texture.

See Also

For examples of specular, diffuse, and ambient light, which are related to these definitions, see About the 3D View Light Sources and Attributes.

Noise

The **Noise** check box is selected by default, with the default **Normal vector noise scale** and **Normal vector noise frequency** taken from the material. Enter other values as required, or click to clear the **Noise** check box.

- Noise is a texture which disturbs the normals when calculating lighting on the surface. This causes the surface to look rough and textured.

- Normal vector noise scale is the power of the noise texture, a high value creates a stronger texture of the surface. A value from 0-1.0 is suitable.
- Normal vector noise frequency is the size of the noise disturbances, a small value creates smaller features on the texture. A value between 0-10 is suitable.

Diffuse and Ambient Color Opacity

The default **Diffuse and ambient color opacity** is 1. Enter a different number as required.

Lighting Model

The default **Lighting model**—**Blinn-Phong** or **Cook-Torrance**—is based on the material. Select **Simple** instead as required.

The different lighting models provide a set of techniques used to calculate the reflection of light from surfaces to create the appropriate shading. For example, a specular highlight is the bright spot of light that appears on shiny objects when illuminated. Specular highlights are important in 3D computer graphics, as they provide a strong visual cue for the shape of an object and its location with respect to light sources in the scene.

If **Blinn-Phong** is selected, the default **Specular exponent** is 64. Enter another value as required. The Specular Exponent is the size of the specular highlight spot depends on the value of the patch and surface object's Specular Exponent property. Typical values for this property range from 1 to 500, with normal objects having values in the range 5 to 20. This model is particularly useful for representing shiny materials.

If **Cook-Torrance** is selected, the default **Reflectance at normal incidence** and **Surface roughness** are taken from the material. Enter other values as required. The Cook-Torrance lighting model accounts for wavelength and color shifting and is a general model for rough surfaces. It is targeted at metals and plastics, although can still be used to represent many other materials.

- Reflectance at normal incidence is the amount of incoming light (0-1) from the normal direction (of the surface) which is reflected.
- Surface roughness is a value which describes microreflectance on the surface. Higher values create a rougher look of the surface, with fewer highlights. A value from 0-1 is suitable.

Working on the Property Group Page

ADDING OUTPUT PROPERTIES TO PREDEFINED MATERIALS

Note	**Output Properties** is only accessible from the **Basic** material properties and with user defined property groups.

1 Add a material to the model.

2 In the **Model Builder**, click **Basic** to open the **Property Group** page. See Figure 6-4.

The **Property Group** tables are linked to other pages related to the selected material. Use the **Move Up** (⇧), **Move Down** (⇩), and **Delete** (✖) buttons to organize the tables as required.

3 Under **Output Properties and Model Inputs**, in the **Quantities** section:

a Browse the **Output Properties** list to locate the property to add.

See Also	About the Output Materials Properties

b Right-click the property to **Add** items to the **Output properties** table or click the **Add** button (✚).

c In the **Output properties** table, edit the **Expression** field directly in the table or by clicking the **Edit** button (✎).

Use the **Move Up** (⇧), **Move Down** (⇩), and **Delete** (✖) buttons to organize the table as required.

ADDING MODEL INPUTS TO PREDEFINED MATERIALS

1 Add a material to the model.

2 In the **Model Builder**, click the property group node (for example, **Basic**) to open the **Property Group** page.

The **Property Group** page tables display on other pages pertaining to the selected material.

3. Browse the **Model Inputs** list. Right-click the property to **Add** items to the **Model inputs** table. Or click the **Add** button (+).

See Also

Model Input Properties

ADDING LOCAL PARAMETERS TO PREDEFINED MATERIALS

1. Add a material to the model.
2. In the **Model Builder**, click the property group node (for example, **Basic**) to open the **Property Group** page.
3. Under **Local Parameters**, enter **Parameters** and **Expressions** as required.

   Use the **Move Up** (↑), **Move Down** (↓), and **Delete** (✖) buttons to organize the table as required.

See Also

Adding a User-Defined Property Group

Adding a User-Defined Property Group

1. Open or create a model file and add a material (predefined or user defined) to the model.
2. In the **Model Builder**, right-click any material node (for example, **Copper**) and select **User defined property group**.

   A **User defined property group** node is added to the **Model Builder**, and a **Property Group** page opens in the settings window.
3. Follow the instructions to define the property group.

See Also

Working on the Property Group Page

User-Defined Materials and Libraries

User-defined materials provide the flexibility needed to design your model and experiments using a combination of existing material properties and properties you define yourself. You can also create your own material database (library) to include materials you use often.

In this section:

- Adding User-Defined Materials to a Model
- Adding an External Material Library
- Creating Your Own User-Defined Libraries
- Restoring a Deleted User-Defined Library

Adding User-Defined Materials to a Model

1 Open or create a model file.

2 In the **Model Builder**, right-click **Materials** and select **Material**.

   A **Material** node is added to the **Model Builder**, and an empty **Material** page opens in the settings window.

3 Follow the instructions as required to define your material:

   a Working on the Material Page

   b Working on the Property Group Page

   c Adding a User-Defined Property Group

   d Adding a Function to the Material

RENAMING MODEL BUILDER NODES OR EDITING NODE PROPERTIES

If required, edit or add comments to the **Node Properties**.

1 In the **Model Builder**, right-click any node and select **Properties** ( ).

2 On the **Node Properties** page, enter the **Name**, **Author**, **Version**, or **Comments** about the node. The **Tag** and **Date created** fields cannot be changed.

   The node’s **Name** can be edited by right-clicking directly on the node and selecting **Rename** ( ).

Adding an External Material Library

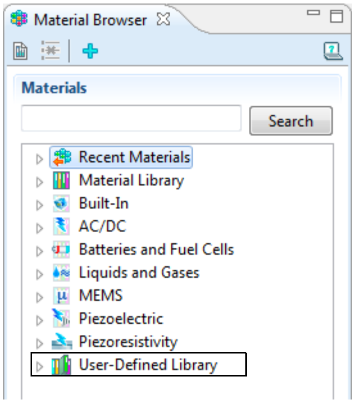
1 In the **Material Browser** window's toolbar, click the **Add Material Library** ( ) button. You can also right-click any database under **Materials** and select **Add Material Library** ( ).

The **Choose Material Library** window opens. From the **Save as type list**, typically choose **XML File (.xml)**, to find material libraries stored as XML files, which is the standard format.

2 Navigate to a material library file and click **Save**. For example, MatWeb provides a service where you can export technical datasheets from MatWeb's collection in the format for a COMSOL material library. For more information about this service, visit www.matweb.com.

Creating Your Own User-Defined Libraries

When you first open the **Material Browser**, an empty **User-Defined Library** is available for you to start creating your own library of materials.



1 In the **Material Browser**, click the **Add Material Library** button ( ) on the toolbar (or right-click **User Defined Library** and select the same option).

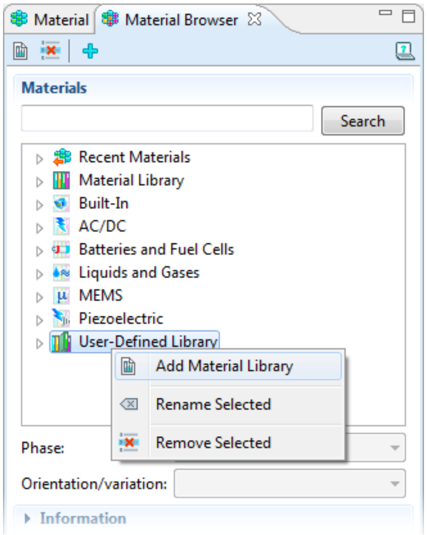


Figure 6-8: Creating and adding multiple user-defined libraries to the Material Browser.

2 In the **Choose Material Library** dialog box, navigate to the folder on the computer where the empty **User Defined Library** database is located. The location of the file varies based on your installation. For example, if the installation is on your hard drive, the file path might be similar to `C:\Users\`*`Your_Name`*`\.comsol\v43\materials`. You can also search for the file name `User_Defined_Library.mph`.

3 Right-click the `User_Defined_Library.mph` and select **Copy**. Rename the copied file.

4 Click **Save**. The empty database, with a new name, is added to the Material Browser.

RENAMING THE DEFAULT USER-DEFINED LIBRARY

In the **Material Browser**, right-click the **User-Defined Library** node and in the **Enter New Name** window, define a **Name** meaningful to the contents of the library. This does not change the library file name, only the name that displays on the screen.

CREATING AND RENAMING ADDITIONAL USER-DEFINED LIBRARIES

Navigate to the folder where the empty database called `User_Defined_Library.mph` is located. Copy the file at its root location and give it a new name, for example, `My_Metals.mph`. Then follow the instructions above to add the library to the **Material Browser**. See Figure 6-8.

ADDING A PREDEFINED MATERIAL TO THE USER-DEFINED LIBRARY

1 Open the **Material Browser** and add any predefined material to a model in the **Model Builder**.

2 Right-click the material node and select **Add Material to Library**.

3 Right-click to **Rename Selected** material that has been added to your library. See Figure 6-9.

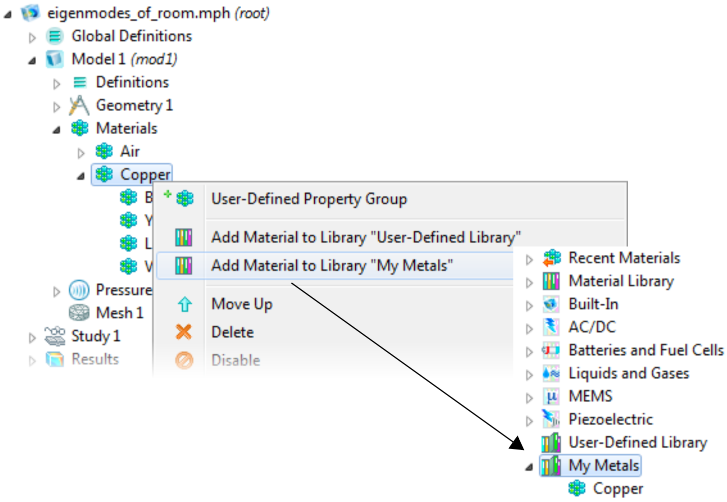


Figure 6-9: Adding a predefined material to a new user-defined library.

REMOVING A MATERIAL FROM A USER-DEFINED LIBRARY

Open the **Material Browser**. Locate the material to remove. Right-click the material and select **Remove Selected** (✖).

Restoring a Deleted User-Defined Library

If the **User-Defined Library** node is deleted in error from the **Material Browser**, you can restore it by following the steps in Creating Your Own User-Defined Libraries.

Material Properties Reference

The material properties for the predefined materials are accessible from most physics interfaces. Using this information, either create a material property group or define a completely new material.

On the **Basic>Property Group** page, add **Output properties** from the **Quantities** subsection. For all **Property Group** pages, **Model inputs** can be added.

In this section:

- About the Output Materials Properties
- Model Input Properties
- Predefined Built-In Materials for all COMSOL Modules
- Material Property Groups Descriptions

About the Output Materials Properties

Note

Some of these properties are only applicable to physics interfaces available with the add-on modules.

BASIC MATERIAL PROPERTIES

These material properties can be added to models from two pages— **Material** and **Property Group**. In both cases the properties belong to the **Basic** property group.

- When this information is accessed from the **Basic>Property Group** page, it is listed under **Quantities>Output Properties** and **Variable** is listed in the table.
- When this information is accessed from the **Material** page, it is listed under **Material Properties>Basic Properties** and **Name** is listed in the tables.

TABLE 6-1: BASIC MATERIAL PROPERTIES

PROPERTY	NAME/VARIABLE	SI UNIT
Absorption Coefficient	kappaR	1/m
Characteristic Acoustic Impedance	Z	Pa·s/m
Coefficient of Thermal Expansion	alpha	1/K

TABLE 6-1: BASIC MATERIAL PROPERTIES

PROPERTY	NAME/VARIABLE	SI UNIT
Compressibility of Fluid	chif	1/Pa
Density	rho	kg/m^3
Diffusion Coefficient	D	m^2/s
Dynamic Viscosity	mu	Pa·s
Electrical Conductivity	sigma	S/m
Electron Mobility	mue	m^2/(Vs)
Heat Capacity at Constant Pressure	Cp	J/(kg·K)
Isotropic Structural Loss Factor	eta s	1
Mass Flux	Mf	kg/(m^2·s)
Mean Molar Mass	Mn	kg/mol
Permeability	kappa	m^2
Poisson's Ratio	nu	1
Porosity	epsilon	1
Ratio of Specific Heats	gamma	1
Relative Permeability	mur	1
Relative Permittivity	epsilonr	1
Resistivity	res	Ωm
Scattering Coefficient	sigmaS	1/m
Speed of Sound	cp	m/s
Storage	S	1/Pa
Surface Emissivity	epsilon rad	1
Thermal Conductivity	k	W/(m·K)
Young's Modulus	E	Pa

Caution

The coefficient of thermal expansion (CTE) and the resistivity temperature coefficient have the SI unit 1/K. COMSOL Multiphysics translates this into the Fahrenheit temperature unit using an offset. This means that you do not get the expected results.

Use caution when a model uses the coefficient of thermal expansion or the resistivity temperature coefficient and the unit system's temperature is not Kelvin.

GAS MODELS

This material property group (including all associated properties) can be added to models from the **Material** page.

TABLE 6-2: GAS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
IDEAL GAS		
Heat capacity at constant pressure	Cp	J/(kg·K)
Ratio of specific heats	gamma	1
Mean molar mass	Mn	kg/mol
Specific gas constant	Rs	J/(kg·K)

SOLID MECHANICS

These material property groups (including all associated properties) can be added to models from the **Material** page.

TABLE 6-3: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
LINEAR ELASTIC MATERIAL		
YOUNG'S MODULUS AND POISSON'S RATIO		
Young's modulus	E	Pa
Poisson's ratio	nu	1
LAMÉ CONSTANTS		
Lamé constant λ	lambLame	Pa
Lamé constant μ	muLame	Pa
BULK MODULUS AND SHEAR MODULUS		
Bulk modulus	K	Pa
Shear modulus	G	Pa
PRESSURE-WAVE AND SHEAR-WAVE SPEEDS		
Pressure-wave speed	cl	m/s
Shear-wave speed	ct	m/s
ORTHOTROPIC		
Young's modulus	Evector	Pa
Poisson's ratio	nuvector	1
Shear modulus	Gvector	Pa
Loss factor for orthotropic Young's modulus	eta_Evector	Pa
Loss factor for orthotropic shear modulus	eta_Gvector	Pa

TABLE 6-3: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ANISOTROPIC		
Elasticity matrix	D	Pa
Loss factor for elasticity matrix D	eta_D	Pa
ELASTOPLASTIC MATERIAL MODEL		
Yield stress level	sigmags	Pa
Isotropic tangent modulus	Et	Pa
Kinematic tangent modulus	Ek	Pa
LINEAR VISCOELASTIC MATERIAL		
Long-term shear modulus	Gv	Pa
Bulk modulus	K	Pa
HYPERELASTIC MATERIAL		
NEO-HOOKEAN		
Lamé constant λ	lambda	Pa
Lamé constant μ	mu	Pa
MOONEY-RIVLIN		
Model parameters	C10	Pa
Model parameters	C01	Pa
MURNAGHAN		
Murnaghan third-order elastic moduli	l	Pa
Murnaghan third-order elastic moduli	m	Pa
Murnaghan third-order elastic moduli	n	Pa
Lamé constant λ	lambLame	Pa
Lamé constant μ	muLame	Pa
POROELASTIC MATERIAL		
Biot-Willis coefficient	alphaB	1
Porosity	epsilon	1
Permeability	kappa	m^2

See Also

The *Geomechanics Module User's Guide* has a list of **Geomechanics Material Model** properties.

Tip

The **Hyperelastic Material** and **Elastoplastic Material** features are available with the Nonlinear Structural Materials Module.

ELECTROMAGNETIC MODELS

These material property groups (including all associated properties) can be added to models from the **Material** page.

TABLE 6-4: ELECTROMAGNETIC MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
LINEARIZED RESISTIVITY		
Reference resistivity	rho0	Ωm
Resistivity temperature coefficient	alpha	1/K
Reference temperature	Tref	K
LOSS TANGENT		
Loss tangent	delta	-
Relative permittivity (real part)	epsilonPrim	1
DIELECTRIC LOSSES		
Relative permittivity (imaginary part)	epsilonBis	1
Relative permittivity (real part)	epsilonPrim	1
Dielectric loss factor	eta_epsilon	
REFRACTIVE INDEX		
Refractive index	n	1
Refractive index, imaginary part	ki	-
MAGNETIC LOSSES		
Relative permeability (real part)	murPrim	-
Relative permeability (imaginary part)	murBis	-
HB CURVE		
Local Parameters	normB	-
Magnetic field norm	normH	A/m
BH CURVE		
Local Parameters	normH	-
Magnetic flux density norm	normB	T

PIEZOELECTRIC MODELS

These material property groups (including all associated properties) can be added to models from the **Material** page.

TABLE 6-5: PIEZOELECTRIC MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
STRESS-CHARGE FORM		
Elasticity matrix cE	cE	Pa
Coupling matrix	eES	C/m^2
Relative permittivity	epsilonrS	1
Loss factor for elasticity matrix	cE	1
Loss factor for coupling matrix	e	1
Loss factor for electrical permittivity	εS	1
STRAIN-CHARGE FORM		
Compliance matrix sE	sE	1/Pa
Coupling matrix	dET	C/N
Relative permittivity	epsilonrT	1
Loss factor for compliance matrix	sE	1
Loss factor for coupling matrix	d	1
Loss factor for electrical permittivity	εT	1

Model Input Properties

The following model input properties can be added to models from the **Property Group** page (vector fields have three components).

TABLE 6-6: MODEL INPUT PROPERTIES

PROPERTY	NAME/VARIABLE
Temperature	T
Absolute Pressure	pA
Electric Field	{E1, E2, E3}
Magnetic Field	{H1, H2, H3}
Magnetic Flux Density	{B1, B2, B3}
Velocity Field	{u1, u2, u3}

TABLE 6-6: MODEL INPUT PROPERTIES

PROPERTY	NAME/VARIABLE
Frequency	freq
Stress Tensor	{F1, F2, F3}
Strain Reference Temperature	Tempref
Concentration	c

Predefined Built-In Materials for all COMSOL Modules

The **Built-In** database contains material data for some common materials of different types. This database is available for all users and has data for these materials.

Note

The material properties for the materials in the Built-in material database are typical for the type of material but do not represent exact values for a specific grade or quality of the material.

3D

A default appearance for 3D models is applied to each material as indicated.

See Also

See Working on the Material Page for more information about customizing the material's appearance in the Graphics window.

TABLE 6-7: BUILT-IN MATERIALS AND 3D MODEL DEFAULT APPEARANCE SETTINGS

MATERIAL	DEFAULT FAMILY	DEFAULT LIGHTING MODEL	CUSTOM DEFAULT SETTINGS					
			NORMAL VECTOR NOISE SCALE	NORMAL VECTOR NOISE FREQUENCY	DIFFUSE AND AMBIENT COLOR OPACITY	SPECULAR EXPONENT	REFLECTANCE AT NORMAL INCIDENCE	SURFACE ROUGHNESS
Air	Air	Simple	0.08	3	0.1	-	-	-
Acrylic plastic	Custom	Blinn-Phong	0	1	1	1000	-	-
Alumina	Aluminum	Cook-Torrance	0	1	1	-	0.9	0.1
Aluminum 3003-H18								
Aluminum 6063-T83								
Aluminum								
American red oak	Custom	Cook-Torrance	0.5	0.3	1	-	0.3	0.6
Beryllium copper UNS C17200	Custom	Cook-Torrance	0	1	1	-	0.9	0.1
Brick	Brick	Cook-Torrance	0.4	10	1	-	0.3	0.6
Cast iron	Custom	Cook-Torrance	0.14	2.1	1	-	0.1	0.1
Concrete	Concrete	Cook-Torrance	0.45	10	1	-	0.3	0.6
Copper	Copper	Cook-Torrance	0	1	1	-	0.9	0.17
FR4 (Circuit Board)	Custom	Cook-Torrance	0.05	2.1	1	-	0.3	0.6
Glass (quartz)	Custom	Cook-Torrance	0	1	0.1	-	0.99	0.02
Granite	Custom	Cook-Torrance	0.6	1.7	1	-	0.3	0.6

TABLE 6-7: BUILT-IN MATERIALS AND 3D MODEL DEFAULT APPEARANCE SETTINGS

MATERIAL	DEFAULT FAMILY	DEFAULT LIGHTING MODEL	CUSTOM DEFAULT SETTINGS					
			NORMAL VECTOR NOISE SCALE	NORMAL VECTOR NOISE FREQUENCY	DIFFUSE AND AMBIENT COLOR OPACITY	SPECULAR EXPONENT	REFLECTANCE AT NORMAL INCIDENCE	SURFACE ROUGHNESS
High-strength alloy steel	Custom	Cook-Torrance	0	1	1	-	0.99	0.12
Iron	Iron	Cook-Torrance	0	1	1	-	0.99	0.14
Magnesium AZ31B	Magnesium	Cook-Torrance	0	1	1	-	0.3	0.1
Mica	Custom	Blinn-Phong	0	1	0.3	20	-	-
Molybdenum	Custom	Cook-Torrance	0	1	1	-	0.3	0.1
Nimonic alloy 90	Custom	Cook-Torrance	0	1	1	-	0.3	0.1
Nylon	Custom	Blinn-Phong	0	1	1	500	-	-
Lead Zirconate Titanate (PZT-5H)	Lead	Cook-Torrance	0	1	1	-	0.3	0.1
Silica glass	Custom	Cook-Torrance	0	1	0.1	-	0.99	0.02
Silicon	Custom	Cook-Torrance	0	1	1	-	0.7	0.5
Polysilicon	Custom	Cook-Torrance	0	1	1	-	0.7	0.5
Solder, 60Sn-40Pb	Custom	Cook-Torrance	0	1	1	-	0.9	0.1
Steel AISI 4340	Steel	Cook-Torrance	0	1	1	-	0.9	0.1
Structural steel	Custom	Cook-Torrance	0	1	1	-	0.9	0.3

TABLE 6-7: BUILT-IN MATERIALS AND 3D MODEL DEFAULT APPEARANCE SETTINGS

MATERIAL	DEFAULT FAMILY	DEFAULT LIGHTING MODEL	CUSTOM DEFAULT SETTINGS					
			NORMAL VECTOR NOISE SCALE	NORMAL VECTOR NOISE FREQUENCY	DIFFUSE AND AMBIENT COLOR OPACITY	SPECULAR EXPONENT	REFLECTANCE AT NORMAL INCIDENCE	SURFACE ROUGHNESS
Titanium beta-21S	Titanium	Cook-Torrance	0	1	1	-	0.9	0.1
Tungsten	Custom	Cook-Torrance	0	1	1	-	0.9	0.15
Water, liquid	Water	Cook-Torrance	0.2	0.2	0.2	-	0.7	0.05

Material Property Groups Descriptions

The following material property groups are available for the entire COMSOL Multiphysics product line (some groups are only applicable to physics interfaces in some of the add-on modules):

Basic: Contains over 25 basic material properties for use with all materials.

See Also

See About the Output Materials Properties for a complete list.

Gas Models

Ideal Gas: In gas models, this property group adds the heat capacity at constant pressure, ratio of specific heats, mean molar mass, and specific gas constant equations.

Electromagnetic Models

Linearized Resistivity: To define the electric resistivity (and conductivity) as a linear function of temperature.

Loss Tangent: Allows the specification of a loss tangent for dielectric losses or takes it from the material. Assumes zero conductivity.

Dielectric Losses: Allows the specification of the real and imaginary parts of the relative permittivity or takes it from the material.

Refractive Index: Allows the specification of the real and imaginary parts of the refractive index or takes it from the material. This assumes a relative permeability of unity and zero conductivity.

Magnetic Losses: Allows the specification of the real and imaginary parts of the relative permeability or takes it from the material.

HB Curve: Allows the use of a curve that relates magnetic flux density B and the magnetic field H.

BH Curve: Allows the use of a curve that relates magnetic flux density B and the magnetic field H.

Solid Mechanics Models

Young's Modulus and Poisson's Ratio: In linear elastic materials, Young’s modulus defines the modulus of elasticity and is the spring stiffness in Hooke’s law. Poisson’s ratio defines the normal strain in the perpendicular direction, generated from a normal strain in the other direction.

Lamé Constants: In linear elastic materials, the first and second Lamé constants λ and μ are linear elastic material property constants.

Bulk Modulus and Shear Modulus: A linear elastic material property where the bulk modulus is a measure of the solid’s resistance to volume changes. The shear modulus is a measure of the solid’s resistance to shear deformations.

Pressure-Wave and Shear-Wave Speeds: A linear elastic material property where the pressure-wave speed (longitudal wave speed) and the shear-wave speed (transverse wave speed) are specified.

0rthotropic: A linear elastic material property that has different properties in orthogonal directions; its stiffness depends on those properties.

Anisotropic: A linear elastic material property that has different material properties in different directions; the stiffness comes from the symmetric elasticity matrix.

Elastoplastic Material Model: An elastoplastic property group adds these equations: yield stress level, kinematic tangent modulus, and isotropic tangent modulus.

Geomechanics Material Model: See the *Geomechanics Module User's Guide* for details.

Linear Viscoelastic Material: Viscoelastic materials are those that exhibit both elastic and viscous behavior when they deform. This property group adds the long-term shear modulus and bulk modulus equations.

Hyperelastic Material: In hyperelastic materials the stresses are computed from a strain energy density function. They are often used to model rubber, but also used in acoustoelasticity.

Neo-Hookean: A hyperelastic material property with two model parameters. The model is based on modified strain invariants.

Mooney-Rivlin: A hyperelastic material property with three model parameters. The model is based on modified strain invariants.

Murnaghan: A hyperelastic material property with five model parameters. The model is based on modified strain invariants and is typically used in acoustoelasticity.

Poroelastic Material: Has three model parameters, Biot-Willis coefficient, porosity, and permeability for porous material.

Piezoelectric Models

Stress-Charge Form: Used to express the constitutive relations using the elasticity matrix, coupling matrix, and relative permittivity equations.

Strain-Charge Form: Used to express the constitutive relations using the compliance matrix, coupling matrix, and relative permittivity equations.

Electrochemistry

Electrode Potential: See the *Batteries & Fuel Cells Module User's Guide* for details.

Electrolyte Conductivity: See the *Batteries & Fuel Cells Module User's Guide* for details.

Linearized Resistivity: See the *Batteries & Fuel Cells Module User's Guide* for details.

Species Properties: See the *Batteries & Fuel Cells Module User's Guide* for details.

Using Functions

In this section:

- Adding a Function to the Material
- Example of Defining an Analytic Function

Adding a Function to the Material

Material functions are either automatically added to the **Model Builder** sequence (usually with materials from the **Model Library**) or functions can be added based on individual requirements:

1. Add a material to the model.
2. In the **Model Builder**, right-click a property group node, for example, **Basic**.
3. Select one of the following from the **Functions** list:
   - Select **Analytic** to extend properties with the ability to bind parameters during function calls.
   - Select **Interpolation** to interpolate from data that can be both structured (defined on a grid) or unstructured (defined on a generic point cloud).
   - Select **Piecewise** if there is one material property that has different definitions on different intervals.The intervals must not overlap, and there cannot be any holes between intervals.

See Also

- Example of Defining an Analytic Function

In the *COMSOL Multiphysics Reference Guide*:

- Analytic Functions
- Interpolation Functions
- Piecewise Functions

Tip

Once a function is created, it can be used for any property in the same property group.

Example of Defining an Analytic Function

Assume that you want to define Young's modulus for a material as a function of pressure and temperature. You can name the function `young(p,T)` and use the expression `2e10+1e6*p-1e8*T` to define the function.

1 Open the **Material Browser**.

2 **Search** or browse for a material to add to the model.

  a In the **Material Library**, open the **Cast Irons** folder.

  b From the list, right-click **A297 HI (UNS J94003)** to **Add Material to Model**.

  The material is added to the **Model Builder** with a **Basic** property group containing Piecewise functions. Observe that Young's Modulus is not automatically added.

3 Add a **Young's Modulus** property to the material.

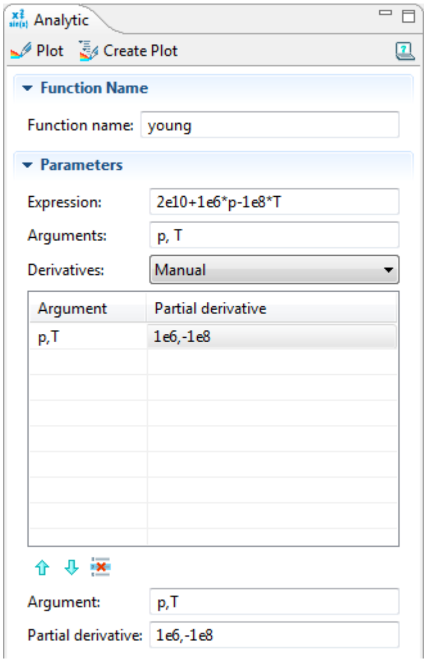
  a In the **Model Builder**, click the material node (in this example, **A297 HI (UNS J94003)**).

  b On the **Material** page, click to expand the **Material Properties** section. Under **Basic Properties**, right-click **Young's Modulus** and **Add to Material**.

  A **Young's modulus** property is added to the undefined properties table in the **Basic** property group.

4 In the **Model Builder**, under the **A297 HI (UNS J94003)** node, right-click **Basic** and select **Functions>Analytic**.

5 On the **Analytic** settings window, enter `young` in **Function Name**.



Function Name

Function name: young

Parameters

Expression: 2e10+1e6*p-1e8*T

Arguments: p, T

Derivatives: Manual

Argument	Partial derivative
p,T	1e6,-1e8

Argument: p,T

Partial derivative: 1e6,-1e8

6 Under **Parameters**:

a In the **Expression** field enter `2e10+1e6*p-1e8*T`.

b In the **Arguments** field enter `p,T`.

c Select **Manual** from the **Derivatives** list.

d In the table, in the **Argument** column, enter `p,T`.

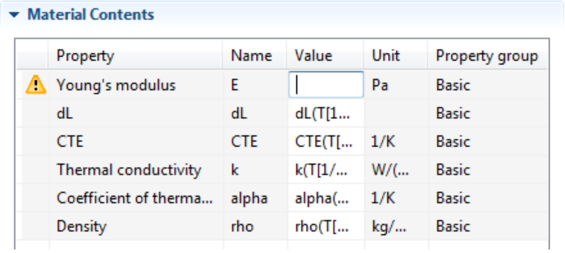
e In the **Partial derivative** column, enter `1e6,-1e8`.

The function `young` can now be used to define the Young's modulus in your material.

COMSOL Multiphysics must know the derivative of a function if a variable is being used that depends on the solution in a function argument. This is why in this example a **Manual** derivative is used. The `d` operator can be used to compute derivatives of a function.

If **Auto** is used while defining the **Derivatives** parameter, COMSOL Multiphysics uses the `d` operator on the expression to get the derivatives. In this case, under **Advanced**, select the **May produce complex output for real arguments** check box if the defined function works similarly to `sqrt`, that is, if it sometimes returns complex values for real-valued input.

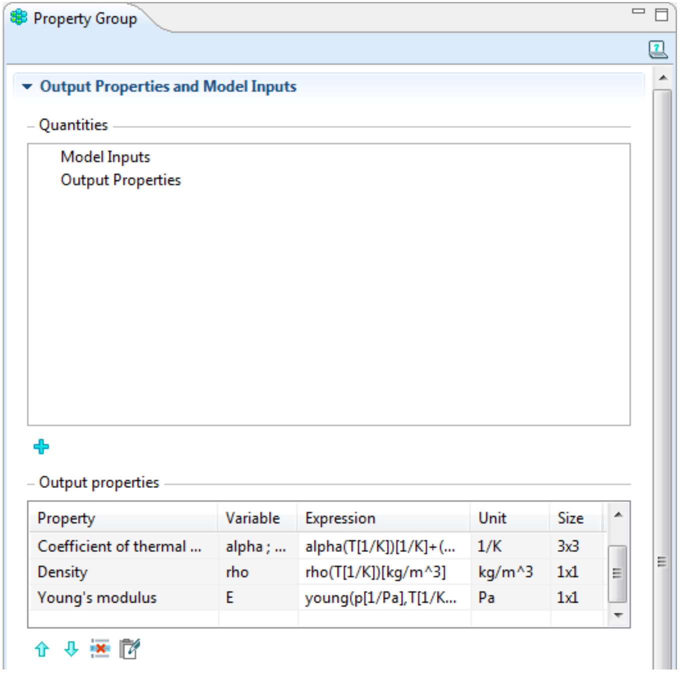
7 Click the material node (in this example, **A297 HI (UNS J94003)**). On the **Material** settings window, under **Material Contents**, enter `young(p[1/Pa],T[1/K])[Pa]` in the **Value** column (in the Young's modulus row).



Material Contents

	Property	Name	Value	Unit	Property group
⚠	Young's modulus	E		Pa	Basic
	dL	dL	dL(T[1...		Basic
	CTE	CTE	CTE(T[...	1/K	Basic
	Thermal conductivity	k	k(T[1/...	W/(...	Basic
	Coefficient of therma...	alpha	alpha(...	1/K	Basic
	Density	rho	rho(T[...	kg/...	Basic

Click the **Basic** node to observe that the **Young's modulus** analytic function is defined on the **Property Group** settings window under **Output properties**. See Figure 6-10.



Property Group

Output Properties and Model Inputs

Quantities

Model Inputs
Output Properties

Output properties

Property	Variable	Expression	Unit	Size
Coefficient of thermal ...	alpha ; ...	alpha(T[1/K])[1/K]+(...	1/K	3x3
Density	rho	rho(T[1/K])[kg/m^3]	kg/m^3	1x1
Young's modulus	E	young(p[1/Pa],T[1/K...	Pa	1x1

Figure 6-10: A Young's Modulus property is defined for an Analytic function.

7

Building a COMSOL Model

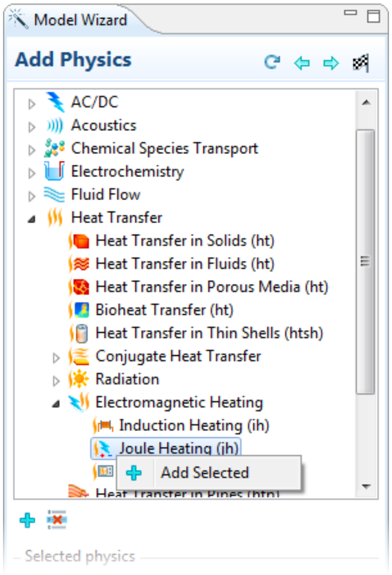
This chapter explains a range of methods and topics used when building models in COMSOL Multiphysics: From working with the Model Wizard and fundamental concepts for the Model Builder to the use of units.

In this chapter:

- The Model Wizard and Model Builder
- Model Administration
- Showing and Expanding Advanced Physics Sections
- The Physics Feature Nodes
- Specifying Model Equation Settings
- Periodic Boundary Conditions
- Calculating Accurate Fluxes
- Using Load Cases
- Using Units

The Model Wizard and Model Builder

When COMSOL is opened to create a model, the **Model Wizard** opens by default to guide you through selecting the space dimension, physics interfaces, and study type.



The physics interfaces contain predefined physics descriptions and equations for a variety of engineering and scientific disciplines, from acoustics to structural mechanics. You can also define your own equations using the options under **Mathematics**. The last step of the **Model Wizard** is to select a **Study Type**—**Stationary**, **Time Dependent**, **Eigenfrequency**, or some other study type, depending on the analysis and the physics.

At any time you can create a new model or add physics. Right-click the **Root** (top) node and select **Add Model** to open the **Model Wizard**, or right-click a **Model** node and select **Add Physics**.

After clicking the **Finish** button (🏁) in the **Model Wizard**, the **Model Builder** window displays a model tree with a set of default nodes—**Definitions**, **Geometry**, **Materials**, **Mesh**, and nodes based on the physics interfaces selected (see Figure 7-1). The **Model** nodes and branches form the sequence of operations that define the model.

In this section:

- The Model Nodes in the Model Builder

- Adding Nodes to the Model Builder
- Deleting, Disabling, and Enabling Nodes

The Model Nodes in the Model Builder

The **Model** nodes define the scope for all parts of a model, and all these nodes together form the input for a model. The position of the nodes in the vertical direction represent the order of execution of operations. Model files can have several **Model** nodes. For example, if you are setting up a system model using a 2D simplification for certain components and full 3D description for other components, these can both be added to the **Model Builder** to represent the different model requirements.

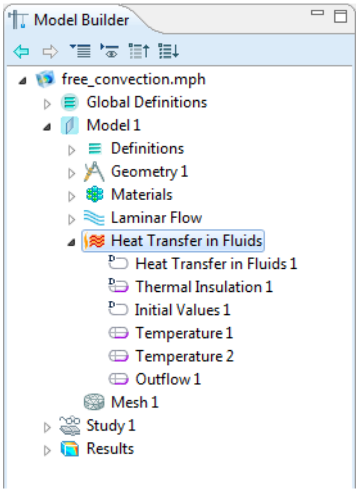


Figure 7-1: An example of the Model Builder default nodes.

These default nodes are normally added under a **Model** node:

- **Definitions**: Contains variables, selections, views, pairs, functions, probes, model couplings, and coordinate systems, which are defined locally for the model. See for information about using these local **Definitions** (≡) and **Global Definitions** (≡). Use **Global Definitions** to define **Parameters**, **Variables**, **Functions**, and **Groups** with a global scope—that is, not specific to one **Model** node.
- **Geometry** ( ): Contains the sequence of geometric objects and operations (or imported CAD data) that defines the model geometry. You can also draw your own geometry in the **Graphics** window.

- **Materials** ( ): Contains the materials and material properties used as sources for material data in the model. See Materials for detailed information.
- **Physics** ( ): Any added physics or mathematical interface displays as a node under **Model** (**Heat Transfer** in Figure 7-1 for example).
- **Mesh** ( ): Contains the sequences of mesh operations that defines the computational meshes for the model. When there is only one mesh in the model, its **Mesh** node appears directly under the **Model** node.

The **Model** settings windows contain the following sections:

MODEL IDENTIFIER

The model identifier is a string used to identify variables in the model. The default identifier for models is `mod1`, `mod2`, and so on, but you can change it in the **Identifier** field.

MODEL SETTINGS

This section contains general settings for the model, which you normally do not need to change:

Unit System

The default setting in the **Unit system** list, **Same as global system**, is to use the global unit system, which you specify in the root node's settings window. If you want to use another unit system in a model, select it from this list.

Spatial Coordinates

The default names for the spatial coordinates are x, y, and z for 3D as well as planar 1D and 2D geometries. For axisymmetric geometries, the default names for the spatial coordinates are r, φ (**phi**), and z. If you use the geometry to represent something else than space, or if you for some other reason want to use other names for the spatial coordinates, you can change the names in the fields for the **First**, **Second**, and **Third** coordinate under **Spatial coordinates**. The field labels include the default spatial coordinate names in parentheses.

Note

You cannot use the variable for the time, t, as a spatial coordinate name.

Geometry Shape Order

The setting in the **Geometry shape order** list determines the order of the curved mesh elements that determine the geometry shape. The default setting is **Automatic**, but it is also possible to select an order such as **Linear**, **Quadratic**, or **Cubic**. The default setting allows for automatic reduction of the order in some cases.

See Also

- Editing Node Properties, Names, and Identifiers
- Setting the Unit System for Individual Models
- Curved Mesh Elements in the *COMSOL Multiphysics Reference Guide*

Adding Nodes to the Model Builder

In the **Model Builder**, right-click any node to open its context menu. Once a node is highlighted, right-click anywhere in the **Model Builder** to open the context menu, which lists all the features available to that particular node on that branch of the tree.

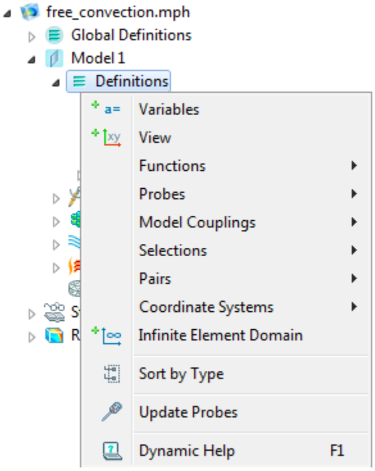


Figure 7-2: A context menu opens when you right-click any node in the Model Builder.

ABOUT THE SEQUENCE OF OPERATIONS

COMSOL Multiphysics evaluates most of the branch nodes in the **Model Builder** from the top-down as a *sequence of operations*. This makes it possible to, for example, to parameterize a model and rerun the simulation. COMSOL then re-evaluates each sequence, updating the geometry, mesh, and solution.

Some nodes under a physics branch can override other nodes higher up in the sequence. How COMSOL Multiphysics treats those nodes depends if they are contributing or exclusive nodes.

See Also	Physics Exclusive and Contributing Feature Node Types

Deleting, Disabling, and Enabling Nodes

You can change the contents, and actions, of the sequences by disabling and deleting nodes. A disabled node does not take part in the evaluation of a sequence. See Figure 7-2. Some nodes, such as container nodes and default nodes in the physics interfaces (see Physics Default Feature Nodes), cannot be disabled and deleted. When this is the case, the context menu does not have these options available.

- To disable selected nodes, right-click it and select **Disable** (⊘) or press F3. The nodes are grayed out in the model tree to indicate that they are disabled.
- To enable disabled nodes, right-click and select **Enable** (●) or press F4.

To delete selected nodes, right-click the nodes and select **Delete** (✕) or press Del (the Delete key). Confirm the deletion of nodes for it to take effect.

Tip	Instead of deleting and enabling variables and physics features to simulate various analysis cases (using different boundary conditions or sources, for example), use the selection of variable and physics features in the study steps’ **Physics and Variables Selection** sections, or use *load cases* for solving cases with varying loads or constraints in a model.

See Also	• Using Load Cases • Load Group • Constraint Group

Model Administration

A variety of tasks can be done to organize and simplify the model building process. In this section:

- About the COMSOL Model File Formats
- Saving COMSOL Multiphysics Model Files
- Reverting to the Last Saved Model File
- Saving a Model Thumbnail Image
- Resetting the Model History
- Saving Recovery Files
- Working with Remote Servers
- Editing Node Properties, Names, and Identifiers
- Opening and Searching the Model Library
- Changing the Model Library Root Folder
- Updating the Model Library Using Model Library Update
- Adding a User Model Library
- Editing Preferences Settings
- Checking and Controlling Products and Licenses Used
- Borrowing Licenses
- Information About Memory Use
- Viewing Node Names, Identifiers, Types, and Tags

See Also

- Showing and Expanding Advanced Physics Sections
- Changing the COMSOL Desktop Layout
- Introduction to the COMSOL Windows

About the COMSOL Model File Formats

There are three COMSOL Multiphysics model file formats: MPH-files, Model Java-files, and Model M-files.

COMSOL MPH-FILES

The default standard file with the extension `.mph`. The file contains both binary and text data. The mesh and solution data are stored as binary data, while all other information is stored as plain text.

You can quickly save and load MPH-files. All the models in the COMSOL Multiphysics Model Library and the model libraries in the add-on modules are MPH-files.

The MPH-files in the COMSOL model libraries can have two formats:

- Full MPH-files, including all meshes and solutions. In the Model Library these models appear with the ● icon. If the MPH-file's size exceeds 25MB, a tooltip with the text "Large file" and the file size appears when you position the cursor at the model's node in the Model Library tree.
- Compact MPH-files with all settings for the model but without built meshes and solution data to save space on the DVD (a few MPH-files have no solutions for other reasons). You can open these models to study the settings and to mesh and re-solve the models. It is also possible to download the full versions—with meshes and solutions—of most of these models through Model Library Update (see Updating the Model Library Using Model Library Update). In the Model Library these models appear with the ● icon. If you position the cursor at a compact model in the Model Library window, a **No solutions stored** message appears. If a full MPH-file is available for download, the corresponding node's context menu includes a **Model Library Update** item.

MODEL JAVA-FILES

Editable script files that contain sequences of COMSOL Multiphysics commands as Java code (see the *COMSOL Java API Reference Guide* for more information about these commands). You can compile these Java files and run them as separate applications. Edit the files in a text editor to add additional commands.

MODEL M-FILES

Model M-files are editable script files, similar to the Model Java-files, for use with MATLAB. Model M-files contain a sequence of COMSOL Multiphysics commands as a MATLAB M-file. Run the Model M-files in MATLAB like any other M-file scripts.

Also edit the files in a text editor to include additional COMSOL Multiphysics commands or general MATLAB commands.

Note

Running files in the Model M-file format requires the COMSOL LiveLink™ for MATLAB®.

See Also

- Saving COMSOL Multiphysics Model Files
- Reverting to the Last Saved Model File
- Saving a Model Thumbnail Image
- Saving Recovery Files

Saving COMSOL Multiphysics Model Files

See Also

- About the COMSOL Model File Formats
- Reverting to the Last Saved Model File
- Saving a Model Thumbnail Image
- To learn how to build your own model library, see Changing the Model Library Root Folder.

SAVING AS A COMSOL MPH-FILE

If this is the first time saving a model or you want to update the file and keep the current name and format:

1. Click the **Save** button (💾), press Ctrl+S, or from the **File** menu, select **Save**.
2. Navigate to a the location where you want to save the model and enter a **File name**.
3. Click **Save**.

If the model has been saved before and you want to save it in a different format or create a copy:

1. From the **File** menu, select **Save As**.
2. In the **Save as type** list, select the file type that you want to use.
3. Navigate to a the location where you want to save the model and enter a **File name**.

4 Click **Save**.

SAVING AS A MODEL JAVA-FILE

1 From the **File** menu, select **Save As Model Java-File**.

2 Navigate to a the location where you want to save the model and enter a **File name**.

3 Click **Save**.

SAVING AS A MODEL M-FILE

1 From the **File** menu, select **Save As Model M-File**.

2 Navigate to a the location where you want to save the model and enter a **File name**.

3 Click **Save**.

Reverting to the Last Saved Model File

To open the last saved version of the file and reinitialize the GUI, from the main menu, select **File>Revert to Saved**. Select **Yes** to continue, and **No** to cancel.

Saving a Model Thumbnail Image

To illustrate the model you can save a model thumbnail image that displays in the **Root Node Properties** and when opening a model in the **Model Library**. To save the current COMSOL Multiphysics plot as a model thumbnail image, from the main menu, select **File>Save Model Thumbnail**. The model thumbnail is a copy of the current plot.

Resetting the Model History

The Model Java-files and Model M-files contain the entire history of the model, including settings that are no longer part of the model. To reset the model history so that the files only include the settings that are part of the current model, from the main menu, select **File>Reset History**.

Saving Recovery Files

COMSOL Multiphysics can store recovery files each time you start a solver. This is a preference setting that is initially active by default. To deactivate or activate the preference setting for automatic save of recovery files, open the **Preferences** dialog box from the **Options** menu and select **General**. Then, under **Recovery**, clear or select the

Save recovery file check box. This change takes effect the next time that you start COMSOL Multiphysics.

The update of the recover file occurs at the following events:

- After completing the solution for each time step specified as the output times in the **Times** field for a time dependent simulation.
- After completing each parameter step in a parametric simulation.
- After completing each successful Newton iteration for a nonlinear stationary simulation.

The recovery files are COMSOL Multiphysics MPH-files that represent the state at the time that they were saved. They make it possible to recover from a solver error, which can be especially useful for long time dependent or parametric runs. To open a recovery file, choose **Open Recovery File** from the **File** menu. The **Open Recovery File** dialog box lists all recovery files found in chronological order. The files are listed with the date and time when they were saved. When a recovery file is selected, click **OK** to open it in the COMSOL Desktop. Click **Delete** to remove the selected recovery file. Click the **Delete All** button to delete all recovery files.

COMSOL Multiphysics keeps track of the computed time steps or parameter steps in the recovery file, so right-click the **Study** node and select **Continue** (⇛) to continue the computation from the point where it was stored in the recovery file. If you are solving a stationary non-parametric problem, the last converged Newton iteration is stored in the recovery file; selecting **Continue** then resumes the solving from this stored state.

See Also

Editing Preferences Settings

Working with Remote Servers

From the main **File** menu, you can connect and disconnect from a server, and import and export models.

CONNECTING AND DISCONNECTING A SERVER

Select **File>Client Server>Connect to Server** or **Disconnect from Server**.

EXPORTING MODELS TO A SERVER

When running a COMSOL Multiphysics server on a remote computer you can export a Model MPH-file to that server.

1 From the **File** menu, select **Client Server>Export Model to Server**.
2 Enter the **Server** and **Port** details.
3 Enter the **Username** and **Password** details. The **Remember password** check box is selected by default. Click to clear as required.
4 Enter a **Model** name.
5 Click **OK**.

IMPORTING MODELS FROM A SERVER

Import a Model MPH-file from a remote computer.

1 From the **File** menu, select **Client Server>Import Model from Server**.
2 Enter the **Server** and **Port** details.
3 Enter the **Username** and **Password** details. The **Remember password** check box is selected by default. Click to clear as required.
4 Enter the **Model** name to import.
5 Click **OK**.

Editing Node Properties, Names, and Identifiers

All nodes, except container nodes, have a common set of node properties, some of which can be changed and some system generated properties that cannot be edited. The root node has additional information that provides an overview of the complete model file.

See Also

- To learn about how some of these properties can be viewed, see Viewing Node Names, Identifiers, Types, and Tags.
- Variable Naming Convention and Scope

EDITING A MODEL IDENTIFIER FOR USE WITH VARIABLES

Use a *model identifier* to access variables throughout the model. The identifier is part of the full reference to variables (for example, when referring to variables in another model).

Note

To edit the model **Name**, see Editing Node Properties.

To edit a model identifier, in the **Model Builder**, click a **Model** node. The **Model** page opens in the settings window. Edit the default identifiers (**mod1**, **mod2**, and so on) as required in the **Identifier** field. See Figure 7-4 for an example.

RENAMING A NODE

To rename a node, in the **Model Builder**, click a node and select **Rename** (⌫) or press F2. Enter a **New name** and click **OK**. The **Name** is both updated in the **Model Builder** and in the **Node Properties** section.

EDITING NODE PROPERTIES

Opening the Properties Window

In the **Model Builder**, right-click a node and select **Properties** from the context menu (see Figure 7-2). The **Properties** window for that specific node replaces the settings window. The **Root** node includes only a settings window where you find a **Node Properties** section with properties that represent the entire model.

Node Properties

This section contains these fields:

- The **Name** field defaults to a system name for the node. Edit as required as a description for the node.
- The **Tag** is a unique system-defined tag for the node, which cannot be changed.
- The **Author** field contains the name of the author (creator) of the node. In addition to editing the author name manually, you can make a change for all nodes that you add later on by selecting **Options>Preferences** from the main menu and then select **User** and edit the username.

- The **Date created** field shows the node creation date and time.
- The **Version** and **Comments** fields are empty by default. Enter version numbers or comments to track model changes or changes to specific node contents.

Used Products

Under **Used Products**, any COMSOL module that is used by that node is listed. This makes it possible to identify which nodes that make use of the license for a certain module.

Returning to the Settings Window

When you are finished editing the properties, right-click the node again and select **Settings** to return to the settings window.

ROOT NODE PROPERTIES

The settings window for the root node includes a **Node Properties** section with additional information about the model file. It also incudes the **Used Products** section and an additional section, **Model Thumbnail**, which contains a thumbnail image of the model.

Opening the Root Node Properties Window

To open the root node's settings window with the **Node Properties** section, click the top **Root** node in the **Model Builder**.

The following information displays in the settings window:

Node Properties

- The **Name** field displays the file name as defined by the user. This field cannot be edited, but if the file name is changed, the new name displays here.
- The **Path** field displays the full path to the current MPH-file. When a new project is started, the field is empty until the MPH-file is saved for the first time. This field cannot be edited, but if the file path changes, the new path displays here.
- The **Program** field displays the name and build number for the version of COMSOL Multiphysics used. This field cannot be edited.
- The **Tag** field has a unique system-defined tag for the node. This field cannot be edited.
- The **Author** field contains the name of the author (creator) of the model (see above). Click the **Reset Author in Model** button to reset the author name for all nodes in the model to the name in the **Author** field for the root node.

- The **Date created** field contains the node creation date and time. This field cannot be edited.
- The **Date modified** field contains the date and time for the last time when the MPH-file was saved. This field cannot be edited.
- The **Modified by** field contains the name of the user who most recently saved the MPH-file. Changing the author name in the **Author** field and then clicking the **Reset Author in Model** button also resets the name in the **Modified by** field.
- The **Version** and **Comments** fields are empty by default. Enter version numbers or comments to track model changes or changes to specific file contents.

Used Products

Under **Used Products**, any COMSOL module that is used by the file is listed.

Model Thumbnail

Under **Model Thumbnail**, and only after some results are plotted, a thumbnail image displays representing the model.

Unit System

Select a **Unit System**. The default is **SI**. Other options are **British engineering units**, **CGSA**, **MPa**, **EMU**, **ESU**, **FPS**, **IPS**, **Gravitational IPS**, and **None**.

Font

Under **Font**, select a font **Family** and font **Size** from the lists.

See Also

- Editing Preferences Settings
- Saving a Model Thumbnail Image
- Using Units

Opening and Searching the Model Library

The Model Library contains sets of models that can be used for a variety of purposes. Each add-on module includes its own model library with models showing how to use the module within its application area. Each model includes full model documentation and a brief model description, including the solution times and information about the computer used for solving the model. These model descriptions appear at the bottom of the **Model Library** window when you select a model.

This section explains how to open the Model Library, search for a model, and open a model and its associated documentation.

1 From the main menu, select **View>Model Library** ( ).

2 Browse the available libraries or enter all or part of the model name or any other phrase or words or and click **Search**. By defaults, COMSOL searches for all words in the **Search** field; to search for a phrase, enclose it in quotation marks (for example, `"plane stress"`). The words can be part of the model's name or its documentation..

Note

COMSOL models are named using an underscore between words (for example, *effective_diffusivity*) because the model name is also the name of the corresponding Model MPH-file. The underscore is required to form a valid filename so it is recommended that you, if you are not sure of the full name, enter only the first word in the **Search** field when searching for a model name.

Tip

To find models that use particular physics interfaces, you can use the physics interface tags. In the **Search** field, type one or more space-separated physics interface tags. For example, search for `jh` to find models that use the Joule Heating interface, or search for `spf chds` to find models that use the Single-Phase Flow and Transport of Diluted Species interfaces. Note that you might, for some searches, also find other models where a tag appears for other reasons somewhere in the model file or documentation.

- If you used a search and the search is successful, or if you just browse the model library folders, double-click a model file name, right-click the name and choose **Open Model**, or click the **Open Model** button to open the model file for the selected model. Click the **Open Model and PDF** button (or choose the corresponding option on the model's context menu) to open the model and its documentation as a PDF file. Click the **Open PDF** button to open the model documentation as a PDF file only. For a selected model you can also access the documentation in the **Help** window.

- If the search does not return any results, the Model Library window contains the message **No Matching Model Found**. Click the **Refresh** button ( ) to return to the root **Model Library** folder list.

See Also

- Changing the Model Library Root Folder
- Updating the Model Library Using Model Library Update
- The Model Library Window
- The Help Window
- Adding a User Model Library

Changing the Model Library Root Folder

After developing and saving your own models, it may be useful to change the **Model Library** root folder. This redirects COMSOL to a different folder where customized models can be stored. To add a user model library while keeping the installed model library, us the ability to add a user model library instead (see Adding a User Model Library).

1 From the main menu, select **View>Model Library** ( ).

2 Click the **Set Model Library Root** button ( ).

3 In the **Model Library Root** box, navigate to the new root folder location or click **Make New Folder**.

4 Click **OK** to save the changes, or **Cancel** to exit without saving.

See Also

- Opening and Searching the Model Library
- Updating the Model Library Using Model Library Update
- The Model Library Window
- Adding a User Model Library

Updating the Model Library Using Model Library Update

Note

Using the **Model Library Update** service requires Internet access. For a default installation, you also need to run COMSOL as administrator. See the section Updates for instructions on how you can modify your installation to avoid this restriction.

See Also

COMSOL MPH-Files

The Model Library Update is a service that provides new and updated models for the model libraries. To update the model libraries, use the **Model Library Update** window, which contains new and updated models for the model libraries of the COMSOL products that your license includes:

1 From the main menu, select **View>Model Library Update** ( ). Alternatively, click the **Model Library Update** toolbar button ( ) in the **Model Library** window or right-click the node for a model with compact MPH-file; see the section COMSOL MPH-Files for an explanation of the concept of "full" and "compact" MPH-files.

Note

If the **Model Library Update** window displays the message **Your Model Library is up-to-date**, no updated or new models are available.

2 In the **Model Library Update** window, a message displays confirming if your Model Library is up-to-date. If it is not, browse the list of new and updated models that appear with a description and image.

3 Choose which models to download by selecting or clearing the check boxes next to the model thumbnail images. By default all check boxes are selected. The latest documentation file as well as any supplementary files are included in the download for each model.

4 At the bottom of the **Model Library Update** window, click the **Download Selected** button to download the selected models, or click the **Download All** button to download all available models. The download time depends on the size of the files, which is listed for each model, and the bandwidth of the Internet connection.

After downloading new and updated models, click the **Refresh** button ( ) in the **Model Library** window to update the lists of models in the model libraries.

See Also

- Opening and Searching the Model Library
- Changing the Model Library Root Folder
- The Model Library Window
- Adding a User Model Library

Adding a User Model Library

Add customized Model Library folders to the Model Library using the **Add Model Library** button ( ).

1 From the main menu, select **View>Model Library** ( ).

2 Click the **Add Model Library** button ( ).

3 In the **User Model Library** box, navigate to a location on your computer where you want to create a custom model library folder. Or click **Make New Folder**.

Note

It is not possible to add a library identical to, containing, or being contained in, any already used library.

4 Click **OK** to save the changes, or **Cancel** to exit without saving.

To remove a user model library from the **Model Library** window, select it and click the **Remove Selected** button ( ).

Editing Preferences Settings

To make changes to how items are displayed throughout COMSOL Multiphysics, from the main menu select **Options>Preferences** and then edit the following settings in the **Preferences** dialog box as required. Click the **Restore Defaults** button to restore the default values for all preference settings.

Important

For some setting changes, a message window displays to tell you that COMSOL Multiphysics needs to be restarted for the changes to take effect.

- General
- User Interface
- Show
- Expand Sections
- Model Builder
- Cluster Computing
- Geometry
- Graphics
- Results
- Builder Tools
- Temporary Files
- Memory and Processors
- Updates
- LiveLink Products

GENERAL

Select **Options>Preferences>General** from the main menu and then:

- Under **Precision**, edit the **Input display precision** or **Table display precision** levels. The defaults are 5 for both fields. If the check box is selected next to the **Input display precision** field, inputs in most fields that take a floating-point value as input always appear with full numerical precision when the field is active. The value in the **Table display precision** field control the display precision for data values in the result tables.
- Select a GUI **Language**—**Chinese (Simplified)**, **Chinese (Traditional)**, **English** (the default), **French**, **German**, **Italian**, **Japanese**, **Korean**, or **Spanish**. Click **OK** to exit and re-open COMSOL in order to display the GUI in the selected language.
- Under **Help**, when the **Sticky help** check box is not selected, it means the **Help** window always updates its contents to shows the relevant context help information

when clicking a window tab or on a model tree node in the **Model Builder**, regardless of which part of the help system is displayed in the **Help** window. When the **Preload help files** check box is selected, it means that the help files are preloaded when launching COMSOL Multiphysics. By preloading the help files the context help opens faster when it is used for the first time. From the **Help mode** list, select **Integrated** (the default) to show the help contents in a **Help** window that is integrated in the COMSOL Desktop environment or select **Web browser** to display the help contents in a separate web browser. See also The Help Window.

- Under **Log** it is possible to change the buffer size (in number of characters) of the **Log** window in the **Log window size (characters)** field. The default size is 300,000 characters. This buffer size also applies to the Log stored in the solvers for the last run.
- Under **Recovery**, the **Save recovery file** check box is selected by default to save recovery files during the solution process for time-dependent, parametric, and nonlinear solvers. Click to clear as required. See Saving Recovery Files for details.
- Under **Username**, edit the **Name** field to change the username.
- Under **History export**, select the **Include author** check box to add the author to the header of Model Java-files and Model M-files that you save.

USER INTERFACE

In the **Look and feel** area, you can specify the **Toolbar style**:

- Select **COMSOL** (the default) to use a toolbar where the main tools appear as left-aligned icons in the blue area below the window's name. The settings window for a node has the same name as the node (**Block**, for example).
- Select **Classic** to use the toolbar look and feel from earlier versions of COMSOL, where the main tools appear as right-aligned icons in a grey area below the window's name. The name of all settings windows is **Settings**.

Note

If you change the toolbar style, you must restart COMSOL for the change to take effect.

SHOW

Note	Any change to these settings are applied to new models or when a model has been created in a previous version of COMSOL (prior to version 4.3). These preferences can also be set by clicking the **Show** button ( ) in the **Model Builder** as detailed in Showing and Expanding Advanced Physics Sections, although those settings are only saved with the current model.

Use this section to control whether to show some more advanced nodes and sections. Select **Options>Preferences>Show** from the Main menu. Click **Select All**, **Deselect All**, **Reset to Default**, or **Copy to Model** as required. If you click **Copy to Model**, the software uses the specified preference settings immediately. The default is to show **Equation sections** and **Override and contribution**.

EXPAND SECTIONS

Note	These settings are applied to new models or when a model has been created in a previous version of COMSOL (prior to version 4.3). These preferences can also be set by clicking the **Show** button ( ) in the **Model Builder** as detailed in Showing and Expanding Advanced Physics Sections, although those settings are only saved with the current model.

Use this section to control if some sections in the settings windows for the physics interface nodes appear initially collapsed or expanded. Select **Options>Preferences>Expand sections** from the Main menu. Click **Select All**, **Deselect All**, **Reset to Default**, or **Copy to Model** as required, or select the individual check boxes for the sections in the physics interface nodes' settings windows that you want to see expanded by default. If you click **Copy to Model**, the software uses the specified preference settings immediately.

The default setting is to not expand any of these sections.

MODEL BUILDER

Under **Sorting**, from one or both of the **Definitions node sorting** or **Physics node sorting** lists, select one of the following—**Disabled in new models** (the default), **Enabled in new models**, **Always enabled**, or **Always disabled.**

Enabled in New Models

When making a new model and **Enabled in new models** is selected:

- The physics interface nodes are organized by default based on the *space dimension* (domains, boundaries, edges, and points).
- The **Global Definitions** and **Definitions** nodes are organized by *type* (for example, **Functions**, **Selections**, and **Coordinate Systems**).

When an existing model is opened, and **Enabled in new models** is selected then:

- Right-click the physics interface node and select **Sort by Space Dimension** to organize the nodes by space dimension.
- Right-click the **Global Definitions** or **Definitions** node and select **Sort by Type** to organize the nodes by type.

For any model and when **Enabled in new models** is selected, right-click the physics interface, **Global Definition**, or **Definitions** node and select **No Sorting**.

Disabled in New Models

When making a new model and **Disabled in new models** is selected, the nodes are not sorted by default. When an existing model is opened, and **Disabled in new models** is selected then:

- Right-click the physics interface node and select **Sort by Space Dimension** to organize the nodes by space dimension.
- Right-click the **Global Definitions** or **Definitions** node and select **Sort by Type** to organize the nodes by type.

Always Enabled

When **Always enabled** is selected, the nodes are always sorted. No options to change the sorting display in the context menu.

Always Disabled

When **Always disabled** is selected, the nodes are never sorted. No options to change the sorting display in the context menu.

See Also

Sorting Nodes by Space Dimension and Type

CLUSTER COMPUTING

Note

See also Show Advanced Study Options where you can control the options for running COMSOL on a cluster from the **Study** node in the **Model Builder**. When **Save as default** is selected from the Study nodes, the **Preferences** window described in this section is where the current settings are saved.

See Also

Details for the cluster type settings are described in Cluster Computing in the *COMSOL Multiphysics Reference Guide*.

The following preference settings for cluster computing are available under **Settings**:

- Select a **Cluster type**—**General** (the default), **HPCS 2008** (Windows HPC Server), **WCCS 2003** (Windows Computer Cluster Server), **OGS/GE** (Open Grid Scheduler/ Grid Engine), or **Not distributed**.
- Select the **MPD is running** check box if required on Linux.
- Enter details about how to start on Linux in the **Host file** and **Rsh** fields.
- The default **Scheduler** is **localhost**.
- Enter a username in the **User** field.
- If you are using OGS/GE enter a **Queue name** for the cluster queue.
- Enter file paths (or use the defaults) for the following:
  - The defaults for the **Batch directory** and **External COMSOL batch directory path** are `home` on Linux and Mac and `Documents` on Windows.
  - The default for the **External COMSOL installation directory path** is the original installation file path.

  Click the **Browse** button for the file path that you want to specify to browse the file system instead of typing the file path.
- Enter a **Prepend command** and **Postpend command** as required for adding commands before and after the main job commands, respectively.
- Enter the **Number of job restarts** if you want the process to restart if failing. The default value is 0 (that is, no job restarts).
- Enter a value for the **Alive time**, the interval time that the running process has before signalling that it is alive. The default is 300 seconds.

GEOMETRY

Select **Options>Preferences>Geometry** from the Main menu and then, in the **Geometry representation area**:

- Select the **Default geometry representation**—**COMSOL kernel** or **CAD Import Module kernel** (the Parasolid kernel).
- Select the **Switch kernel automatically** check box to make the software switch geometry kernel automatically when needed.

Under **Automatic rebuild** you can control if COMSOL automatically rebuilds the geometry sequence when you click a node outside of the **Geometry** branch. Select **On** from the **Default in new geometries** to make an automatic rebuild of the geometry the default behavior, or select **Off** to make no rebuilding the default behavior.

In the **Visualization** area, select the **Draw on work plane in 3D** check box to activate interactive drawing of 2D geometries directly on work planes in the 3D geometry.

Note	The CAD Import Module kernel requires the CAD Import Module.

See Also	The Geometry Node and Its Settings in the *COMSOL Multiphysics Reference Guide*

GRAPHICS

Select **Options>Preferences>Graphics** from the Main menu and then:

Visualization

Under **Visualization**:

- Select a **Rendering** type—**OpenGL** (the default), **DirectX 9**, or **Software**. The software might change the rendering to use software rendering if the graphics hardware do not support the chosen rendering type.

Note

If you change the renderer, you must restart COMSOL for the change to take effect.

- From the **Optimize for** list, **Performance** is the default. Select **Quality** if you are using the material appearance to display the details and "texture" that make water look like a liquid, for example. If **Performance** is selected, only the color appearance is displayed. Click **OK** to exit and re-open COMSOL in order to apply the changes.

See Also

Customizing the Appearance of a Material in the Graphics Window

- Select a **Multisampling**—**Off**, **Medium** (the default), or **High**. Multisampling uses multiple samples per pixel to improve the graphics by providing better antialiasing of edges. The exact number of samples for a setting depends on the graphics card and driver. Select **Off** to use the lowest number of samples, select **Medium** to use a value in-between the lowest and the highest, and select **High** to use the maximum supported samples per pixel.

Note

Multisampling is only available with hardware rendering using graphics cards that support multiple samples per pixel.

- Select a **Detail**—**Normal** (the default), **Wireframe**, **Coarse**, or **Fine**. This controls the level of graphics rendering detail. Select **Wireframe** to speed up rendering of complex models or to improve visual appearance.
- The **Show logo on canvas** check box is selected by default and displays the COMSOL logo on the **Graphics** window. Click to clear the check box as required.

- The **Show materials** check box is selected by default and displays the material texture and color as defined in Customizing the Appearance of a Material in the Graphics Window.
- Select the **Show physics symbols** check box to display boundary condition symbols that are available with structural mechanics features. The check box is not selected by default.

See Also

Physics Symbols for Boundary Conditions

Precision

Under **Precision** you can specify the precision of the numerical values that appear as axis labels and color legend values:

- In the **Graph** field enter an integer between 1 and 15 for the precision (number of digits) for the values on the axes in 1D plots and graphs. The default setting is 5 digits.
- In the **2D axis** field, enter an integer between 1 and 15 for the precision (number of digits) for the values on the axes in 2D plots. The default setting is 4 digits.
- In the **3D grid** field, enter an integer between 1 and 15 for the precision (number of digits) for the values on the axes of the grid in 3D plots. The default setting is 3 digits.
- In the **Color legend** field, enter an integer between 1 and 15 for the precision (number of digits) for the values displayed on the color legend. The default setting is 5 digits.

Animation

Under **Animation**, select a **Codec**—**MJPEG Compressor** or **DV Video Encoder**. The default is the first codec that the program finds on the computer.

Font

Under **Font**:

- Select the **Load system fonts** check box to always load system fonts.
- Select a font **Family** and font **Size** (the default is 9).

Note

The **Font** option is for the text that displays for plots in the **Graphics** window. The change is applied when creating a new model. You can also change the setting used for a model from the root node's settings window: in the **Model Builder** click the root node and edit as above.

RESULTS

Select **Options>Preferences** from the Main menu and then click **Results**:

Automatic Update of Plots

Under **Automatic update of plots** you can set the preferences for the updating result plots:

- Select the **Disable for new models** check box to always disable automatic update of plots for new models that you create.
- Select the **Disable for models loaded from file (override saved)** check box to always disable automatic update of plots for models that you open from file, such as models in the model libraries. The automatic update of plots is then disabled initially, regardless of the setting in the model.

If you want to activate automatic updates of plots for the current model, select the **Automatic update of plots** check box in the **Result** node's settings window.

Convergence Plot

Under **Convergence plot**, you can specify preferences for the convergence plots that you can use to monitor the solver convergence:

- The **Generate convergence plots** check box is selected by default. Clear that check box if you do not want the software to generate convergence plots while computing the solution.
- For the convergence plots, enter a buffer size (in number of steps) in the **Convergence plot buffer size (steps)** field (default value: 10,000 steps).

Table

Under **Table**, the **Buffer size (rows)** controls the size of the buffer for storing table data. The default value is 10,000 rows.

Report

These preference settings control some properties for the report generator. In the **Default report style-sheet file** field type the full path and filename to a style sheet (.css-file) that you want to use as the default style sheet for reports of HTML format. Click **Browse** to browse to a file that contains the style sheet.

In the **Logotype file** field, type the full path and filename for an image file (on PNG or JPEG format) to use as the logotype. Click **Browse** to browse to the logotype file. If left empty, the COMSOL Multiphysics logotype appears in the report.

In the **Company** field, type the name of a company associated with the report if desired.

From the **Default image size** list you can select the default size for report images; choose between **Extra small**, **Small**, **Medium**, and **Large**. Similarly, use the **Default image type** list to specify the default type for the report images: **PNG**, **JPEG**, or **BMP**.

Select the **Prompt for update of table of contents in Microsoft Word** check box to make Microsoft® Word ask whether you want to update the table of contents when you first launch a report of this format. Such an update is necessary to generate page numbers in the table of contents, but you can choose to do the update after you have opened the document in Microsoft Word. By default this check box is not selected.

BUILDER TOOLS

These preference settings control some properties for the Physics Builder and its tools.

In the **Settings** area, select the **Enable Physics Builder** check box to make the Physics Builder available from the **File** menu.

In the **Editor** area you can specify some properties for the code editor:

- Select the **Auto indent** check box to activate automatic indentation (this is active by default)
- Select the **Spaces for tabs** check box to use spaces for the tabs, replacing tab characters with empty spaces (this is the default setting).
- Select the **Auto-close quotes** check box to automatically add a closing quotation mark when needed (this is the default setting).
- Specify the tab width in number of characters in the **Tab width** field. The default is 2 characters.

In the **Comments and documentation** area you can control whether to show the nodes for adding comments and documentation to the physics interfaces and applications that you create:

- From the **Show comments** list, choose one of the following options:
  - **All** to display all **Comments** nodes for adding comments about the functionality in the specific feature or functionality that the node implements (the default setting).
  - **Changed** to display only **Comments** nodes where the contents has changed (that is, empty **Comments** nodes are not displayed).
  - **None** to not display any **Comments** nodes.
- From the **Show documentation** list, choose one of the following options:
  - **All** to display all **Documentation** nodes and their subnodes for adding documentation about the functionality (the default setting).
  - **Changed** to display only **Documentation** nodes where the contents has changed (that is, empty **Documentation** nodes are not displayed).
  - **None** to not display any **Documentation** nodes.
- From the **Enumerate sections to level** list, choose which section levels, if any, in the documentation should be numbered. The options are
  - **None** for unnumbered sections.
  - **Level 1**(default), **Level 2**, …, **Level 6** to enumerate sections up to a specific level, where level 1 indicates the top level in a hierarchical documentation structure.
  - **All** to enumerate sections at all levels.
- In the **Section levels in table of contents** list you can choose how many section levels, between 1 and 5 (2_levels is the default), to include in the table of contents if you have included one in the documentation.
- The setting in the **Generate separate files at section level** list determines the highest section level that starts a new HTML file in the generated documentation. For example, if you specify **Level 2** (the default), a new file is generated when sections at level 1 and level 2 begin, while sections at level 3 or higher are placed in the same HTML files as their ancestor of level 2.
- The **Documentation plug-in prefix** field lets you specify a default namespace for documentation in a plug-in format that is applicable to your organization (for

example, COMSOL uses the plug-in prefix `com.comsol`). This setting can be overridden by the **Plug-in prefix** field in the **Documentation** settings window.

- Specify your organization's name in the **Vendor** field. This name appears in the manifest file inside the generated help plug-in file.

TEMPORARY FILES

Recovery

Select the **Save recovery file** check box to save recovery files to disk (this is the default setting). In the **Folder for recovery files** field you can specify another folder than the default to, for example, use a folder where there is more disk space for storing large recovery files. Click **Browse** to browse to a recovery file folder.

Temporary files

In the **Folder for temporary files** field you can specify another folder than the default to, for example, use a folder where there is more disk space for storing large temporary files. Click **Browse** to browse to a folder for temporary files.

If you run COMSOL in a client-server configuration, you can specify a **Folder for recovery files on server**, **Folder for temporary files on client**, and **Folder for temporary files on server**.

MEMORY AND PROCESSORS

Parametric Sweep

To reduce the size of MPH-files for models using parametric sweeps, you have the option of storing only the last solution in the sweep in the file. If you want to use this setting as the default, choose **Only last** from the **Keep solutions in memory** list. You then have the further option of saving each solution as a model file. To do this, select the **Save each solution as model file** check box. The default option in the **Keep solutions in memory** list, **All**, stores all solutions in the file.

Processors

By default, COMSOL uses all available processors (cores) on a multicore platform and displays that number of processors. If you want to control how many processors that COMSOL can use, select the **Number of processors** check box and enter the number of processors in the associated field. This corresponds to using the `-np` option for the COMSOL commands from the command line.

UPDATES

Product Updates

Select the **Check for updates at launch** check box to make the program check for updates each time COMSOL Multiphysics is launched.

Proxy Server Settings for Communication with COMSOL's Model Library Update Server

If you connect to the Internet through a web proxy, you can use the controls in this section to specify which proxy server settings COMSOL Multiphysics should use when communicating with COMSOL's website for performing a Model Library Update.

The **Configuration** list provides the following alternatives:

- **No proxy server**: Connect to the Model Library Update server directly, bypassing any proxies. This is the default setting.
- **Use system settings**: Use the system-wide proxy server settings defined on your computer.
- **Manual**: Choose this alternative if you want to specify a proxy server by entering the name (or IP address) in the **Server** field and the port number in the **Port number** field. The default port number is 443, which is the default for HTTP secure (HTTPS). If the proxy server requires authentication, you will be asked to provide username and password the first time you call Model Library Update in each COMSOL modeling session.

Place Model Library Update Files In

The **Destinations** list provides two options for specifying which model and documentation directories are synchronized with COMSOL's server when you launch a Model Library Update request:

- **Current model and documentation directories** (default): Synchronize with MPH-files under the Model Library root set in the **Model Library** window and with model documentation files under the directory specified in the **Documentation root directory** field on the **Preferences** dialog's **General** page (see the General section in this chapter).
- **Specify custom directories**: Choosing this option lets you specify model and documentation root directories separate from those of your current COMSOL Desktop environment.

By default, both the Model Library and documentation root directories lie directly under the COMSOL installation root directory, in models/ and doc/ respectively.

This typically implies that special permissions are required for saving downloaded files, and it can therefore be beneficial to move or copy the directories to a different location. The settings referred to in this section are provided to let you customize Model Library Update to the IT environment of your organization.

LIVELINK PRODUCTS

These settings makes it possible to specify the installation and common file folders for some of the COMSOL LiveLink products:

LiveLink for MATLAB

If needed, specify the **MATLAB installation folder** in the text field, or click **Browse** to locate the folder. By default, the MATLAB installation folder is the one you specified when installing COMSOL, but you may need to change the location of that folder (for example, after installing another MATLAB version).

On Windows only, if you change the MATLAB installation folder, also click the **Register MATLAB As COM Server** button to update the path of the MATLAB version that is connected to COMSOL that was defined during the initial COMSOL installation.

LiveLink for Pro/ENGINEER®

If needed, specify the **Pro/ENGINEER® installation folder** in the text field, or click **Browse** to locate the folder. By default, the Pro/ENGINEER® installation folder is the one you specified when installing COMSOL.

LiveLink for Creo Parametric™

If needed, specify the **Creo Parametric™ custom file folder** in the text field, or click **Browse** to locate the folder. By default, the Creo Parametric™ installation folder is the one you specified when installing COMSOL.

If you have a license for both these CAD LiveLink packages, select **Pro/ENGINEER®** or **Creo Parametric™** from the **Use** list.

Checking and Controlling Products and Licenses Used

Open the **Licenses** window to list or block the products your license includes. Blocking a license can be useful for consultants who want to duplicate a client's environment while building a model or when collaborating with other users who do not have access to the same set of COMSOL products. You can also use these settings to prevent the use of a module when sharing a floating-network license, for example.

1 To view a list of licenses or block the use of a product, from the main menu, select **Options>Licenses**. The **Licenses** window opens.

By default the use of all products is active and the check boxes for all products are selected. The licenses currently in use are unavailable (you cannot block the use of products with functionality already in use).

2 Click to clear the check box next to a product to hide or block it from use. Click the **Select All** button to activate all products. Click the **Deselect All** button to block all products (except that ones that are already in use).

Note

The settings are local for each COMSOL Multiphysics session. The program does not store these settings for future modeling sessions.

3 Click **OK** to save the changes or **Cancel** do discard any changes and close the window.

Tip

Under **Other products** is a list of COMSOL products the license does not include. Click the **Product Information** button to go to the product information pages on the COMSOL website, where you find information about all COMSOL products.

You can also get information about the licensed products from the **About COMSOL Multiphysics** box on the **Help** menu. Select **Licensed products** from the list and click the **Show Information** button to display a list of used licenses and the total number of licenses for each product.

Borrowing Licenses

If you have a floating network license (FNL) or a class kit license (CKL) and your license file has been enabled for borrowing you can use the **Borrow License** dialog box to borrow licenses from your license server. Select **Options>Licenses** from the main menu and then click **Borrow**. Select the licenses you want to borrow from the list and specify the number of days you want to keep them. Click **OK**.

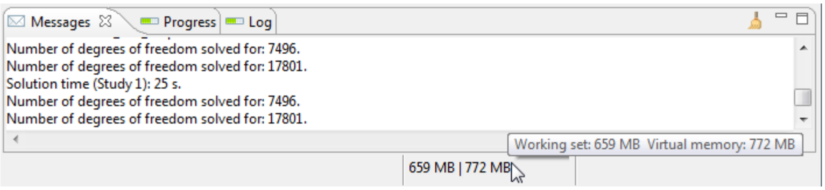
Important

Remember that other users cannot use the licenses that you have checked out. If you try to borrow a license that has already been borrowed, you receive an error message that shows for which products the license has been borrowed.

Information About Memory Use

In the bottom-right corner of the COMSOL Desktop is information about how much memory COMSOL Multiphysics currently uses. The two numbers, display as, for example, **405 MB | 453 MB**, represent the physical memory (working set) and the virtual memory, respectively. If you position the cursor above these numbers, the tooltip that appears includes the numbers with the type of memory explicitly stated:

- The **Physical memory** or **Working set** (on Windows) number is the amount of physical memory (RAM) or the amount of virtual memory in "active" use (the amount of physical memory that is required so that all used virtual memory is in RAM).
- The **Virtual memory** number is the is the current size of the virtual address space that COMSOL Multiphysics uses.



```
Number of degrees of freedom solved for: 7496.
Number of degrees of freedom solved for: 17801.
Solution time (Study 1): 25 s.
Number of degrees of freedom solved for: 7496.
Number of degrees of freedom solved for: 17801.
```

Working set: 659 MB Virtual memory: 772 MB

659 MB | 772 MB

Figure 7-3: An example of memory use displayed in COMSOL Multiphysics.

Viewing Node Names, Identifiers, Types, and Tags

Select the node labels to display in the **Model Builder**. You can view combinations of the following label types:

- *Names* are only used in the Model Builder for names (descriptions) of the nodes.
- *Tags* are unique for each node and are assigned automatically. They are used when working with a model through the COMSOL Java API or the optional LiveLink for MATLAB.

- *Identifiers* are primarily used as names for functions and operators when called in models and as identifiers for the **Model** nodes and physics interfaces that you use for variable scoping.
- *Types* are the unique descriptions for each type of node.

Names and some identifiers can be changed by the user. You can change the name of a node for all levels (except the root node, which get its name from the model filename), but the identifier can only be changed for the top **Model** and physics interface nodes and as a name for functions and operators. The unique tag and type are automatically assigned by the software and cannot be changed.

You can specify the name of a node in its **Properties** section or by right-clicking the node and choosing **Rename** or pressing F2, which opens a window where a **New name** is entered. The identifier serves as the name of functions and operators, which you can define in the settings windows' **Function Name** or **Operator Name** section for those nodes. You can also specify it in the **Model Identifier** and **Interface Identifier** sections for the main **Model** nodes and physics interface nodes, respectively.

SELECTING THE LABELS TO VIEW IN THE MODEL BUILDER

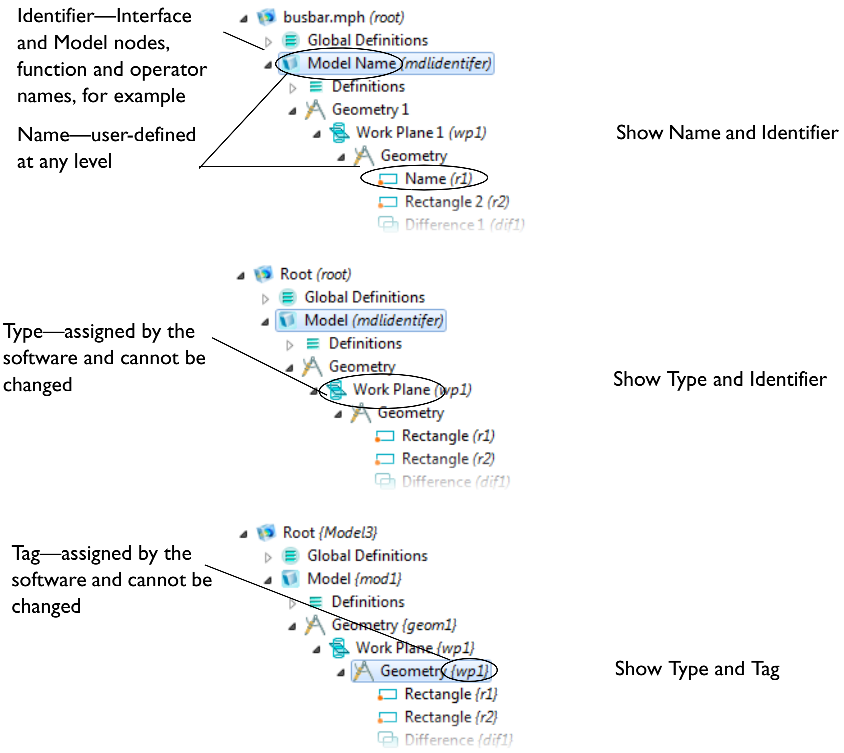
1. In the **Model Builder**, click the top node level where you want to display specific labels. Each node level can display different labels.
2. Select **View>Model Builder Node Label** to select from the list: **Show Name Only**, **Show Name and Identifier**, **Show Name and Tag**, **Show Type and Identifier**, or **Show Type and Tag**. See Figure 7-4 for examples of the different label types.

The options in the list modify the view and options in the **Model Builder** in the following way:

- Select **Show Name Only** to only display the name.
- Select **Show Name and Identifier** to display the name with the identifier in parentheses using an italic font. The identifier appears only where it is defined, that is, as for model nodes, physics interface nodes, mathematical interface nodes, functions, model couplings, and geometry features. This is the default setting.
- Select **Show Name and Tag** to display each node's feature name with the predefined tag in curly braces using an italic font.
- Select **Show Type and Identifier** to display each node's feature type (predefined name) with the identifier in parentheses using an italic font.

- Select **Show Type and Tag** to display each node’s feature type (predefined name) with the predefined tag in curly braces using an italic font.

  The previous selections are mutually exclusive: only one of them can be active. The following two settings can be activated or turned off individually:



Identifier—Interface and Model nodes, function and operator names, for example

Name—user-defined at any level

Show Name and Identifier

Type—assigned by the software and cannot be changed

Show Type and Identifier

Tag—assigned by the software and cannot be changed

Show Type and Tag

Figure 7-4: Examples of the available label combinations on the View menu.

See Also

Editing Node Properties, Names, and Identifiers

Showing and Expanding Advanced Physics Sections

To display the features described in this section, click the **Show** button ( ) on the **Model Builder** and then select the applicable option. You can also click the **Expand Sections** button ( ) to always show some sections (see Expanding Sections). From either menu, select **Reset to Default** to reset the sections to display only the **Equation Sections** and **Override and Contribution**.

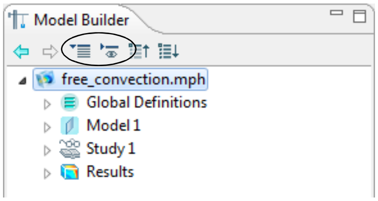


Figure 7-5: The location of the Expand Sections and Show buttons in the Model Builder.

Tip

These options are also accessed (and activated) by selecting the check boxes from the **Options>Preferences>Show** and **Options>Preferences>Expand sections** windows. See Editing Preferences Settings.

- Resetting to Default
- Resetting from Preferences
- Setting as Preferences
- Expanding Sections
- Expanding the Equation Sections
- Show Equation Sections
- Show Equation View
- Show Override and Contribution
- Show Discretization
- Show Stabilization
- Show Advanced Physics Options

- Show Advanced Physics Options—Context Menu Equation-Based Feature Nodes
- Show Advanced Study Options
- Show Advanced Results Options

Resetting to Default

From the **Model Builder**, click the **Show** button ( ) and select **Reset to Default** to reset the sections to display only the **Equation Sections** and **Override and Contribution**.

From the **Model Builder**, click the **Expand Sections** button ( ) and select **Reset to Default** to reset it so all sections are collapsed (not expanded).

Resetting from Preferences

From the **Model Builder**, click the **Show** button ( ) and select **Reset from Preferences** to reset the sections to display the system preferences defined from the **Options>Preferences>Show** window.

From the **Model Builder**, click the **Expand Sections** button ( ) and select **Reset from Preferences** to reset the sections to the system preferences defined from the **Options>Preferences>Show** window.

Setting as Preferences

From the **Model Builder**, click the **Show** button ( ) and select **Set as Preferences** to make these selections the default system preference (also defined from the **Options>Preferences>Show** window).

From the **Model Builder**, click the **Expand Sections** button ( ) and select **Set as Preferences** to make these selections the default system preference (also defined from the **Options>Preferences>Show** window).

Expanding Sections

To always display sections on a physics interface from the **Model Builder**, click the **Expand Sections** button ( ) and choose as many of these options as required—**Equation**, **Override and Contribution**, **Discretization**, **Stabilization**, and **Advanced Physics Options**.

See Expanding the Equation Sections, Show Equation Sections, Show Override and Contribution, Show Discretization, Show Stabilization, and Show Advanced Physics Options for details about these features.

Tip

These options are also accessed (and activated) by selecting the check boxes from the **Options>Preferences>Expand sections** window. See Editing Preferences Settings.

Expanding the Equation Sections

To always display this section in its expanded view, click the **Expand Sections** button (≡) and select **Equations**. Selecting this option expands all the Equation sections on physics interface settings windows.

See Also

Physics Nodes—Equation Section

Show Equation Sections

To enable this feature, click the **Show** button (≡) and select **Equation Sections**. This displays the **Equation** section on all physics feature nodes settings windows.

See Also

Physics Nodes—Equation Section

Show Equation View

To display the **Equation View** node under all physics interface nodes in the **Model Builder**, click the **Show** button ( ) and select **Equation View**.

See Also

- The Equation View Node
- Physics Nodes—Equation Section

Show Override and Contribution

To enable this feature, click the **Show** button ( ) and select **Override and Contribution**. The **Override and Contribution** section is then included in all physics interface feature nodes (for material modes, sources, boundary conditions, and so on). For a specific node, the **Override and Contribution** section in its settings window contains lists of other nodes that the node is overridden by, other nodes that the node overrides, and other nodes that contributes with the node.

See Also

- Listing Overrides and Contributions
- Overridden Selections

Show Discretization

Note

There are two categories of discretization—a section on the physics interface settings window (described here) and adding The Discretization Node for global equation based features.

To enable this feature, click the **Show** button ( ) and select **Discretization**. The options available for each feature node are described individually throughout the documentation.

DISCRETIZATION

Element Order and Shape Function Type

The **Element order** (or, more precisely, the order of the shape function) directly affects the number of degrees of freedom in the solution and the accuracy of the solution. Increasing the order of the elements roughly corresponds to a uniform mesh refinement. Most physics interfaces uses Lagrange elements, which can be of order 1 to 4 (in 3D) or 5 (in 1D and 2D), with 2 being the default order. You can change the order using the **Element order** list. The software adapts the order of the numerical integration to the element orders for the physics interfaces in the model. Some physics interfaces use special element types or a reduced element order for some of the field variables.

See Also

Some additional information is included in Element Types and Discretization and Finite Elements in the *COMSOL Multiphysics Reference Guide*.

Discretization of Fluids

The following is an example of the choices of element order for the **Laminar Flow** interface:

- **P1+P1** means linear elements for both the velocity components and the pressure field. Linear elements are computationally cheaper than higher-order elements and are also less prone to introducing spurious oscillations, thereby improving the numerical robustness.

- **P2+P1** means second-order elements for the velocity components and linear elements for the pressure field. Second-order elements work well for low flow velocities.
- **P3+P2** means third-order elements for the velocity components and second-order elements for the pressure field. This can add additional accuracy but it also adds additional degrees of freedom compared to P2+P1 elements.

Note

The abbreviation P_mP_n is often used to indicate the polynomial order of, in this case, the shape functions (elements) for the velocity components (m) and the pressure (n) when using tetrahedral or triangular elements. Here a corresponding nomenclature is used for all element shapes.

The theory about this is in P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley & Sons, 2000.

Complex Variables

Under **Value types when using splitting of complex variables**, you can specify the **Value type** (**Real** or **Complex)** of dependent variables when **Split complex variables in real and imaginary parts** setting is activated in the Compile Equations node of any solver sequence used. The default is the complex value type, but you can specify that the value of a dependent variable is real to make sure that it does not get affected by small imaginary contributions, which can occur, for example, when combining a time-dependent or stationary study with a frequency-domain study. If split complex variables setting is not active the value type is ignored.

See Also

For information about how to specify the splitting of complex variables, see Compile Equations in the *COMSOL Multiphysics Reference Guide*.

Show Stabilization

To enable this feature, click the **Show** button ( ) and select **Stabilization**. Selecting this option displays the **Consistent Stabilization** and **Inconsistent Stabilization** sections on the settings windows. If you have access to the **Level Set** and **Mixture Model** interfaces, it displays a **Stabilization** section instead.

Numerical stabilization is available for physics interfaces that model transport such as fluid flow or convective heat transfer, where the fundamental governing equations are less stable than, for example, conduction-dominated models, solid mechanics models, and wave propagation in the frequency domain.

Several physics interfaces have these settings available, and below you find the common information about the stabilization settings. Differences not described below are noted for the individual interface documentation.

See Also

- Stabilization Techniques and Numerical Stabilization in the *COMSOL Multiphysics Reference Guide*.
- Additional information is also described for the Heat Transfer interfaces (see Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces) and Fluid Flow branches (see Numerical Stability—Stabilization Techniques for the Fluid Flow Interfaces).

CONSISTENT STABILIZATION

There are two consistent stabilization methods— **Streamline diffusion** and **Crosswind diffusion**. Usually, both check boxes are selected by default and should remain selected for optimal performance. These consistent stabilization methods do not perturb the original transport equation.

The crosswind diffusion method specifies the smallest allowable concentration change across an element. As the concentration gradient appears in the denominator in the equations describing crosswind diffusion, the gradient ensures that unreasonable values do not occur in regions with small to negligible concentration changes.

Crosswind Diffusion and Tuning Parameter

On most interfaces, when the **Crosswind diffusion** check box is selected, enter a **Tuning parameter** C_k. The default is most often 0.5. The **Tuning parameter** controls the amount of crosswind diffusion introduced. It recommended that it is kept in the order of 1 in order to not introduce excessive amounts of diffusion. The value used must also neither be space dependent nor time dependent.

Crosswind Diffusion and Lower Gradient Limit

In some interfaces, if the **Crosswind diffusion** check box is selected, the **Lower gradient limit** g_lim (SI unit: K/m) field is available. This variable corresponds to the smallest concentration change across an element considered by the stabilization, and is used to make sure that the crosswind diffusion expressions are valid also in regions with small to negligible concentration changes.

Residual

For some interfaces, and for both consistent stabilization methods, select a **Residual** (or **Equation Residual**). **Approximate residual** is the default setting and it means that derivatives of the diffusion tensor components are neglected. This setting is usually accurate enough and is faster to compute. If required, select **Full residual** instead.

INCONSISTENT STABILIZATION

There is usually just one inconsistent stabilization method— **Isotropic diffusion**. This method is equivalent to adding a term to the diffusion coefficient in order to dampen the effect of oscillations. If possible, minimize the use of the inconsistent stabilization method because by using it, the original problem is no longer solved.

By default there is no isotropic diffusion selected. If required, select the **Isotropic diffusion** check box and enter a **Tuning parameter** δ_{id}. The default is 0.25. Increase or decrease this number to increase or decrease the amount of isotropic stabilization, and thus oscillation damping.

Show Advanced Physics Options

To enable this feature, click the **Show** button ( ) and select **Advanced Physics Options**. This displays the **Advanced** section on physics interface settings windows. This selection also activates the Show Advanced Physics Options—Context Menu Equation-Based Feature Nodes, which are described separately.

ADVANCED SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. This section does not normally show unless the physics interface contains some advanced options.

Show Advanced Physics Options—Context Menu Equation-Based Feature Nodes

In the **Model Builder**, when you right-click any physics interface node and open the context menu, there are additional feature nodes that can be added to your model. To enable this feature and add the following nodes, click the **Show** button ( ) and select **Advanced Physics Options** and then:

From the **More** menu add these nodes:

- **Weak Contribution.** See The Weak Contribution Node and The Weak Contribution (ODEs and DAEs) Node.
- **Weak Constraint**. See The Weak Constraint Node.
- **Pointwise Constraint**. See The Pointwise Constraint Node and Using Weak Constraints.
- **Weak Contribution on Mesh Boundaries.** This feature is similar to Weak Contribution but is active on mesh boundaries. See The Weak Contribution Node and The Weak Contribution (ODEs and DAEs) Node.

From the **Global** menu add these nodes:

- **Global Equations.** See The Global Equations Node and Adding ODEs, DAEs, and Other Global Equations.
- **Global Constraint**. See The Global Constraint Node, Using Weak Constraints, Bidirectional and Unidirectional Constraints, and Constraint.
- **Weak Contribution**. See The Weak Contribution Node and The Weak Contribution (ODEs and DAEs) Node.
- **Discretization**. See The Discretization Node and Element Types and Discretization.

Note

Clicking the **Show** button ( ) and selecting **Advanced Physics Options** also displays the **Advanced** section on physics interface settings windows, which is described in Show Advanced Physics Options.

Tip

These same settings are also activated from the **Preferences** dialog box. See Editing Preferences Settings.

Show Advanced Study Options

Under the **Study** node, click the **Show** button ( ) and select **Advanced Study Options** to enable this feature and add these nodes to the **Model Builder**:

- **Job Configurations**. See Generating Solver Configurations and Job Configurations in this guide and Job Configurations in the *COMSOL Multiphysics Reference Guide*.
- **Solver Configurations**. See Generating Solver Configurations and Job Configurations in this guide and Solver Configurations in the *COMSOL Multiphysics Reference Guide*.

Note

Both the **Job Configurations** and **Solver Configurations** nodes also display if they have content.

Click the **Show** button ( ), select **Advanced Study Options** and then right-click the **Study** node to enable these options from the context menu:

- **Cluster Computing**. See Cluster Computing in the *COMSOL Multiphysics Reference Guide*.
- **Cluster Sweep**. See Cluster Computing in the *COMSOL Multiphysics Reference Guide*.
- **Batch Sweep**. See Cluster Computing in the *COMSOL Multiphysics Reference Guide*.
- **Batch**. See Batch in the *COMSOL Multiphysics Reference Guide*.

Click the **Show** button ( ), select **Advanced Study Options** and then right-click the **Solver Configurations** node to enable this option from the context menu:

- **Create Custom Solver**. See The Solver Configurations Node in the *COMSOL Multiphysics Reference Guide*.

Click the **Show** button ( ), select **Advanced Study Options** and then right-click any of the **Study Step** nodes to enable this option from the context menu:

- **Multigrid Level**. See Multigrid Level in the *COMSOL Multiphysics Reference Guide*.

Show Advanced Results Options

To display the **Views** node under **Results**, click the **Show** button ( ) and select **Advanced Results Options**.

See Also

- User-Defined Views
- Processing and Analyzing Results

The Physics Feature Nodes

An important part of building a model is where you add **Physics** branches. These contain the nodes that define the material properties, equations, loads, initial values, boundary conditions, and other part of the physics that the model describes. This section describes some common behavior for these nodes and includes these topics:

- Specifying Physics Settings
- Physics Node Context Menu Layout
- Physics Nodes by Space Dimension
- Physics Exclusive and Contributing Feature Node Types
- Physics Default Feature Nodes
- Physics Node Status
- Physics Boundary Types
- Continuity on Interior Boundaries
- Physics Axial Symmetry Node
- Specifying Boundary Conditions for Identity Pairs
- The Equation View Node

Specifying Physics Settings

Each physics interface provide features for specifying all input data for a specific physics in a model:

- Material properties. See also Materials.
- Boundary and interface conditions
- Equations (for equation-based modeling)
- Initial conditions

In addition, specify weak form contributions and element types for additional flexibility.

Specifically, the settings are available on the following parts of the geometry:

- Domains
- Boundaries

- Edges
- Points
- Additional properties that are independent of the geometry

Not all of these options are available for all geometry types and physics interfaces.

All settings windows for the specification of the physics and equations accept parameters and variables as input data (see).

Physics Node Context Menu Layout

The context menu opens when you right-click a physics interface node. This menu is divided into four sections for most physics interfaces: the first section contains domain settings, the second boundary settings, the third edge settings, and the fourth has point settings.

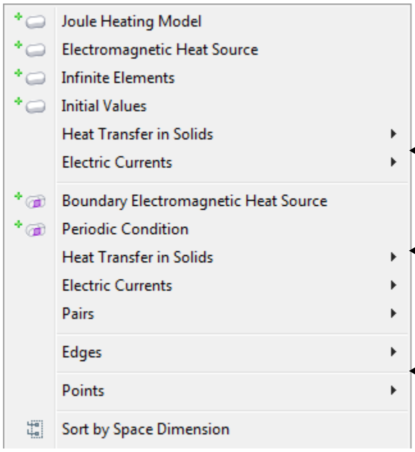
Important

It is important to remember this division as sometimes there are menu items with the same name but applied at different geometric entity levels.

As shown in Figure 7-6, the context menu layout is also based on whether the nodes are not sorted (the default) or if **Sort by Space Dimension** is selected.

See Also

Sorting Nodes by Space Dimension and Type



Joule Heating Model
Electromagnetic Heat Source
Infinite Elements
Initial Values
Heat Transfer in Solids
Electric Currents
Boundary Electromagnetic Heat Source
Periodic Condition
Heat Transfer in Solids
Electric Currents
Pairs
Edges
Points
Sort by Space Dimension

The context help menu when the nodes are not sorted by space dimension (the default).

Domain level settings

Boundary level settings

Pairs, Edges, and Points submenus

Figure 7-6: The physics context menu layout for a 3D model with unsorted nodes.

Physics Nodes by Space Dimension

The physics nodes indicate the geometric entity level (domains, boundaries, edges, points, or pairs) based on the space dimension of the model. See Table 7-1 and Figure 7-7. The nodes also correspond to The Graphics Toolbar Buttons, some of which are also based on space dimension. See Physics Exclusive and Contributing Feature Node Types and Physics Node Status for examples of other differences to how the nodes display in the **Model Builder**.

TABLE 7-1: PHYSICS NODES BY SPACE DIMENSION

NODE	NAME
1D and 1D Axisymmetric Models	
—	Domain Level
D—	Domain Level, default node
—■	Boundary Level
D—■	Boundary Level, default node
M—■	Boundary Level, Pairs

TABLE 7-1: PHYSICS NODES BY SPACE DIMENSION

NODE	NAME
2D and 2D Axisymmetric Models	
	Domain Level
D	Domain Level, default node
	Boundary Level
D	Boundary Level, default node
	Boundary Level, Pairs
	Point Level
3D Models	
	Domain Level
D	Domain Level, default node
	Edge Level
	Boundary Level
D	Boundary Level, default node
	Boundary Level, Pairs
	Point Level

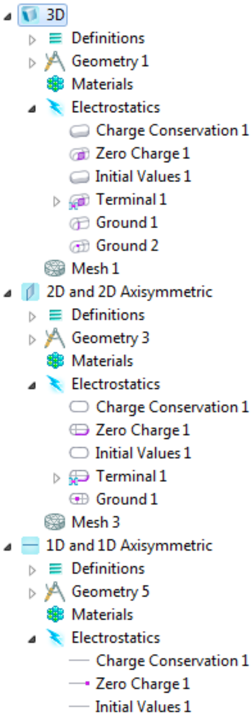
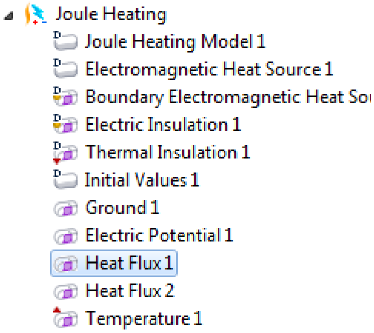


Figure 7-7: An example of the physics nodes listed in Table 7-1 *and as displayed in the Model Builder based on model space dimension.*

Physics Exclusive and Contributing Feature Node Types

The nodes for the physics interfaces are in a sequence, which acts like a macro that the software runs in a top-down order. Depending on the selection for each node, a node can totally or partially override, or shadow, a node earlier in the sequence. How the software treats these nodes depends on their relationship. There are two different types of nodes: exclusive and contributing.



Joule Heating
Joule Heating Model 1
Electromagnetic Heat Source 1
Boundary Electromagnetic Heat Soı
Electric Insulation 1
Thermal Insulation 1
Initial Values 1
Ground 1
Electric Potential 1
Heat Flux 1
Heat Flux 2
Temperature 1

Figure 7-8: An example of exclusive and contributing nodes in a 3D model physics operating sequence and when Ungroup Nodes is selected.

Note	The exclusive and contributing nodes maintain the described behavior only in relation to similar types of nodes within the same physics (for example, you can have a temperature constraint and a pressure constraint for the same boundary in the same model).

Tip	What the node looks like in the **Model Builder** is based on the space dimension. See Physics Nodes by Space Dimension.

EXCLUSIVE NODES

The use of an *exclusive node* means that only one can be active for a given selection. That is, if you add another exclusive node (for example, an identical feature node) with the same selection, the first exclusive node is overridden and thus has no effect.

Typical exclusive nodes include model equations, initial values, and boundary conditions that are constraints, such as prescribed values for displacements, temperatures, pressures, and so on, including special variants of these such as ground conditions in electromagnetics and fixed constraints in structural mechanics. Also some boundary conditions that are not constraints but have a definitive meaning are exclusive nodes—for example, electric insulation, thermal insulation, and no-flow conditions. Depending on the selections for each node, an exclusive node can override another node partially. Nodes are exclusive only within their specific physics. When a node is selected in the Model Builder tree, nodes that are overridden by the selected node are indicated using a red arrow in the lower-left corner of the icon ( ), and nodes that override the selected node are indicated using red arrow in the upper-left corner of the icon ( ).

CONTRIBUTING NODES

A *contributing node* means you can have several of these nodes with the same selection—the software adds these together when evaluating the model. Typical contribution nodes are loads, fluxes, and source terms, where you can have more than one of each type that is active on the same domain or boundary, for example. When a node is selected in the Model Builder tree, COMSOL Multiphysics shows the nodes the current nodes contributes with indicated using a yellow circle to the left of the icon, for example, in this boundary level icon . See also Figure 7-8 for an example.

LISTING OVERRIDES AND CONTRIBUTIONS

If your preferences include showing the **Override and Contribution** section in the settings windows for physics interface feature nodes, you can find the following information about how exclusive and contribution nodes interact in the model. click the **Show** button ( ) and select **Override and Contribution** from the **Model Builder** to display the section. To expand this section on all nodes, click **Expand Sections** ( ) and select **Override and Contribution** from the **Model Builder** and click the physics nodes to display the information as in Figure 7-9 and described below.

- The **Overridden by** list contains the names of the nodes that the selected node are overridden by. The selected node is then overridden by these nodes at least partially, and the **Selection** list contains (overridden) for the geometric entities (boundaries, for example) where it is overridden. The nodes that the selected node is overridden by are indicated using a red arrow in the lower-left corner of the icon, for example, in this boundary level icon .
- The **Overrides** list contains the names of the nodes that the selected node overrides (where the current node is active). The nodes that the selected node overrides are indicated using a red arrow in the upper-left corner of the icon, for example, in this boundary level icon .
- The **Contributes with** list contains the names of the nodes that the selected node contributes with for at least some shared selection. The nodes that the selected node

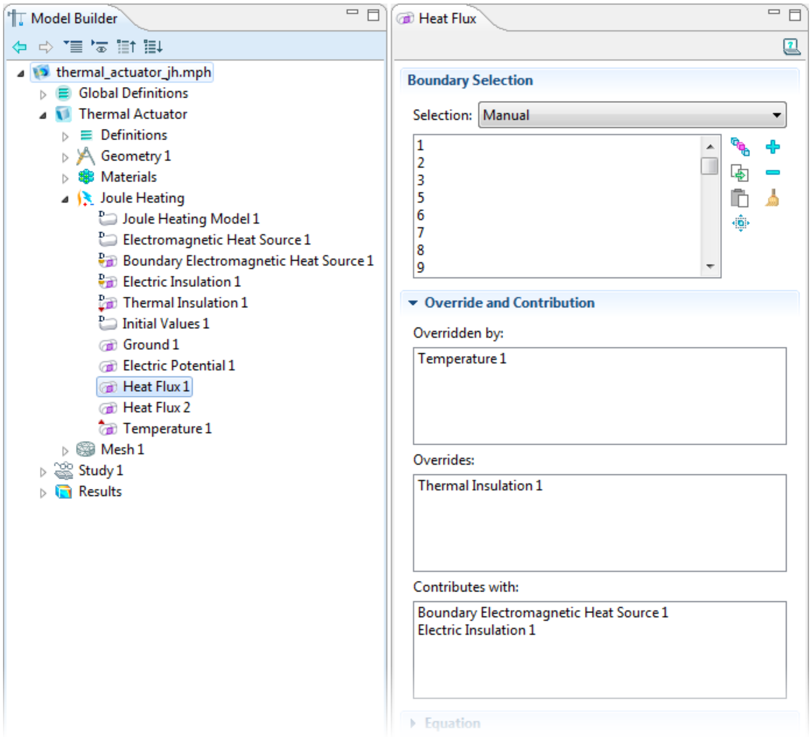
contributes with are indicated using a yellow circle to the left of the icon, for example, in this boundary level icon .

Note

If you disable physics features locally in a study step using the **Physics and Variables Selection** section in the study step's settings window, the indications of overrides and contributions in the Model Builder are unchanged (but disabled physics nodes get an asterisk to indicate that their state has been changed in at least one study step). The local variables and physics tree in the study step's settings window, on the other hand, does display the overrides and contributions taking the disabled nodes into account.

See Also

- Physics Node Status
- Key to the Nodes and Toolbar Buttons
- Physics Default Feature Nodes



Model Builder

- thermal_actuator_jh.mph
  - Global Definitions
  - Thermal Actuator
    - Definitions
    - Geometry 1
    - Materials
    - Joule Heating
      - Joule Heating Model 1
      - Electromagnetic Heat Source 1
      - Boundary Electromagnetic Heat Source 1
      - Electric Insulation 1
      - Thermal Insulation 1
      - Initial Values 1
      - Ground 1
      - Electric Potential 1
      - Heat Flux 1
      - Heat Flux 2
      - Temperature 1
    - Mesh 1
  - Study 1
  - Results

Heat Flux

Boundary Selection

Selection: Manual

1
2
3
5
6
7
8
9

Override and Contribution

Overridden by:

Temperature 1

Overrides:

Thermal Insulation 1

Contributes with:

Boundary Electromagnetic Heat Source 1
Electric Insulation 1

Equation

Figure 7-9: The Override and Contribution section lists other physics nodes that the selected node is overridden by, overrides, or contributes with. This is an example for a 3D model and with Ungroup Nodes selected.

Physics Node Status

The status of a physics interface depends on if it is a default feature node, the selection that it applies to, and other nodes in the same branch that can override nodes earlier in the sequence. You can change the order of nodes other than the default feature nodes by moving them up or down.

OVERRIDDEN SELECTIONS

A node can be partially or completely *overridden* by another node further down in the same branch of the model tree that is of a similar, exclusive type. For example, if you specify a temperature boundary condition on boundary 1 and boundary 3, and then add another temperature boundary condition for boundary 3, the first temperature boundary condition is overridden on boundary 3. In the settings window for the **Temperature** nodes that define the temperature boundary condition, the **Selection** list then shows **3 (overridden)** to indicate that the temperature boundary condition defined on this selection is overridden for boundary 3 but is still active on boundary 1. Deleting or disabling the other temperature boundary condition on boundary 3 reactivates the original temperature boundary condition, and then shows **3** (without the **(overridden)** indication).

SELECTIONS THAT ARE NOT APPLICABLE

For selections that are not applicable for a feature node (interior boundaries for an boundary condition that is only applicable for exterior boundaries, for example), the **Selection** list then shows **(not applicable)** next to the entries in the list that are, in this case, interior boundaries.

ENABLING AND DISABLING NODES

By enabling or disabling physics nodes, you can activate and inactivate (shadow) other physics interface nodes that appear higher up in the physics inter face branches.

See Also

- Physics Default Feature Nodes
- Physics Exclusive and Contributing Feature Node Types
- Physics Nodes by Space Dimension
- Key to the Nodes and Toolbar Buttons
- Deleting, Disabling, and Enabling Nodes

Physics Default Feature Nodes

When a physics interface is added, it automatically adds a main physics branch, which typically includes a number of default feature nodes, including but not limited to:

- A model equation or material model node, typically on the domain level. This node defines the domain equations (except optional sources, loads, reactions, and similar contributing domain quantities) and the related material properties or coefficients.
- A boundary condition node. For multiphysics interfaces there is one boundary condition for each physics.

- For axisymmetric models, the symmetry axis has an **Axial Symmetry** boundary condition (see Physics Axial Symmetry Node).
- An **Initial Values** node for specifying initial values for a time dependent simulation or an initial guess for the solution to a nonlinear model.

The default nodes' initial selections include all domains or all boundaries (or all instances of another geometric entity level). Their selection is always every instance that has no selection defined by another node on the same geometric entity level that is not contributing to the default node. All default nodes include a *D* (for "default") in the upper-left corner (for example,) to indicate their special status. It is not possible to delete the default nodes, but you can copy and duplicate default nodes. The copy or duplicate is a node of the same type as the default node but is a normal feature node with an initially empty selection.

For example, for a geometry with four boundaries, the default boundary condition's initial selection includes all four boundaries. If another exclusive boundary condition for Boundary 3 is added, that boundary becomes overridden (inactive) in the default boundary condition's selection. If you disable or remove that boundary condition, the default boundary condition becomes active for Boundary 3 again. You cannot change a default node's selection.

See Also

- Physics Nodes by Space Dimension and Table 7-1
- Deleting, Disabling, and Enabling Nodes
- Key to the Nodes and Toolbar Buttons
- Physics Node Context Menu Layout

Note

Some physics interfaces also add standard feature nodes directly when you add them to a model. They represent functionality that is likely to be useful throughout the model but that you might want to make active on a part of the geometry only or delete. Such nodes do not include a *D* in the upper-left corner.

Physics Boundary Types

There are different types of boundaries for the physics, which all support different types of boundary conditions:

- Exterior boundaries, where most boundary conditions are applicable—see below.

- Interior boundaries, where special interface conditions can be applicable—see below.
- Axial symmetry boundaries, which are artificial boundaries representing the symmetry axis in axisymmetric models. See Physics Axial Symmetry Node.

If a selection for a boundary condition node, for example, contains boundaries of a type that is not applicable or supported, the **Selection** list has **(not applicable)** next to those boundary numbers.

INTERIOR AND EXTERIOR BOUNDARIES

When specifying boundary and interface conditions, COMSOL Multiphysics differentiates between exterior and interior boundaries:

- An *exterior boundary* is an outer boundary of the modeling domain.
- A *interior boundary* is a dividing interface between two domains in the modeling domain.

If an equation or physics interface is deactivated in one domain, the interior boundary between the active and inactive domain becomes an exterior boundary for its variables because it then borders on the outside of the active domain for those fields. The boundaries of the inactive domain are then void.

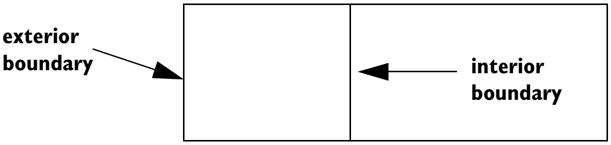


Figure 7-10: Examples of exterior and interior boundaries.

See Also

- Physics Nodes by Space Dimension
- Key to the Nodes and Toolbar Buttons

Continuity on Interior Boundaries

Unless a specific boundary condition is specified on interior boundaries (such as a contact resistance condition), COMSOL ensures continuity in the field variables across interior boundaries. For assembly geometries with identity pairs, select a **Continuity** node on the **Pairs** menu in the boundary part of the context menu for most physics interfaces. The **Continuity** condition is only suitable for pairs where the boundaries match.

BOUNDARY SELECTION

The selection list in this section shows the boundaries for the selected pairs.

PAIR SELECTION

Select the pairs where you want to impose continuity across the pair boundaries. Select the pairs from the **Pairs** list (Ctrl-click to deselect).

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Physics Axial Symmetry Node

In axisymmetric models, boundaries on the symmetry axis are boundaries where only a condition for the axial symmetry exists. COMSOL Multiphysics adds a default **Axial Symmetry** node that is active on all boundaries on the symmetry axis. The condition on the symmetry axis is typically a zero Neumann or no-flux condition.

BOUNDARY SELECTION

The selection list for boundaries is not available because this is a default boundary condition. In the list, boundaries that are not on the symmetry axis have **(not applicable)** added after the boundary number.

See Also

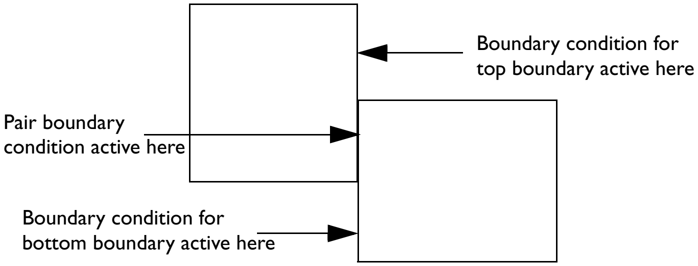
- Physics Nodes by Space Dimension
- Key to the Nodes and Toolbar Buttons

Specifying Boundary Conditions for Identity Pairs

The identity pairs connect parts of an assembly. You can specify boundary conditions for these pairs from the **Pairs** submenu at the bottom of the boundary condition part of the context menu for the physics. For physics defined on boundaries, a **Pairs** submenu is available for conditions on edges (in 3D) or points (in 2D). The **Pairs** submenu includes conditions suitable for pair boundaries, including **Continuity**, which provides continuity in the physics field across the pair boundary (continuity is the default behavior for interior boundaries in geometries formed as a union). The nodes in the **Model Builder** that represent pair boundary conditions use an icon with a pair symbol in the lower-left corner: .

BOUNDARY CONDITIONS ON NON-OVERLAPPING PARTS

For identity pairs where parts of the boundaries do not overlap you need to specify boundary conditions for the non-overlapping parts, which typically represent exterior boundaries outside of the overlapping area. These boundary conditions appear as subnodes to the pair's boundary condition node in the **Model Builder**. By default, the default boundary condition for exterior boundaries apply to the non-overlapping parts. If you want to use another boundary condition for any of the non-overlapping parts, right-click the pair's boundary condition node and select any of the standard boundary conditions from the context menu. In the settings window, the selection include all applicable boundaries by default, but a separate boundary condition can be added for only a subset of the pair boundaries. The boundary conditions added as subnodes to the pair's boundary condition apply on the non-overlapping parts only. In the following illustration, which shows a simple example with two partially overlapping rectangles, there is one identity pair that consists of two boundaries, each with a non-overlapping part. Then right-click the pair's boundary condition node and add one boundary condition for the top boundary and another boundary condition for the bottom boundary if desired.



See Also

- Physics Nodes by Space Dimension
- Key to the Nodes and Toolbar Buttons

The Equation View Node

Equation View ($\frac{\partial u}{\partial t}$:f) is a subnode available for all the physics nodes. To display these subnodes, click the **Show** button (👁) and select **Equation View** from the **Model Builder**.

The **Equation View** settings window contains detailed information about the implementation of the physics feature: variables, shape functions, weak-form equation expressions, and constraints.

To update the values in the **Equation View** node's settings window to reflect the latest changes in the physics feature, click the **Refresh Equations** button (⟳) in the settings window's toolbar.

⚠ *Caution*

Editing the predefined expressions for variables, equations, and constraints means that the equations are altered and that COMSOL Multiphysics solves for the values of variables.

Edit the values of variables, weak-form expressions, and constraints in the corresponding tables. This makes it possible to introduce custom changes to the equations and variable definitions.

For a changed definition of a variable or a change to a weak-form expression or constraint, a warning icon (⚠) appears in the leftmost column, and a small padlock is added to the lower-right corner of the icon for the physics feature node where you have made modifications in its equation view. To restore only the change in the selected variable, weak-form expression, or constraint, click the **Reset Selected** button (↺) under the table in the **Variables**, **Weak Expression**, or **Constraints** section. To reset all changes in the equation view, click the **Reset All** button (↺) in the settings window's toolbar. If no changes remain, the padlock disappears from the corresponding physics feature node.

See Also

For information about the Equation displays available, see Physics Nodes—Equation Section.

VARIABLES

This section has a table with the variables that the physics feature defines. The table includes these columns:

- **Name**: the name of the variable
- **Expression**: the expression, using COMSOL Multiphysics syntax, that defines the variables
- **Unit**: the unit for the variable (in the active unit system)
- **Description**: a description of the variable
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the variable is defined (**Domain 1**, for example)
- **Details**: this column contains some details about the variable's behavior. See About the Details Column below.

SHAPE FUNCTIONS

This section has a table with the dependent variables that the physics feature defines and their shape functions. This is primarily applicable to equation model features like the main material nodes; for most physics features such as boundary conditions, the table is empty. The table has these columns:

- **Name**: the name of the variable.
- **Shape function**: the type of shape function (element) for the variable (for example, **Lagrange** for Lagrange elements, which are the most common elements).

See Also

Selecting an Element Type

- **Unit**: the unit for the variable (in the active unit system).

- **Description**: a description of the variable.
- **Shape frame**: the frame type (typically either a spatial or a material frame) for the shape function.
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the shape function is defined (**Domain 1**, for example)
- **Details**: This column contains some details about the shape function's behavior. See About the Details Column below.

WEAK EXPRESSIONS

This section has a table with the weak-formulation equation contributions that the physics feature generates. The table consist of the following columns:

- **Weak expression**, the equation expressed in a weak formulation
- **Integration frame**, the frame type (typically either a spatial or a material frame) used when integrating the expression.
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the weak expression is defined (**Domain 1**, for example)

Each equation contribution appears on its own row under **Weak expression**, but the order is not significant.

Note

The PDE interfaces and the ODEs an DAEs interface do not display any weak expressions. They are either implemented using strong formulations, directly display the weak formulation, or define equations discretized in the time domain only.

CONSTRAINTS

This section has a table with the constraints that the physics feature generates. This is typically the case for boundary conditions of constraint types, such as prescribed displacements, temperature, or velocities. Many other physics features do not generate any constraints, and the table is then empty. The table consists of the following columns:

- **Constraint**: the expression for the constraint
- **Constraint force**: the expression that defines the associated constraint force, which is typically the test function of the constraint

- **Shape function**: the type of shape function (element) for the constraint (for example, **Lagrange** for Lagrange elements)
- **Selection**: the geometric entities (domains, boundaries, edges, or points) where the constraint is defined (**Boundaries 1–5**, for example)

ABOUT THE DETAILS COLUMN

The **Details** column shows some details about the behavior of variables and shape functions. For variables:

- An empty cell indicates that overlapping contributions are overridden.
- **+ operation** indicates that overlapping contributions are added
- In rare cases, other operations (*** operation**, for example) can occur

For shape functions:

- **Slit** means that the shape function creates a slit for the degree of freedom

See Also

- Physics Nodes by Space Dimension
- Key to the Nodes and Toolbar Buttons

Specifying Model Equation Settings

The model equation settings appear in the main feature for a physics interface and describe the physics on the domains. Use different values for each domain using settings of the following types:

- Coefficients that define the PDE on the domain. The PDE coefficients are available for the mathematics (equation-based) interfaces.
- Material properties that define the physics in the domain. These properties are available in the physics interfaces.

In this section:

- Specifying Initial Values
- Modeling Anisotropic Materials

Specifying Initial Values

An **Initial Values** node is added by default to each physics interface.

In some types of analyses you must provide initial values:

- As the initial condition for a time-dependent analysis
- As an initial guess for the nonlinear stationary solver
- As a linearization (equilibrium) point when solving a linearized stationary model or when performing an eigenvalue study.

To enter initial values, in the **Model Builder**, click the **Initial Values** node under a physics interface node. In the settings window, enter the **Initial Values** for all dependent variables (fields) in the physics interface. The default initial values are zero.

For some physics interfaces you can also enter initial values for the first time derivative of the dependent variables. These are used when solving time dependent problems containing second time derivatives (wave-type applications). Like other default settings, these initial values apply to all domains where you have not specified any other values.

If you want to use different initial values in different domains, right-click the main physics node and select **Initial Values** to create additional nodes as needed.

Note

By default, the **Override and Contribution** section is displayed on the **Initial Values** settings window. Also click the **Show** button ( ) and select **Equation View** from the **Model Builder** to display the **Equation View** node under the node in the **Model Builder.** See Showing and Expanding Advanced Physics Sections for more information.

See Also

For more information about using initial conditions, see Dependent Variables in the *COMSOL Multiphysics Reference Guide*.

Modeling Anisotropic Materials

The *c* coefficient in the coefficient form PDE corresponds to physical properties that can be anisotropic such as:

- Diffusion coefficient
- Permittivity
- Thermal conductivity
- Electrical conductivity

For an anisotropic material, the *c* coefficient is a tensor with at most four components in 2D and at most nine components in 3D:

$$c = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

When specify and of the *c* coefficient components—as well as for certain material properties in some of the other physics interfaces—you can choose from the following types from the list underneath the fields:

- **Isotropic** (the default)—enter only one value.
- **Diagonal**—enter the diagonal components for an anisotropic material with the main axes aligned with the model’s coordinate system.

- **Symmetric**—enter a symmetric matrix using the diagonal components and the upper off-diagonal components.
- **Anisotropic**—enter the full 2-by-2 (2D) or 3-by-3 (3D) matrix for an anisotropic material.

Periodic Boundary Conditions

In this section:

- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Using Periodic Boundary Conditions

Use *periodic boundary conditions* to make the solution equal on two different (but usually equally shaped) boundaries.

To add a periodic boundary condition, in the **Model Builder**, right-click a physics interface node and select **Periodic Condition**. The periodic boundary condition typically implements standard periodicity (that is, the value of the solution is the same on the periodic boundaries), but in most cases you can also choose antiperiodicity so that the solutions have opposing signs. For fluid flow interfaces, the **Periodic Flow Condition** provides a similar periodic boundary condition but without a selection of periodicity. Typically, the periodic boundary conditions determines the source and destination boundaries automatically (an displays them, under **Model>Definitions**, in an **Explicit** selection node ( ), which is "read only"), but you can also define feasible destination boundaries manually by adding a **Destination Selection** subnode.

Note

For some physics interfaces you can choose the direction in which you want a periodic boundary condition. For a description of the standard periodic boundary condition, which most physics interfaces use, see Periodic Condition.

Model

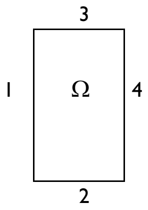
This model makes use of periodic boundary conditions: The KdV Equation and Solitons: Model Library path **COMSOL_Multiphysics/Equation-Based_Models/kdv_equation**

Periodic Boundary Condition Example

MODEL DEFINITION

Consider an eigenvalue equation on a rectangle of dimension $1\times\sqrt{\pi}$.

$$\begin{array}{ll} -\Delta u = \lambda u & \text{on } \Omega \\ u(x_1, y) = u(x_4, y) & \text{on Boundaries 1 and 4} \\ u(x, y_2) = u(x, y_3) & \text{on Boundaries 2 and 3} \end{array}$$



RESULTS AND DISCUSSION

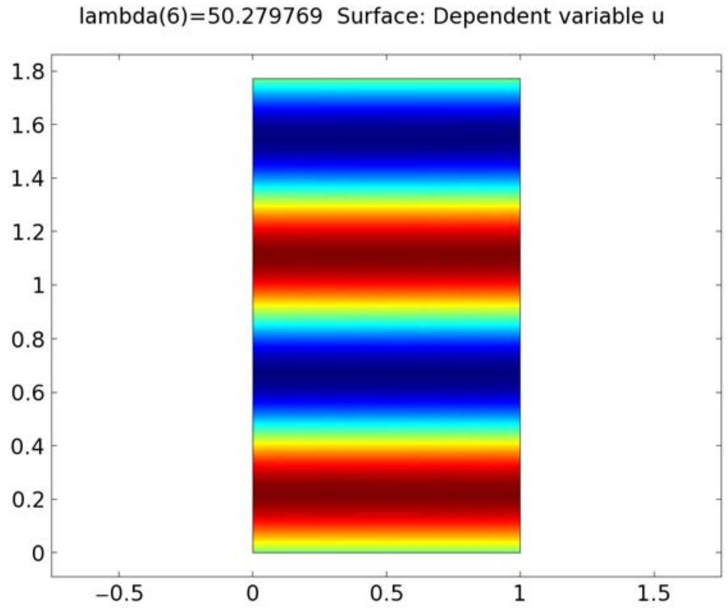


Figure 7-11: The periodic pattern in the solution (6th eigenmode).

The wavelength of the sinusoidal solutions is the rectangle side length divided by a series of integers, each of which corresponds to an eigensolution. The locations of the maxima are random because the model does not constrain the absolute value of the solution anywhere.

MODELING INSTRUCTIONS

Model Wizard

1 Start COMSOL Multiphysics.

2 In the **Model Wizard**, select **2D** from the **Space dimension** list. Click the **Next** button (⇨).

3 From the list of physics interfaces on the **Add Physics** page, select **Mathematics>PDE Interfaces>Coefficient Form PDE (c)**. Click the **Next** button (⇨).

4 On the **Select Study Page**, select **Preset Studies>Eigenvalue**.

5 Click the **Finish** button (🏁).

Geometry Modeling

1 Right-click **Geometry 1** and select **Rectangle** (▭).

2 Type `sqrt(pi)` in the **Height** field.

3 Click the **Build All** button (▦) in the settings window toolbar.

PDE Settings

The default PDE coefficient values are correct.

Periodic Boundary Conditions

1 Right-click **PDE** and select **Periodic Condition**.

2 Select boundaries 1 and 4 in the **Graphics** window and click the **Add to Selection** button (✚).

3 Right-click **PDE** and select **Periodic Condition**.

4 Select boundaries 2 and 3 in the **Graphics** window and click the **Add to Selection** button (✚).

In this case, COMSOL Multiphysics automatically identifies one boundary as the source and the other as the destination. Two **Explicit** selection nodes (⧉) contain the destination boundaries for the two periodic boundary condition.

Mesh Generation

1 Right-click **Mesh 1** and select **Free Triangular** (◭).

2 Click the **Build All** button (▦) in the settings window toolbar.

Computing the Solution

The eigenvalue algorithm does not work reliably when the parameter specified in the **Search for eigenvalues about** field is equal to an eigenvalue. This particular problem has a zero eigenvalue, so the value of this parameter must be changed.

1. Select **Study 1>Eigenvalue 1** in the Model Tree.
2. In the settings window, locate the **Study Settings** section.
3. Type `10` in the **Search for eigenvalues around** field.
4. Right-click **Study 1** and select **Compute** to compute the solution.

Analysis and Results

By default COMSOL Multiphysics plots the first eigenmode, which in this case is a flat solution associated with a zero eigenvalue.

1. Select **2D Plot Group 1** under **Results** in the Model Tree.
2. In the settings window, locate the **Data** section.
3. Look at the solution for different eigenvalues by making a corresponding selection in the **Eigenvalue** list. Start with the last eigenvalue.
4. Click the **Plot** button ( ) in the settings window toolbar to plot each eigenvalue

Calculating Accurate Fluxes

COMSOL Multiphysics provides two ways of calculating accurate reaction forces and fluxes:

- The first approach involves the reaction force operator (`reacf`) that makes it possible to compute integrals of reaction forces or fluxes during analysis. Use the reaction force operator when calculating integrals of reaction forces or fluxes during analysis. See The Reacf Operator for details.
- The second, more general approach for calculating reaction forces and fluxes is to use weak constraints. Use this approach when you need reaction forces or fluxes in other contexts than calculating integrals of reaction forces or fluxes.

See Also

Find further information on the use of weak constraints in the section Using Weak Constraints.

When using weak constraints in physics interfaces, the Lagrange multipliers are additional dependent variables in those physics interfaces. When using the reaction force operator, the reaction force operator of a certain dependent variable corresponds to the Lagrange multiplier of that dependent variable. The Lagrange multipliers correspond to the following quantities (but with the reversed sign) in the physics interfaces:

TABLE 7-2: PHYSICS INTERFACE INTERPRETATION OF LAGRANGE MULTIPLIERS

PHYSICS INTERFACE	QUANTITY
Electrostatics	Surface charge density
Magnetic Fields	Surface current
Electric Currents	Current flux
Heat Transfer	Heat flux
Transport of Diluted Species	Flux
Solid Mechanics	Force
Laminar Flow	Total force per area

COMSOL Multiphysics computes only the part of the boundary flux captured by the Lagrange multiplier. You might have additional flux coming from boundary sources or nonidentity constraint matrices. This should not happen in the physics interfaces, though.

Note

Lagrange multipliers in axial symmetry are not equal to the reaction flux per area but rather per length and full revolution.

See Also

See the following examples:

- Flux Computation Example—Equation-Based Model
- Flux Calculation Example—Heat Transfer Model

Flux Computation Example—Equation-Based Model

Consider calculating the flux over the constrained boundary of a rectangle. The PDE of the rectangle is Poisson's equation, $\Delta u = 1$, with $u = 0$ on the boundary. This example computes the flux using an expression, a reaction force operator, and a Lagrange multiplier.

MODEL WIZARD

1 On the **Select Space Dimension** page, click the **2D** button. Then click the **Next** button.

2 In the list of physics interface, open **Mathematics>PDE Interfaces** and then select **Coefficient Form PDE (c)**. Then click the **Next** button.

3 From the **Studies** list, select **Preset Studies>Stationary**.

4 Click **Finish**.

GEOMETRY MODELING

Draw a single rectangle of any size.

COEFFICIENT FORM PDE

The default equation is Poisson' equation, but the default boundary condition is a zero flux (homogeneous Neumann boundary condition). Instead, add a homogeneous Dirichlet boundary condition: $u = 0$:

1 In the **Model Builder** window, right-click the **PDE (c)** node and select **Dirichlet Boundary Condition**.

2 In the **Dirichlet Boundary Condition** node's settings window, select **All boundaries** from the **Selection** list in the **Boundary Selection** section.

COMPUTING THE SOLUTION

In the **Model Builder** window, right-click **Study 1** and choose **Compute**.

RESULTS—FLUX EXPRESSION

1 In the **Model Builder** window, choose **Results>Derived Values>Surface Integration**.

2 In the **Surface Integration** node's settings window, type `1` in the **Expression** field.

3 Select domain 1 and click the **Evaluate** button (☰) at the top of the settings window. The rectangle's area appears in the **Results** window.

4 Choose **Results>Derived Values>Line Integration**.

5 In the **Line Integration** node's settings window, type `nx*ux+ny*uy` in the **Expression** field.

6 Select all boundaries and click the **Evaluate** button (☰) at the top of the settings window. COMSOL Multiphysics computes the flux over the boundaries as the normal component of the gradient of u.

The influx is approximately equal to the integral of the negative source term in Poisson's equation.

RESULTS—REACTION FORCE OPERATOR

1 In the **Line Integration** node's settings window, type `reacf(u)` in the **Expression** field.

2 Select all boundaries and click the **Evaluate** button (☰) at the top of the settings window. COMSOL Multiphysics computes the flux over the boundaries using the reaction force operator for the solution u. The computed value is much closer to the expected value. To see the value as a double floating-point number with full precision, click the **Full Precision** button in the **Results** window's toolbar.

COEFFICIENT FORM PDE—LAGRANGE MULTIPLIER

1 Under **Physics**, select **Dirichlet Boundary Condition 1**. Clear the **Prescribed value of u** check box to remove this constraint.

2 Right-click **PDE (c)** and choose **More>Weak Constraint** on the boundary level.

3 In the **Weak Constraint** node's settings window, select **All boundaries** from the **Selection** list.

4 In the **Weak Constraint** section, type `u` in the **Constraint expression** field to constrain *u* to 0 at the boundary.

5 Expand the **Lagrange Multiplier** section and replace `lm` with `flux` in the **Lagrange multiplier variable** field.

COMPUTING THE SOLUTION—LAGRANGE MULTIPLIER

Right-click **Study 1** and choose **Compute**.

RESULTS—LAGRANGE MULTIPLIER

1 In the **Line Integration** node's settings window, type `flux` in the **Expression** field.

2 Select all boundaries and click the **Evaluate** button (≡) at the top of the settings window. COMSOL Multiphysics computes the flux over the boundaries using the Lagrange multiplies variable. The integral of the flux exactly equals the integral of the finite-element projection of the source term in Poisson's equation.

Flux Calculation Example—Heat Transfer Model

Consider a heat transfer model where a heat flux of 1 W/m^2 flows in through one boundary of a square 2D region. All other boundaries are kept at a fixed temperature of 293.15 K. The material is copper. This example verifies that the flux is conserved exactly using a Lagrange multiplier for computing the total flux over the bondaries with a fixed temperature.

MODEL WIZARD

1 On the **Select Space Dimension** page, click the **2D** button. Then click the **Next** button.

2 In the list of physics interface, select open **Heat Transfer>Heat Transfer in Solids**. Then click the **Next** button.

3 From the **Studies** list, select **Preset Studies>Stationary**.

4 Click **Finish**.

GEOMETRY MODELING

Draw a unit square (1-by-1 m).

MATERIALS

In the **Material Browser**, locate **Built-in>Copper** and then right-click it and select **Add Material to Model**.

HEAT TRANSFER

The **Heat Transfer in Solids** node defines the material properties to be those from the material (copper) and does not need to be changed, but the default boundary condition is thermal insulation. Instead, add a heat flux to the bottom boundary and a fixed temperature on the other three boundaries.

1 In the **Model Builder** window, right-click **Heat Transfer** node and select **Heat Flux**.

2 Click boundary 2 (the bottom boundary) and then right-click it to confirm the selection

3 In the **Heat Flux** node's settings window, enter `1` in the **General inward heat flux** field for q_0.

4 In the **Model Builder** window, right-click **Heat Transfer** node and select **Temperature**.

5 Select the other three boundaries (1, 3, and 4) and add them to the selection for the temperature condition.

6 To display the weak constraint option to add the Lagrange multipliers, click the **Show** button ( ) and select **Advanced Physics Options**. In the **Temperature** node's settings window, keep the default value for the temperature, 293.15 K, but select the **Use weak constraints** check box in the **Constraint Settings** section. This adds a Lagrange multiplier for the heat flux as an extra variable to compute.

COMPUTING THE SOLUTION

In the **Model Builder** window, right-click **Study 1** and choose **Compute**. The resulting plot shows the temperature distribution in the domain.

RESULTS—FLUX EXPRESSION AND LAGRANGE MULTIPLIER

1 In the **Model Builder** window, choose **Results>Derived Values>Line Integration**.

2 In the **Line Integration** node's settings window, select the three boundaries with a fixed temperature (1, 3, and 4).

3 Click the **Replace Expression** button ( ) and select **Heat Transfer>Total heat flux magnitude** (the variable `ht.tfluxMag`).

4 Click the **Evaluate** button ( ) at the top of the settings window. The total heat flux across these boundaries appears in the **Results** window under **Total heat flux magnitude** and is about 0.986, which is close to but not exactly equal to the influx of 1. This value approaches 1 if you refine the mesh.

5 Click the **Replace Expression** button ( ) and select **Heat Transfer>Lagrange multiplier for temperature** (the variable `T_lm`).

6 Click the **Evaluate** button (≡) at the top of the settings window. The total heat flux across these boundaries appears in the **Results** window under **Lagrange multiplier for temperature** and is 1—exactly equal to the influx without the need for a computationally expensive extremely fine mesh.

Using Load Cases

For linear stationary problems it can be of interest to see the solution for several different loads $\boldsymbol{F}$ (right-hand side of the basic PDE) on the same structure (a model where the geometry and materials are defined and do not change). Typically this is used for studies using linear combinations of different loads—*load cases*. It is then possible to solve for these load cases in a computationally efficient way because there is no need to reassemble the stiffness matrix. Varying constraints can also be part of a general load case definition, and COMSOL Multiphysics supports load cases that are combination of loads, with optional weights, and constraints.

In this section:

- Defining Load Groups and Constraint Groups
- Defining and Evaluating Load Cases

See Also

- Load Group
- Constraint Group

Defining Load Groups and Constraint Groups

For boundary conditions that represent loads and constraints, as well as other loads and constraints such as body loads, you can define *load groups* and *constraint groups*, which contain the loads and constraints, respectively, that you want to use as parts of load cases. All loads and constraints for solid mechanics support load groups and constraint groups. You can create load groups and constraint groups in two ways:

- Add a a **Load Group** ( ) or **Constraint Group** ( ) under **Global Definitions** to create groups to which you can later assign loads and constraints. If you sort the nodes, the load and constraint groups appear under the **Load and Constraint Groups** node ( )
- Right-click the physics interface node for a load or constraint and choose **Load Group>New Load Group** or **Constraint Group>New Constraint Group**, respectively. In this case, the software creates a load group or constraint group and at the same time assigns that physics interface feature (a load or a constraint) to that group.

To assign a load or a constraint to a load group or constraint group, right-click the physics interface node for a load or constraint and from the **Load Group** or **Constraint Group** submenu choose one of the following:

- **Active in All Load Groups** (or **Active in All Constraint Groups**). This is the default setting, which you can use for some boundary conditions or other parts of the physics that take part in all load cases.
- One of the defined load groups or constraint groups such as **Load Group 1**, **Load Group 2**, and so on.
- **New Load Group** (or **New Constraint Group**) to create a new group as described earlier in this section.

Define as many groups as you need for the load cases that you want to study. Each load or constraint can only belong to one group. The next step is then to define the actual load cases as combinations of these groups.

Model

- Tapered Cantilever with Two Load Cases: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/tapered_cantilever**
- If you have the Structural Mechanics Module, also see Pratt Truss Bridge: Model Library path **Structural_Mechanics_Module/Civil_Engineering/pratt_truss_bridge**

Defining and Evaluating Load Cases

You define load cases in the **Stationary** study step's settings window. Follow these steps to create load cases:

1. In the **Stationary** study step's settings window, go to the **Study Extensions** section.
2. Select the **Define load cases** check box.
3. In the **Define load cases** area, click the **Add** (+) button underneath the table to add a load case.
4. The added load case appears last in the table of load cases. Use the **Move Up** (⇧), **Move Down** (⇩), and **Delete** (✖) buttons to rearrange the load cases in the table, and click the **Add** (+) button to add more load cases.
5. For each load case you can change its name from the default (Load case 2, for example) in the **Load case** column.

6 Include the load groups and constraint groups for each load case by clicking the ✗ in the columns for the groups to include. The symbol then changes to ✓ to indicate that the group participates in the load case.

7 For load groups, optionally change the weight from its default value of 1.0 in the corresponding **Weight** column (the **Weight** column to the right of the load group that it affects). Use a positive value other than 1 to increase or decrease the magnitude of the load; a negative value also reverses the loads direction.

When you have defined all load cases you can compute the solution. COMSOL Multiphysics then solves for all load cases directly. In the plot groups that are created, a **Load case** list in the **Data** section of the **Plot Group** nodes' settings windows contains all load cases. To plot using the solution for a specific load case, select the load case of interest from the **Load case** list, and then click **Plot** ().

Using Units

COMSOL Multiphysics supports a number of consistent unit systems, including the SI unit system, which is the default unit system. The user interface displays the unit for the physical quantities entered in the selected unit system, but by Using Standard Unit Prefixes and Syntax you can use any available and applicable unit or SI prefix to define your input quantities. In addition to SI units, many English units and units from the CGS (or cgs) system are also available, regardless of the unit system used in the model. All data in the material databases and Material Library product use SI units with declared units using the unit syntax (see Materials). This makes it possible to use the material data also in models with non-SI unit systems. Regardless of the selected unit system, you can always choose from a list of applicable units for plotting and results evaluation.

In this section:

- Unit Systems in COMSOL Multiphysics, with links to the different unit systems
- Selecting a Unit System
- Using Standard Unit Prefixes and Syntax
- Switching Unit System
- About Temperature Units
- About Editing Geometry Length and Angular Units
- Indication of Unexpected, Unknown, or Inconsistent Units
- Units and Space Dimensions

In the tables with units, “N/A” means that a symbol for the unit is not available.

Unit Systems in COMSOL Multiphysics

COMSOL Multiphysics supports the following unit systems:

METRIC UNIT SYSTEMS

- SI units, the International System of Units (SI, Système International d’Unités). This is the default unit system (sometimes also called MKS). For a list of SI units in COMSOL Multiphysics, see SI Base, Derived, and Other Units.
- CGSA units. The CGS system uses centimeter, gram, and second as basic units of length, mass, and time, respectively. The remaining basic units are identical to the

SI units. The CGS unit system gives nice values for small lengths, masses, forces, pressures, and energies when working on a microscale and with weak electromagnetic forces. The derived units of force, pressure, and energy have well-known and widely used names: dyne, barye, and erg, respectively. CGSA adds *ampere* as the basic unit for electric current. For a list of the CGSA units, see Special CGSA Units.

- Electromagnetic units (EMU). This system is based on Ampère's law, which defines the unit of electric current once you select an appropriate value for the constant C. When dealing exclusively with magnetic effects, it is convenient to set $C = 1$. If CGS units are used for the remaining basic dimensions, the current unit is called an *abampere*, and the corresponding coherent unit system is called electromagnetic units. Unique names for derived units have been introduced by prefixing the SI name with *ab-*. For a list of EMU units, see Special EMU Units.
- Electrostatic units (ESU). Based on Coulomb's law for the force between point charges, ESU uses a unit of charge called the *statcoulomb* with CGS units for length, mass, and time. From there, the *statampere*, or *franklin*, and other derived units of the electrostatic unit system follow. For a list of ESU units, see Special ESU Units.
- MPa units. For stationary structural mechanics, where the density does not appear in the equations, it can be convenient to use a system where the newton and megapascal (hence the name: the MPa system) are naturally derived units of force and pressure. Keeping the SI unit for time, the basic units of length and mass become millimeter and tonne. Except for the force and pressure units, other derived units are nameless. For a list of MPa units, see Special MPa Units.

ENGLISH UNIT SYSTEMS

- Foot-pound-second unit system (FPS units). The original foot-pound-second system seems to be the absolute system using the pound as a unit of mass. This version of the FPS system is in agreement with the IEEE standard (the pound is a unit of mass and not of force). The natural derived unit of force is the *poundal*. For a list of FPS units, see Special FPS Units.
- British engineering units. An alternative to the standard FPS system is the British engineering unit system (also called gravitational foot-pound-second system or foot-slug-second system). Here, the pound force is the natural unit of force, which causes the introduction of the mass unit *slug* such that a pound force is a slug-foot per second squared. For a list of British engineering units, see Special British Engineering Units.

- Inch-pound-second unit system (IPS units). It is possible to define varieties of the FPS and British engineering systems based on the inch instead of the foot as basic unit of length. This gives rise to two distinct inch-pound-second systems: the *absolute IPS system* (just called IPS) and the *gravitational IPS system*. For a list of IPS units, see Special IPS Units.
- Gravitational IPS units. This alternative IPS unit system considers the pound a unit of weight rather than a unit of mass. For a list of Gravitational IPS units, see Special Gravitational IPS Units.

OTHER

- None. No units appear in the settings, which can be useful in non-dimensionalized models.

Selecting a Unit System

SETTING THE UNIT SYSTEM ON THE GLOBAL LEVEL

To set the unit system for the entire MPH-file on a global level:

1. In the **Model Builder**, click the root node (the top node in the model tree). The root node's name is the name of the MPH-file or **Untitled.mph** before you have saved your work.
2. In the root node's settings window, select the unit system from the list in the **Unit System** section or **None** to turn off unit support.

SETTING THE UNIT SYSTEM FOR INDIVIDUAL MODELS

By default, all models in the MPH-file use the same global unit system, but it is possible to use different unit systems in each model. To do so, follow these steps:

1. In the **Model Builder**, click the top node for a model branch (**Model 1**, for example).
2. In the model node's settings window, locate the **Model Settings** section.

3 Select the **Override global system** check box, and then select the unit system from the list of unit systems that becomes available.

Tip

To disable unit support in a model, choose **None** from the list in the **Unit System** section in the root node's settings window and make sure that the corresponding setting for each individual model is **Same as global system**. If unit support is turned off only for some models (or if the settings at global and model level differ for a single-model MPH-file) unexpected side effects may occur.

Using Standard Unit Prefixes and Syntax

STANDARD UNIT PREFIXES

For SI units you can scale data using the standard prefixes for powers of 10—*kilo*, *mega*, *milli*, or *micro*, for example. Either the full prefix or the symbol can be used but you must use the same form for the prefix and the unit—that is, `[milliampere]` and `[mA]` are valid but not `[mampere]` or `[milliA]`). In the settings windows for plotting and numerical results, the **Unit** list contains the SI unit for the quantity, including the most common prefixes. The lists also contain applicable non-SI units, which in a few cases also support these prefixes.

Use Table 7-3 as a guide for the format to enter.

TABLE 7-3: SI PREFIXES

FULL PREFIX	SYMBOL	FACTOR
yotta	Y	10^{24}
zetta	Z	10^{21}
exa	E	10^{18}
peta	P	10^{15}
tera	T	10^{12}
giga	G	10^{9}
mega	M	10^{6}
kilo	k	10^{3}
hekto	h	10^{2}
deca	da	10^{1}
deci	d	10^{-1}

TABLE 7-3: SI PREFIXES

FULL PREFIX	SYMBOL	FACTOR
centi	c	10^{-2}
milli	m	10^{-3}
micro	u	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}
atto	a	10^{-18}
zepto	z	10^{-21}
yocto	y	10^{-24}

STANDARD UNIT SYNTAX

You can use the unit syntax to specify a quantity with any applicable unit. To do so, append the unit to any constant or variable in a model using a syntax where you enclose the unit in brackets, for example, `200[ft]` and `3e6[kg/m^3]`.

Both the name and the symbol can be used for a unit. For example, `2.4[ampere]` and `2.4[A]` are both valid to indicate an electric current in SI units (see SI Base, Derived, and Other Units). The SI units can also contain standard prefixes. Appending a unit means that you *multiply* the constant or variable to the left of the unit declaration with this unit. This multiplication takes precedence over other operators so, for example, `1/2[m]` evaluates to 0.5 m^{-1} (`0.5[1/m]`) whereas both `(1/2)[m]` and `1/2*1[m]` evaluate to 50 cm (`0.5[m]` or `50[cm]`). Also, if `L` is a variable defined as `2[m]`, `L[1/s]` evaluates to `2[m/s]`.

The following examples show how to apply the unit syntax:

- Adding two quantities of the same kind that use different units: `0.5[ft]+33[mm]`. COMSOL Multiphysics converts the result to the SI length unit (meters).
- Using multiplication with a unit to get consistent units for two quantities that you want to add, for example, `14[kg]+ht.rho[m^3]`, which works if `ht.rho` represents the density for a model.

Note

For unit names with spaces and hyphens, such as *British thermal unit* and *pound-force*, only use the symbols when declaring units.

It is possible to add constants (without units) to any quantity. COMSOL Multiphysics then assumes that this value has the unit of the SI unit for that quantity. You can also concatenate several units, for example, `3.6[N][m]`, which is equivalent to typing `3.6[N*m]` and evaluates to `3.6 N·m`.

All data in the material databases and Material Library product use SI units and this unit syntax (see Materials).

DECLARING UNITS FOR PARAMETERS, VARIABLES, AND FUNCTIONS

Important

It is important to be aware of the following aspects of unit handling.

When using parameters, variables, and functions in expressions:

- If parameters or variables are used in the physics features (see), it is good practice to use the unit syntax to define them. The settings windows for parameters and variables display the resulting unit, in SI units, of user-defined parameters and expressions. It is important to verify that the variables have the expected unit before using them in the physics settings. The unit of parameters and variables are otherwise undefined.
- Functions expect dimensionless inputs and outputs, so it is good practice to use make inputs such as time dimensionless using unit syntax. If the input is not dimensionless, COMSOL Multiphysics marks the expression in an orange color and reports an unexpected unit of input. For example, to use the time *t* as input to a Rectangle function `rect1`, use `[1/s]` to make the input dimensionless: `rect1(t[1/s])`.
- Using properties with undefined units in a model does not affect the numerical results during the analysis, but undefined units are required in the results and visualization stages—expressions involving such parameters and variables are also unitless.
- To declare the SI unit for electric resistance, ohm, use `[ohm]`. COMSOL Multiphysics then displays this as `[Ω]`.
- If non-SI units or SI prefixes are used, the conversion to SI units also affects the value (quantity) using a scaling factor (and an offset in the case of temperature units). The **Value** column in a **Parameter** node's settings window displays the quantity and unit in the base SI unit system so that you can see the result of the unit

conversion. For example, a parameter is defined as `3[ft]`, the result in the **Value** column is **0.9144 m**. See .

SI Base, Derived, and Other Units

The SI units form an internationally accepted system with seven units for base quantities and a large number of derived units. Use the symbols for these and other units when declaring units in COMSOL Multiphysics (for example, `10[m/s]` for a velocity).

- Table 7-4 lists the SI units for the seven base quantities.
- Table 7-5 lists the SI derived units supported in COMSOL Multiphysics.
- Table 7-6 lists additional units available in COMSOL regardless of the unit system in the model. If more than one name or symbol is available, use any of them, except when names contain more than one word or a hyphen. See also the tables with special units for other unit systems than the SI system; special units that are not listed in Table 7-6 are only available when using such a non-SI unit system.
- Table 7-7 lists other SI derived units without special names or symbols

TABLE 7-4: BASE SI UNITS

BASE QUANTITY	NAME	SYMBOL
length	meter, metre*	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
temperature	kelvin**	K
amount of substance	mole	mol
luminous intensity	candela	cd

* See About Editing Geometry Length and Angular Units

**See About Temperature Units

TABLE 7-5: SI DERIVED UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
absorbed dose	gray	Gy
capacitance	farad	F
conductance	siemens	S
dose equivalent	sievert	Sv

TABLE 7-5: SI DERIVED UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
electric charge	coulomb	C
electric resistance, impedance, reactance	ohm	Ω
electric potential difference, voltage	volt	V
energy, work, heat	joule	J
force, weight	newton	N
frequency	hertz	Hz
illuminance	lux	lx
inductance	henry	H
luminous flux	lumen	lm
magnetic flux	weber	Wb
magnetic flux density, magnetic induction	tesla	T
plane angle	radian	rad
power	watt	W
pressure	pascal	Pa
radioactivity	becquerel	Bq

TABLE 7-6: ADDITIONAL UNITS IN COMSOL MULTIPHYSICS

DERIVED QUANTITY	NAME	SYMBOLS	VALUE
acceleration	galileo	Gal	0.01 m/s^2
dipole moment	debye	D	$3.33564095\cdot 10^{-30}$ C·m
energy	British thermal unit*	BTU, Btu	1055.05585 J
energy	calorie*	cal	4.184 J
energy	electronvolt	eV	$1.6021765314\cdot 10^{-19}$ J
energy	erg	erg	10^{-7} J
force	dyne	dyn	10^{-5} N
force	poundal	pdl	0.138254954376 N
force	pound-force	lbf	4.4482216152605 N
frequency	rpm	RPM	1/60 Hz
length	angstrom	Å	10^{-10} m
length	inch	in	0.0254 m
length	foot	ft	0.3048 m

TABLE 7-6: ADDITIONAL UNITS IN COMSOL MULTIPHYSICS

DERIVED QUANTITY	NAME	SYMBOLS	VALUE
length	mile*	mi	1609.344 m
length	microinch	uin	$0.0254\cdot 10^{-6}$ m
length	milliinch	mil, thou	$0.0254\cdot 10^{-3}$ m
length	nautical mile*, nautimile	nmi	1852 m
length	yard	yd	0.9144 m
magnetic field strength	oersted	Oe	$10^{3}/(4\cdot\pi)$ A/m
magnetic flux density	gauss	G	10^{-4} T
mass	atomic mass unit, dalton	u, amu, Da	$1.660538782\cdot 10^{-27}$ kg
mass	gram	g	0,001 kg
mass	pound, pound-mass	lb, lbm	0.45359237 kg
mass	stone	st	6.35029318 kg
mass	slug	slug	approx. 14.5939 kg
mass	ton, tonne	t	1000 kg
permeability	millidarcy*	mD	$9.869233\cdot 10^{-16}$ m^{2}
plane angle	degree	deg	$\pi/180$
pressure	atmosphere	atm	101325 Pa
pressure	bar	bar	100000 Pa
pressure	barye	ba	0.1 Pa
pressure	psi	psi	$6.894757\cdot 10^{3}$ Pa
pressure	torr	Torr, mmHg	133.322 Pa
speed	mph, MPH	mph	0.44704 m/s
speed	knot*	knot	1852 km/h (approx. 0.614 m/s)
temperature	Celsius**	degC	T+273.15
temperature	Fahrenheit**	degF	$5/9\cdot T+459.67$
temperature	Rankine**	R, Ra	$5/9\cdot T$
time	year*	a, yr	31556952 s
time	day	d	86400 s
time	hour	h	3600 s
time	minute	min	60 s

TABLE 7-6: ADDITIONAL UNITS IN COMSOL MULTIPHYSICS

DERIVED QUANTITY	NAME	SYMBOLS	VALUE
volume	gallon*	gal	0.003785411784 m^3
volume	imperialgallon	impgal	0.00454609 m^3
volume	liter, litre	L, l	0.001 m^3
volume	pint*	pt	0.000473176473 m^3
volume	quart*	qt	0.000946352946 m^3
volume flow rate	cubic feet per minute	CFM, cfm	$4.719474 \cdot 10^{-4}$ m^3/s

* See the additional notes following this table.

** See About Temperature Units

ADDITIONAL NOTES ABOUT Table 7-6

UNIT NAME	NOTE
British thermal unit	An energy unit defined as the amount of heat required to raise the temperature of one pound (pound-mass) of water by one degree from 60° to 61° Fahrenheit at a constant pressure of one atmosphere. Refer to the British thermal unit using the symbol only (Btu or BTU): for example, `0.28[Btu/(h*in*degF)]` for a thermal conductivity.
calorie	Small calorie or gram calorie, which equals 4.184 J. A large calorie or kilogram calorie is 1000 calories (4.184 kJ). Use [kcal] for large calories.
millidarcy (mD)	Widely used for permeability in petroleum engineering. Typical values for the permeability of porous media are in the range of a few to a few hundred mD. The symbol D represents the debye, a unit for the magnetic dipole moment, and not the darcy unit.
mile	The international statute mile, which equals 1609.344 m.
nautimile	The nautical mile equals 1852 m.
knot	The same as nautical miles per hour.
year	A Gregorian year, which equals 365.2425 days.
gallon (gal)	This is the U.S. liquid gallon which equals 0.003785411784 m^3; the Imperial (UK) gallon (imperialgallon, impgal) is equal to 0.00454609 m^3.
pint and quart	The U.S. liquid pint and U.S. liquid quart, respectively.

TABLE 7-7: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

QUANTITY	NAME	SYMBOL
acceleration	meter per second squared	m/s^2
amount-of-substance concentration	mole per cubic meter	mol/m^3
area	square meter	m^2
current density	ampere per square meter	A/m^2
heat capacity, specific heat	joule per kilogram kelvin	J/(kg·K)
magnetic field strength	ampere per meter	A/m
mass density	kilogram per cubic meter	kg/m^3
permeability	henry per meter	H/m
speed, velocity	meter per second	m/s
wave number	reciprocal meter	m^{-1}
volume	cubic meter	m^3

Special British Engineering Units

The base units in the British engineering unit system are identical to the SI units with the following exceptions:

TABLE 7-8: SPECIAL BASE UNITS IN THE BRITISH ENGINEERING UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	slug	N/A
temperature	Fahrenheit	degF

There is one derived unit that differs from the corresponding SI unit:

TABLE 7-9: DERIVED BRITISH ENGINEERING UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	pound-force	lbf

The British thermal unit is also available as Btu or BTU.

Note

If the British engineering unit system is the base unit system, COMSOL Multiphysics constructs derived units from the applicable SI base units and the units listed in Table 7-8 and Table 7-9. This means, for example, that the unit for voltage displayed in the user interface is lbf·ft/As rather than V (volt). In a text field that expects a voltage as input, you need to use the unit syntax when entering a numerical value, for example, `10[V]`.

Special CGSA Units

The base units in the CGSA unit system are identical to the SI units with the following exceptions:

TABLE 7-10: SPECIAL BASE UNITS IN THE CGSA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g

The CGSA unit system includes the following derived units that differ from the corresponding SI units:

TABLE 7-11: DERIVED CGSA UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
energy	erg	N/A
force	dyne	dyn
pressure	barye	N/A
speed	kyne	N/A

Special EMU Units

The base units in the EMU unit system are identical to the SI units with the following exceptions:

TABLE 7-12: SPECIAL BASE UNITS IN THE EMU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm

TABLE 7-12: SPECIAL BASE UNITS IN THE EMU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
mass	gram	g
electric current	abampere, biot	N/A

The EMU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 7-13: DERIVED EMU UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
capacitance	abfarad	N/A
conductance	absiemens	N/A
electric charge	abcoulomb	N/A
electric resistance	abohm	N/A
electric potential difference, voltage	abvolt	N/A
energy	erg	N/A
force	dyne	dyn
inductance	abhenry	N/A
magnetic flux	abweber	N/A
magnetic flux density	abtesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special ESU Units

The base units in the ESU unit system are identical to the SI units with the following exceptions:

TABLE 7-14: SPECIAL BASE UNITS IN THE ESU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
mass	gram	g
electric current	statampere, franklin	N/A

The ESU unit system includes the following derived units that differ from the corresponding SI units:

TABLE 7-15: DERIVED ESU UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
capacitance	statfarad	N/A
conductance	statsiemens	N/A
electric charge	statcoulomb	N/A
electric resistance	statohm	N/A
electric potential difference, voltage	statvolt	N/A
energy	erg	N/A
force	dyne	dyn
inductance	stathenry	N/A
magnetic flux	statweber	N/A
magnetic flux density	stattesla	N/A
pressure	barye	N/A
speed	kyne	N/A

Special FPS Units

The base units in the FPS unit system are identical to the SI units with the following exceptions:

TABLE 7-16: SPECIAL BASE UNITS IN THE FPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	foot	ft
mass	pound	lb
temperature	Fahrenheit	degF

There is one derived unit that differs from the corresponding SI unit:

TABLE 7-17: DERIVED FPS UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	poundal	N/A

Special IPS Units

The base units in the IPS unit system are identical to the SI units with the following exceptions:

TABLE 7-18: SPECIAL BASE UNITS IN THE IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	pound	lb
temperature	Fahrenheit	degF

Special MPa Units

The base units in the MPa unit system are identical to the SI units with the following exceptions:

TABLE 7-19: SPECIAL BASE UNITS IN THE MPA UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	millimeter	mm
mass	tonne, ton	t

There is one derived unit that differs from the corresponding SI unit:

TABLE 7-20: DERIVED MPA UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
pressure	megapascal	MPa

Special Gravitational IPS Units

The base units in the Gravitational IPS unit system are identical to the SI units with the following exceptions:

TABLE 7-21: SPECIAL BASE UNITS IN THE GRAVITATIONAL IPS UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	inch	in
mass	GIPS_mass	N/A
temperature	Fahrenheit	degF

The following derived units differ from the corresponding SI units:

TABLE 7-22: DERIVED GRAVITATIONAL IPS UNITS IN COMSOL MULTIPHYSICS

BASE QUANTITY	NAME	SYMBOL
force	pound-force	lbf
pressure	psi	psi

Switching Unit System

If the unit system is switched during modeling, or a unit system other than SI is used, COMSOL Multiphysics does not convert the data in the model. All physical constants and data in the materials libraries are in SI units and defined using the unit syntax, so you can use them with any unit system because COMSOL Multiphysics converts these values to the corresponding values in the model's unit system.

Note

The units of other input data must be declared for the model using the unit syntax (or convert the numerical values to the new unit system manually).

About Temperature Units

The relationship between different temperature units involve an offset in addition to the usual scale factor. In many cases, the offset is not important to the physics equations because these equations are concerned only with temperature differences. There are, however, some cases where an absolute or thermodynamic temperature measure must be used. The most common example is the Stefan-Boltzmann law for black-body radiation used in radiation boundary conditions.

The SI unit system uses the Kelvin, which is an absolute temperature, as the basic unit of temperature. English unit systems use degree Fahrenheit as the basic unit of temperature, which, because the Fahrenheit scale is not absolute, is fine for most purposes except radiation. For such purposes, the Rankine scale provides the corresponding absolute temperature unit. See Table 7-23 for a list of acceptable unit syntax.

TABLE 7-23: TEMPERATURE UNITS

SCALE	UNIT
Celsius	`[degC]`
Fahrenheit	`[degF]`

TABLE 7-23: TEMPERATURE UNITS

SCALE	UNIT
Kelvin	[K]
Rankine	[R] or [Ra]

DIFFERENTIAL VS. ABSOLUTE TEMPERATURE

If the dimension of an expression that includes a unit is temperature or 1/temperature, COMSOL Multiphysics interprets the dimension as an absolute temperature. If the dimension is something other than temperature but the unit expression includes temperature, the temperature is a differential temperature; that is, COMSOL Multiphysics uses no offset when converting between different temperature units.

The following examples show how the unit conversion works for different expressions that include temperature units:

- `100[degC]` is an expression that has temperature as the dimension. COMSOL Multiphysics interprets it as an absolute temperature and evaluates it as 373.15 K.
- `373.15[1/K]` is interpreted as an absolute temperature (but no conversion is necessary from kelvin to kelvin).
- `100[degC/K]` is dimensionless, and the temperature is therefore a differential temperature; that is, the result is 100 because the conversion uses no offset.
- To make COMSOL Multiphysics interpret `100[degC/K]` as an absolute temperature, split the expression using two separate expressions such as `100[degC]*1[1/K]`, which equals 373.15. This is also what occurs when you use a variable (`TC`, for example) defined as `100[degC]`. `TC[1/K]` is then also two expressions where both are interpreted as absolute temperature.

About Editing Geometry Length and Angular Units

The default units are meters for length and degrees for angles. For many applications an independent length unit for the geometry might be required. For example, if the model describes a MEMS device, the natural length unit might be µm (micrometers), or the geometry imported from a CAD file might use another unit than meters. It can also be useful to specify the angular unit in radians instead.

Note

The length unit for the geometry does not affect the units that include length in the physics interfaces or any other part of COMSOL Multiphysics.

1 Create or open a model file.

2 In the **Model Builder**, under a **Model** node, click the **Geometry** node.

3 Under **Units**, select a **Length unit** from the list.

4 Select an **Angular unit**—**Degrees** or **Radians**.

5 Select the **Scale values when changing units** check box to automatically scale for dimensions in the existing geometry.

6 Enter a **Default relative repair tolerance** and select a **Geometry representation.**

Note

For details about the geometry settings, see The Geometry Node and Its Settings in the *COMSOL Multiphysics Reference Guide*.

When importing 3D CAD geometries, you can choose to use the length unit from the CAD file or the length unit from COMSOL.

Indication of Unexpected, Unknown, or Inconsistent Units

The unit display appears orange for the properties in the settings for the physics and materials that have invalid or inconsistent units or a different unit than expected. An inconsistent unit can occur by summing terms with units that represent different physical quantities, such as `273[K]+3[ft]`, for example. A tooltip displays a message at the corresponding field. In the case of a valid but unexpected unit, this message contains the deduced and expected units in the current unit system.

Note

If an unexpected or inconsistent unit appears in a text field for a physical property, COMSOL Multiphysics ignores the unit and uses the numerical value, including an SI prefix if present, as the input to the model. For example, in a text field for density using SI units, the software interprets `2930[K]` as 293 kg/m^3 and `2930[mK]` as 2.930 kg/m^3.

A unit display that appears red contains a syntax error, which can be due to, for example, missing or misplaced parentheses.

Units and Space Dimensions

Most physics interfaces support 2D (and in some cases also 1D) models in addition to 3D models. The units for intensive physical quantities such as density in the physics interfaces are the same regardless of the space dimension (for density, kg/m^3 in SI units). This makes it possible to use common material property values also in models with other space dimensions than 3D using their well-known, physical units regardless of the dimension you are modeling in. In planar 2D, this means that the implementation includes an implicit unit depth in the out-of-plane direction, except for some physics interfaces (for solid mechanics and electric currents, for example), where the thickness is a user-defined property that defines the volume of the model domain. In axisymmetric models, the volume of the domain is defined by the 2D cross section in the *rz*-plane that is the geometry you define for such models. The volume that it defines is the area of the 2D cross section integrated a full 360 degrees in the circumferential direction.

8

Overview of the Physics Interfaces

This chapter gives an overview of the available physics interfaces as well as some general guidelines for effective modeling.

In this chapter:

The COMSOL Multiphysics Physics Interfaces

This section includes these topics:

- Introduction to the Physics Interfaces
- Physics Interfaces Groups in the Model Wizard
- Physics List in COMSOL Multiphysics
- Selecting a Physics Interface
- Show More Physics Options

See Also

- Showing and Expanding Advanced Physics Sections
- The Physics Feature Nodes
- Specifying Model Equation Settings
- Periodic Boundary Conditions
- The Model Builder Window
- Sorting Nodes by Space Dimension and Type
- Opening a Context Menu to Add Nodes
- Key to the Nodes and Toolbar Buttons

Introduction to the Physics Interfaces

Solving PDEs generally means you must take the time to set up the underlying equations, material properties, and boundary conditions for a given problem. COMSOL Multiphysics, however, relieves you of much of this work. The package provides a number of *physics interfaces* that consist of predefined templates and user interfaces already set up with equations and variables for specific areas of physics. The physics interfaces consist of a number of features with associated windows on the COMSOL Desktop for the physics in domains and on boundaries, edges, and points along with predefined PDEs. A set of interface-dependent variables makes it easy to visualize and evaluate the important physical quantities using conventional terminology and notation.

Tip

Suites of physics interfaces that are optimized for specific disciplines together with specialized model libraries are available in a group of optional products. See The COMSOL Modules for a list.

A complement to the physics interfaces, special interfaces for equation-based modeling simplify the setup of PDEs for modeling that does not explicitly refer to any particular application field. In addition a number of interfaces supplement the physics interfaces with special functionality such as the Sensitivity and Moving Mesh interfaces.

Physics Interfaces Groups in the Model Wizard

The **Add Physics** page in the **Model Wizard** contains the main groups of physics interfaces and mathematics interfaces as in Table 8-1(some items only display if a license includes the add-on modules).

TABLE 8-1: PHYSICS INTERFACE GROUPS IN THE MODEL WIZARD

ICON	BRANCH NAME	DESCRIPTION OF PHYSICS INTERFACES IN THE BRANCH
	Recently Used	Contains the most recently used interfaces for easy access to the ones used the most.
	AC/DC	For low-frequency electromagnetics such as electrostatics and electric currents.
	Acoustics	Interfaces for acoustics.
	Chemical Species Transport	Interfaces for chemical species transport by, for example, convection and diffusion, solving for the species concentrations.
	Electrochemistry	Interfaces for electrochemistry and modeling of electrochemical components such as batteries and fuel cells. This branch is only available if a license includes the Batteries & Fuel Cells Module, the Electrodeposition Module, or the Corrosion Module.
	Fluid Flow	Interfaces for fluid flow such as laminar single-phase flow and, with add-on modules, multiphase flow and turbulent flow.
	Heat Transfer	Interfaces for heat transfer in solids and fluids and thermal multiphysics applications such as Joule heating.

TABLE 8-1: PHYSICS INTERFACE GROUPS IN THE MODEL WIZARD

ICON	BRANCH NAME	DESCRIPTION OF PHYSICS INTERFACES IN THE BRANCH
	Plasma	Interfaces for plasma modeling. This branch is only available if a license includes the Plasma Module.
	Radio Frequency	Interfaces for high-frequency electromagnetic field simulations solving the full Maxwell equations. This branch is only available if a license includes the RF Module.
	Structural Mechanics	Interfaces for structural mechanics, studying displacements and stresses in solids, for example.
Δu	Mathematics	Mathematics interfaces for solving PDEs, ODEs, and DAEs, for optimization (requires the Optimization Module) and sensitivity analysis, and for modeling moving meshes and parameterized geometry.

Physics List in COMSOL Multiphysics

The table lists the physics interfaces in COMSOL Multiphysics and their availability for 1D, 1D axisymmetric, 2D, 2D axisymmetric, and 3D geometries.

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
AC/DC				
Electrostatics		es	all dimensions	stationary; time dependent
Electric Currents		ec	all dimensions	stationary
Magnetic Fields		mf	2D, 2D axisymmetric	stationary; frequency domain
Acoustics				
Pressure Acoustics				
Pressure Acoustics Frequency Domain		acpr	all dimensions	eigenfrequency; frequency domain

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
Chemical Species Transport				
Transport of Diluted Species		chds	all dimensions	stationary; time dependent
Fluid Flow				
Single-Phase Flow				
Single-Phase Flow, Laminar Flow		spf	3D, 2D, 2D axisymmetric	stationary; time dependent
Heat Transfer				
Heat Transfer in Solids		ht	all dimensions	stationary; time dependent
Heat Transfer in Fluids		ht	all dimensions	stationary; time dependent
Electromagnetic Heating				
Joule Heating		jh	all dimensions	stationary; time dependent
Structural Mechanics				
Solid Mechanics		solid	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent
Mathematics				
Wall Distance		wd	all dimensions	stationary; time dependent
PDE Interfaces				
Coefficient Form PDE	Δu	c	all dimensions	stationary; eigenfrequency; time dependent

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
General Form PDE	Δu	g	all dimensions	stationary; eigenfrequency; time dependent
Weak Form PDE	∫du	w	all dimensions	stationary; eigenfrequency; time dependent
Δu *Lower Dimensions*				
Coefficient Form Boundary PDE	Δu	cb	all dimensions	stationary; eigenfrequency; time dependent
Coefficient Form Edge PDE	Δu	ce	3D	stationary; eigenfrequency; time dependent
Coefficient Form Point PDE	Δu	cp	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent
General Form Boundary PDE	Δu	gb	all dimensions	stationary; eigenfrequency; time dependent
General Form Edge PDE	Δu	ge	3D	stationary; eigenfrequency; time dependent
General Form Point PDE	Δu	gp	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent
Weak Form Boundary PDE	∫du	wb	all dimensions	stationary; eigenfrequency; time dependent
Weak Form Edge PDE	∫du	we	3D	stationary; eigenfrequency; time dependent
Weak Form Point PDE	∫du	wp	3D, 2D, 2D axisymmetric	stationary; eigenfrequency; time dependent

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
$\frac{d}{dt}$ *ODE and DAE Interfaces*				
Global ODEs and DAEs	$\frac{d}{dt}$	ge	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
Domain ODEs and DAEs	$\frac{d}{dt}$	dode	all dimensions	stationary; time dependent; eigenvalue
Boundary ODEs and DAEs	$\frac{d}{dt}$	bode	all dimensions	stationary; time dependent; eigenvalue
Edge ODEs and DAEs	$\frac{d}{dt}$	eode	3D	stationary; time dependent; eigenvalue
Point ODEs and DAEs	$\frac{d}{dt}$	pode	3D, 2D, 2D axisymmetric	stationary; time dependent; eigenvalue
Events	∿	ev	all dimensions	time dependent
🎯 *Optimization and Sensitivity*				
Optimization	🎯	opt	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
Sensitivity	📊	sens	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
∇^2 *Classical PDEs*				
Helmholtz Equation	∇^2	hzeq	all dimensions	stationary
Laplace Equation	∇^2	lpeq	all dimensions	stationary
Poisson's Equation	∇^2	poeq	all dimensions	stationary

PHYSICS	ICON	TAG	SPACE DIMENSION	PRESET STUDIES
Wave Equation	∇^2	waeq	all dimensions	time dependent
Heat Equation	∇^2	hteq	all dimensions	stationary; time dependent
Convection-Diffusion Equation	∇^2	cdeq	all dimensions	stationary; time dependent
Deformed Mesh				
Deformed Geometry		dg	all dimensions	stationary; time dependent; frequency domain; eigenvalue
Moving Mesh		ale	all dimensions	stationary; time dependent; frequency domain; eigenvalue

Selecting a Physics Interface

When creating a model in COMSOL Multiphysics, you can select a single physics interface that describes one type of physics or select several physics interfaces for multiphysics modeling and coupled-field analyses.

MODELING USING A SINGLE PHYSICS INTERFACE

Most of the physics interfaces contain stationary, eigenvalue, and time dependent (dynamic) study types. As already mentioned, these physics interfaces provide features and windows where you can create models using material properties, boundary conditions, and initial conditions. Each physics interfaces comes with a template that automatically supplies the appropriate underlying PDEs.

If you cannot find a physics interface that matches a given problem, try one of the mathematics interfaces, which allow you to define a custom model in general mathematical terms. Indeed, COMSOL Multiphysics can model virtually any scientific phenomenon or engineering problem that originates from the laws of science.

MULTIPHYSICS MODELING USING MULTIPLE PHYSICS INTERFACES

When modeling real-world systems, you often need to include the interaction between different kinds of physic: multiphysics. For instance, an electric current produces heat, and the properties of an electronic component such as an inductor vary with temperature. To solve such a problem, combine two or several physics interfaces into a single model using the multiphysics capabilities of COMSOL Multiphysics. For the example just mentioned, you can use the predefined Joule Heating interface, which is a combination of the Electric Currents and Heat Transfer physics interfaces. This way you create a system of two PDEs with two dependent variables: V for the electric potential and T for the temperature. There are many other predefined multiphysics couplings that provide a unified interface that combines two or more coupled physics interfaces for common multiphysics applications.

You can also combine physics interfaces and equation-based modeling for maximum flexibility.

To summarize the proposed strategy for modeling processes that involve several types of physics: Look for physics interfaces suitable for the phenomena of interest. If you find them among the physics interfaces, use them; if not, add one or more PDE interfaces for equation-based modeling.

When coupling multiple physics interfaces in a multiphysics model (without using a predefined multiphysics interface), the couplings can occur in domains and on boundaries. COMSOL Multiphysics automatically identifies potential *model inputs* for quickly forming couplings between physics. For example, a velocity field from a fluid-flow interface is a model input for convection in a heat transfer interface.

Show More Physics Options

There are several features available on many physics interfaces or individual nodes. This section is a short overview of the options and includes links to the *COMSOL Multiphysics User's Guide* or *COMSOL Multiphysics Reference Guide* where additional information is available.

Important

The links to the features described in the *COMSOL Multiphysics Reference Guide* do not work in the PDF, only from within the online help.

Tip

To locate and search all the documentation for this information, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

To display additional features for the physics interfaces and feature nodes, click the **Show** button ( ) on the **Model Builder** and then select the applicable option.

After clicking the **Show** button ( ), some sections display on the settings window when a node is clicked and other features are available from the context menu when a node is right-clicked. For each, the additional sections that can be displayed include **Equation**, **Advanced Settings**, **Discretization**, **Consistent Stabilization**, and **Inconsistent Stabilization**.

You can also click the **Expand Sections** button ( ) in the **Model Builder** to always show some sections or click the **Show** button ( ) and select **Reset to Default** to reset to display only the **Equation** and **Override and Contribution** sections.

For most physics nodes, both the **Equation** and **Override and Contribution** sections are always available. Click the **Show** button ( ) and then select **Equation View** to display the **Equation View** node under all physics nodes in the **Model Builder**.

Availability of each feature, and whether it is described for a particular physics node, is based on the individual physics selected. For example, the **Discretization**, **Advanced Settings**, **Consistent Stabilization**, and **Inconsistent Stabilization** sections are often described individually throughout the documentation as there are unique settings.

SECTION	CROSS REFERENCE	LOCATION IN COMSOL MULTIPHYSICS USER'S GUIDE OR REFERENCE GUIDE
Show More Options and Expand Sections	• Showing and Expanding Advanced Physics Sections • The Model Builder Window	User's Guide
Discretization	• Show Discretization • Element Types and Discretization	User's Guide
	• Finite Elements • Discretization of the Equations	Reference Guide
Discretization - Splitting of complex variables	Compile Equations	Reference Guide

SECTION	CROSS REFERENCE	LOCATION IN COMSOL MULTIPHYSICS USER'S GUIDE OR REFERENCE GUIDE
Pair Selection	• Identity and Contact Pairs • Specifying Boundary Conditions for Identity Pairs	User's Guide
Consistent and Inconsistent Stabilization	Show Stabilization	User's Guide
	• Stabilization Techniques • Numerical Stabilization	Reference Guide
Geometry	Working with Geometry	User's Guide
Constraint Settings	Using Weak Constraints	User's Guide

Modeling Guidelines

To allow you to model large-scale problems and for successful modeling in general, COMSOL Multiphysics lets you tune solver settings and use symmetries and other model properties to reach a solution or—failing that—interrupt the solution process to retrieve a partial solution. This section provides some tips and guidelines when modeling.

In this section:

- Using Symmetries
- Effective Memory Management
- Selecting an Element Type
- Analyzing Model Convergence and Accuracy
- Achieving Convergence When Solving Nonlinear Equations
- Avoiding Strong Transients
- Physics-Related Checks and Guidelines
- Results With Small Unphysical Values

Using Symmetries

By using symmetries in a model you can reduce its size by one-half or more, making this an efficient tool for solving large problems. This applies to the cases where the geometries and modeling assumptions include symmetries.

The most important types of symmetries are axial symmetry and symmetry and antisymmetry planes or lines:

- *Axial Symmetry* is common for cylindrical and similar 3D geometries. If the geometry is axisymmetric, there are variations in the radial (r) and vertical (z) direction only and not in the angular (θ) direction. You can then solve a 2D problem in the rz-plane instead of the full 3D model, which can save considerable memory and computation time. Many COMSOL Multiphysics physics interfaces are available in axisymmetric versions and take the axial symmetry into account.
- *Symmetry and Antisymmetry Planes or Lines* are common in both 2D and 3D models. *Symmetry* means that a model is identical on either side of a dividing line or plane. For a scalar field, the normal flux is zero across the symmetry line. In

structural mechanics, the symmetry conditions are different. *Antisymmetry* means that the loading of a model is oppositely balanced on either side of a dividing line or plane. For a scalar field, the dependent variable is 0 along the antisymmetry plane or line. Structural mechanics applications have other antisymmetry conditions. Many physics interfaces have symmetry conditions directly available as features.

To take advantage of symmetry planes and symmetry lines, all of the geometry, material properties, and boundary conditions must be symmetric, and any loads or sources must be symmetric or antisymmetric. You can then build a model of the symmetric portion, which can be half, a quarter, or an eighth of the full geometry, and apply the appropriate symmetry (or antisymmetry) boundary conditions.

Effective Memory Management

Especially in 3D modeling, extensive memory usage dictates some extra precautions. First, check that you have selected an iterative linear system solver. Normally you do not need to worry about which solver to use, because the physics interface makes an appropriate default choice. In some situations, though, it might be necessary to make additional changes to the solver settings and the model.

ESTIMATING THE MEMORY USE FOR A MODEL

Out-of-memory messages can occur when COMSOL Multiphysics tries to allocate an array that does not fit sequentially in memory. It is common that the amount of available memory seems large enough for an array, but there might not be a contiguous block of that size due to memory fragmentation.

In estimating how much memory it takes to solve a specific model, the following factors are the most important:

- The number of node points
- The number of dependent and independent variables
- The element order
- The sparsity pattern of the system matrices. The sparsity pattern, in turn, depends on the shape of the geometry and the mesh. For example, an extended ellipsoid gives sparser matrices than a sphere.

The MUMPS and PARDISO out-of-core solvers can make use of available disk space to solve large models that do not fit in the available memory.

You can monitor the memory use in the bottom-right corner of the COSMOL Desktop, where the program displays the amount of physical memory (working set) and total virtual memory used.

CREATING A MEMORY-EFFICIENT GEOMETRY

A first step when dealing with large models is to try to reduce the model geometry as much as possible. Often you can find symmetry planes and reduce the model to a half, a quarter or even an eighth of the original size. Memory usage does not scale linearly but rather polynomially (Cn^k, $k > 1$), which means that the model needs less than half the memory if you find a symmetry plane and cut the geometry size by half. Other ways to create a more memory-efficient geometry include:

- Avoiding small geometry objects where not needed and using Bézier curves instead of polygon chains.
- Using linear elements if possible (this is the default setting in many physics interfaces). See Selecting an Element Type.
- Making sure that the mesh elements are of a high quality. Mesh quality is important for an iterative linear system solver. Convergence is faster and more robust if the element quality is high.
- Avoiding geometries with sharp, narrow corners. Mesh elements get thin when they approach sharp corners, leading to poor element quality in the adjacent regions. Sharp corners are also unphysical and can lead to very large (even infinite, in theory) stress concentrations, for example.

Selecting an Element Type

As the default element type for most physics interfaces, COMSOL Multiphysics uses typically uses first-order or second-order Lagrange elements or shape functions. Second-order elements and other higher-order elements add additional degrees of freedom on midpoint and interior nodes in the mesh elements. These added degrees of freedom typically provide a more accurate solution but also require more memory due to the reduced sparsity of the discretized system. For many application areas, such as stress analysis in solid mechanics, the increased accuracy of a second-order element is important because quantities such as stresses involve space derivatives and become constant within an element when using first-order elements.

COMSOL recommends that you use the default element types. For some applications, it might be possible to use a lower-order element than the default element type, but you must then use care to ensure that the important quantities are resolved.

See Also

For information about editing shape functions, see The Equation View Node.

Analyzing Model Convergence and Accuracy

It is important that the finite element model accurately captures local variations in the solution such as stress concentrations. In some cases you can compare your results to values from handbooks, measurements, or other sources of data. Many examples in the COMSOL Multiphysics Model Library include comparisons to established results or analytical solutions. Look for these *benchmark models* as a means of checking results.

If a model has not been verified by other means, a *convergence test* is useful for determining if the mesh density is sufficient. Here you refine the mesh and run the study again, and then check if the solution is converging to a stable value as the mesh is refined. If the solution changes when you refine the mesh, the solution is mesh dependent, so the model requires a finer mesh. You can use adaptive mesh refinement, which adds mesh elements based on an error criterion to resolve those areas where the error is large.

For convergence, it is important to avoid singularities in the geometry.

See Also

Avoiding Singularities and Degeneracies in the Geometry

Achieving Convergence When Solving Nonlinear Equations

Nonlinear problems are often difficult to solve. In many cases, no unique solution exists. COMSOL Multiphysics uses a Newton-type iterative method to solve nonlinear systems of PDEs. This solution method can be sensitive to the initial estimate of the solution. If the initial conditions are too far from the desired solution, convergence might be impossible, even though it might be simple from a different starting value.

You can do several things to improve the chances for finding the relevant solutions to difficult nonlinear problems:

- Provide the best possible initial values.
- Solve sequentially and iterate between single-physics equations; finish by solving the fully coupled multiphysics problem when you have obtained better starting guesses.
- Ensure that the boundary conditions are consistent with the initial solution and that neighboring boundaries have compatible conditions that do not create singularities.
- Refine the mesh in regions of steep gradients.
- For convection-type problem, introduce artificial diffusion to improve the problem's numerical properties (see Stabilization Techniques in the *COMSOL Multiphysics Reference Guide*). Most physics interfaces for modeling of fluid flow and chemical species transport provide artificial diffusion as part of the default settings.
- Scaling can be an issue when one solution component is zero. In those cases, the automatic scaling might not work.
- Turn a stationary nonlinear PDE into a time-dependent problem. Making the problem time dependent generally results in smoother convergence. By making sure to solve the time-dependent problem for a time span long enough for the solution to reach a steady state, you solve the original stationary problem.
- Use the parametric solver and vary a material property or a PDE coefficient starting from a value that makes the equations less nonlinear to the value at which you want to compute the solution. This way you solve a series of increasingly difficult nonlinear problems. The solution of a slightly nonlinear problem that is easy to solve serves as the initial value for a more difficult nonlinear problem.

Avoiding Strong Transients

If you start solving a time-dependent problem with initial conditions that are inconsistent, or if you use boundary conditions or sources that switch instantaneously at a certain time, you induce strong transient signals in a system. The time-stepping algorithm then takes very small steps to resolve the transient, and the solution time might be very long, or the solution process might even stop. Stationary problems can run into mesh-resolution issues such as overshooting and undershooting of the solution due to infinite flux problems.

Unless you want to know the details of these transients, start with initial conditions that lead to a consistent solution to a stationary problem. Only then turn on the boundary values, sources, or driving fluxes over a time interval that is realistic for your model.

In most cases you should turn on your sources using a smoothed step over a finite time. What you might think of as a step function is, in real-life physics, often a little bit smoothed because of inertia. The step or switch does not happen instantaneously. Electrical switches take milliseconds, and solid-state switches take microseconds.

Physics-Related Checks and Guidelines

There are some important checks and guidelines that primarily apply to different areas of physics. Making these checks ensure that the model input is sufficient and increase the chances for successful modeling. See also the modeling sections of the documentation for the physics interfaces and the modules for more information related to modeling different physics.

FLUID FLOW AND TRANSPORT PHENOMENA

The following checks and guidelines primarily apply to fluid-flow modeling but also to modeling of other transport phenomena:

- If none of the boundary conditions include the pressure (most outlet conditions do, however), then you should specify the pressure at some point in the fluid domain. Without a specified pressure, the problem is underconstrained and it is difficult to get convergence.
- Make sure that the mesh if sufficiently fine, so that it contains at least 4–6 mesh elements across the thickness of a channel, for example.
- Make sure that the boundary conditions and the initial conditions match for time-dependent problems. For example, instead of starting with a full velocity on the wall, compared to a zero initial velocity field in the fluid, ramp up the velocity with a smoothed step function or a ramp function that takes the inlet velocity from zero, which matches the initial value for the velocity field, to the full velocity. See also Avoiding Strong Transients.
- For fluid-flow models it is important to estimate the flow regime (laminar or turbulent) using the Reynolds number, for example. If the flow is in the turbulent regime, a turbulence model is typically required.

ACOUSTIC, STRUCTURAL, AND ELECTROMAGNETIC WAVE PROPAGATION

For models that describe wave propagation, it is important to fully resolve the wave in both time and space. In practice that means using a maximum mesh element size that provides about 10 linear or five second-order elements per wavelength and also, for transient simulations, a fixed time step that is small enough.

STRUCTURAL MECHANICS

The following checks and guidelines primarily apply to modeling of structural mechanics:

- Make sure that the model is fully constrained. At a minimum, you typically need to constrain the model to avoid all rigid-body movement, which for a 3D solid mechanics model means 6 constraints for three translations and three rotations. Otherwise the solution is not well defined and does not converge. It is not possible to add all 6 constraints in a single point, where you can constrain at most three translational degrees of freedom. For a 3D solid model you can use a “3–2–1 approach” to constrain 3 degrees of freedom at one point (a fixed constraint), 2 at another point, and 1 at a third point. To do so, select three convenient points (vertices) that are well separated. Then fix the first point in all three directions. Constrain the second point in the two directions orthogonal (normal) to the vector from point one to point two making sure that there is no restriction to deformation along the line from point one to point two. Finally constrain the third point in a direction normal to the plane formed by the three points. To test this approach, the model should expand or contract under temperature loading and have small stresses throughout with no stress concentrations. The corresponding minimum constraints for a 2D model are a fixed constraint at one point for the 2 translational degrees of freedom and an additional constraint in one direction at another point to constrain the single rotational degree of freedom.
- Consider if you can assume that the material is linear elastic and that the deformations are small. If not, consider using a nonlinear material model.
- Avoid sharp corners in the geometry, which are unphysical and lead to unbounded stress concentrations.

Results With Small Unphysical Values

WHERE AND WHY DO UNPHYSICAL VALUES APPEAR?

In some models small unphysical values can occur due to numerical artifacts. Examples include:

- Negative concentrations in mass transfer.

- A temperature that is slightly higher than the initial condition in time-dependent heat transfer studies.
- Small reaction forces that appear in unloaded directions in structural mechanics models.
- Small negative gaps in a contact analysis.
- Small negative effective plastic strain values.
- Stresses above the yield limit for an ideally plastic material in solid mechanics.

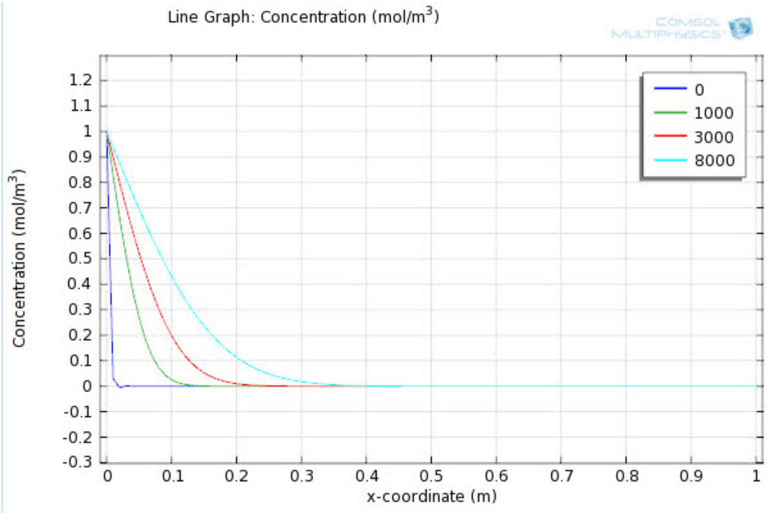
Some reasons for why these unphysical values occur:

- Numerical noise is a common cause. When the values of the dependent variables approach zero, the numerical noise can become relatively significant and cause some of the results to be slightly negative even if that is not physically possible.
- Interpolation and extrapolation of values can cause some values to become unphysical. For example, results for an elastoplastic material are correct (within some tolerance) at the integration points (Gauss points) inside the finite elements, but values might become unphysical when extrapolating the data to the element boundaries.

Tip

The **Plasticity** feature is available as a subnode to a **Linear Elastic Material** with the Nonlinear Structural Materials Module.

- Discontinuities in the model is another source of, for example, small negative concentrations due to a discontinuous initial value. With an initial value that is zero along a boundary for a convective transport models, for example, the physical interpretation is an initially sharp, gradually diffusing front moving away from the boundary. However, for the default shape function (second-order Lagrange elements), only continuous functions are admissible as solutions for the finite element model. The program then modifies the discontinuous initial value before the time stepping can begin. This often results in a small dip in the solution at the start time. In the example model that the following figure shows, the concentration is locally slightly negative at $t = 0$:



- Lack of mesh resolution is another cause of unphysical values such as negative concentrations. The resulting convergence problems are often the underlying issue when negative concentrations are observed in high convection regimes (high Peclet number) and in those with large reaction terms or fast kinetics (high Damkohler number).
- Incorrect physics in the model can also cause these types of problems. For mass transfer, for example, the use of a constant sink in a reaction term is an approximation that only works for large concentrations. When the concentration reaches zero, the reaction term continues to consume the species, finally resulting in a negative concentration.

AVOIDING UNPHYSICAL VALUES

This section contains some ways to avoid computing or displaying unphysical values:

- In some cases it is possible to add a baseline to the dependent variable so that the numerical noise does not affect the solution in the same way as when the values of the dependent variable approach zero. This scaling is not possible with, for example, a reaction term that depends on the concentration because then the scale and origin do matter.
- Avoid discontinuities in the model by using, for example, the available smoothed step functions instead.

- Formulate logarithmic variables as a way of eliminating mesh resolution problems and negative dips by using the logarithm of the original dependent variable (the concentration, for example) as the dependent variable. The reason for this is that a linearly varying mesh can sometimes not capture the exponential behavior of the changes in the dependent variable. In addition, modeling the logarithm of the dependent variable ensures that the real concentration, for example, cannot become negative during the solution process.
- Avoid displaying small unphysical values due to numerical noise by clipping the values for the plot. You can do this by plotting, for example, `c*(c>0)` instead of `c`, which evaluates to 0 everywhere where *c* is smaller than 0. You can also adjust the range of the plot data and colors to only show values that are nonzero. In the case that the data range changes, parts of the plots where the values are outside the range become empty.
- It can also be useful to check how the mesh affects the solution by refining the mesh and check if the problem with unphysical values gets better or worse. If it gets better, then continue to refine the mesh. If it gets worse, you probably need to check the physics of the model.

Equations for the Physics Interfaces

In this section:

- Physics Nodes—Equation Section
- Physics Nodes Advanced Settings

Physics Nodes—Equation Section

For each physics interface and feature node there is an **Equation** section always available on the settings window. This has options to display mathematical equations applicable to the feature.

See Also	The Equation View Node

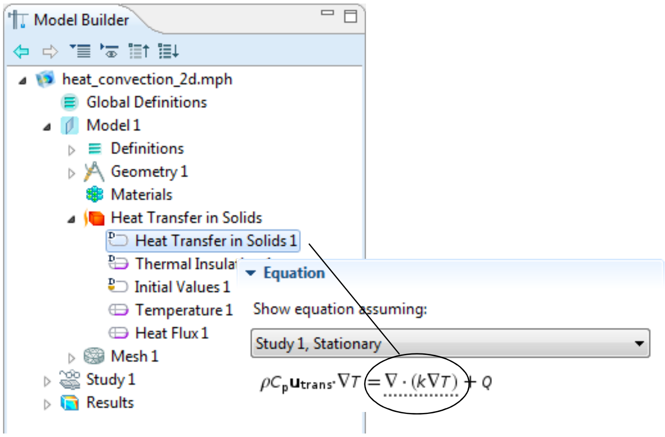
Tip	You can expand the section at any time from the settings window. Click the **Expand Sections** button (▾☰) in the **Model Builder** and select **Equations**.

The display options available from the drop-down lists are dependent on the study types and other physics specific factors. See Figure 8-2 for an example comparing the equations that display for a **Stationary** or **Time Dependent** study for a **Heat Transfer** interface. Some settings windows do not have any options and only display the relevant equation and other windows have additional sections that become available for the **Equation** display based on the study type selected.

See Also	Study Types in the *COMSOL Multiphysics Reference Guide*

Node Contributions Display a Dotted Line Under Part of the Equation

For all feature nodes (excluding the interface node level), the equation that displays includes a dotted line underneath where the node's contribution is made to the equation. See Figure 8-1 for an example where a section of the heat transfer equation is underlined, indicating where the **Heat Transfer in Solids** feature contributes to it.



Show equation assuming: Study 1, Stationary

$\rho C_p \mathbf{u}_{\text{trans}} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q$

Figure 8-1: An example of the Heat Transfer in Solids feature contribution to the equation for a 3D model.

Equation Form

When you add physics to a model, the default **Study types** are listed in the **Equation form** list. **Study controlled** is the default; select another option as required.

Show Equation Assuming

The **Show equation assuming** option is available by default when **Study controlled** is selected (or left as the default) as the **Equation form**. Availability of the options are based on the studies added and defined for the model.

For the following options—frequency and mode analysis frequency—you also have the option to use another frequency than the one used by the solver. This can be necessary if you need two different frequencies for two interfaces.

Frequency

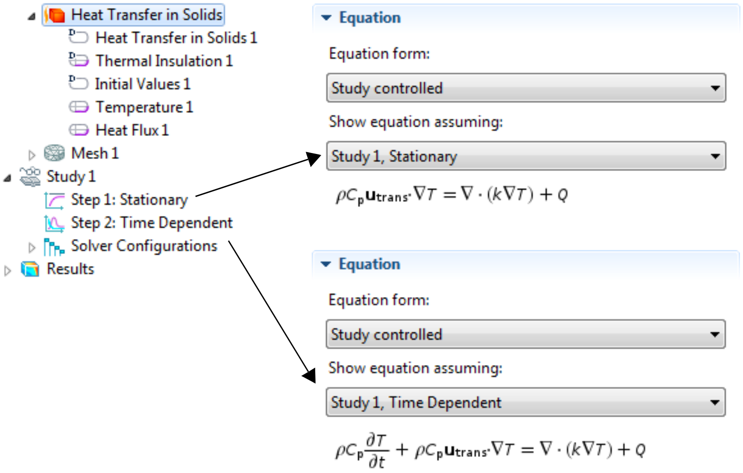
This option is available if **Frequency domain** is selected as the **Equation form**. The default uses the frequency **From solver**. If **User defined** is selected, enter another value or expression (SI unit: Hz).

Mode Analysis Frequency

This option is available if **Mode Analysis** or **Boundary Mode Analysis** is selected as the **Equation Form**. Enter a value or expression in the field (SI unit: Hz). Specify a frequency (it is not present as a solver variable).

Port Name

This option is available with the RF Module **Electromagnetic Waves** interface and if **Boundary Mode Analysis** is selected as the **Equation Form**. Enter a value in the field (unitless).



▾ Equation

Equation form:

Study controlled

Show equation assuming:

Study 1, Stationary

$$\rho C_p \mathbf{u}_{\text{trans}} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q$$

▾ Equation

Equation form:

Study controlled

Show equation assuming:

Study 1, Time Dependent

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u}_{\text{trans}} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q$$

Figure 8-2: An example of the Equation section on a Heat Transfer interface. Selecting the study type updates the equation accordingly.

Physics Nodes Advanced Settings

Various settings are available by clicking the **Show** button ( ) and selecting from the **Advanced>** menus. See Showing and Expanding Advanced Physics Sections for more information.

9

The AC/DC Branch

This chapter explains the physics interfaces for modeling electromagnetics, which you find under the **AC/DC** branch (⚡) in the **Model Wizard**. It also contains sections about general fundamentals and theory for electric fields.

In this chapter:

The Electromagnetics Interfaces

For simulating electromagnetic fields, COMSOL Multiphysics has three physics interfaces.

With the first two, you can perform static simulations to solver for electric properties:

- Electrostatics
- Electric Currents
- Magnetic Fields

These interfaces are available in all space dimensions.

2D

2D Axi

For 2D axisymmetric and 2D geometries, you can use the Magnetic Fields interface to perform magnetostatic simulations.

This section begins with a brief introduction to electromagnetics and a definition of the electromagnetic quantities. Then it describes each of the physics interfaces in detail.

Note

The optional AC/DC Module contains specialized and extended physics interfaces for electromagnetic simulations, for example, for computations of inductors and capacitors. The optional RF Module includes physics interfaces for simulating electromagnetic wave propagation that are especially useful in microwave engineering and photonics.

Fundamentals of Electromagnetics

In this section:

- Maxwell's Equations
- Constitutive Relations
- Potentials
- Material Properties
- About the Boundary and Interface Conditions
- Electromagnetic Forces
- References for Electromagnetic Theory

Maxwell's Equations

The problem of electromagnetic analysis on a macroscopic level is that of solving *Maxwell's equations* subject to certain boundary conditions. Maxwell's equations are a set of equations, written in differential or integral form, stating the relationships between the fundamental electromagnetic quantities. These quantities are:

- Electric field intensity **E**
- Electric displacement or electric flux density **D**
- Magnetic field intensity **H**
- Magnetic flux density **B**
- Current density **J**
- Electric charge density ρ

The equations can be formulated in differential form or integral form. The differential form is presented here because it leads to differential equations that the finite element method can handle. For general time-varying fields, Maxwell's equations can be written as

$$\begin{aligned} \nabla\times\mathbf{H} &= \mathbf{J}+\frac{\partial\mathbf{D}}{\partial t} \\ \nabla\times\mathbf{E} &= -\frac{\partial\mathbf{B}}{\partial t} \\ \nabla\cdot\mathbf{D} &= \rho \\ \nabla\cdot\mathbf{B} &= 0 \end{aligned}$$

The first two equations are also referred to as *Maxwell-Ampère's law* and *Faraday's law*, respectively. Equation three and four are two forms of *Gauss' law*: the electric and magnetic form, respectively.

Another fundamental equation is the *equation of continuity*

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t}$$

Out of the five equations mentioned, only three are independent. The first two combined with either the electric form of Gauss' law or the equation of continuity form such an independent system.

Constitutive Relations

To obtain a closed system, the equations include *constitutive relations* that describe the macroscopic properties of the medium. They are given as

$$\begin{aligned} \mathbf{D} &= \varepsilon_0 \mathbf{E} + \mathbf{P} \\ \mathbf{B} &= \mu_0(\mathbf{H} + \mathbf{M}) \\ \mathbf{J} &= \sigma \mathbf{E} \end{aligned} \tag{9-1}$$

where ε_0 is the *permittivity of vacuum*, μ_0 is the *permeability of vacuum*, and σ the *electrical conductivity.* In the SI system, the permeability of vacuum is chosen to be $4\pi\cdot 10^{-7}$ H/m. The velocity of an electromagnetic wave in vacuum is given as c_0 and the permittivity of vacuum is derived from the relation

$$\varepsilon_0 = \frac{1}{c_0^2 \mu_0} = 8.854 \cdot 10^{-12} \text{ F/m} \approx \frac{1}{36\pi} \cdot 10^{-9} \text{ F/m}$$

The electromagnetic constants ε_0, μ_0, and c_0 are available in COMSOL Multiphysics as predefined *physical constants.*

The *electric polarization vector* **P** describes how the material is polarized when an electric field **E** is present. It can be interpreted as the volume density of electric dipole moments. **P** is generally a function of **E**. Some materials can have a nonzero **P** also when there is no electric field present.

The *magnetization vector* **M** similarly describes how the material is magnetized when a magnetic field **H** is present. It can be interpreted as the volume density of magnetic dipole moments. **M** is generally a function of **H**. Permanent magnets, for instance, have a nonzero **M** also when there is no magnetic field present.

For linear materials, the polarization is directly proportional to the electric field, $\mathbf{P} = \varepsilon_0\chi_e\mathbf{E}$, where χ_e is the *electric susceptibility*. Similarly in linear materials, the magnetization is directly proportional to the magnetic field, $\mathbf{M} = \chi_m\mathbf{H}$, where χ_m is the *magnetic susceptibility*. For such materials, the constitutive relations are

$$\mathbf{D} = \varepsilon_0(1+\chi_e)\mathbf{E} = \varepsilon_0\varepsilon_r\mathbf{E} = \varepsilon\mathbf{E}$$
$$\mathbf{B} = \mu_0(1+\chi_m)\mathbf{H} = \mu_0\mu_r\mathbf{H} = \mu\mathbf{H}$$

The parameter ε_r is the *relative permittivity* and μ_r is the *relative permeability* of the material. Usually these are scalar properties but can, in the general case, be 3-by-3 tensors when the material is anisotropic. The properties ε and μ (without subscripts) are the *permittivity* and *permeability* of the material.

GENERALIZED CONSTITUTIVE RELATIONS

See Also

The Charge Conservation feature describes the macroscopic properties of the medium (relating the electric displacement **D** with the electric field **E**) and the applicable material properties.

For nonlinear materials, a generalized form of the constitutive relationships is useful. The relationship used for electric fields is $\mathbf{D} = \varepsilon_o\varepsilon_r\mathbf{E} + \mathbf{D}_r$ where $\mathbf{D}_r$ is the *remanent displacement*, which is the displacement when no electric field is present.

Similarly, a generalized form of the constitutive relation for the magnetic field is

$$\mathbf{B} = \mu_0\mu_r\mathbf{H} + \mathbf{B}_r$$

where $\mathbf{B}_r$ is the *remanent magnetic flux density*, which is the magnetic flux density when no magnetic field is present.

For some materials, there is a nonlinear relationship between **B** and **H** such that

$$\mathbf{B} = f(|\mathbf{H}|)$$

The relation defining the current density is generalized by introducing an externally generated current $\mathbf{J}_e$. The resulting constitutive relation is $\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_{e.}$

Potentials

Under certain circumstances it can be helpful to formulate the problems in terms of the *electric scalar potential* V and the *magnetic vector potential* $\mathbf{A}$. They are given by the equalities

$$\mathbf{B} = \nabla\times\mathbf{A}$$

$$\mathbf{E} = -\nabla V - \frac{\partial\mathbf{A}}{\partial t}$$

The defining equation for the magnetic vector potential is a direct consequence of the magnetic Gauss' law. The electric potential results from Faraday's law.

Material Properties

Until now, there has only been a formal introduction of the constitutive relations. These seemingly simple relations can be quite complicated at times. There are four main groups of materials where they require some consideration. A given material can belong to one or more of these groups. The groups are:

- Inhomogeneous materials
- Anisotropic materials
- Nonlinear materials
- Dispersive materials

A material can belong to one or more of these groups.

INHOMOGENEOUS MATERIALS

Inhomogeneous materials are the least complicated. An inhomogeneous medium is one in which the constitutive parameters vary with the space coordinates so that different field properties prevail at different parts of the material structure.

ANISOTROPIC MATERIALS

For anisotropic materials the field relationships at any point differ for different directions of propagation. This means that a 3-by-3 tensor is necessary to properly define the constitutive relationships. If this tensor is symmetric, the material is often referred to as *reciprocal*. In such cases, rotate the coordinate system such that a diagonal matrix results. If two of the diagonal entries are equal, the material is *uniaxially anisotropic*. If none of the elements has the same value, the material is *biaxially anisotropic* (Ref. 2). Anisotropic parameters are needed, for example, to examine permittivity in crystals (Ref. 2) and when working with conductivity in solenoids.

NONLINEAR MATERIALS

Nonlinearity is the effect of variations in permittivity or permeability with the intensity of the electromagnetic field. Nonlinearity also includes hysteresis effects, where not only the current field intensities influence the physical properties of the material, but also the history of the field distribution.

DISPERSIVE MATERIALS

Dispersion describes changes in a wave's velocity with wavelength. In the frequency domain dispersion is expressed with a frequency dependence of the constitutive laws.

About the Boundary and Interface Conditions

To get a full description of an electromagnetics problem, boundary conditions must be specified at material interfaces and physical boundaries. At interfaces between two media, the boundary conditions can be expressed mathematically as

$$\begin{aligned}
\mathbf{n}_2 \times (\mathbf{E}_1 - \mathbf{E}_2) &= \mathbf{0} \\
\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) &= \rho_s \\
\mathbf{n}_2 \times (\mathbf{H}_1 - \mathbf{H}_2) &= \mathbf{J}_s \\
\mathbf{n}_2 \cdot (\mathbf{B}_1 - \mathbf{B}_2) &= 0
\end{aligned}$$

where ρ_s and $\mathbf{J}_s$ denote *surface charge density* and *surface current density*, respectively, and $\mathbf{n}_2$ is the outward normal from medium 2. Of these four conditions, only two are independent. This is an overdetermined system of equations, so it needs to be reduced. First select either equation one or equation four. Then select either equation two or equation three. Together these selections form a set of two independent conditions.

From these relationships, the interface condition is derived for the current density,

$$\mathbf{n}_2 \cdot (\mathbf{J}_1 - \mathbf{J}_2) = -\frac{\partial \rho_s}{\partial t}$$

INTERFACE BETWEEN A DIELECTRIC AND A PERFECT CONDUCTOR

A perfect conductor has infinite electrical conductivity and thus no internal electric field. Otherwise, it would produce an infinite current density according to the third fundamental constitutive relation. At an interface between a dielectric and a perfect conductor, the boundary conditions for the $\mathbf{E}$ and $\mathbf{D}$ fields are simplified. Assume that subscript 1 corresponds to a perfect conductor; then $\mathbf{D}_1 = \mathbf{0}$ and $\mathbf{E}_1 = \mathbf{0}$ in the relationships just given. If, in addition, it is a time-varying case, then $\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{H}_1 = \mathbf{0}$, as well, as a consequence of Maxwell's equations. The result is the following set of boundary conditions for the fields in the dielectric medium for the time-varying case:

$$\begin{aligned} -\mathbf{n}_2 \times \mathbf{E}_2 &= 0 \\ -\mathbf{n}_2 \times \mathbf{H}_2 &= \mathbf{J}_s \\ -\mathbf{n}_2 \cdot \mathbf{D}_2 &= \rho_s \\ -\mathbf{n}_2 \cdot \mathbf{B}_2 &= 0 \end{aligned}$$

Electromagnetic Forces

The Magnetic Field interface contains a predefined domain-level variable for calculating the Lorentz force, which gives the force distribution exerted on a current-carrying conductor placed in magnetic flux density $\mathbf{B}$. The Lorentz force is defined as $\mathbf{F} = \mathbf{J} \times \mathbf{B}$.

The Lorentz force gives very good accuracy for electromagnetic force calculations in conducting domains. For nonconducting domains, use a more general method—integrating the Maxwell stress tensor variables over the boundaries of the object for which to calculate the total force. The Maxwell surface stress tensor is available as a boundary variable.

References for Electromagnetic Theory

1. D.K. Cheng, *Field and Wave Electromagnetics*, Addison-Wesley, Reading, Massachusetts, 1989.

2. J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 1993.

3. B.D. Popovic, *Introductory Engineering Electromagnetics*, Addison-Wesley, Reading, Massachusetts, 1971.

Theory of Electric Fields

COMSOL Multiphysics includes physics interfaces for the modeling of static electric fields and currents. Deciding what specific physics interface and study type to select for a particular modeling situation requires a basic understanding of the charge dynamics in conductors. This section is a brief introduction to Charge Relaxation Theory.

Tip

Physics interfaces for the modeling of dynamic, quasi-static (that is, without including wave propagation effects) electric fields and currents are available with the AC/DC Module and MEMS Module.

Charge Relaxation Theory

The different physics interfaces involving only the scalar electric potential can be interpreted in terms of the charge relaxation process. The fundamental equations involved are *Ohm's law* ($\mathbf{J} = \sigma\mathbf{E}$) the *equation of continuity*

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\mathbf{J} = 0$$

and *Gauss' law*

$$\nabla\cdot(\varepsilon\mathbf{E}) = \rho$$

By combining these, one can deduce the following differential equation for the space charge density in a homogeneous medium

$$\frac{\partial\rho}{\partial t} + \frac{\sigma}{\varepsilon}\rho = 0$$

This equation has the solution

$$\rho(t) = \rho_0 e^{-t/\tau}$$

where

$$\tau = \frac{\varepsilon}{\sigma}$$

is called the charge relaxation time. For a good conductor like copper, τ is of the order of 10^{-19} s whereas for a good insulator like silica glass, it is of the order of 10^3 s. For a pure insulator, it becomes infinite.

When modeling real world devices, there is not only the intrinsic time scale of charge relaxation time but also an external time scale t at which a device is energized or the observation time. It is the relation between the external time scale and the charge relaxation time that determines what physics interface and study type to use. The results are summarized in Table 9-1 below,

TABLE 9-1: SUITABLE PHYSICS INTERFACE AND STUDY TYPE FOR DIFFERENT TIME-SCALE REGIMES.

CASE	PHYSICS INTERFACE	STUDY TYPE
τ>>t	Electrostatics	Stationary
τ<<t	Electric Currents	Stationary
τ~t	Electric Currents	Time Dependent or Frequency Domain (in AC/DC Module or MEMS Module)

FIRST CASE: τ >> T

If the external time scale is short compared to the charge relaxation time, the charges do not have time to redistribute to any significant degree.Thus the charge distribution can be considered as given model input and the best approach is to solve the *Electrostatics* formulation using the electric potential V.

By combining the definition of the potential with Gauss' law, 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 a variant of Poisson's equation

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

This equation is used in the Electrostatics interface. It is worth noting that Gauss' law does not require the charge distribution to be static. Thus, provided dynamics are slow enough that induced electric fields can be neglected and hence a scalar electric potential is justified, the formulation can be used also in the Time Dependent study type. That typically involves either prescribing the charge dynamics or coupling a separate formulation for this.

Tip

Such separate charge transport formulations can be found in the Plasma Module and the Chemical Reaction Engineering Module.

SECOND CASE: $\tau << T$

If the external time scale is long compared to the charge relaxation time, the stationary solution to the equation of continuity has been reached. In a stationary coordinate system, a slightly more general form than above of *Ohm's law* states that

$$\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}^{\mathrm{e}}$$

where $\mathbf{J}^{\mathrm{e}}$ is an externally generated current density. The static form of the equation of continuity then reads

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma\nabla V - \mathbf{J}^{\mathrm{e}}) = 0$$

To handle current sources the equation can be generalized to

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

This equation is used in the static study type for the Electric Currents interface.

Theory for the Electrostatics Interface

The Electrostatics Interface is available for 3D, 2D in-plane, and 2D axisymmetric models. Applications with Electrostatics Equations include high-voltage apparatus, electronic devices, and capacitors. The term "statics" is not to be interpreted literally—it is the observation time or time scale at which the applied excitation changes is short compared to the charge relaxation time and that the electromagnetic wavelength and skin depth are very large compared to the size of the domain of interest.

If you do not know whether to use the Electric Currents or the Electrostatics interface, which both solve for the scalar electric potential V, consider using an explicit charge transport model. See Charge Relaxation Theory.

Electrostatics Equations

Under static conditions the electric potential, V, is defined by the relationship

$$\mathbf{E} = -\nabla V$$

Combining this equation with the constitutive relationship $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$ between the electric displacement $\mathbf{D}$ and the electric field $\mathbf{E}$, it is possible to represent Gauss' law as the following equation:

$$-\nabla \cdot (\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

In this equation, the physical constant, ε_0 (SI unit: F/m) is the permittivity of vacuum, $\mathbf{P}$ (SI unit: C/m^2) is the electric polarization vector, and ρ (SI unit: C/m^3) is a space charge density. This equation describes the electrostatic field in dielectric materials.

For in-plane 2D modeling, the Electrostatics interface 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. With this symmetry, the same equation is solved as in the 3D case. The interface solves the following equation where d is the thickness in the z direction:

$$-\nabla \cdot d(\varepsilon_0 \nabla V - \mathbf{P}) = \rho$$

The axisymmetric version of the interface considers the situation where the fields and 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.

Theory for the Electric Currents Interface

The Electric Currents Interface solves a current conservation problem for the scalar electric potential V and is available for 3D, 2D in-plane, and 2D axisymmetric models. Electrolysis and the computation of resistances of grounding plates are examples that involve conductive media with electrical conductivities and electric currents. If you are uncertain whether to use the Electric Currents interface or the Electrostatics interface, which both solve for the scalar electric potential V, refer to the section on Charge Relaxation Theory. In this section Electric Currents Equations in Steady State are discussed.

Electric Currents Equations in Steady State

When handling stationary electric currents in conductive media you must consider the stationary equation of continuity. In a stationary coordinate system, the point form of *Ohm's law* states that $\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_\mathrm{e}$ where σ is the electrical conductivity (SI unit: S/m), and $\mathbf{J}_\mathrm{e}$ is an *externally generated current density* (SI unit: $\mathrm{A/m^2}$). The static form of the equation of continuity then states

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma\nabla V - \mathbf{J}_\mathrm{e}) = 0$$

To handle *current sources*, you can generalize the equation to

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

In planar 2D the Electric Currents interface assumes that the 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 xy-plane. The Electric Currents interface then solves the following equation where d is the thickness in the z direction:

$$-\nabla \cdot d(\sigma\nabla V - \mathbf{J}_\mathrm{e}) = dQ_j \tag{9-2}$$

In 2D axisymmetry, the Electric Currents interface considers the situation where the fields and 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.

Theory of Magnetic and Electric Fields

Quasi-static analysis of magnetic and electric fields is valid under the assumption that $\partial\mathbf{D}/\partial t = 0$. In this section:

- Maxwell's Equations
- Magnetic and Electric Potentials
- Gauge Transformations
- Selecting a Particular Gauge
- The Gauge and the Equation of Continuity for Dynamic Fields
- Time-Harmonic Magnetic Fields

Maxwell's Equations

This implies that it is possible to rewrite Maxwell's equations in the following manner:

$$\begin{aligned}
\nabla\times\mathbf{H} &= \mathbf{J} = \sigma(\mathbf{E}+\mathbf{v}\times\mathbf{B})+\mathbf{J}_\mathrm{e}\\
\nabla\times\mathbf{E} &= -\frac{\partial\mathbf{B}}{\partial t}\\
\nabla\cdot\mathbf{B} &= 0\\
\nabla\cdot\mathbf{D} &= \rho\\
\nabla\cdot\mathbf{J} &= 0
\end{aligned}$$

Here $\mathbf{J}_\mathrm{e}$ is an externally generated current density and $\mathbf{v}$ is the velocity of the conductor. The crucial criterion for the quasi-static approximation to be valid is that the currents and the electromagnetic fields vary slowly. This means that the dimensions of the structure in the problem need to be small compared to the wavelength.

Magnetic and Electric Potentials

Using the definitions of the potentials,

$$\begin{aligned}
\mathbf{B} &= \nabla\times\mathbf{A}\\
\mathbf{E} &= -\nabla V-\frac{\partial\mathbf{A}}{\partial t}
\end{aligned}$$

and the constitutive relation $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$, Ampère's law can be rewritten as

$$\sigma\frac{\partial \mathbf{A}}{\partial t} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma\mathbf{v} \times (\nabla \times \mathbf{A}) + \sigma\nabla V = \mathbf{J}_\mathrm{e} \qquad (9\text{-}3)$$

The equation of continuity, which is obtained by taking the divergence of the above equation, adds the following equation:

$$\nabla \cdot \left(-\sigma\frac{\partial \mathbf{A}}{\partial t} + \sigma\mathbf{v} \times (\nabla \times \mathbf{A}) - \sigma\nabla V + \mathbf{J}_\mathrm{e}\right) = 0 \qquad (9\text{-}4)$$

Equation 9-3 and Equation 9-4 form a system of equations for the two potentials **A** and *V*.

Gauge Transformations

The electric and magnetic potentials are not uniquely defined from the electric and magnetic fields through

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V$$
$$\mathbf{B} = \nabla \times \mathbf{A}$$

Introducing two new potentials

$$\tilde{\mathbf{A}} = \mathbf{A} + \nabla\Psi$$
$$\tilde{V} = V - \frac{\partial \Psi}{\partial t}$$

gives the same electric and magnetic fields:

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V = -\frac{\partial(\tilde{\mathbf{A}} - \nabla\Psi)}{\partial t} - \nabla\left(\tilde{V} + \frac{\partial \Psi}{\partial t}\right) = -\frac{\partial \tilde{\mathbf{A}}}{\partial t} - \nabla\tilde{V}$$
$$\mathbf{B} = \nabla \times \mathbf{A} = \nabla \times (\tilde{\mathbf{A}} - \nabla\Psi) = \nabla \times \tilde{\mathbf{A}}$$

The variable transformation of the potentials is called a *gauge transformation*. To obtain a unique solution, choose the gauge, that is, put constraints on Ψ that make the solution unique. Another way of expressing this additional condition is to put a constraint on $\nabla \cdot \mathbf{A}$. A vector field is uniquely defined up to a constant if both $\nabla \cdot \mathbf{A}$ and $\nabla \times \mathbf{A}$ are given. This is called *Helmholtz's theorem*.

One particular gauge is the *Coulomb gauge* given by the constraint: $\nabla \cdot \mathbf{A} = 0$.

Selecting a Particular Gauge

Important observations are that in the dynamic case **A** and V are coupled via the selected gauge. For a dynamic formulation, it is also possible to select a Ψ such that the scalar electric potential vanishes and only the magnetic vector potential has to be considered. The dynamic formulations (frequency domain and time dependent study types) of the Magnetic Fields interface are operated in this gauge as it involves only **A**. In the static limit, **A** and V are not coupled via the gauge selection and thus any gauge can be chosen for **A** when performing magnetostatic modeling.

The Gauge and the Equation of Continuity for Dynamic Fields

After eliminating the electric potential by choosing the appropriate gauge and disregarding the velocity term. The equation of continuity obtained by taking the divergence of Ampère's law reads:

$$\nabla\cdot\left(-\sigma\frac{\partial\mathbf{A}}{\partial t}+\mathbf{J}_{\mathrm{e}}\right) = 0$$

It is clear that unless the electrical conductivity is uniform, the particular gauge used to eliminate V cannot be the Coulomb gauge as that would violate the equation of continuity and would thereby also violate Ampère's law.

Time-Harmonic Magnetic Fields

In the time-harmonic case, there is no computational cost for including the displacement current in Ampère's law (then called Maxwell-Ampère's law):

$$\nabla\times\mathbf{H} = \mathbf{J} = \sigma(\mathbf{E}+\mathbf{v}\times\mathbf{B})+j\omega\mathbf{D}+\mathbf{J}^{\mathrm{e}}$$

In the transient case the inclusion of this term would lead to a second-order equation in time, but in the harmonic case there are no such complications. Using the definition of the electric and magnetic potentials, the system of equations becomes

$$-\nabla\cdot((j\omega\sigma-\omega^2\varepsilon_0)\mathbf{A}-\sigma\mathbf{v}\times(\nabla\times\mathbf{A})+(\sigma+j\omega\varepsilon_0)\nabla V-(\mathbf{J}^{\mathrm{e}}+j\omega\mathbf{P})) = 0$$

$$(j\omega\sigma-\omega^2\varepsilon_0)\mathbf{A}+\nabla\times(\mu_0^{-1}\nabla\times\mathbf{A}-\mathbf{M})-\sigma\mathbf{v}\times(\nabla\times\mathbf{A})+(\sigma+j\omega\varepsilon_0)\nabla V = \mathbf{J}^{\mathrm{e}}+j\omega\mathbf{P}$$

The constitutive relation $\mathbf{D}=\varepsilon_0\mathbf{E}+\mathbf{P}$ has been used for the electric field.

To obtain a particular gauge that reduces the system of equation, choose $\Psi = -jV/\omega$ in the gauge transformation. This gives

$$\tilde{\mathbf{A}} = \mathbf{A} - \frac{j}{\omega}\nabla V \qquad\qquad \tilde{V} = 0$$

When $\tilde{V}$ vanishes from the equations, only the second one is needed,

$$(j\omega\sigma - \omega^2\varepsilon_0)\tilde{\mathbf{A}} + \nabla \times (\mu_0^{-1}\nabla \times \tilde{\mathbf{A}} - \mathbf{M}) - \sigma\mathbf{v} \times (\nabla \times \tilde{\mathbf{A}}) = \mathbf{J}^{\mathrm{e}} + j\omega\mathbf{P}$$

Working with $\tilde{\mathbf{A}}$ is often the best option when it is possible to specify all source currents as external currents $\mathbf{J}^{\mathrm{e}}$ or as surface currents on boundaries.

Theory for the Magnetic Fields Interface

Simulation of magnetic fields is of interest when studying magnets, motors, transformers, and conductors carrying static or alternating currents. The Magnetic Fields Interface is used for 3D, 2D in-plane, and 2D axisymmetric models. Unless you have a license for the AC/DC Module, only 2D modeling involving out-of-plane currents and axisymmetric modeling involving azimuthal currents are supported.

In this section:

- Magnetostatics Equation
- Frequency Domain Equation
- Transient Equation

See Also

For a deeper theoretical background to the magnetic vector potential used below, see the section starting with Maxwell's Equations.

Magnetostatics Equation

To derive the magnetostatic equation, start with Ampère's law for static cases $\nabla \times \mathbf{H} = \mathbf{J}$. The current is

$$\mathbf{J} = \sigma \mathbf{v} \times \mathbf{B} + \mathbf{J}^{\mathrm{e}}$$

where $\mathbf{J}^{\mathrm{e}}$ is an externally generated current density, and $\mathbf{v}$ is the velocity of the conductor.

Using the definitions of magnetic potential, $\mathbf{B} = \nabla \times \mathbf{A}$ and the constitutive relationship, $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$, rewrite Ampère's law as

$$\nabla \times (\mu_0^{-1} \nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}^{\mathrm{e}}$$

which is the equation used in magnetostatics.

Frequency Domain Equation

To derive the time harmonic equation this physics interface solves, start with Ampère's law including displacement currents (then called Maxwell-Ampère's law) as these do not involve any extra computational cost in the frequency domain,

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} = \sigma \mathbf{E} + \sigma \mathbf{v} \times \mathbf{B} + \mathbf{J}^{e} + \frac{\partial \mathbf{D}}{\partial t}$$

Now assume time-harmonic fields and use the definitions of the fields,

$$\begin{aligned} \mathbf{B} &= \nabla \times \mathbf{A} \\ \mathbf{E} &= -j\omega \mathbf{A} \end{aligned}$$

and combine them with the constitutive relationships $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$ and $\mathbf{D} = \varepsilon_0 \mathbf{E}$ to rewrite Ampère's law as

$$(j\omega\sigma - \omega^2\varepsilon_0)\mathbf{A} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}^{\mathrm{e}}$$

Transient Equation

The transient equation this physics interface solves is Ampère's law, here illustrated with the constitutive relation $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$.

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}_{\mathrm{e}}$$

The Electrostatics Interface

The **Electrostatics** interface ( ), found under the **AC/DC** branch ( ) in the **Model Wizard**, has the equations, boundary conditions, and space charges for modeling electrostatic fields, solving for the electric potential.

Charge Conservation is the main feature, which adds the equation for the electric potential and has a settings window for defining the constitutive relation for the electric displacement field and its associated properties such as the relative permittivity.

When this interface is added, these default nodes are also added to the **Model Builder**—**Charge Conservation**, **Zero Charge** (default boundary condition), and **Initial Values**. Right-click the **Electrostatics** node to add other features that implement, for example, boundary conditions and space charges.

Model

Electric Sensor: Model Library path **COMSOL_Multiphysics/Electromagnetics/electric_sensor**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `es`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the electric potential and the equations that describe the potential field for dielectrics. To choose specific domains, select **Manual** from the **Selection** list.

THICKNESS

1D

Enter a default value for the **Cross-section area** A (SI unit: m^2). The default value of 1 is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models. See also Change Cross-Section.

2D

Enter a default value for the **Out-of-plane thickness** d (SI unit: m). The default value of 1 is typically not representative for a thin dielectric medium, for example. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models. See also Change Thickness (Out-of-Plane).

DISCRETIZATION

To display this section, select click the **Show** button ( ) and select **Discretization**. Select an element order for the **Electric potential**—**Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or (in 2D only) **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

DEPENDENT VARIABLES

The dependent variable is the **Electric potential** V. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another electric potential field in the model, the interfaces will share degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Conditions for the Electrostatics Interface
- Theory for the Electrostatics Interface

Domain, Boundary, Edge, Point, and Pair Conditions for the Electrostatics Interface

The Electrostatics Interface has these domain, boundary, edge, point, and pair features available.

About the Boundary Conditions

The relevant interface condition at interfaces between different media is

$$\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

In the absence of surface charges, this condition is fulfilled by the natural boundary condition

$$\mathbf{n} \cdot [(\varepsilon_0 \nabla V - \mathbf{P})_1 - (\varepsilon_0 \nabla V - \mathbf{P})_2] = -\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

Available Features

These features are available for this interface and listed in alphabetical order. Also see Table 9-2 for a list of interior and exterior boundary conditions, including edge, point, and pair availability.

- Change Cross-Section
- Change Thickness (Out-of-Plane)
- Charge Conservation
- Electric Displacement Field
- Electric Potential
- External Surface Charge Accumulation
- Floating Potential
- Ground
- Initial Values
- Line Charge
- Line Charge (on Axis)
- Line Charge (Out-of-Plane)
- Periodic Condition
- Point Charge
- Point Charge (on Axis)
- Space Charge Density
- Surface Charge Density
- Thin Low Permittivity Gap
- Zero Charge (the default boundary condition)

Table 9-2 lists the interior and exterior boundaries available with this interface. It also includes edge, point, and pair availability.

TABLE 9-2: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTROSTATICS INTERFACE

FEATURE	INTERIOR	EXTERIOR	ALSO AVAILABLE FOR
Change Cross-Section	x	x	pairs
Change Thickness (Out-of-Plane)	x	x	pairs
Electric Displacement Field	x	x	pairs
Electric Potential	x	x	edges, points, and pairs
External Surface Charge Accumulation		x	pairs
Floating Potential	x	x	pairs
Ground	x	x	edges, points, and pairs
Periodic Condition		x	not applicable
Surface Charge Density	x	x	pairs
Thin Low Permittivity Gap	x		not applicable
Zero Charge (the default)	x	x	pairs

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** feature to the model that is valid on the axial symmetry boundaries only. There are also **Line Charge (on Axis)** and **Point Charge (on Axis)** features available.

See Also

- Continuity on Interior Boundaries
- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Charge Conservation

The **Charge Conservation** node adds the equations for charge conservation according to Gauss' law for the electric displacement field. It provides an interface for defining the constitutive relation and its associated properties such as the relative permittivity.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the electric potential and the equation based on Gauss' law that describes the potential field.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid**, or **From material**.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

ELECTRIC FIELD

Select a **Constitutive relation** to describe the macroscopic properties of the medium (relating the electric displacement $\mathbf{D}$ with the electric field $\mathbf{E}$) and the applicable material properties, such as the relative permittivity. Select:

- **Relative permittivity** to use the constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E}$ (the default).
- **Polarization** to use the constitutive relation $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$.
- **Remanent electric displacement** to use constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E} + \mathbf{D}_r$, where $\mathbf{D}_r$ is the remanent displacement (the displacement when no electric field is present).
  - If **Relative permittivity** is selected, the default is to take the **Relative permittivity** (ε_r) values **From material**. If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix.
  - If **Polarization** is selected, enter the components based on space dimension for the **Polarization** vector $\mathbf{P}$ (SI unit: C/m^2).
  - If **Remanent electric displacement** is selected, the default is to take the **Relative permittivity** (ε_r) values **From material**. If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or

matrix. Then enter the components based on space dimension for the **Remanent electric displacement** $\mathbf{D}_\mathrm{r}$ (SI unit: C/m^2).

Space Charge Density

The **Space Charge Density** node adds a space charge density ρ, which appears on the right-hand side of the equation that the interface defines.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define a space charge density.

SPACE CHARGE DENSITY

Enter a value or expression for the **Space charge density** ρ_q (SI unit: C/m^3).

Initial Values

The **Initial Values** node adds an initial value for the electric potential V that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the **Electric potential** V. The default value is 0 V.

Ground

The **Ground** node is the default boundary condition and implements ground as the boundary condition $V = 0$.

Ground means that there is a zero potential on the boundary. This boundary condition is also applicable at symmetry boundaries where the potential is known to be antisymmetric with respect to the boundary.

For some interfaces, also select additional **Ground** features from the **Edges** (3D models) or **Points** (2D and 3D models) submenus. For 2D axisymmetric models, it can be applied on the Symmetry axis.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (boundaries, edges, or points) to apply a ground (zero potential) boundary condition.

⚠ Caution

Beware that constraining the potential on edges or points in 3D or on points in 2D usually yields a current outflow that is mesh dependent.

PAIR SELECTION

If **Ground** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Electric Potential

The **Electric Potential** node provides an electric potential V_0 as the boundary condition $V = V_0$.

Because the electric potential is being solved for in the interface, the value of the potential is typically defined at some part of the geometry. For some interfaces, also select additional **Electric Potential** features from the **Edges** (3D models) or **Points** (2D and 3D models) submenus. For 2D axisymmetric models, it can be applied on the Symmetry axis.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (boundaries, edges, or points) to apply an electric potential as the boundary condition.

⚠ Caution

Beware that constraining the potential on edges or points in 3D or on points in 2D usually yields a current outflow that is mesh dependent.

PAIR SELECTION

If **Electric Potential** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

ELECTRIC POTENTIAL

Enter the value or expression for the **Electric potential** V_0 (SI unit: V).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Surface Charge Density

The **Surface Charge Density** node provides the following surface-charge boundary condition for exterior boundaries (left) and interior boundaries (right):

$$-\mathbf{n}\cdot\mathbf{D} = \rho_s, \qquad \mathbf{n}\cdot(\mathbf{D}_1-\mathbf{D}_2) = \rho_s$$

Specify the surface charge density ρ_s at an outer boundary or at an interior boundary between two nonconducting media.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a surface charge density.

PAIR SELECTION

If **Surface Charge Density** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

SURFACE CHARGE DENSITY

Enter the value or expression for the **Surface charge density** ρ_s (SI unit: C/m^2).

External Surface Charge Accumulation

The **External Surface Charge Accumulation** node implements the boundary condition

$$-\mathbf{n}\cdot\mathbf{D} = \rho_s$$

where ρ_s is the solution of the following distributed ODE on the boundary:

$$\frac{d\rho_s}{dt} = \mathbf{n}\cdot\mathbf{J}_i + \mathbf{n}\cdot\mathbf{J}_e$$

where $\mathbf{n}\cdot\mathbf{J}_i$ is the normal component of the total ion current density on the wall and $\mathbf{n}\cdot\mathbf{J}_e$ is the normal component of the total electron current density on the wall, which are feature inputs.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a surface charge accumulation.

PAIR SELECTION

If **External Surface Charge Accumulation** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

EXTERNAL SURFACE CHARGE ACCUMULATION

Enter values or expressions for the **Normal ion current density** $\mathbf{n}\cdot J_i$ (SI unit: A/m^2) and the **Normal electron current density** $\mathbf{n}\cdot J_e$ (SI unit: A/m^2).

Floating Potential

The **Floating Potential** node is used when modeling a metallic electrode at floating potential. The electrode may have a charge Q_0 deposited on it.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the floating electrode.

PAIR SELECTION

If **Floating Potential** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

FLOATING POTENTIAL

Specify a an optionally non zero **Charge** Q_0 (SI unit: C).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Electric Displacement Field

The **Electric Displacement Field** node adds the following electric-displacement boundary condition:

$$\mathbf{n} \cdot \mathbf{D} = \mathbf{n} \cdot \mathbf{D}_0$$

It specifies the normal component of the electric displacement field at a boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to use the normal component of the displacement field as the boundary condition.

PAIR SELECTION

If **Displacement Field** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

ELECTRIC DISPLACEMENT FIELD

Enter the coordinates of the **Boundary electric displacement field** $\mathbf{D}_0$ (SI unit: $\mathrm{C/m^2}$).

Periodic Condition

The **Periodic Condition** node defines periodicity or antiperiodicity between two boundaries. If required, activate periodic conditions on more than two boundaries, in which case the Periodic Condition tries to identify two separate surfaces that can each consist of several connected boundaries.

Tip

For more complex geometries it might be necessary to use the **Destination Selection** subnode. With this subnode the boundaries which constitute the source and destination surfaces can be manually specified. To add the subnode, right-click the **Periodic Condition** node and select **Destination Selection**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a periodic condition.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** or **Antiperiodicity**.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

- Periodic Condition
- Destination Selection
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Zero Charge

The **Zero Charge** node adds the condition that there is zero charge on the boundary so that $\mathbf{n} \cdot \mathbf{D} = 0$. This boundary condition is also applicable at symmetry boundaries where the potential is known to be symmetric with respect to the boundary. This is the default boundary condition at exterior boundaries. At interior boundaries, it means that no displacement field can penetrate the boundary and that the electric potential is discontinuous across the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a zero charge condition.

PAIR SELECTION

If **Zero Charge** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Thin Low Permittivity Gap

Use the **Thin Low Permittivity Gap** node

$$\mathbf{n} \cdot \mathbf{D}_1 = \frac{\varepsilon_0 \varepsilon_\mathrm{r}}{d_s}(V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{D}_2 = \frac{\varepsilon_0 \varepsilon_\mathrm{r}}{d_s}(V_2 - V_1)$$

to model a thin gap of a material with a small permittivity compared to the adjacent domains. The layer has the thickness d_s and the relative permittivity ε_r. The indices 1 and 2 refer to the two sides of the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a thin low permittivity gap condition.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid**, or **From material**.

PAIR SELECTION

When **Thin Low Permittivity Gap** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

THIN LOW PERMITTIVITY GAP

The default is to take the **Relative permittivity** (ε_r) values **From material**. Select **User defined** to enter a different value or expression.

Enter a **Surface thickness** d_s (SI unit: m).

Line Charge

3D

For 3D models, use the **Line Charge** node to specify line charges along the edges of a geometry.

EDGE SELECTION

From the **Selection** list, choose the edges to add a line charge.

Caution

Beware that constraining the potential on edges usually yields a current outflow that is mesh dependent.

LINE CHARGE

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length.

See Also

- Line Charge (on Axis)
- Line Charge (Out-of-Plane)

Line Charge (on Axis)

2D Axi

For 2D axisymmetric models, use the **Line Charge (on Axis)** node to specify line charges along the symmetry axis.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries on axis to add a line charge.

LINE CHARGE (ON AXIS)

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length.

See Also

- Line Charge
- Line Charge (Out-of-Plane)

Line Charge (Out-of-Plane)

2D

2D Axi

For 2D and 2D axisymmetric models, points are selected and this is the same as a line out-of-plane.

Use the **Line Charge (Out-of-Plane)** node to specify line charges along the points of a geometry for 2D and 2D axisymmetric models.

POINT SELECTION

From the **Selection** list, choose the points to add a line charge.

Caution

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

LINE CHARGE (OUT-OF-PLANE)

Enter a value or expression to apply a **Line charge** Q_L (SI unit: C/m). This source represents electric charge per unit length.

See Also

- Line Charge
- Line Charge (on Axis)

Point Charge

3D

The **Point Charge** node adds a point source to 3D models. The point charge represents an electric displacement field flowing out of the point.

POINT SELECTION

From the **Selection** list, choose the points to add a point charge.

Caution

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CHARGE

Enter a value or expression to apply a **Point charge** Q_P (SI unit: C) to points. This source represents an electric displacement field flowing out of the point.

See Also	- Point Charge (on Axis) - Line Charge (Out-of-Plane)

Point Charge (on Axis)

2D Axi	The **Point Charge (on Axis)** node adds a point source to 2D axisymmetric models. The point charge represents an electric displacement field flowing out of the point.

POINT SELECTION

From the **Selection** list, choose the points to add a point charge.

Caution	Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CHARGE (ON AXIS)

Enter a value or expression to apply a **Point charge** Q_P (SI unit: C) to points on axis. This source represents an electric displacement field flowing out of the point.

See Also	- Point Charge - Line Charge (Out-of-Plane)

Change Cross-Section

1D	This feature is available with 1D models. This setting overrides the global **Thickness** setting made in any interface that uses this feature.

Use the **Change Cross-Section** feature to set the cross-section area for specific geometric entities.

DOMAIN OR BOUNDARY SELECTION

From the **Selection** list, choose the geometric entity (domains or boundaries) to define the change cross-section.

PAIR SELECTION

When **Change Cross-Section** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CHANGE CROSS-SECTION

Enter a value or expression for the **Cross-section area** A (SI unit: m^2). The default value of 1 unit length is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models.

See Also	For 2D models, see Change Thickness (Out-of-Plane).

Change Thickness (Out-of-Plane)

2D	This feature is available for 2D models. This setting overrides the global **Thickness** setting made in any interface that uses this feature.

Use the **Change Thickness (Out-of-Plane)** feature to set the out-of-plane thickness for specific geometric entities.

DOMAIN OR BOUNDARY SELECTION

From the **Selection** list, choose the geometric entity (domains or boundaries) to define the change thickness.

PAIR SELECTION

When **Change Thickness (Out-of-Plane)** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CHANGE THICKNESS (OUT-OF-PLANE)

Enter a value or expression for the **Out-of-plane thickness** d (SI unit: m). The default value of 1 unit length is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models.

See Also	For 1D models, see Change Cross-Section.

Infinite Elements

Note	For information about this feature, see About Infinite Element Domains and Perfectly Matched Layers.

The Electric Currents Interface

The **Electric Currents** interface ( ), found under the **AC/DC** branch ( ) in the **Model Wizard**, has the equations, boundary conditions, and current sources for modeling electric currents in conductive media, solving for the electric potential.

Current Conservation is the main feature, which adds the equation for the electric potential and provides a settings window for defining the electrical conductivity as well as the constitutive relation for the electric displacement field and its associated material properties such as the relative permittivity.

When this interface is added, these default nodes are also added to the **Model Builder**—**Current Conservation**, **Electric Insulation** (the default boundary condition), and **Initial Values**. Right-click the **Electric Currents** node to add other features that implement, for example, boundary conditions and current sources.

Model

Pacemaker Electrode: Model Library path **COMSOL_Multiphysics/Electromagnetics/pacemaker_electrode**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `ec`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the electric potential and the equations that describe the potential field for conductive media. To choose specific domains, select **Manual** from the **Selection** list.

Thickness (1D)/ Out-of-Plane Thickness (2D)

1D

Enter a default value for the **Cross-section area** A (SI unit: m^2). The default value of 1 is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 1D equation identical to the equation used for 3D models. See also Change Cross-Section (described for the **Electrostatics** interface).

2D

Enter a default value for the **Out-of-plane thickness** d (SI unit: m) (see Equation 9-2). The default value of 1 is typically not representative for a thin dielectric medium, for example. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models. See also Change Thickness (Out-of-Plane) (described for the **Electrostatics** interface).

Dependent Variables

The dependent variable is the **Electric potential** V. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another electric potential field in the model, the interfaces will share degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

Discretization

To display this section, click the **Show** button ( ) and select **Discretization**. Select an **Electric potential**—**Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or (in 2D only) **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Features for the Electric Currents Interface
- Theory for the Electric Currents Interface

Domain, Boundary, Edge, Point, and Pair Features for the Electric Currents Interface

The Electric Currents Interface has these domain, boundary, edge, point, and pair conditions available.

About the Boundary Conditions

The exterior and interior boundary conditions listed in Table 9-2 are available. The relevant interface condition at interfaces between different media and interior boundaries is continuity; that is,

$$\mathbf{n}_2 \cdot (\mathbf{J}_1 - \mathbf{J}_2) = 0$$

which is the natural boundary condition.

Available Features

These features are available for this interface and listed in alphabetical order. Also see Table 9-2 for a list of interior and exterior boundary conditions, including edge, point, and pair availability.

- Boundary Current Source
- Contact Impedance
- Current Conservation
- Current Source
- Distributed Impedance
- Electric Insulation
- External Current Density
- Floating Potential
- Initial Values
- Line Current Source
- Line Current Source (on Axis)
- Normal Current Density
- Point Current Source
- Sector Symmetry

These features are described for the **Electrostatics** interface:

- Change Cross-Section
- Change Thickness (Out-of-Plane)

- Electric Potential
- Ground
- Periodic Condition

See Also

- Continuity on Interior Boundaries
- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Table 9-2 lists the interior and exterior boundaries available with this interface. It also includes edge, point, and pair availability.

TABLE 9-3: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTRIC CURRENTS INTERFACE

FEATURE	INTERIOR	EXTERIOR	ALSO AVAILABLE FOR
Boundary Current Source	x		pairs
Contact Impedance	x		pairs
Distributed Impedance	x	x	not applicable
Electric Insulation	x	x	pairs
Electric Potential	x	x	edges, points, and pairs
Floating Potential	x	x	pairs
Ground	x	x	edges, points, and pairs
Normal Current Density		x	not applicable
Periodic Condition		x	not applicable

Current Conservation

The **Current Conservation** node adds the continuity equation for the electrical potential and provides an interface for defining the electric conductivity as well as the constitutive relation and the relative permittivity for the displacement current.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the electric potential and the continuity equation that describes the potential field.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty. If a linear temperature relation is added for the conductivity, then the source for the temperature T can be defined. From the **Temperature** list, select an existing temperature variable (from another physics interface) if available, or select **User defined** to define a value or expression for the temperature (SI unit: K) in the field that appears underneath the list.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

CONDUCTION CURRENT

By default, the **Electrical conductivity** σ (SI unit: S/m) for the media is defined **From material**. Or select **User defined**, **Linearized resistivity**, or **Porous media**.

- If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** depending on the characteristics of the electrical conductivity, and then enter values or expressions in the field or matrix.

Note

If another type of temperature dependence is used other than a linear temperature relation, enter any expression for the conductivity as a function of temperature.

- Select **Linearized resistivity** for a temperature-dependent conductivity (this occurs in, for example, Joule heating, and is also called resistive heating). The equation describing the conductivity:

$$\sigma = \frac{1}{\rho_0(1+\alpha(T-T_0))}$$

where ρ_0 is the resistivity at the reference temperature T_0, and α is the temperature coefficient of resistance, which describes how the resistivity varies with temperature.

  - The default **Reference temperature** T_{ref} (SI unit: K), **Resistivity temperature coefficient** α (SI unit: 1/K), and **Reference resistivity** ρ_0 (SI unit: Ωm) are taken **From material**, which means that the values are taken from the domain material.

Note

Only certain material models (see Materials) support the **Linearized resistivity**.

  - To specify other values for any of these properties, select **User defined** from the list and then enter a value or expression. T is the current temperature, which can be a value that is specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field appears in the **Model Inputs** section.

ELECTRIC FIELD

Note

See **Electric Field** as described for the Charge Conservation node for the Electrostatics interface.

Floating Potential

The **Floating Potential** node is useful when modeling a metallic electrode at floating potential. The electrode may have a charge Q_0 deposited on it.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the floating electrode.

PAIR SELECTION

If **Floating Potential** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

FLOATING POTENTIAL

Specify a an optionally non zero **Charge** Q_0 (SI unit: C).

Note

For the **Electric Currents** interface, enter a **Terminal current** I_0 (SI unit: A). Specify zero current for a disconnected electrode.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

External Current Density

The **External Current Density** node adds an externally generated current density $\mathbf{J}_\mathrm{e}$, which appears in Ohm's law

$$\mathbf{J} = \sigma\mathbf{E} + \mathbf{J}_\mathrm{e}$$

and in the equation that the interface defines.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an external current density.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

EXTERNAL CURRENT DENSITY

Based on space dimension, enter the coordinates (**x**, **y**, and **z** for 3D models for example) of the **External current density** $\mathbf{J}_e$ (SI unit: A/m^2).

Current Source

The **Current Source** node adds a distributed current source Q_{j} in the equation that the interface defines. Use this feature with caution as it may violate the current conservation law that is inherent in Maxwell-Ampère’s law.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define a current source.

CURRENT SOURCE

Enter a value or expression for the **Current source** Q_j (SI unit: A/m^3).

Initial Values

The **Initial Values** node adds an initial value for the electric potential that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. If more than one set of initial values is required, right-click to add additional **Initial Values** features.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the **Electric potential** V (SI unit: V). The default value is 0 V.

Boundary Current Source

The **Boundary Current Source** node adds a current source Q_{j} on the boundary.

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = Q_{\mathrm{j}}$$

It is applicable to interior boundaries that represent either a source or a sink of current.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a current source.

PAIR SELECTION

If **Boundary Current Source** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

BOUNDARY CURRENT SOURCE

Enter a value or expression for the **Boundary current source** Q_j (SI unit: $\mathrm{A/m^2}$).

Normal Current Density

The **Normal Current Density** node is applicable to exterior boundaries that represent either a source or a sink of current. It provides a condition for specifying the normal current density as an inward or outward current flow:

$$-\mathbf{n}\cdot\mathbf{J} = J_n$$

Or, alternatively, as a current density $\mathbf{J}_0$:

$$\mathbf{n}\cdot\mathbf{J} = \mathbf{n}\cdot\mathbf{J}_0$$

The normal current density is positive when the current flows inward in the domain.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a current flow as the boundary condition using the normal current density.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

NORMAL CURRENT DENSITY

Select an option from the **Type** list—**Inward current density** or **Current density**.

- If **Inward current density** is selected, enter a value or expression for the **Normal current density** J_n (SI unit: $\mathrm{A/m^2}$). Use a positive value for an inward current flow or a negative value for an outward current flow.
- If **Current density** is selected, enter values or expressions for the components of the **Current density** (SI unit: $\mathrm{A/m^2}$) in the $\mathbf{J}_0$ fields.

Distributed Impedance

The **Distributed Impedance** node adds a distributed impedance boundary condition to a model.

Tip

Use this boundary condition to model a thin sheet of a resistive material, connected to a reference potential V_{ref}.

The layer impedance can be specified either with the bulk material conductivity σ_s, the relative permittivity ε_{r} and the layer thickness d_s, or directly with the surface resistance ρ_{s} and capacitance C_s. Assuming DC currents, the equation is:

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = \frac{\sigma_s}{d_s}(V-V_{\mathrm{ref}})$$

$$\mathbf{n}\cdot(\mathbf{J}_1-\mathbf{J}_2) = \frac{1}{\rho_s}(V-V_{\mathrm{ref}})$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a distributed impedance.

DISTRIBUTED IMPEDANCE

Enter the **Reference potential** V_{ref} (SI unit: V).

Select a potentially complex valued **Layer specification**—**Thin layer** (the default) or **Surface impedance**.

- If **Thin layer** is selected, enter values or expressions for the:
  - **Surface thickness** d_{s} (SI unit: m). The default is $5{\cdot}10^{-3}$ m (5 mm).
  - **Electrical conductivity** σ (SI unit: S/m) and **Relative permittivity** ε_{r}. The defaults take values **From material**. Select **User defined** to enter different values or expressions.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: $\Omega\mathrm{m}^2$) and for the **Surface capacitance** C_{s} (SI unit: $\mathrm{F/m}^2$).

Electric Insulation

Electric Insulation is the default boundary condition and this feature adds electric insulation as the boundary condition:

$$\mathbf{n} \cdot \mathbf{J} = 0$$

This boundary condition means that no electric current flows into the boundary. At interior boundaries, it means that no current can flow through the boundary and that the electric potential is discontinuous across the boundary. It is also applicable at symmetric boundaries where the potential is known to be symmetric with respect to the boundary.

Tip

To add electric insulation to an interior boundary, add an **Electric Insulation** node in addition to the one that represents the default boundary condition. Electric insulation as the default boundary condition is not applicable to interior boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply electric insulation.

PAIR SELECTION

If **Electric Insulation** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Contact Impedance

Use the **Contact Impedance** node on interior boundaries to model a thin layer of resistive material. It can also be added as a pair.

$$\mathbf{n} \cdot \mathbf{J}_1 = \frac{\sigma}{d_s}(V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{J}_2 = \frac{\sigma}{d_s}(V_2 - V_1)$$

$$\mathbf{n} \cdot \mathbf{J}_1 = \frac{1}{\rho_s}(V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{J}_2 = \frac{1}{\rho_s}(V_2 - V_1)$$

The layer impedance can be specified either with the bulk material conductivity σ_s, the relative permittivity ε_r and the layer thickness d_s, or directly with the surface resistance ρ_s and capacitance C_s. The indices 1 and 2 refer to the two sides of the boundary.

Note

These parameters work the same as with Distributed Impedance.

Model

Thin-Film Resistance: Model Library path **COMSOL_Multiphysics/ Electromagnetics/thin_film_resistance**

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a contact resistance.

PAIR SELECTION

If **Contact Impedance** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

SURFACE IMPEDANCE

Select a potentially complex valued **Layer specification**—**Thin layer** (the default) or **Surface impedance**.

- If **Thin layer** is selected, enter values or expressions for the:
  - **Surface thickness** d_s (SI unit: m). The default is $5\cdot10^{-3}$ m (5 mm).
  - **Electrical conductivity** σ (SI unit: S/m) and **Relative permittivity** ε_r. The defaults take values **From material**. Select **User defined** to enter different values or expressions.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: Ωm^2) and for the **Surface capacitance** C_s (SI unit: F/m^2).

Sector Symmetry

Select **Sector Symmetry** at interfaces between rotating objects where sector symmetry is used. It is only available for pairs.

Note

This feature assumes rotation around the origin.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries from an existing identity pair. This pair first has to be created.

PAIR SELECTION

When **Sector Symmetry** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

SECTOR SETTINGS

Enter the **Number of sectors (<50)** n_{sect}. The default is 2.

Select a **Type of periodicity**—**Continuity** or **Antiperiodicity**.

Based on space dimension, enter values or expressions in the table for the **Axis of rotation** $\mathbf{a}_{\text{rot}}$.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Line Current Source

2D

2D Axi

3D

The **Line Current Source** node adds a line source to edges in 3D models and to points in 2D and 2D axisymmetric models. The line source represents electric current per unit length.

EDGE OR POINT SELECTION

From the **Selection** list, choose the edges or points to add a current source.

Caution

Beware that constraining the potential on edges or points usually yields a current outflow that is mesh dependent.

LINE CURRENT SOURCE

Enter a value or expression to apply a **Line current source** Q_j (SI unit: A/m). This source represents electric current per unit length.

See Also

Line Current Source (on Axis)

Line Current Source (on Axis)

2D Axi

The **Line Current Source (on Axis)** node adds a line source to boundaries in 2D axisymmetric models. The line source represents electric current per unit length.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to add a line current source (on axis).

LINE CURRENT SOURCE (ON AXIS)

Enter a value or expression to apply a **Line current source** Q_j (SI unit: A/m) to boundaries. This source represents electric current per unit length.

See Also

Line Current Source

Point Current Source

3D

The **Point Current Source** node adds a point source and represents an electric current flowing out of the point. Add point sources to 3D models from the **Points** menu.

POINT SELECTION

From the **Selection** list, choose the points to add a current source.

Caution

Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CURRENT SOURCE

Enter a value or expression to apply a **Point current source** Q_j (SI unit: A) to points. This source represents an electric current flowing out of the point.

See Also

- Line Current Source to apply it to points for 2D models.
- Point Current Source (on Axis) to apply to points for 2D axisymmetric models.

Point Current Source (on Axis)

2D Axi	The **Point Current Source (on Axis)** node adds a point source and represents an electric current flowing out of the point in 2D axisymmetric models.

POINT SELECTION

From the **Selection** list, choose the points to add a current source.

Caution	Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

POINT CURRENT SOURCE

Enter a value or expression to apply a **Point current source** Q_j (SI unit: A) to points. This source represents an electric current flowing out of the point.

See Also	• Point Current Source to apply this feature to points on 3D models. • Line Current Source to apply it to points for 2D models.

The Magnetic Fields Interface

The **Magnetic Fields** interface ( ), found under the **AC/DC** branch ( ) in the **Model Wizard**, has the equations, boundary conditions, and currents for modeling magnetic fields, solving for the magnetic vector potential. The main feature is the **Ampère's Law** feature, which adds the equation for the magnetic vector potential and provides an interface for defining the constitutive relations and its associated properties such as the relative permeability.

When this interface is added, these default nodes are also added to the **Model Builder**—**Magnetic Fields**, **Ampère's Law**, **Magnetic Insulation** (the default boundary condition), and **Initial Values**. Right-click the **Magnetic Fields** node to add other features that implement, for example, boundary conditions and external currents.

Model

Quadrupole Lens: Model Library path **COMSOL_Multiphysics/ Electromagnetics/quadrupole**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `mf`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the magnetic vector potential and the equations that describe the potential field for magnetic fields. To choose specific domains, select **Manual** from the **Selection** list.

BACKGROUND FIELD

Select an option from the **Solve for** list—**Full field** (the default) or **Reduced field**. If **Reduced field** is selected, specify a **Background magnetic vector potential** $\mathbf{A}_b$ (SI unit: Wb/m). The total field used in the physics and equations are given by the sum of the reduced and background fields.

COMPONENTS

Note

The current vector has the same direction as the magnetic vector potential, so this setting also controls the direction in which applied and induced currents can flow in the model. The default option is to solve for the out-of-plane component only.

Select **Components**—**Out-of-plane vector potential** (the default), **In-plane vector potential**, or **Three-component vector potential** for the magnetic vector potential. From the practical viewpoint this choice is equivalent to deciding in what directions the electric current is allowed to flow (out-of-plane currents, in-plane currents or currents flowing in all three coordinate directions).

THICKNESS

2D

Enter a value or expression for the global **Out-of-plane thickness** d (SI unit: m). The default value of 1 m is typically not representative for a thin domain. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models.

Use the Change Thickness (Out-of-Plane) node (described for the **Electrostatics** interface) to define specific geometric entities (for example, domains) instead of a global setting for the thickness.

DEPENDENT VARIABLES

The dependent variable is the **Magnetic vector potential** A. You can change both its field name and the individual component variable names. If the new field name coincides with the name of another magnetic vector potential field in the model, the interfaces will share degrees of freedom and component names. The new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model, except for fields of the same type sharing a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select a **Magnetic vector potential**—**Quadratic** (the default), **Linear**, **Cubic**, or **Quartic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

See Also

- Show More Physics Options
- Domain, Boundary, Point, and Pair Conditions for the Magnetic Fields Interface
- Theory for the Magnetic Fields Interface

Domain, Boundary, Point, and Pair Conditions for the Magnetic Fields Interface

The Magnetic Fields Interface has these domain, boundary, point, and pair features available. Features are listed in alphabetical order.

About the Boundary Conditions

With no surface currents present the interface conditions

$$\mathbf{n}_2 \times (\mathbf{A}_1 - \mathbf{A}_2) = \mathbf{0}$$
$$\mathbf{n}_2 \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{0}$$

need to be fulfilled. Because **A** is being solved for, the tangential component of the magnetic potential is always continuous, and thus the first condition is automatically fulfilled. The second condition is equivalent to the natural boundary condition and is hence also fulfilled unless surface currents are explicitly introduced.

Available Features

These features are available for this interface and listed in alphabetical order. Also see Table 9-2 for a list of interior and exterior boundary conditions.

- Ampère’s Law
- Change Thickness (Out-of-Plane) (described for the **Electrostatics** interface)
- External Current Density
- Initial Values
- Line Current (Out-of-Plane)
- Magnetic Field

- Magnetic Insulation (the default boundary condition)
- Magnetic Potential
- Perfect Magnetic Conductor
- Surface Current
- Velocity (Lorentz Term)

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Table 9-2 lists the interior and exterior boundaries available with this interface.

TABLE 9-4: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS FOR THE MAGNETIC FIELDS INTERFACE

FEATURE	INTERIOR	EXTERIOR
Change Thickness (Out-of-Plane)	x	x
Magnetic Field	x	x
Magnetic Insulation	x	x
Magnetic Potential	x	x
Perfect Magnetic Conductor	x	x
Periodic Condition		x
Surface Current	x	x

2D Axi

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

See Also

About Infinite Element Domains and Perfectly Matched Layers

Ampère's Law

The **Ampère's Law** feature adds Ampère's law for the magnetic field and provides an interface for defining the constitutive relation and its associated properties such as the relative permeability as well as electric properties.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the magnetic vector potential and the equation based on Ampère's law that defines the potential.

MODEL INPUTS

This section contains field variables that appear as model inputs, if the current settings include such model inputs. By default, this section is empty. If a linear temperature relation is added for the conductivity, then define the source for the temperature T. From the **Temperature** list, select an existing temperature variable (from another physics interface) if available, or select **User defined** to define a value or expression for the temperature (SI unit: K) in the field that appears underneath the list.

MATERIAL TYPE

Select a **Material type**—**Non-solid** (the default), **Solid**, or **From material**.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

CONDUCTION CURRENT

By default, the **Electrical conductivity** σ (SI unit: S/m) for the media is defined **From material**. Also select **User defined** or **Linearized resistivity from the list.**

- If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the electrical conductivity, and then enter values or expressions in the field or matrix.
- If **Linearized resistivity** is selected, it defines the electric resistivity (and conductivity) as a linear function of temperature and this 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 resistance, which describes how the resistivity varies with temperature.

- The default **Reference temperature** T_{ref} (SI unit: K), **Resistivity temperature coefficient** α (SI unit: 1/K), and **Reference resistivity** ρ_0 (SI unit: Ω·m) are taken **From material**, which means that the values are taken from the boundary material.
- To specify other values for any of these properties, select **User defined** from the list and then enter a value or expression. T is the current temperature, which can be a value specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field is in the **Model Inputs** section.

ELECTRIC FIELD

The default **Relative permittivity** ε_{r} (unitless) for the media is used **From material** and defined on the shell domain. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the permittivity and then enter values or expressions in the field or matrix.

MAGNETIC FIELD

Specify the **Constitutive relation** that describe the macroscopic properties of the medium (relating the magnetic flux density **B** and the magnetic field **H**) and the applicable material properties, such as the relative permeability.

Select a **Constitutive relation**—**Relative permeability** (the default), **HB curve**, **Magnetic losses**, **Remanent flux density**, or **Magnetization**.

Note

The equation for the selected constitutive relation displays under the list. For all options, the default uses values **From material**, or select **User defined** to enter a different value or expression.

- Select **Relative permeability** μ_{r} (unitless) to use the constitutive relation $\mathbf{B} = \mu_0\mu_{\mathrm{r}}\mathbf{H}$. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix.
- Select **HB curve** $|\mathbf{H}|$ (SI unit: A/m) to use a curve that relates magnetic flux density **B** and the magnetic field **H** as $|\mathbf{H}| = f(|\mathbf{B}|)$.

- Select **Magnetic losses** μ' and μ'' (unitless) to describe the relative permeability as a complex-valued quantity: $\mu_r = \mu' + i\mu''$, where μ' and μ'' are the real and imaginary parts, respectively.
- Select **Remanent flux density** $\mathbf{B}_r$ (SI unit: T) to use the constitutive relation $\mathbf{B} = \mu_0\mu_r\mathbf{H} + \mathbf{B}_r$, where $\mathbf{B}_r$ is the remanent flux density (the flux density when no magnetic field is present).
  - The default relative permeability μ_r (unitless) uses values **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the relative permeability and enter another value or expression in the field or matrix.
  - Enter **x** and **y** components for the for the remanent flux density $\mathbf{B}_r$.
- Select **Magnetization** $\mathbf{M}$ (SI unit: A/m) to use the constitutive relation $\mathbf{B} = \mu_0\mathbf{H} + \mu_0\mathbf{M}$. Enter **x** and **y** components.

External Current Density

The **External Current Density** feature adds an externally generated current density $\mathbf{J}_e$, which appears on the right-hand side of the equation that the **Magnetic Fields** interface defines.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an external current density.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

EXTERNAL CURRENT DENSITY

Enter a value or expression for each component of the **External current density** $\mathbf{J}_e$ (SI unit: A/m^2).

Velocity (Lorentz Term)

The **Velocity (Lorentz term)** feature adds velocity $\mathbf{v}$. The external current is equal to $\sigma\mathbf{v} \times \mathbf{B}$. This feature is only valid as follows

- In 2D and 2D axisymmetry when solving for only the out-of-plane component of the magnetic vector potential.

Important	To use the velocity feature correctly requires deep physical insight. In situations when the moving domain is of bounded extent in the direction of the motion or material properties vary in this direction or it contains magnetic sources that also move, the Lorentz term must not be used.

An operational definition of when it can be used is that the moving domain should only contain an induced magnetic source (magnetization + eddy currents) that has to be stationary with respect to the motion. Thus, it cannot be used for modeling projectiles of finite length or projectiles containing magnets. It can be used to model conductive, homogeneous spinning disks (magnetic brakes), magnets over a moving infinite homogenous plane (maglev trains), a flow of homogeneous conducting fluid past a magnet (liquid metal pumps, Hall generators/thrusters).

Tip	If in doubt, contact COMSOL Technical Support www.comsol.com/support

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the velocity.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

VELOCITY (LORENTZ TERM)

Select **User defined** to enter the components for the **Velocity** vector **v** (SI unit: m/s) or, if present, select any velocity field—**Velocity field (spf/pf1)**, for example—defined in the model.

Tip

For example, using the velocity field is useful when coupling to the velocity field of a fluid for a magnetohydrodynamic model.

Initial Values

The **Initial Values** feature adds an initial value for the magnetic vector potential A that can serve as an initial value for a transient simulation or as an initial guess for a nonlinear solver.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define an initial value.

INITIAL VALUES

Enter values or expressions for the initial value of the **Magnetic vector potential** $\mathbf{A}$ (SI unit: Wb/m). The default is 0.

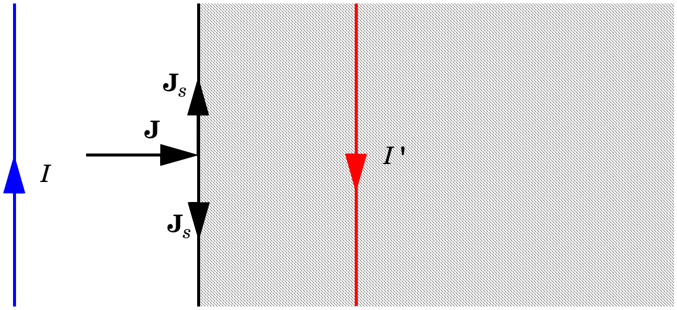
Magnetic Insulation

The **Magnetic Insulation** node is the default boundary condition for the **Magnetic Fields** interface and adds a boundary condition that sets the tangential components of the magnetic potential to zero at the boundary $\mathbf{n} \times \mathbf{A} = 0$.

Tip

Magnetic insulation is a special case of the magnetic potential boundary condition that sets the tangential component of the magnetic potential to zero.

It is used for the modeling of a lossless metallic surface, for example a ground plane or as a symmetry type boundary condition. It imposes symmetry for magnetic fields and “magnetic currents.” In the transient and time harmonic formulations it also imposes antisymmetry for electric fields and electric currents. It supports induced electric surface currents and thus any prescribed or induced electric currents (volume, surface, or edge currents) flowing into a perfect electric conductor boundary is automatically balanced by induced surface currents.



The magnetic insulation boundary condition is used on exterior and interior boundaries representing the surface of a lossless metallic conductor or (on exterior boundaries) representing a symmetry cut. The shaded (metallic) region is not part of the model but still carries effective mirror images of the sources. Note also that any current flowing into the boundary is perfectly balanced by induced surface currents. The tangential vector potential (and electric field) vanishes at the boundary.

BOUNDARY SELECTION

Note

For the default node no user selection is required. **All boundaries** is automatically selected, it applies to all the *external* boundaries.

When additional nodes are added, from the **Selection** list choose the boundaries to define the magnetic insulation.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Magnetic Field

The **Magnetic Field** feature adds a boundary condition for specifying the tangential component of the magnetic field at the boundary:

$$\mathbf{n} \times \mathbf{H} = \mathbf{n} \times \mathbf{H}_0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify the magnetic field.

PAIR SELECTION

If **Magnetic Field** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

MAGNETIC FIELD

Enter the value or expression for the **Magnetic Field** $\mathbf{H}_0$ (SI unit: A/m) coordinates.

Surface Current

The **Surface Current** feature adds a boundary condition for a surface current density $\mathbf{J}_\mathrm{s}$:

$$-\mathbf{n}\times\mathbf{H} = \mathbf{J}_s$$

$$\mathbf{n}\times(\mathbf{H}_1-\mathbf{H}_2) = \mathbf{J}_s$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify a surface current.

PAIR SELECTION

If **Surface Current** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

SURFACE CURRENT

Enter values or expressions for the **Surface current density** $\mathbf{J}_\mathrm{s}$ (SI unit A/m) coordinates.

Magnetic Potential

The **Magnetic Potential** feature adds a boundary condition for the magnetic vector potential:

$$\mathbf{n}\times\mathbf{A} = \mathbf{n}\times\mathbf{A}_0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify the magnetic potential.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

PAIR SELECTION

If **Magnetic Potential** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

MAGNETIC POTENTIAL

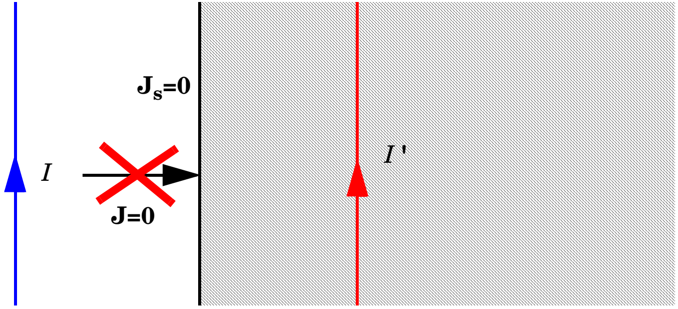
Enter a value or expression for the **Magnetic vector potential** $\mathbf{A}_0$ (SI unit: Wb/m) coordinates.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Perfect Magnetic Conductor

The **Perfect Magnetic Conductor** boundary condition $\mathbf{n} \times \mathbf{H} = 0$ is a special case of the surface current boundary condition that sets the tangential component of the magnetic field and thus also the surface current density to zero. On external boundaries, this can be interpreted as a "high surface impedance" boundary condition or used as a symmetry type boundary condition. It imposes symmetry for electric fields and electric currents. Electric currents (volume, surface, or edge currents) are not allowed to flow into a perfect magnetic conductor boundary as that would violate current conservation. On interior boundaries, the perfect magnetic conductor boundary condition literally sets the tangential magnetic field to zero which in addition to setting the surface current density to zero also makes the tangential magnetic vector potential (and in dynamics the tangential electric field) discontinuous.



The perfect magnetic conductor boundary condition is used on exterior boundaries representing the surface of a high impedance region or a symmetry cut. The shaded (high impedance) region is not part of the model but nevertheless carries effective mirror images of the sources. Note also that any electric current flowing into the boundary is forbidden as it cannot be balanced by induced electric surface currents. The tangential magnetic field vanishes at the boundary. On interior boundaries, the perfect magnetic conductor boundary condition literally sets the tangential magnetic field to zero which in addition to setting the surface current density to zero also makes the tangential magnetic vector potential (and in dynamics the tangential electric field) discontinuous.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to model as perfect magnetic conductors.

PAIR SELECTION

If **Perfect Magnetic Conductor** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Line Current (Out-of-Plane)

Use the **Line Current (Out-of-Plane)** feature, selected from the **Points** menu, to specify a line current out of the modeling plane. In axially symmetric geometries this is the rotational direction, in 2D geometries this is the z-direction.

POINT SELECTION

From the **Selection** list, choose the points to add a line current.

LINE CURRENT (OUT-OF-PLANE)

Enter a value or expression for the **Out-of-plane current** I_0 (SI unit: A).

10

The Acoustics Branch

This chapter describes how to use the Pressure Acoustics, Frequency Domain interface, found under the **Acoustics>Pressure Acoustics** branch () in the **Model Wizard**, for modeling and simulation of acoustics and vibrations.

In this chapter:

Fundamentals of Acoustics

There are certain difficulties that often arise when modeling acoustics, such as the rather severe requirements on the mesh resolution, the modeling of artificial boundaries, and the modeling of real-world damping materials. This section starts with a brief introduction to acoustics and gives some examples of standard acoustics problems.

See Also

Overview of the Physics Interfaces and Building a COMSOL Model

In this section:

- Acoustics Explained
- Examples of Standard Acoustics Problems
- Mathematical Models for Acoustic Analysis

Acoustics Explained

Acoustics is the physics of *sound*. Sound is the sensation, as detected by the ear, of very small rapid changes in the air pressure above and below a static value. This static value is atmospheric pressure (about 100,000 pascals), which varies slowly. Associated with a sound pressure wave is a flow of energy. Physically, sound in air is a longitudinal wave where the wave motion is in the direction of the movement of energy. The wave crests are the pressure maxima, while the troughs represent the pressure minima.

Sound results when the air is disturbed by some source. An example is a vibrating object, such as a speaker cone in a sound system. It is possible to see the movement of a bass speaker cone when it generates sound at a very low frequency. As the cone moves forward it compresses the air in front of it, causing an increase in air pressure. Then it moves back past its resting position and causes a reduction in air pressure. This process continues, radiating a wave of alternating high and low pressure at the speed of sound.

Examples of Standard Acoustics Problems

These standard problems or scenarios occur frequently when analyzing acoustics:

Depending on the basic dependent variable used to model the acoustic field, the acoustical interfaces can be divided in two main categories:

- *Pressure Acoustics*—The dependent variable is the acoustic pressure, p.
- A*eroacoustics*—The dependent variable is the potential, ϕ , for the acoustic particle-velocity field, $\mathbf{v}= \nabla\phi$. In the typical situation, the fluid is in motion with a total velocity, $\mathbf{v}_{\text{tot}} = \mathbf{V} + \mathbf{v}$, split into a stationary background-flow velocity, $\mathbf{V}$, and the particle velocity, $\mathbf{v}$, associated with the acoustic waves. Analysis of this type of problem requires the Acoustics Module.

The Radiation Problem

A vibrating structure (a speaker, for example) radiates sound into the surrounding space. A far-field boundary condition or a PML (perfectly matched layer) is necessary to model the unbounded domain.

The Scattering Problem

An incident wave impinges on a body and creates a scattered wave. A far-field radiation boundary condition or a PML is necessary.

The Sound Field in an Interior Space (for Example, a Room)

The acoustic waves stay in a finite volume so no radiation condition is necessary.

Coupled Fluid-Elastic Structure Interaction (Structural Acoustics)

If the radiating or scattering structure consists of an elastic material, the interaction must be considered between the body and the surrounding fluid. In the multiphysics coupling, the acoustic analysis provides a load (the sound pressure) to the structural analysis, and the structural analysis provides accelerations to the acoustic analysis.

The Transmission Problem

An incident sound wave propagates into a body, which can have different acoustic properties. Pressure and acceleration are continuous on the boundary.

Aeroacoustics Problems

Sound (noise) is generated by turbulent fluid motion or by the interaction between a fluid and a surface. Analysis of this type of problem requires the Acoustics Module.

Mathematical Models for Acoustic Analysis

Sound waves in a lossless medium are governed by the following equation for the (differential) pressure p (SI unit: $\mathrm{N/m^2}$):

$$\frac{1}{\rho_0 c_s^2}\frac{\partial^2 p}{\partial t^2}+\nabla\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q})\right) = Q$$

Here ρ_0 (SI unit: $\mathrm{kg/m^3}$) refers to the density and c_s (SI unit: m/s) denotes the speed of sound. The ***dipole source*** $\mathbf{q}$ (SI unit: $\mathrm{N/m^3}$), and the ***monopole source*** Q (SI unit: $1/\mathrm{s^2}$), are both optional additional sources. The combination $\rho_0 c_s^2$ is called the ***bulk modulus***, commonly denoted β (SI unit: $\mathrm{N\ /m^2}$).

A special case is a time-harmonic wave, for which the pressure varies with time as

$$p(\mathbf{x},t) = p(\mathbf{x})e^{i\omega t}$$

where $\omega = 2\pi f$ (rad/s) is the angular frequency, f (Hz) as usual denoting the frequency. Assuming the same harmonic time-dependence for the source terms, the wave equation for acoustic waves reduces to an inhomogeneous Helmholtz equation:

$$\nabla\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q})\right)-\frac{\omega^2 p}{\rho_0 c_s^2} = Q \qquad (10\text{-}1)$$

With the source terms removed, this equation can also be treated as an eigenvalue PDE to solve for eigenmodes and eigenfrequencies. Typical boundary conditions are sound-hard boundaries (walls), sound-soft boundaries, impedance boundary conditions, and radiation boundary conditions.

These are described in more detail below.

In lossy media, an additional term of first order in the time derivative needs to be introduced to model attenuation of the sound waves:

$$\frac{1}{\rho_0 c_s^2}\frac{\partial^2 p}{\partial t^2}-d_a\frac{\partial p}{\partial t}+\nabla\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q})\right) = Q$$

Note also that even when the sound waves propagate in a lossless medium, attenuation frequently occurs by interaction with the surroundings at the boundaries of the system.

Theory for the Pressure Acoustics, Frequency Domain Interface

The Pressure Acoustics, Frequency Domain Interface is designed for the analysis of various types of pressure acoustics problems in the frequency domain, all concerning pressure waves in a fluid. An acoustics model can be part of a larger multiphysics model that describes, for example, the interactions between structures and acoustic waves. This interface is suitable for modeling acoustics phenomena that do not involve fluid flow.

This interface solves for the acoustic pressure, p. It is available in all space dimensions—for 3D, 2D, and 1D Cartesian geometries as well as for 2D and 1D axisymmetric geometries.

These studies are discussed briefly in this section:

- Frequency Domain Study
- Eigenfrequency Study
- References for the Pressure Acoustics, Frequency Domain Interface

Frequency Domain Study

The frequency domain—or time-harmonic—formulation uses the following inhomogeneous Helmholtz equation:

$$\nabla\cdot\left(-\frac{1}{\rho_c}(\nabla p-\mathbf{q})\right)-\frac{\omega^2 p}{\rho_c c_c^2}=Q \tag{10-2}$$

In this equation, $p = p\,(\mathbf{x},\,\omega)$ (the dependence on ω is henceforth not explicitly indicated). With this formulation compute the frequency response with a parametric sweep over a frequency range using a harmonic load.

When there is damping, ρ_c and c_c are complex-valued quantities. The available damping models and how to apply them is described in the section Damping Models

Equation 10-2 is the equation that the software solves for 3D geometries. In lower-dimensional and axisymmetric cases, restrictions on the coordinate dependence mean that the equations differ from case to case. Here is a brief summary of the situation.

2D

In 2D, the pressure is of the form

$$p(\mathbf{r}) = p(x, y)e^{-ik_z z}$$

which inserted in Equation 10-2 gives

$$\nabla \cdot \left(-\frac{1}{\rho_c}(\nabla p - \mathbf{q})\right) - \frac{1}{\rho_c}\left(\frac{\omega^2}{c_c^2} - k_z^2\right)p = Q \qquad (10\text{-}3)$$

The *out-of-plane wave number*, k_z, can be set on the Pressure Acoustics page. By default its value is zero. In the mode analysis type, $-ik_z$ is used as the eigenvalue.

2D AXISYMMETRY

For 2D axisymmetric geometries the independent variables are the radial coordinate, r, and the axial coordinate, z. The only dependence allowed on the azimuthal coordinate, φ , is through a phase factor,

$$p(r, \phi, z) = p(r, z)e^{-im\varphi} \qquad (10\text{-}4)$$

where m denotes the *circumferential wave number*. Because the azimuthal coordinate is periodic, m must be an integer. Just like k_z in the 2D case, m can be set on the Pressure Acoustics page.

As a result of Equation 10-4, the equation to solve for the acoustic pressure in 2D axisymmetric geometries becomes

$$\frac{\partial}{\partial r}\left[-\frac{r}{\rho_c}\left(\frac{\partial p}{\partial r} - q_r\right)\right] + r\frac{\partial}{\partial z}\left[-\frac{1}{\rho_c}\left(\frac{\partial p}{\partial z} - q_z\right)\right] - \left[\left(\frac{\omega}{c_c}\right)^2 - \left(\frac{m}{r}\right)^2\right]\frac{rp}{\rho_c} = rQ$$

1D AXISYMMETRY

In 1D axisymmetric geometries,

$$p(r, \phi, z) = p(r)e^{-i(k_z z + m\varphi)}$$

leading to the radial equation

$$\frac{\partial}{\partial r}\left[-\frac{r}{\rho_c}\left(\frac{\partial p}{\partial r}-q_r\right)\right]-\left[\left(\frac{\omega}{c_c}\right)^2-\left(\frac{m}{r}\right)^2-k_z^2\right]\frac{rp}{\rho_c} = rQ$$

where both the circumferential wave number m, and the *axial wave number* k_z, appear as parameters.

1D

The equation for the 1D case is obtained by taking the pressure to depend on a single Cartesian coordinate, x:

$$\frac{d}{dx}\left(-\frac{1}{\rho_c}\left(\frac{dp}{dx}-q\right)\right)-\frac{\omega^2}{\rho_c c_c^2}p = Q$$

Eigenfrequency Study

In the eigenfrequency formulation the source terms are absent, the eigenmodes and eigenfrequencies are solved for:

$$\nabla\cdot\left(-\frac{1}{\rho_c}\nabla p\right)+\frac{\lambda^2 p}{\rho_c c^2} = 0 \qquad (10\text{-}5)$$

The eigenvalue λ introduced in this equation is related to the eigenfrequency f, and the angular frequency ω, through $\lambda = i2\pi f = i\omega$. Because they are independent of the pressure, the solver ignores any dipole and monopole sources unless a coupled eigenvalue problem is being solved.

Equation 10-5 applies to the 3D case. The equations solved in eigenfrequency studies in lower dimensions and for axisymmetric geometries are obtained from their time-harmonic counterparts, given in the previous subsection, by the substitution $\omega^2 \rightarrow -\lambda^2$.

Switch between specifying the eigenvalues, the eigenfrequencies, or the angular frequencies by selecting from the Eigenvalue transformation list in the solver sequence's Eigenvalue feature node's settings window.

References for the Pressure Acoustics, Frequency Domain Interface

1. D. Givoli and B. Neta, "High-order Non-reflecting Boundary Scheme for Time-dependent Waves," *J. Comput. Phys.*, vol. 186, pp. 24–46, 2004.

2. A. Bayliss, M. Gunzburger, and E. Turkel, “Boundary Conditions for the Numerical Solution of Elliptic Equations in Exterior Regions,” *SIAM J. Appl. Math.*, vol. 42, no. 2, pp. 430–451, 1982.

3. A.B. Bauer, “Impedance Theory and Measurements on Porous Acoustic Liners,” *J. Aircr.*, vol. 14, pp. 720–728, 1977.

4. S. Temkin, *Elements of Acoustics*, Acoustical Society of America, 2001.

The Pressure Acoustics, Frequency Domain Interface

The **Pressure Acoustics, Frequency Domain** interface ( ) has the equations, boundary conditions, and sources for modeling acoustics, solving for the sound pressure. Select the interface from the **Pressure Acoustics** branch ( ) in the **Model Wizard**.

The sound pressure, p, which is solved for in pressure acoustics, represents the acoustic variations (or excess pressure) to the ambient pressure. The ambient pressure is in the absence of flow simply the static absolute pressure.

When the geometrical dimensions of the acoustic problems are reduced from 3D to 2D (planar symmetry or axisymmetric) or to 1D axisymmetric, it is possible to specify an out-of-plane wave number, k_z, and circumferential wave number, m, when applicable. The wave number used in the equations, k_{eq}, contains both the ordinary wave number, k, as well as the out-of-plane wave number and circumferential wave number, when applicable.

The following table lists the names SI units for the most important physical quantities in the **Pressure Acoustics, Frequency Domain** interface:

TABLE 10-1: PRESSURE ACOUSTICS, FREQUENCY DOMAIN INTERFACE PHYSICAL QUANTITIES

QUANTITY	SYMBOL	SI UNIT	ABBREVIATION
Pressure	p	pascal	Pa
Density	ρ	kilogram/meter^3	kg/m^3
Frequency	f	hertz	Hz
Wave number	k	1/meter	1/m
Dipole source	$\mathbf{q}$	newton/meter^3	N/m^3
Monopole source	Q	1/second^2	1/s^2
Speed of sound	c_{s}	meter/second	m/s
Acoustic impedance	Z	pascal-second/meter	Pa·s/m
Normal acceleration	a_n	meter/second^2	m/s^2
Source location	r_0	meter	m
Wave direction	$\mathbf{n}_{\mathrm{k}}$	(dimensionless)	1

In the following descriptions of the functionality in this interface, the subscript c in ρ_c and c_c (the density and speed of sound, respectively) denotes that these can be complex-valued quantities in models with damping.

When this interface is added, these default nodes are also added to the **Model Builder**—**Pressure Acoustics Model**, **Sound Hard Boundary (Wall)**, and **Initial Values**.

Right-click the **Pressure Acoustics** node to other features that implement, for example, boundary conditions.

Model

Eigenmodes of a Room: Model Library path **COMSOL_Multiphysics/Acoustics/eigenmodes_of_room**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `acpr`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define a sound pressure field and the associated acoustics equation. To choose specific domains, select **Manual** from the **Selection** list.

SOUND PRESSURE LEVEL SETTINGS

The zero level on the dB scale varies with the type of fluid. That value is a reference pressure that corresponds to 0 dB. This variable occurs in calculations of the sound pressure level based on the root mean square (rms) pressure,

$$\bar{p} = \sqrt{\frac{1}{2}p\,\mathrm{conj}(p)}$$

an expression valid for the case of harmonically time-varying acoustic pressure, p.

Based on the fluid type, select a **Reference pressure for the sound pressure level**. Select:

- **Use reference pressure for air** to use a reference pressure of 20 μPa ($20\cdot10^{-6}$ Pa).

- **Use reference pressure for water** to use a reference pressure of 1 μPa ($1 \cdot 10^{-6}$ Pa).
- **User-defined reference pressure** to enter a reference pressure $p_{\mathrm{ref,\ SPL}}$ (SI unit: Pa). The default value is the same as for air, 20 μPa.

TYPICAL WAVE SPEED

Enter a value or expression for the **Typical wave speed for perfectly matched layers** c_{ref} (SI unit m/s).

DEPENDENT VARIABLES

This interface defines one dependent variable (field), the **Pressure** p. If required, edit the name, which changes both the field name and the dependent variable name. If the new field name coincides with the name of another pressure field in the model, the interfaces will share degrees of freedom and dependent variable name. The new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

ADVANCED SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed.

1D Axi

2D Axi

Edit the following as required:

- **Circumferential wave number** m is an integer-valued number used in axisymmetry and is by default 0.
- **Out-of-plane wave number** k_z is used in 2D and 1D axisymmetry and is by default 0.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select **Quadratic** (the default), **Linear**, **Cubic**, or **Quartic** for the **Pressure**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Features for the Pressure Acoustics, Frequency Domain Interface
- Theory for the Pressure Acoustics, Frequency Domain Interface

Domain, Boundary, Edge, Point, and Pair Features for the Pressure Acoustics, Frequency Domain Interface

The Pressure Acoustics, Frequency Domain Interface has these domain, boundary, edge, point, and pair features available and listed in alphabetical order.

Domain Features

- Pressure Acoustics Model
- Monopole Source
- Dipole Source
- Initial Values

Boundary Conditions

> *Note*
>
> Continuity in pressure is the default condition on interior boundaries.

- Cylindrical Wave Radiation
- Destination Selection
- Impedance
- Interior Sound Hard Boundary (Wall)
- Normal Acceleration
- Periodic Condition
- Plane Wave Radiation
- Pressure
- Sound Hard Boundary (Wall)
- Sound Soft Boundary
- Spherical Wave Radiation

Edge, Point, and Pair Conditions

- Continuity
- Pressure

- Sound Hard Boundary (Wall)
- Sound Soft Boundary

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Pressure Acoustics Model

The **Pressure Acoustics Model** feature adds the equations for time-harmonic and eigenfrequency acoustics modeling in the frequency domain. In the settings window, define the properties for the acoustics model and model inputs including temperature.

DOMAIN SELECTION

From the **Selection** list, choose the domains to compute the acoustic pressure field and the equation that defines it.

PRESSURE ACOUSTICS MODEL

The default **Fluid Model** for pressure acoustics is a **Linear elastic** fluid. By default the values for the **Density** ρ (SI unit: $\mathrm{kg/m^3}$) and the **Speed of sound** c (SI unit: m/s) are taken **From material**. Select **User defined** to enter other values for these properties.

Monopole Source

Use the **Monopole Source** feature to radiate a uniform sound field in all directions.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the monopole source.

MONOPOLE SOURCE

Enter a **Monopole source** Q (SI unit: $1/s^2$). The default is 0.

Dipole Source

Use the **Dipole Source** feature to radiate a sound field that is typically stronger in two opposite directions.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the dipole source.

DIPOLE SOURCE

Enter coordinates for the **Dipole source** q (SI unit: N/m^3). These are the individual components of the dipole source vector. The defaults are 0.

Initial Values

The **Initial Values** feature adds initial values for the sound pressure and the pressure time derivative that can serve as an initial guess for a nonlinear solver. If more than one initial value is needed, right-click the interface node to add more **Initial Values** nodes.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial values for the **Pressure** p (SI unit: Pa) and the **Pressure time derivative**, dp/dt (SI unit: Pa/s). The defaults are 0.

Sound Hard Boundary (Wall)

The **Sound Hard Boundary (Wall)** adds a boundary condition for a sound hard boundary or wall, which is a boundary at which the normal component of the acceleration is zero:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p-\mathbf{q})\right) = 0$$

For zero dipole charge and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary:

$$\frac{\partial p}{\partial n} = 0$$

Sound-hard boundaries are available for all study types.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define as sound hard boundaries (wall).

Normal Acceleration

The **Normal Acceleration** adds an inward normal acceleration a_n:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p - \mathbf{q})\right) = a_\mathrm{n}$$

Alternatively, specify the acceleration a of the boundary. The part in the normal direction is used to define the boundary condition:

$$\mathbf{n}\cdot\left(-\frac{1}{\rho_0}(\nabla p - \mathbf{q})\right) = \mathbf{n}\cdot\mathbf{a}_0$$

This feature represents an external source term. It can also be used to couple acoustics with a structural analysis for modeling acoustic-structure interaction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify a normal acceleration boundary condition.

NORMAL ACCELERATION

Select a **Type**—Select **Inward Acceleration** to enter the value of the inward normal acceleration a_n (SI unit: m/s^2). Use a positive value for inward acceleration or a negative value for outward acceleration. Select **Acceleration** to enter values for the components of the acceleration a_0 (SI unit: m/s^2).

Sound Soft Boundary

The **Sound Soft Boundary** adds a boundary condition for a sound soft boundary, where the acoustic pressure vanishes: $p = 0$.

This boundary condition is an appropriate approximation for a liquid-gas interface and in some cases for external waveguide ports.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define as sound soft boundaries. If the feature is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PAIR SELECTION

If **Sound Soft Boundary** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this option, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Pressure

The **Pressure** node creates a boundary condition that acts as a pressure source at the boundary, which means a constant acoustic pressure p_0 is specified and maintained at the boundary: $p = p_0$. In the frequency domain, p_0 is the amplitude of a harmonic pressure source.

The feature is also available from the **Pairs** submenu as an option at interfaces between parts in an assembly.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify a pressure boundary condition. If the feature is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PRESSURE

Enter the value of the **Pressure** p_0 (SI unit: Pa) at the boundary.

PAIR SELECTION

If **Pressure** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this option, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Impedance

The **Impedance** node adds an impedance boundary condition, which is a generalization of the sound-hard and sound-soft boundary conditions:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right) = -\frac{i\omega p_t}{Z_i}$$

Here Z_i is the acoustic input impedance of the external domain. From a physical point of view, the acoustic input impedance is the ratio between pressure and normal particle velocity.

The **Impedance** boundary condition is a good approximation for a locally reacting surface—a surface for which the normal velocity at any point depends only on the pressure at that exact point.

Note

In the two opposite limits $Z_i \to \infty$ and $Z_i \to 0$, this boundary condition is identical to the **Sound Hard** boundary condition and the **Sound Soft** boundary condition, respectively.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to specify an impedance boundary condition.

IMPEDANCE

Enter the value of the **Input impedance** Z_i (SI unit: Pa·s/m). The default value is 1.2 kg/m^3·343 m/s.

Plane, Spherical, and Cylindrical Radiation Boundary Conditions

Specify a **Plane Wave Radiation**, **Spherical Wave Radiation**, or **Cylindrical Wave Radiation** boundary condition to allow an outgoing wave to leave the modeling domain with minimal reflections. The condition can be adapted to the geometry of the modeling domain. The **Plane** wave type is suitable for both far-field boundaries and ports.

Because many waveguide structures are only interesting in the plane-wave region, it is particularly relevant for ports.

Radiation boundary conditions are available for all types of studies. For the frequency domain study, Givoli and Neta's reformulation of the Higdon conditions (Ref. 1) for plane waves has been implemented to the second order. For **Cylindrical** and **Spherical** waves COMSOL uses the corresponding 2nd-order expressions from Bayliss, Gunzburger, and Turkel (Ref. 2). The Eigenfrequency study implement the same expansions to the first order.

The first-order radiation boundary conditions in the frequency domain read

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right)+(ik+\kappa(r))\frac{p}{\rho_c} = (ik+\kappa(r))\frac{p_\mathrm{i}}{\rho_c}+\mathbf{n}\cdot\left(\frac{\nabla p_\mathrm{i}}{\rho_c}\right)$$

where k is the wave number and $\kappa(r)$ is a function whose form depends on the wave type:

- Plane wave: $\kappa(r)=0$ (see Plane Wave Radiation)
- Cylindrical wave: $\kappa(r)=1/(2r)$ (see Cylindrical Wave Radiation)
- Spherical wave: $\kappa(r)=1/r$ (see Spherical Wave Radiation)

In the cylindrical and spherical wave cases, r is the shortest distance from the point $\mathbf{r}=(x, y, z)$ on the boundary to the source. The right-hand side of the equation represents an optional incoming pressure field p_i (see Incident Pressure Field).

The second order radiation boundary conditions in the frequency domain are defined below. In these equations, Δ_T at a given point on the boundary denotes the Laplace operator in the tangent plane at that particular point.

- **Plane wave**:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right)+i\frac{k}{\rho_c}p+\frac{i}{2k\rho_c}\Delta_\mathrm{T}p = \frac{i}{2k\rho_c}\Delta_\mathrm{T}p_i+i\frac{k}{\rho_c}p_i+\mathbf{n}\cdot\frac{1}{\rho_c}\nabla p_i$$

  In the notation of Givoli and Neta (Ref. 1), the above expressions correspond to the parameter choices $C_0=C_1=C_2=\omega/k$. For normally incident waves this gives a vanishing reflection coefficient.

- **Cylindrical wave**:

  This boundary condition is based on a series expansion of the outgoing wave in cylindrical coordinates (Ref. 2), and it assumes the field is independent of the axial coordinate. Specify the axis of this coordinate system by giving an orientation

(n_x, n_y, n_z) and a point (x_0, y_0, z_0) on the axis. In axisymmetric geometries the symmetry axis is the natural and only choice.

- **Spherical wave**:

$$-\mathbf{n}\cdot\left(-\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right)+\left(ik_{\mathrm{eq}}+\frac{1}{r}\right)\frac{p}{\rho_c}-\frac{r\Delta_{\mathrm{T}}p}{2\rho_c(ik_{\mathrm{eq}}r+1)}$$
$$=-\frac{r\Delta_{\mathrm{T}}p_{\mathrm{i}}}{2\rho_{0c}(ik_{\mathrm{eq}}r+1)}+\left(ik_{\mathrm{eq}}+\frac{1}{r}\right)\frac{p_{\mathrm{i}}}{\rho_c}+\mathbf{n}\cdot\frac{1}{\rho_c}\nabla p_{\mathrm{i}}$$

Use this option to allow a radiated or scattered wave—emanating from an object centered at the point (x_0, y_0, z_0) that is specified—to leave the modeling domain without reflections. The boundary condition is based on an expansion in spherical coordinates from Bayliss, Gunzburger, and Turkel (Ref. 2), implemented to the second order.

Plane Wave Radiation

The **Plane Wave Radiation** node adds a radiation boundary condition for a plane wave (see Plane, Spherical, and Cylindrical Radiation Boundary Conditions for equations). If required, right-click the main node to add an Incident Pressure Field.

Model

Acoustics of a Muffler: Model Library path **COMSOL_Multiphysics/ Acoustics/automotive_muffler**

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to use a plane wave radiation condition.

Spherical Wave Radiation

The **Spherical Wave Radiation** feature adds a radiation boundary condition for a spherical wave, for which you define the source location (see Plane, Spherical, and Cylindrical Radiation Boundary Conditions for equations). If required, right-click the main node to add an Incident Pressure Field.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to use a spherical wave radiation condition.

SPHERICAL WAVE RADIATION

Enter coordinates for the **Source location** r_0 (SI unit: m).

Cylindrical Wave Radiation

The **Cylindrical Wave Radiation** feature adds a radiation boundary condition for a cylindrical wave, for which you define the source location and the source axis direction (see Plane, Spherical, and Cylindrical Radiation Boundary Conditions for equations). If required, right-click the main node to add an Incident Pressure Field.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to use a cylindrical wave radiation condition.

CYLINDRICAL WAVE RADIATION

Enter coordinates for the **Source location** r_0 (SI unit: m) and the **Source axis** direction r_{axis} (unitless).

Periodic Condition

The **Periodic Condition** feature adds a periodic boundary condition.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a periodic boundary condition. The software automatically identifies the boundaries as either source boundaries or destination boundaries.

Note

This feature works well for cases like opposing parallel boundaries. In other cases use a **Destination Selection** subnode to control the destination. By default it contains the selection that COMSOL Multiphysics identifies.

PERIODICITY SETTINGS

Select a **Type of periodicity**—**Continuity** (the default), **Antiperiodicity**, **Floquet periodicity**, or **Cyclic symmetry**.

- If **Floquet Periodicity** is selected, enter a **k-vector for Floquet periodicity** $\mathbf{k}_\mathrm{F}$ (SI unit: rad/m) for the **x**, **y**, and **z** coordinates (3D models), or the **r** and **z** coordinates (2D axisymmetric models), or **x** and **y** coordinates (2D models).

- If **Cyclic symmetry** is selected, select a **Sector angle**—**Automatic** (the default), or **User defined**. If **User defined** is selected, enter a value for θ_S (SI unit: rad). For any selection, also enter a **Mode number** m (unitless).

See Also

- Periodic Condition
- Destination Selection
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Incident Pressure Field

The **Incident Pressure Field** node is a subnode to all non reflecting boundary conditions (plane, cylindrical, or spherical wave radiation, for example). Add this node by right-clicking the boundary condition and selecting **Incident Pressure Field**. If the incident pressure field is a predefined plane wave, it is of the type:

$$p_\mathrm{i} = p_0 e^{-i(\mathbf{k}\cdot\mathbf{r})} = p_0 e^{-ik_\mathrm{eq}\left(\frac{\mathbf{r}\cdot\mathbf{e}_k}{\|\mathbf{e}_k\|}\right)}$$

where p_0 is the wave amplitude, $\mathbf{k}$ is the wave vector (with amplitude $k_\mathrm{eq}=|\mathbf{k}|$ and wave direction vector $\mathbf{e}_k$), and $\mathbf{r}$ is the location on the boundary. The incident pressure field can also be a user-defined value or expression.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to include an incident pressure field p_i in the boundary condition. By default, this feature node inherits the selection from its parent node, and only a selection that is a subset of the parent node's selection can be used.

INCIDENT PRESSURE FIELD

From the **Incident pressure field type** list, select **Plane wave** to define an incident pressure field of plane wave type. Enter a **Pressure amplitude** p_0 (SI unit: Pa) and **Wave direction** $\mathbf{e}_k$ (SI unit: m).

Select **User defined** to enter the expression for the **Incident pressure field** p_i (SI unit: Pa) as a function of space. The default is 0.

Interior Sound Hard Boundary (Wall)

The **Interior Sound Hard Boundary (Wall)** node adds a boundary condition for a sound hard boundary or wall on interior boundaries. A sound-hard boundary is a boundary at which the normal component of the acceleration is zero:

$$-\mathbf{n}\cdot -\left(\frac{1}{\rho_c}(\nabla p-\mathbf{q})\right)_1 = 0 \qquad -\mathbf{n}\cdot\left(\left(-\frac{1}{\rho_c}\right)(\nabla p-\mathbf{q})\right)_2 = 0$$

For zero dipole charge and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the interior boundaries to define as sound hard boundaries (wall).

Axial Symmetry

1D Axi

2D Axi

The **Axial Symmetry** node is a default node added for all 1D and 2D axisymmetric models. The boundary condition is active on all boundaries on the symmetry axis.

BOUNDARY SELECTION

Note

The boundaries section shows on which boundaries the feature is active. All boundaries on the symmetry axis are automatically selected.

Continuity

Continuity is available as an option at interfaces between parts in a pair.

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

This condition gives continuity in pressure and in the normal acceleration over the pair (subscripts 1 and 2 in the equation refer to the two sides in the pair):

$$-\mathbf{n}\cdot\left[-\left(\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right)_1-\left(-\frac{1}{\rho_c}(\nabla p_t-\mathbf{q})\right)_2\right]=0$$

BOUNDARY SELECTION

Note

This list cannot be edited. It shows the boundaries in the selected pairs.

PAIR SELECTION

When **Continuity** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this option, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

11

The Chemical Species Transport Branch

This chapter explains how to use the Transport of Diluted Species interface, found under the **Chemical Species Transport** branch () in the Model Wizard, to model and simulate mass transfer by diffusion and convection based on Fick's law of diffusion.

In this chapter:

Theory for the Transport of Diluted Species Interface

The Transport of Diluted Species Interface provides a predefined modeling environment for studying the evolution of chemical species transported by diffusion and convection. The interface assumes that all species present are dilute; that is, that their concentration is small compared to a solvent fluid or solid. As a rule of thumb, a mixture containing several species can be considered dilute when the concentration of the solvent is more than 90 mol%. Due to the dilution, mixture properties such as density and viscosity can be assumed to correspond to those of the solvent.

Fick's law governs the diffusion of the solutes dilute mixtures or solutions. The Transport of Diluted Species interface supports the simulation of chemical species transport by convection and diffusion in 1D, 2D, and 3D as well as for axisymmetric models in 1D and 2D. In this section:

- Mass Balance Equation
- Convective Term Formulation
- Solving a Diffusion Equation Only

Mass Balance Equation

The default node attributed to The Transport of Diluted Species Interface assumes chemical species transport through diffusion and convection and implements the mass balance equation:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot (D \nabla c) + R \qquad (11\text{-}1)$$

Equation 11-1 includes these quantities (with the SI unit in parentheses):

- c is the concentration of the species (mol/m^3)
- D denotes the diffusion coefficient (m^2/s)
- R is a reaction rate expression for the species (mol/(m^3·s))
- $\mathbf{u}$ is the velocity vector (m/s)

The first term on the left-hand side of Equation 11-1 corresponds to the accumulation (or indeed consumption) of the species.

The second term accounts for the convective transport due to a velocity field **u**. This field can be expressed analytically or be obtained from coupling this physics interface to one that describes fluid flow (momentum balance). To include convection in the mass balance equation, an expression that includes the spatial and time variables, or else the velocity vector component variable names from a fluid flow interface can be entered into the appropriate field. The velocity fields from existing fluid flow interfaces are available directly as predefined fields.

On the right-hand side of the mass balance equation (Equation 11-1), the first term describes the diffusion transport, accounting for interaction between the dilute species and the solvent. A field for the ***diffusion coefficient*** is available in the interface and any equation that relates to another variable, such as temperature, can be entered there. The node has a matrix that is used to describe the diffusion coefficient, if it is vectorized or is a tensor. Anisotropic diffusion can therefore be simulated here.

Finally, the second term on the right-hand side of Equation 11-1 represents a source or sink term, typically due to a chemical reaction. In order for the chemical reaction to be specified, another node must be added to the Transport of Diluted Species interface—the Reaction node, which has a field to specify a reaction equation using the variable names of all participating species.

Tip

With the Chemical Reaction Engineering Module, and if the Reaction Engineering interface has been added to set up the mass balance, then the reaction equations show up in this node.

Convective Term Formulation

The default node attributed to The Transport of Diluted Species Interface assumes chemical species transport through diffusion and convection and implements the mass balance equation in Equation 11-1.

There are two ways to present a mass balance where chemical species transport occurs through diffusion and convection. These are the non-conservative and conservative formulations of the convective term:

$$\text{non-conservative: } \frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c = \nabla\cdot(D\nabla c) + R \tag{11-2}$$

$$\text{conservative: } \frac{\partial c}{\partial t} + \nabla\cdot(c\mathbf{u}) = \nabla\cdot(D\nabla c) + R \tag{11-3}$$

and each is treated slightly differently by the solver algorithms. In these equations D (SI unit: m^2/s) is the diffusion coefficient, R (SI unit: $mol/(m^3 \cdot s)$) is a production or consumption rate expression, and $\mathbf{u}$ (SI unit: m/s) is the solvent velocity field. The diffusion process can be anisotropic, in which case D is a tensor.

If the conservative formulation is expanded using the chain rule, then one of the terms from the convection part, $c\nabla\cdot\mathbf{u}$, would equal zero for an incompressible fluid and would result in the non-conservative formulation above. This is in fact the default formulation in this interface and ensures that nonphysical source terms cannot come from the solution of a flow field.

Tip

To switch between the two formulations, click the **Show** button ( ) and select **Advanced Physics Options**. See Show Advanced Physics Options for information.

Solving a Diffusion Equation Only

Remove the convection term from Equation 11-2 and Equation 11-3 by clearing the **Convection** check box in the **Transport Mechanisms** section for The Transport of Diluted Species Interface. The equation then becomes

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c) + R$$

The Transport of Diluted Species Interface

Mass transfer is an important part of chemical engineering because this is the field that considers the conversion of one type of substance into another. A lot of this occurs through chemical reactions, although separation and other unit operations are an important part. You can use the Transport of Diluted Species interface to model transport of a diluted species in chemical systems by convection and diffusion.

In the Transport of Diluted Species interface, 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. Assuming a binary mixture of solute A in solvent B, concentrations of up to 10 mol% of A can be considered dilute.

Tip

Other optional modules have an extension of the physics for modeling multicomponent convection, diffusion, and migration (electrokinetic flow).

The **Transport of Diluted Species** interface ( ), found under the **Chemical Species Transport** branch ( ) in the **Model Wizard**, has the equations, boundary conditions, and rate expression terms for modeling mass transport of diluted species in mixtures, solutions, and solids, solving for the species concentration. This interface is applicable for solutions (either fluid or solid) where the transported species have concentrations at least one order of magnitude less than the solvent. The settings for this physics interface can be chosen so as to simulate chemical species transport through diffusion (Fick's law) and convection (when coupled to fluid flow).

The interface supports simulation of transport by convection and diffusion in 1D, 2D, and 3D as well as for axisymmetric models in 1D and 2D. The dependent variable is the molar concentration, c.

When this interface is added, these default nodes are also added to the **Model Builder**—**Convection and Diffusion**, **No Flux**, and **Initial Values**. Right-click the **Transport of Diluted**

Species node to add other features with appropriate boundary conditions and rate expression terms.

Model

- Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**
- Thin-Layer Diffusion: Model Library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `chds`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the dependent variables and the equations. To choose specific domains, select **Manual** from the **Selection** list.

Tip

There may be a domain in the model that is not described by mass transfer, such as a reactor's solid wall. In this case, remove that domain selection from here.

TRANSPORT MECHANISMS

Control which of the following transport mechanism are to be included in the model:

- Diffusion is always included.
- Convection is active by default. To model using the diffusion equation only, clear the **Convection** check box.

Dependent Variables

Important

The species are dependent variables, and their names must be unique with respect to all other dependent variables in the model.

Advanced Settings

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed.Select a **Convective term**—**Non-conservative** or **Conservative**. The conservative formulation should be used for a compressible flow.

See Also

Convective Term Formulation

Discretization

To display this section, click the **Show** button ( ) and select **Discretization**. Select an element order for the **Concentration**—**Linear** (the default), **Quadratic**, **Cubic**, or **Quartic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

Consistent and Inconsistent Stabilization

To display these sections, click the **Show** button ( ) and select **Stabilization**. Any settings unique to this interface are listed below.

- When the **Crosswind diffusion** check box is selected, the **Lower gradient limit** g_{lim} (SI unit: mol/m^4) field defaults to `0.1[mol/m^3)/chds.helem`, where `chds.helem` is the local element size.
- For both consistent stabilization methods, select an **Equation residual. Approximate residual** is the default setting and it means that derivatives of the diffusion tensor

components are neglected. This setting is usually accurate enough and is faster to compute. If required, select **Full residual** instead.

See Also

- Show More Physics Options
- Domain, Boundary, and Pair Features for the Transport of Diluted Species Interface
- Theory for the Transport of Diluted Species Interface

Domain, Boundary, and Pair Features for the Transport of Diluted Species Interface

The Transport of Diluted Species Interface has these domain, boundary, and pair features available and listed in alphabetical order.

- Concentration
- Dynamic Transport Feature Node
- Flux
- Flux Discontinuity
- Inflow
- Initial Values
- Reactions
- No Flux (the default boundary condition)
- Open Boundary
- Outflow
- Periodic Condition
- Symmetry
- Thin Diffusion Barrier

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** node to the model that is valid on boundaries that represent the symmetry axis.

See Also

- Continuity on Interior Boundaries
- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Tip

A boundary pair occurs when the solutions on two separate surfaces within a simulation are related (such as when two components in an assembly are touching)

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Dynamic Transport Feature Node

This node is dynamic in that it is dependent on the transport mechanisms that have been chosen in the **Transport of Diluted Species** interface (convection and diffusion), and includes only the input fields required by the activated transport mechanisms. It has all the equations defining the transport of diluted species as well as inputs for the material properties. The name of the node also changes the activated transport mechanisms, and can be one of the following:

- Diffusion
- Convection and Diffusion

DOMAIN SELECTION

From the **Selection** list, choose the domains to define material properties and other parameters that govern the transport equations.

Note

If there is more than one type of domain, with different material properties, it may be necessary to deselect these domains. These are then defined in an additional **Convection and Diffusion** node.

MODEL INPUTS

If transport by convection is active the velocity field of the solvent needs to be specified as a model input.

Select the source of the **Velocity field** $\mathbf{u}$:

- Select **User defined** to enter values or expressions for the velocity components (SI unit: m/s) in the fields or table that appears below the drop-down menu. This input option is always available.
- Select the velocity field solved by a fluid flow physics interface that has also been added to the model. These physics interfaces have their own tags or Interface Identifier, and they are available to choose in the Velocity field drop-down menu, if they are also active in the domains being defined here. This lists the variable names related to the fluid flow physics interface in the table underneath the drop-down menu.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used for interpreting directions for an anisotropic diffusion coefficient.

DIFFUSION

Enter the **Diffusion coefficient** D_c (SI unit: m^2/s), which can be a scalar value for isotropic diffusion or a tensor describing anisotropic diffusion. Select the appropriate tensor type that describes the diffusion transport, and then enter the values in the corresponding field.

Reactions

Use the **Reactions** node to account for the consumption or production of species. Define the rate expression as required, which display on the right-hand side of the species transport equations in the **Convection and Diffusion** node. In other words, these two nodes are integrated.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define rate expression that govern the source term in the transport equations.

Note

If there is more than one type of domain, with subsequent and different reactions occurring within them, it may be necessary to remove these domains from the selection. These are then defined in an additional **Reactions** node.

REACTIONS

Add a rate expression, R_c (SI unit: mol/(m^3·s)), for the species to be solved for. Enter a value or expression in the field.

Initial Values

The **Initial Values** node adds the initial values for the concentration of each species to be specified. These serve as an initial guess for a stationary solver or as an initial condition for a transient simulation.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

Note

If there is more than one type of domain, each with different initial values defined, it may be necessary to remove these domains from the selection. These are then defined in an additional **Initial Values** node.

INITIAL VALUES

Enter a value or expression for the initial value of the concentration c, in the **Concentration** field. The default value is 0 mol/m^3.

Concentration

The **Concentration** boundary node adds a boundary condition for the species concentration. For example, a $c = c_0$ condition specifies the concentration of species c.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply the concentration boundary condition.

PAIR SELECTION

If **Concentration** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONCENTRATION

Specify the concentration for each species individually. Select the check box for the **Species** to specify the concentration, and then enter a value or expression in the corresponding field. To use another boundary condition for a specific species, click to clear the check box for that species' concentration.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

Flux

The **Flux** node can be used to specify the total species flux across a boundary. The total flux of species c is defined accordingly:

$$-\mathbf{n}\cdot(c\mathbf{u}-D\nabla c) = N_0$$

where N_0 is an arbitrary user-specified flux expression (SI unit: mol/(m^2·s)). For example, N_0 can represent a flux into a much larger surrounding environment, a phase change, or a flux due to chemical reactions.

When diffusion is the only transport mechanism present, the flux condition is extended to include a mass transfer term to describe flux into a surrounding environment:

$$-\mathbf{n}\cdot(-D\nabla c) = N_0+k_c(c_b-c)$$

where k_c is a mass transfer coefficient (SI unit: m/s), and c_b is the concentration (SI unit: mol/m^3) in the surroundings of the modeled system (the bulk concentration). The mass transfer coefficient is to be specified, often given by boundary-layer theory.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a flux boundary condition.

INWARD FLUX

Specify the flux of each species individually. Select the check box for the **Species** to specify the flux, and enter a value or expression in the corresponding field. To use another boundary condition for a specific species, click to clear the check box for that species' mass fraction.

Important

Use a minus sign appropriately when specifying a flux leaving the system.

Inflow

The **Inflow** node adds a boundary condition for an inflow boundary, where the concentration of all species $c = c_0$ is specified. This condition is similar to the **Concentration** feature, except the concentrations of all species must be specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply an inflow boundary condition.

CONCENTRATION

Enter a value or expression for the concentration of each species (SI unit: $\mathrm{mol/m^3}$).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

No Flux

The **No Flux** node, which is the default boundary condition on exterior boundaries, represents boundaries where no mass flows in or out of this boundary, such that the total flux is zero:

$$-\mathbf{n}\cdot(c\mathbf{u}-D\nabla c) = N_0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a no-flux boundary condition.

NO FLUX

Select **Apply for all species** to specify that the boundary is completely impervious for all species. If **Apply for...** is selected, click to select the check box for the species to specify the condition.

Outflow

Note

In models where only diffusion occurs, this boundary condition is not available.

Set the **Outflow** condition at outlets where species are transported out of the model domain by a fluid flow. It is assumed that convection is the dominating transport mechanism across the outflow boundary, and therefore that diffusive transport can be ignored, that is:

$$\mathbf{n}\cdot(-D\nabla c) = 0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply an outflow boundary condition.

Symmetry

The **Symmetry** node can be used to represent boundaries where the concentration of species is symmetric, that is, where there is no mass flux in the normal direction across the boundary.

This boundary condition is identical to that of the **No Flux** feature, but applies to all species and cannot be applied to individual species.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a symmetry condition.

Flux Discontinuity

The **Flux Discontinuity** node represents a discontinuity in the mass flux across an interior boundary:

$$-\mathbf{n}\cdot(\mathbf{N}_{\mathrm{d}}-\mathbf{N}_{\mathrm{u}}) = N_0 \qquad \mathbf{N} = (c\mathbf{u}-D\nabla c)$$

where the value N_0 (SI unit: $\mathrm{mol}/(\mathrm{m}^2\cdot\mathrm{s})$) specifies the jump in flux evaluated the boundary. This boundary condition is only available on interior boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the interior boundaries to apply a flux discontinuity boundary condition.

PAIR SELECTION

If **Flux Discontinuity** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

FLUX DISCONTINUITY

Specify the jump in species mass flux.

Tip

Use a positive value for increasing flux when going from the downside to the upside of the boundary. The boundary normal points in the direction from the downside to the upside of an interior boundary and can be plotted for visualization.

Select the check boxes for the species to specify a flux discontinuity, and enter a value or expression for the mass flux jump in the corresponding field. To use a different boundary condition for a specific species, click to clear the check box for that species' flux discontinuity.

Periodic Condition

The **Periodic Condition** node can be used to define periodicity or antiperiodicity between two boundaries. The feature can be activated on more than two boundaries, in which case the feature tries to identify two separate surfaces that can each consist of several connected boundaries. For more complex geometries it might be necessary to use the **Destination Selection** subnode. With this subnode the boundaries that constitute the source and destination surfaces can be manually specified. To add the subnode, right-click the **Periodic Condition** node and select **Destination Selection**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a periodic condition.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** or **Antiperiodicity**.

Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

- Periodic Condition
- Destination Selection
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Open Boundary

Use the **Open Boundary** node to set up mass transport across boundaries where both convective inflow and outflow can occur. Use this boundary condition to specify an exterior species concentration on parts of the boundary where fluid flows into the domain. A condition equivalent to the **Outflow** feature applies to parts of the boundary where fluid flows out of the domain.

Tip

The direction of the flow across the boundary is typically calculated by a **Fluid Flow** branch interface and is provided as a model input to the Transport of Diluted Species interface.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply the open boundary condition.

EXTERIOR CONCENTRATION

Enter a value or expression for the **Exterior concentration** $c_{0,c}$ (SI unit: mol/m^3).

Thin Diffusion Barrier

Use the **Thin Diffusion Barrier** boundary condition to model a thin layer through which mass is transported by diffusion only. To set up the node, specify the layer thickness and a diffusion coefficient for each transported species.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply the thin diffusion barrier.

PAIR SELECTION

When **Thin Diffusion Barrier** is selected from the **Pairs** menu, choose the **Pairs** to define. An identity pair has to be created first. Ctrl-click to deselect.

Note

A **No Flux** node is added by default. Right-click to add other pair nodes.

THIN DIFFUSION BARRIER

Enter a **Layer thickness** d_s (SI unit: m). The default is 0.005 m.

Enter a **Diffusion coefficient** $D_{s,c}$ (SI unit: mol/(m^3·s)). The default is 0.

12

The Fluid Flow Branch

This chapter explains how to use the Laminar Flow interface, found under the **Fluid Flow>Single-Phase Flow** branch (≋) in the Model Wizard, to model and simulate fluid mechanics for laminar, incompressible fluids. The engineering community often uses the term *CFD*, *computational fluid dynamics*, to refer to the numerical simulation of fluids.

In this chapter:

Theory for the Laminar Flow Interface

The Laminar Flow Interface theory is described in this section:

- General Single-Phase Flow Theory
- Compressible Flow
- The Mach Number Limit
- Incompressible Flow
- The Reynolds Number
- Numerical Stability—Stabilization Techniques for the Fluid Flow Interfaces
- Pseudo Time Stepping for Laminar Flow Models
- The Projection Method for the Navier-Stokes Equations
- The Boussinesq Approximation
- Theory for the Slip Wall Boundary Condition
- Theory for the Sliding Wall Boundary Condition
- Prescribing Inlet and Outlet Conditions
- Theory for the Pressure, No Viscous Stress Boundary Condition
- Theory for the Normal Stress Boundary Condition
- Discontinuous Galerkin Formulation
- Particle Tracing in Fluid Flow
- References for the Single-Phase Flow, Laminar Flow Interfaces

Note

The Laminar Flow interface has many shared features that are only available with some modules. For the theory relating to the advanced features, see the individual module documentation.

General Single-Phase Flow Theory

The single-phase fluid-flow interfaces are based on the Navier-Stokes equations, which in their most general form read

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{12-1}$$

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I}+\tau]+\mathbf{F} \tag{12-2}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T\right) = -(\nabla\cdot\mathbf{q}) + \tau{:}\mathbf{S} - \frac{T}{\rho}\frac{\partial\rho}{\partial T}\bigg|_p\left(\frac{\partial p}{\partial t} + (\mathbf{u}\cdot\nabla)p\right) + Q \tag{12-3}$$

where

- ρ is the density (SI unit: kg/m^3)
- $\mathbf{u}$ is the velocity vector (SI unit: m/s)
- p is pressure (SI unit: Pa)
- τ is the viscous stress tensor (SI unit: Pa)
- $\mathbf{F}$ is the volume force vector (SI unit: N/m^3)
- C_p is the specific heat capacity at constant pressure (SI unit: J/(kg·K))
- T is the absolute temperature (SI unit: K)
- $\mathbf{q}$ is the heat flux vector (SI unit: W/m^2)
- Q contains the heat sources (SI unit: W/m^3)
- $\mathbf{S}$ is the strain-rate tensor:

$$\mathbf{S} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$$

The operation “:” denotes a contraction between tensors defined by

$$\mathbf{a}{:}\mathbf{b} = \sum_n\sum_m a_{nm}b_{nm} \tag{12-4}$$

This is sometimes referred to as the double dot product.

Equation 12-1 is the continuity equation and represents the conservation of mass. Equation 12-2 is a vector equation and represents the conservation of momentum. Equation 12-3 describes the conservation of energy, formulated in terms of temperature. This is an intuitive formulation that facilitates boundary condition specification.

To close the equation system Equation 12-1 through Equation 12-3, some constitutive relations are needed. A common relation is derived by assuming that the fluid is Newtonian. Together with Stokes' assumption, the viscous stress tensor becomes:

$$\tau = 2\mu\mathbf{S} - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I} \tag{12-5}$$

The dynamic viscosity μ (SI unit: Pa·s) is allowed to depend on the thermodynamic state but not on the velocity field. All gases and many liquids can be considered Newtonian. Examples of non-Newtonian fluids are honey, mud, blood, liquid metals, and most polymer solutions.

Tip

When you have the CFD Module or the Microfluidics Module, you can model flows of non-Newtonian fluids using the predefined power law and Carreau models, which describes the dynamic viscosity for non-Newtonian fluids.

The Heat Transfer Module treats all fluids as Newtonian according to Equation 12-5.

Other commonly used constitutive relations are Fourier's law of heat conduction and the ideal gas law.

In theory, the same equations describe laminar as well as turbulent flows. In practice, however, the mesh resolution required to simulate turbulence with the Laminar Flow interface makes such an approach impractical.

See Also

There are several books where derivations of the Navier-Stokes equations and detailed explanations of concepts such as Newtonian fluids and the Stokes assumption are found. See, for example, the classical text by Batchelor (Ref. 1) and the more recent work by Panton (Ref. 2).

Many applications describe isothermal flows where Equation 12-3 is decoupled from Equation 12-1 and Equation 12-2.

Tip

If you have the Heat Transfer Module, the temperature equation is described in the section The Heat Equation found in the Heat Transfer Theory chapter.

Compressible Flow

The Navier-Stokes equations solved by default in all single-phase flow interfaces are the compressible formulation of the continuity:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{12-6}$$

and the momentum equations:

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right) + \mathbf{F} \tag{12-7}$$

These equations are applicable for incompressible as well as compressible flows where the density varies.

The Mach Number Limit

An important dimensionless number in fluid dynamics is the Mach number, Ma, defined by

$$\mathrm{Ma} = \frac{|\mathbf{u}|}{a}$$

where a is the speed of sound. A flow is formally incompressible when $\mathrm{Ma} = 0$. This is theoretically achieved by letting the speed of sound tend to infinity. The Navier-Stokes equations then have the numerical property that a disturbance anywhere in the computational domain is instantaneously spread to the entire domain. This results in a parabolic equation system.

The fully compressible Navier-Stokes equations, Equation 12-1 through Equation 12-3, have a finite speed of sound and hence a Mach number larger than zero. This has no numerical significance as long as the Mach number is well below one. However, when the Mach number approaches unity, the equations turn from parabolic to hyperbolic. When this happens, the numerical properties of the equation change, one of several implications being that the boundary conditions used for incompressible Navier-Stokes equations become invalid. The compressible formulation of the laminar and turbulent interfaces uses the same boundary conditions as the incompressible interfaces, which implies that the compressible interfaces cannot be used for flows with Mach number larger than or equal to one.

The practical Mach number limit is lower than one, however. The first reason for this is that the sound wave transport term that has been neglected in the heat equation.

This term becomes important already at moderate Mach numbers. The second reason is that already at moderate Mach number, the fully compressible Navier-Stokes equations start to display very sharp gradients. To handle these gradients, special numerical techniques are needed. It is impossible to give an exact limit where the low Mach number regime ends and the moderate Mach number regime begins, but a rule of thumb is that the Mach number effects start to appear at $\mathrm{Ma} = 0.3$. For this reason the compressible formulation is referred to as *compressible flow (Ma<0.3)* in COMSOL.

Tip

The CFD Module includes physics interfaces for modeling high Mach number compressible flows.

Incompressible Flow

When the temperature variations in a flow are small, a single-phase fluid can often be assumed incompressible; that is, ρ is constant or nearly constant. This is the case for all liquids under normal conditions and also for gases at low velocities. For constant ρ, Equation 12-6 reduces to

$$\rho\nabla\cdot\mathbf{u} = 0 \qquad (12\text{-}8)$$

and Equation 12-7 becomes

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F} \qquad (12\text{-}9)$$

The Reynolds Number

Fundamental to the analysis of fluid flow is the Reynolds number:

$$\mathrm{Re} = \frac{\rho UL}{\mu}$$

where U denotes a velocity scale, and L denotes a representative length. The Reynolds number represents the ratio between inertial and viscous forces. At low Reynolds numbers, viscous forces dominate and tend to damp out all disturbances, which leads to laminar flow. At high Reynolds numbers, the damping in the system is very low giving small disturbances the possibility to grow by nonlinear interactions. If the

Reynolds number is high enough, the fluid flow field eventually ends up in a chaotic state called turbulence. The Navier-Stokes interface automatically calculates the local cell Reynolds number $\mathrm{Re}^{\mathrm{c}} = \rho|\mathbf{u}|h/(2\mu)$ using the element length h for L and the magnitude of the velocity vector u for the velocity scale U. The cell Reynolds number is a predefined quantity available for visualization and evaluation (typically it is available as: `spf.cellRe`).

Numerical Stability—Stabilization Techniques for the Fluid Flow Interfaces

The momentum equations (Equation 12-7 and Equation 12-9) are (nonlinear) convection-diffusion equations. Such equations are unstable if discretized using the Galerkin finite element method. Stabilized finite element methods are therefore necessary in order to obtain physical solutions. The stabilization settings are found in the main fluid-flow features. To display this section, click the **Show** button ( ) and select **Stabilization**.

There are three types of stabilization methods available for Navier-Stokes—*streamline diffusion*, *crosswind diffusion*, and *isotropic diffusion*. Streamline diffusion (GLS) and crosswind diffusion are consistent stabilization methods, whereas isotropic diffusion is an inconsistent stabilization method.

For optimal functionality, the exact weak formulations and constants of GLS and crosswind diffusion depend on the order of the basis functions (elements). The values of constants of GLS and crosswind diffusion follow Ref. 3 and Ref. 4.

See Also

In the *COMSOL Multiphysics Reference Guide:*

- Stabilization Techniques
- Numerical Stabilization
- Preconditioners for Iterative Solvers
- Isotropic Diffusion

STREAMLINE DIFFUSION

For strongly coupled systems of equations, the streamline diffusion must be applied to the whole system of equations, not only to each equation separately. These ideas were first explored by Hughes and Mallet (Ref. 5) and were later extended to Galerkin least-squares (GLS) applied to the Navier-Stokes equations (Ref. 6), which is the form that COMSOL Multiphysics supports. The time-scale tensor is the diagonal tensor presented in Ref. 7.

Streamline diffusion is active by default because it is necessary when convection is the dominating part of the flow.

The unstabilized incompressible Navier-Stokes equations are subject to the Babuska-Brezzi condition, which states that the basis functions for the pressure must be of lower order than the basis functions for the velocity. If the incompressible Navier-Stokes equations are stabilized by GLS, it is possible to use equal-order interpolation. Hence, streamline diffusion is necessary when using first-order elements. This applies also if the model is solved using geometric multigrid (either as a solver or as a preconditioner) and at least one multigrid hierarchy level uses linear Lagrange elements.

CROSSWIND DIFFUSION

Crosswind diffusion can also be formulated for systems of equations, and when applied to the Navier-Stokes equations it becomes a shock-capturing operator. COMSOL supports the formulation in Ref. 6 with shock capturing viscosity taken from Ref. 8.

Incompressible flows do not contain shock waves, but crosswind diffusion is still useful for introducing extra diffusion in sharp boundary layers and shear layers that otherwise would require a very dense mesh to resolve.

The tuning parameter, C_k, controls the amount of crosswind diffusion introduced in the model. The recommended range for low Mach number flows and incompressible flows is $0 < C_k < 1.0$ ($C_k = 0$ means no diffusion at all). The value must not depend on space or time.

Crosswind diffusion is active by default as it makes it easier to obtain a solution even if the problem is fully resolved by the mesh.

! Important	Crosswind diffusion also enables the iterative solvers to use inexpensive presmoothers such as SSOR. If crosswind diffusion is deactivated, any SSOR solver preconditioners must be changed to the Vanka preconditioner.

ISOTROPIC DIFFUSION

Isotropic diffusion adds diffusion to the momentum equations of the Navier-Stokes equations The stability of the continuity equation is not improved.

Pseudo Time Stepping for Laminar Flow Models

A stationary formulation has per definition no time derivatives and Equation 12-9 reduces to:

$$\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]+\mathbf{F} \tag{12-10}$$

Solving Equation 12-10 requires a starting guess that is close enough to the final solution. If no such guess is at hand, the fully transient problem can be solved instead. This is, however, a rather costly approach in terms of computational time. An intermediate approach is to add a fictitious time derivative to Equation 12-10:

$$\rho\frac{\mathbf{u}-\mathrm{nojac}(\mathbf{u})}{\Delta\tilde{t}}+\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)]+\mathbf{F}$$

where $\Delta\tilde{t}$ is a *pseudo time step*. Since $\mathbf{u}$–`nojac`($\mathbf{u}$) is always zero, this term does not affect the final solution. It does, however, affect the discrete equation system and effectively transforms a nonlinear iteration into a time step of size $\Delta\tilde{t}$.

Pseudo time stepping is not active per default. However, the pseudo time step $\Delta\tilde{t}$ can be chosen individually for each element based on the local CFL number:

$$\Delta\tilde{t} = \mathrm{CFL}_{\mathrm{loc}}\frac{h}{|\mathbf{u}|}$$

where h is the mesh cell size. A small CFL number means a small time step. It is hence practical to start with a small CFL number and gradually increase it as the solution approaches steady state. If the automatic expression for $\mathrm{CFL}_{\mathrm{loc}}$ is

$$\begin{gathered}1.3^{\min(\mathrm{niterCMP},\,9)}+\\ \mathrm{if}(\mathrm{niterCMP}>20,\,9\cdot1.3^{\min(\mathrm{niterCMP}-20,\,9)},\,0)+\\ \mathrm{if}(\mathrm{niterCMP}>40,\,90\cdot1.3^{\min(\mathrm{niterCMP}-40,\,9)},\,0)\end{gathered} \tag{12-11}$$

The variable `niterCMP` is the nonlinear iteration number. It is equal to one for the first nonlinear iteration. $\mathrm{CFL}_{\mathrm{loc}}$ starts at 1.3 and increases with 30% each iteration until it reaches $1.3^9\approx10.6$. It remains there until iteration 20 where it starts to increase unit it reaches approximately 106. A final increase after iteration 40 then takes it to 1060.

The continuity equation has a time derivative only in the incompressible case and it can then be written as

$$\frac{\partial\rho}{\partial p}\frac{\partial p}{\partial t}$$

A pseudo time derivative can be written as

$$\frac{\partial\rho}{\partial p}\frac{p-\mathrm{nojac}(p)}{\Delta\tilde{t}}$$

For incompressible flow, the following approximation is used:

$$\frac{\partial\rho}{\partial p}\approx\frac{0.1\rho}{1[\mathrm{atm}]}$$

while for compressible flows

$$\frac{\partial\rho}{\partial p}\approx\mathrm{if}\left(\frac{\partial\rho}{\partial p}=0,\frac{0.1\rho}{1[\mathrm{atm}]},\frac{\partial\rho}{\partial p}\right)$$

Equation 12-11 can for some advanced flows increase $\mathrm{CFL}_{\mathrm{loc}}$ too slow or too quickly. $\mathrm{CFL}_{\mathrm{loc}}$ can then be prescribed manually.

The Projection Method for the Navier-Stokes Equations

A well-known approach to solve the Navier-Stokes equations is the pressure-correction method. This type of method is a so-called segregated method, and it generally requires far less memory than the COMSOL Multiphysics default formulation. Several variants of the original version have been developed (see Ref. 11, for example). COMSOL Multiphysics uses incremental pressure-correction schemes.

Important

This formulation is only available for time-dependent problems and requires the time discrete solver. It is available for Laminar Flow and Turbulent Flow, k-ε interfaces.

This method reformulates the Navier-Stokes equations so that it is possible to solve for one variable at a time in sequence. Let $\mathbf{u}$ and p be the velocity and pressure variables and $\mathbf{u}_{\mathrm{c}}$ and p_{c} the corrected velocity and pressure variables, respectively. The pressure-correction algorithm solves the Navier-Stokes equations using the following steps:

1 Solve in sequence for all $\mathbf{u}$ components following equation:

$$\rho\frac{\partial \mathbf{u}}{\partial t}+\rho\mathbf{u}_{\mathrm{c}}^{n}\cdot\nabla\mathbf{u}^{n+1} = -\nabla p^{n}+\nabla\cdot\left(\mu(\nabla\mathbf{u}^{n+1}+(\nabla\mathbf{u}^{n+1})^{T})-\frac{2}{3}\mu(\nabla\cdot\mathbf{u}^{n+1})\mathbf{I}\right)+\mathbf{F}$$

where the superscript index stands for the time-step index, and $\frac{\partial \mathbf{u}}{\partial t}$ is discretized using a BDF method up to the second order where $\mathbf{u}$ values from previous time steps are replaced by $\mathbf{u}_{\mathrm{c}}$ values. At the first order it is discretized it reads:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}_{\mathrm{c}}^{n}}{\text{timestep}}$$

2 Solve Poisson's equation to adjust the pressure:

$$\text{timestep}\Delta(p^{n+1}-p^{n}) = \frac{\partial\rho}{\partial t}-\nabla\cdot\rho\mathbf{u}^{n+1} \qquad (12\text{-}12)$$

3 Update the corrected velocity:

$$\mathbf{u}_{\mathrm{c}}^{n+1} = \mathbf{u}^{n+1}-\frac{\text{timestep}}{\rho}\nabla(p^{n+1}-p^{n})$$

Note

For incompressible flows, the $\frac{2}{3}\mu(\nabla\cdot\mathbf{u}^{n+1})\mathbf{I}$ term in Equation 12-11 and $\frac{\partial\rho}{\partial t}$ term in Equation 12-12 are suppressed.

Due to this specific discretization, this algorithm is only available with the time discrete solver.

Because velocity components and pressure are solved in a segregated way, boundary conditions might have a different implementation or might not be available with the projection method. In this case, this is mentioned in the documentation for each boundary condition.

When the projection method is used for turbulent flows or multiphysics couplings, the same algorithm is used for the velocity and pressure variables. Extra steps are needed to solve the other variables. By default the equation form used for these variables is the time-dependent form, and the time derivative is automatically discretized using a second-order BDF method.

The Boussinesq Approximation

The Boussinesq approximation is a way to treat some simple cases of buoyant flows without having to use a compressible formulation of the Navier-Stokes equations.

The Boussinesq approximation assumes that variations in density have no effect on the flow field except that they give rise to buoyant forces. The density is taken to be a reference value, ρ_0, except in the body force term, which is set to

$$\mathbf{F} = (\rho_0 + \Delta\rho)\mathbf{g} \tag{12-13}$$

where **g** is the gravity vector. Enter an expression for Equation 12-13 in the Volume force fields in the Volume Force feature; however, further simplifications are often possible. Because **g** can be written in terms of a potential, Φ, it is possible to write Equation 12-13 as:

$$\mathbf{F} = -\nabla(\rho_0\Phi) + \Delta\rho\mathbf{g}$$

The first part can be canceled out by splitting the true pressure, p, as a sum of a hydrodynamic component, P, and a hydrostatic component, $-\rho_0\Phi$. Then write Equation 12-8 and Equation 12-9 in terms of the hydrodynamic pressure $P = p + \rho_0\Phi$:

$$\rho\nabla\cdot\mathbf{u} = 0 \tag{12-14}$$

$$\rho_0\frac{\partial\mathbf{u}}{\partial t} + (\rho_0\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla P + \nabla\cdot(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)) + \mathbf{g}\Delta\rho \tag{12-15}$$

To obtain the Boussinesq approximation in this form, enter the expression for $\mathbf{g}\Delta\rho$ for the Volume Force feature.

In practice, the shift from p to P can be ignored except where the pressure appears in boundary conditions. The pressure that is specified at boundaries is the hydrodynamic pressure in this case. For example, on a vertical outflow or inflow boundary, the hydrodynamic pressure is typically a constant, while the true pressure is a function of the vertical coordinate.

The system that Equation 12-14 and Equation 12-15 form has its limitations. The main assumption is that the density fluctuations must be small; that is, $\Delta\rho/\rho_0 << 1$. There are also some more subtle constraints that, for example, makes the Boussinesq approximation unsuitable for systems of very large dimensions. An excellent discussion of the Boussinesq approximation and its limitations appears in Chapter 14 of Ref. 9.

See Also	See Volume Force for the node settings.

Theory for the Slip Wall Boundary Condition

The Slip condition assumes that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this may be a reasonable approximation if the important effect of the wall is to prevent fluid from leaving the domain. Mathematically, the constraint can be formulated as:

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad (-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = \mathbf{0}$$

The no penetration will take precedence over the Neumann part of the condition and the above expression is therefore equivalent to

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad \mathbf{K}-(\mathbf{K}\cdot\mathbf{n})\mathbf{n} = \mathbf{0}$$
$$\mathbf{K} = \mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)\mathbf{n}$$

expressing that there is no flow across the boundary and no viscous stress in the tangential direction.

Tip

If you have the CFD Module or Heat Transfer Module, boundary conditions for k and ε are

$$\nabla k\cdot\mathbf{n} = 0 \qquad \nabla\varepsilon\cdot\mathbf{n} = 0$$

When you have the Microfluidics Module, the boundary condition for ν is $\nabla\tilde{\nu}\cdot\mathbf{n} = 0$.

See Also

Wall for the node settings.

Theory for the Sliding Wall Boundary Condition

The Sliding Wall boundary condition is appropriate if the wall behaves like a conveyor belt; that is, the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

2D

In 2D, the tangential direction is unambiguously defined by the direction of the boundary, but the situation becomes more complicated in 3D. For this reason, this boundary condition has slightly different definitions in the different space dimensions.

2D

2D Axi

For 2D and 2D axisymmetric models, the velocity is given as a scalar U_w and the condition prescribes

$$\mathbf{u}\cdot\mathbf{n} = 0, \qquad \mathbf{u}\cdot\mathbf{t} = U_\mathrm{w}$$

where $\mathbf{t} = (n_y – n_x)$ for 2D and $\mathbf{t} = (n_z – n_r)$ for axial symmetry.

3D

For 3D models, the velocity is set equal to a given vector $\mathbf{u}_\mathrm{w}$ projected onto the boundary plane:

$$\mathbf{u} = \frac{\mathbf{u}_\mathrm{w} - (\mathbf{n}\cdot\mathbf{u}_\mathrm{w})\mathbf{n}}{\|\mathbf{u}_\mathrm{w} - (\mathbf{n}\cdot\mathbf{u}_\mathrm{w})\mathbf{n}\|}\|\mathbf{u}_\mathrm{w}\|$$

The normalization makes $\mathbf{u}$ have the same magnitude as $\mathbf{u}_\mathrm{w}$ even if $\mathbf{u}_\mathrm{w}$ is not exactly parallel to the wall.

See Also

Wall for the node settings.

Prescribing Inlet and Outlet Conditions

Navier-Stokes equations can show a large variety of numerical behaviors, ranging from almost completely elliptic to almost completely hyperbolic. This has implications when it comes to prescribing admissible boundary conditions. There is also a discrepancy between mathematically valid boundary conditions and practically useful boundary conditions.

INLET CONDITIONS

An inlet requires specification of the velocity field components. The most robust way to do this is to prescribe a velocity field.

A common alternative to prescribing the complete velocity field is to prescribe a pressure, in which case the normal velocity component will be specified indirectly via the continuity equation. The pressure can either be specified pointwise, which is mathematically an over-constraining but a numerically robust formulation. Alternatively, the pressure can be specified via a stress condition:

$$-p + \mu\frac{\partial u_n}{\partial n} = F_n \tag{12-16}$$

where $\partial u_n/\partial n$ is the normal derivative of the normal velocity field component. Equation 12-16 is mathematically more correct compared to specifying the pressure pointwise, but can on the same time not guarantee that p obtains the desired value.

OUTLET CONDITIONS

The most common approach is to prescribe a pressure on the outlet. As in the case of inlets, the pressure can either be specified pointwise or via a stress condition.

Specifying the pressure only is however mathematically not sufficient unless the Reynolds number is infinity. In practice, it can suffice if the Reynolds number is high enough. Otherwise, the pressure boundary condition must be supplemented by conditions on the tangential velocity components. This is often achieved by prescribing vanishing tangential stress:

$$\mu\frac{\partial u_t}{\partial n} = 0$$

where $\partial u_t/\partial n$ is the normal derivative of the tangential velocity field. It is also possible to prescribe u_t to be zero. The latter option should be used with care since it can have a significant effect on the upstream solution.

The elliptic characteristic of the Navier-Stokes equations permit mathematically a complete velocity field to be specified on an outlet. This can however difficult to apply in practice. The reason being that it is hard to specify the outlet velocities so that they at each point are consistent with the interior solution. The adjustment to the specified velocity will then occur across an outlet boundary layer. The thickness of this boundary layer will depend on the Reynolds number: the higher the Reynolds number, the thinner the boundary layer.

See Also

Inlet and Outlet for the node settings.

Theory for the Pressure, No Viscous Stress Boundary Condition

The Pressure, No Viscous Stress boundary condition specifies vanishing viscous stress along with a Dirichlet condition on the pressure:

$$\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\mathbf{n} = \mathbf{0}, \qquad p = p_0$$

$$\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}, \qquad p = p_0$$

using the compressible and the incompressible formulation, respectively.

This boundary condition is physically equivalent to a boundary that is adjacent to a large container (inlets) or exiting into a large container (outlets). It is numerically stable and admits total control of the pressure level along the entire boundary; however, it can give artifacts on the boundary if the viscous stresses just downstream from an inlet or inside the outlet are not zero. In such situations there are two choices. Either move the boundary farther away to a location where the artifacts do not matter or use a another stress type boundary condition present in the Boundary Stress feature.

While the Pressure, no viscous stress boundary condition is numerically more robust than the Normal stress condition (which also specifies the pressure), it is also theoretically an over-constraint of the flow (Ref. 2). This theoretical “flaw” is often ignored since it in most cases has no practical implication.

See Also

Inlet and Outlet for the node settings.

Theory for the Normal Stress Boundary Condition

The total stress on the boundary is set equal to a stress vector of magnitude f_0, oriented in the negative normal direction:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = -f_0\mathbf{n}$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}$$

using the compressible and the incompressible formulation, respectively.

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 is

$$p = 2\mu\frac{\partial u_n}{\partial n}+f_0 \tag{12-17}$$

If $\partial u_n/\partial n$ is small, Equation 12-17 states that $p \approx f_0$.

The Normal Stress condition is the mathematically correct version of the Pressure, No Viscous Stress condition (Ref. 2), but it is numerically less stable.

See Also

Inlet, Outlet, Open Boundary, and Boundary Stress for the individual node settings.

Discontinuous Galerkin Formulation

Some boundary conditions are implemented using a discontinuous Galerkin formulation. These boundary conditions include

- Wall – Slip
- Periodic Flow Condition
- Flow Continuity

The formulation used in the fluid flow interfaces in COMSOL is the Symmetric Interior Penalty Method (SIPG). The SIPG method can be regarded as fulfilling the boundary conditions in an integral sense rather than pointwise. More information on SIPG can be found in Ref. 12.

In particular, the SIPG formulation includes a penalty parameter that must be large enough for the formulation to be coercive. The higher the value, the more exactly the boundary condition is fulfilled, but a too high value results in an ill-conditioned equation system. The penalty parameter in COMSOL is implemented according to Ref. 13.

Particle Tracing in Fluid Flow

Tip

The Particle Tracing Module is available to assist with these types of modeling problems.

It is possible to do particle tracing with COMSOL Multiphysics provided that the impact of the particles on the flow field is negligible. It is then possible to first compute the flow field, then, as an analysis step, calculate the motion of particles. The motion of a particle is defined by Newton's second law

$$m\frac{d^2\mathbf{x}}{dt^2} = \mathbf{F}\left(t, \mathbf{x}, \frac{d\mathbf{x}}{dt}\right)$$

where $\mathbf{x}$ is the position of the particle, m the particle mass, and $\mathbf{F}$ is the sum of all forces acting on the particle. Examples of forces acting on a particle in a fluid are the drag force, the buoyancy force, and the gravity force. The drag force represents the force that a fluid exerts on a particle due to a difference in velocity between the fluid and the particle. It includes the viscous drag, the added mass, and the Basset history term. Several empirical expression have been suggested for the drag force. One of those is the one proposed by Khan and Richardson (Ref. 10). That expression is valid for spherical particles for a wide range of particle Reynolds numbers. The Reynolds particle number is defined as

$$\mathrm{Re}_p = \frac{|\mathbf{u} - \mathbf{u}_p| 2r\rho}{\mu}$$

where $\mathbf{u}$ is the velocity of the fluid, $\mathbf{u}_p$ the particle velocity, r the particle radius, ρ the fluid density, and μ the fluid dynamic viscosity. The empirical expression for the drag force according to Khan and Richardson is

$$\mathbf{F} = \pi r^2 \rho|\mathbf{u}-\mathbf{u}_p|(\mathbf{u}-\mathbf{u}_p)[1.84\mathrm{Re}_p^{-0.31}+0.293\mathrm{Re}_p^{0.06}]^{3.45}$$

Model

The model Flow Past a Cylinder (Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**) demonstrates how to add and set up particle tracing in a plot group using the **Particle Tracing with Mass** node. It uses the predefined Khan-Richardson model for the drag force and neglects gravity and buoyancy forces.

References for the Single-Phase Flow, Laminar Flow Interfaces

1. G.K. Batchelor, *An Introduction To Fluid Dynamics*, Cambridge University Press, 1967.

2. R.L. Panton, *Incompressible Flow*, 2nd ed., John Wiley & Sons, 1996.

3. I. Harari and T.J.R. Hughes, “What are C and h? Inequalities for the Analysis and Design of Finite Element Methods,” *Comp. Meth. Appl. Mech. Engrg*, vol. 97, pp. 157–192, 1992.

4. Y. Bazilevs, V.M. Calo, T.E. Tezduyar, and T.J.R. Hughes, “$YZ\beta$ Discontinuity Capturing for Advection-dominated Processes with Application to Arterial Drug Delivery,” *Int.J.Num. Meth. Fluids*, vol. 54, pp. 593–608, 2007.

5. T.J.R. Hughes and M. Mallet, “A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive System,” *Comp. Meth. Appl. Mech. Engrg*, vol. 58, pp. 305–328, 1986.

6. G. Hauke and T.J.R. Hughes, “A Unified Approach to Compressible and Incompressible Flows,” *Comp. Meth. Appl. Mech. Engrg*, vol. 113, pp. 389–395, 1994.

7. G. Hauke, “Simple Stabilizing Matrices for the Computation of Compressible Flows in Primitive Variables,” *Comp. Meth. Appl. Mech. Engrg*, vol. 190, pp. 6881–6893, 2001.

8. A. Soulaïmani and M. Fortin, “Finite Element Solution of Compressible Viscous Flows Using Conservative Variables,” *Comp. Meth. Appl. Mech. Engrg*, vol. 118, pp. 319–350, 2001.

9. D.J. Tritton, *Physical Fluid Dynamics*, 2nd ed., Oxford University Press, 1988.

10. J.M. Coulson and J.F. Richardson, “Particle Technology and Separation Processes,” *Chemical Engineering, Volume 2*, Butterworth-Heinemann, 2002.

11. J.L. Guermond, P. Minev, and J. Shen, “An overview of projection methods for incompressible flows,” *Comp. Meth. Appl. Mech. Engrg*, vol. 195, pp. 6011–6045, 2006.

12. B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*, SIAM, 2008.

13. Y. Epshteyn and B. Rivière, “Estimation of penalty parameters for symmetric interior penalty Galerkin methods”, *Journal of Computational and Applied Mathematics*, Vol. 206, pp. 843–872, 2007.

The Laminar Flow Interface

The **Laminar Flow** interface (≋), found under the **Single-Phase Flow** branch (≋) in the **Model Wizard**, has the equations, boundary conditions, and volume forces for modeling freely moving fluids using the Navier-Stokes equations, solving for the velocity field and the pressure. The main feature is **Fluid Properties**, which adds the Navier-Stokes equations and provides an interface for defining the fluid material and its properties.

Model

- Flow Past a Cylinder: Model Library path **COMSOL_Multiphysics/ Fluid_Dynamics/cylinder_flow**
- Terminal Falling Velocity of a Sand Grain: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/falling_sand**

When this interface is added, these default nodes are also added to the **Model Builder**—**Fluid Properties**, **Wall** (the default boundary condition is **No slip**), and **Initial Values**. Right-click the **Laminar Flow** node to add other features that implement, for example, boundary conditions and volume forces.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `spf`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the fluid pressure and velocity and the Navier-Stokes equations that describe those fields. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

Control the properties of the **Laminar Flow** interface, which control the overall type of fluid flow model.

Compressibility

By default the interface uses the **Compressible flow (Ma<0.3)** formulation of the Navier-Stokes equations. Select **Incompressible flow** to use the incompressible (constant density) formulation.

DEPENDENT VARIABLES

This interface defines these dependent variables (fields)—**Velocity field** $\mathbf{u}$ (SI unit: m/s) and **Pressure** p (SI unit: Pa).

If required, edit the field, component, and dependent variable names. Editing the name of a scalar dependent variable changes both its field name and the dependent variable name. If a new field name coincides with the name of another field of the same type, the fields will share degrees of freedom and dependent variable names. A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two fields share a common field name.

ADVANCED SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed.

Select the **Use pseudo time stepping for stationary equation form** check box to add pseudo time derivatives to the equation when the **Stationary equation** form is used. When selected, also choose a **CFL number expression**—**Automatic** (the default) or **Manual**. **Automatic** calculates the local CFL number (from the Courant–Friedrichs–Lewy condition) from a built-in expression. If **Manual** is selected, enter a **Local CFL number** $\mathrm{CFL}_{\mathrm{loc}}$.

See Also

The Projection Method for the Navier-Stokes Equations and Pseudo Time Stepping for Laminar Flow Models

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. It controls the discretization (the element types used in the finite element formulation). From the

Discretization of fluids list select the element order for the velocity components and the pressure: **P1+P1** (the default), **P2+P1**, or **P3+P2**.

- **P1+P1** (the default) means linear elements for both the velocity components and the pressure field. This is the default element order for the Laminar Flow and Turbulent Flow flow interfaces. Linear elements are computationally cheaper than higher-order elements and are also less prone to introducing spurious oscillations, thereby improving the numerical robustness.
- **P2+P1** means second-order elements for the velocity components and linear elements for the pressure field. This is the default for the Creeping Flow interface because second-order elements work well for low flow velocities.
- **P3+P2** means third-order elements for the velocity components and second-order elements for the pressure field. This can add additional accuracy but it also adds additional degrees of freedom compared to P2+P1 elements.

Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

See Also

- Show More Physics Options
- Domain, Boundary, Point, and Pair Features for the Single-Phase Flow Interfaces
- Theory for the Laminar Flow Interface

Domain, Boundary, Point, and Pair Features for the Single-Phase Flow Interfaces

The domain, boundary, point, and pair features are described in this section and listed in alphabetical order.

Note

The theory about most boundary conditions is found in P.M. Gresho and R.L. Sani, *Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley & Sons, 2000.

- Boundary Stress
- Flow Continuity
- Fluid Properties

- Initial Values
- Inlet
- Open Boundary
- Outlet
- Periodic Flow Condition
- Pressure Point Constraint
- Symmetry
- Volume Force
- Wall

2D Axi

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** feature to the model that is valid on the axial symmetry boundaries only.

Fluid Properties

The **Fluid Properties** feature adds the momentum equations solved by the interface, except for volume forces which are added by the Volume Force feature. The node also provides an interface for defining the material properties of the fluid.

DOMAIN SELECTION

From the **Selection** list, choose the domains to apply the fluid properties.

MODEL INPUTS

Edit input variables to the fluid-flow equations if required. For fluid flow, these are typically introduced when a material requiring inputs has been applied.

FLUID PROPERTIES

Density

The default **Density** ρ (SI unit: $\mathrm{kg/m^3}$) uses the value **From material**. Select **User defined** to enter a different value or expression.

Dynamic Viscosity

The default **Dynamic viscosity** μ (SI unit: Pa·s) uses the value **From material** and describes the relationship between the shear rate and the shear stresses in a fluid.

Intuitively, water and air have a low viscosity, and substances often described as thick (such as oil) have a higher viscosity.

- Select **User defined** to define a different value or expression. Using a built-in variable for the shear rate magnitude, `spf.sr`, makes it possible to define arbitrary expressions of the dynamics viscosity as a function of the shear rate.

Volume Force

The **Volume Force** feature specifies the volume force **F** on the right-hand side of the incompressible flow equation. Use it, for example, to incorporate the effects of gravity in a model.

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F}$$

If several volume force features are added to the same domain, then the sum of all contributions are added to the momentum equations.

DOMAIN SELECTION

From the **Selection** list, choose the domains where the volume force acts on the fluid.

VOLUME FORCE

Enter the components of the **Volume force F** (SI unit: N/m^3).

See Also

The Boussinesq Approximation

Initial Values

The **Initial Values** feature adds initial values for the velocity field and the pressure that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define initial values.

INITIAL VALUES

Enter values or expressions for the initial value of the **Velocity field** $\mathbf{u}$ (SI unit: m/s) and for the **Pressure** p (SI unit: Pa). The default values are 0.

Wall

The **Wall** feature includes a set of boundary conditions describing the fluid flow condition at a wall.

- No Slip (the default for laminar flow)
- Slip
- Sliding Wall
- Moving Wall
- Leaking Wall

See Also

- Theory for the Slip Wall Boundary Condition
- Theory for the Sliding Wall Boundary Condition
- The Moving Mesh Interface

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that represent solid walls.

PAIR SELECTION

If **Wall** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

BOUNDARY CONDITION

Select a **Boundary condition** for the wall. The boundary conditions available vary by interface.

No Slip

No slip is the default boundary condition for a stationary solid wall. The condition prescribes $\mathbf{u} = 0$, that is, that the fluid at the wall is not moving.

Slip

The **Slip** condition combines a no-penetration condition, $\mathbf{u}\cdot\mathbf{n}=0$. It hence implicitly assumes that there are no viscous effects at the slip wall and hence, no boundary layer develops. From a modeling point of view, this may be a reasonable approximation if the important effect of the wall is to prevent fluid from leaving the domain.

Sliding Wall

The **Sliding wall** boundary condition is appropriate if the wall behaves like a conveyor belt; that is, the surface is sliding in its tangential direction. The wall does not have to actually move in the coordinate system.

2D

In 2D, the tangential direction is unambiguously defined by the direction of the boundary, but the situation becomes more complicated in 3D. For this reason, this boundary condition has slightly different definitions in the different space dimensions. Enter the components of the **Velocity of the tangentially moving wall** U_w (SI unit: m/s).

3D

Enter the components of the **Velocity of the sliding wall** $\mathbf{u}_\mathrm{w}$ (SI unit: m/s). If the velocity vector entered is not in the plane of the wall, COMSOL projects it onto the tangential direction. Its magnitude is adjusted to be the same as the magnitude of the vector entered.

Moving Wall

If the wall moves, so must the fluid. Hence, this boundary condition prescribes $\mathbf{u} = \mathbf{u}_\mathrm{w}$. Enter the components of the **Velocity of moving wall** $\mathbf{u}_\mathrm{w}$ (SI unit: m/s).

Important

Specifying this boundary condition does not automatically cause the associated wall to move. An additional **Moving Mesh** interface needs to be added to physically track the wall movement in the spatial reference frame.

Leaking Wall

Use this boundary condition to simulate a wall where fluid is leaking into or leaving through a perforated wall $\mathbf{u} = \mathbf{u}_\mathrm{l}$. Enter the components of the **Fluid velocity** $\mathbf{u}_\mathrm{l}$ (SI unit: m/s).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional** for the following wall boundary conditions: No slip, Moving wall, and Leaking wall. The other types of wall boundary conditions with constraints use unidirectional constraints only. Select the **Use weak constraints** check box (available for all boundary conditions except Sliding wall, which does not add any constraints) to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

Inlet

The **Inlet** node includes a set of boundary conditions describing the fluid flow condition at an inlet. The **Velocity** boundary condition is the default.

Tip	In most cases the inlet boundary conditions appear, some of them slightly modified, in the Outlet type as well. This means that there is nothing in the mathematical formulations to prevent a fluid from leaving the domain through boundaries where the **Inlet** type is specified.

- Velocity (the default)
- Pressure, No Viscous Stress
- Normal Stress

See Also	• Theory for the Pressure, No Viscous Stress Boundary Condition • Theory for the Normal Stress Boundary Condition

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that represent inlets. The boundary conditional available is based on interface.

BOUNDARY CONDITION

Select a **Boundary condition** for the inlet. After selecting a **Boundary Condition** from the list, a section with the same name displays underneath. For example, if **Velocity** is selected, a **Velocity** section displays where further settings are defined for the velocity.

VELOCITY

Select **Normal inflow velocity** (the default) to specify a normal inflow velocity magnitude $\mathbf{u} = -\mathbf{n}\mathrm{U}_0$ where $\mathbf{n}$ is the boundary normal pointing out of the domain. Enter the velocity magnitude U_0 (SI unit: m/s).

If **Velocity field** is selected, it sets the velocity equal to a given velocity vector $\mathbf{u}_0$ when $\mathbf{u} = \mathbf{u}_{0.}$ Enter the velocity components $\mathbf{u}_0$ (SI unit: m/s) to set the velocity equal to a given velocity vector.

Note

This section displays when **Velocity** is selected as the **Boundary condition**. The option is available for the **Inlet** and **Outlet** boundary features.

PRESSURE, NO VISCOUS STRESS

The **Pressure, no viscous stress** boundary condition specifies vanishing viscous stress along with a Dirichlet condition on the pressure. Enter the **Pressure** p_0 (SI unit: Pa) at the boundary.

Note

This section displays when **Pressure, no viscous stress** is selected as the **Boundary condition**. The option is available for the **Inlet** and **Outlet** boundary features. Depending on the pressure field in the rest of the domain, an inlet boundary with this condition can become an outlet boundary.

NORMAL STRESS

Enter the magnitude of **Normal stress** f_0 (SI unit: N/m^2). Implicitly specifies that $p \approx f_0$.

Note

This section displays when **Normal Stress** is selected as the **Boundary condition**. The option is available for the **Inlet**, **Outlet**, **Open Boundary**, and **Boundary Stress** features.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. Select the **Use weak constraints** check box as required.

Outlet

The **Outlet** feature includes a set of boundary conditions describing fluid flow conditions at an outlet. The **Pressure, no viscous stress** boundary condition is the default. Other options are based on individual licenses. Selecting appropriate outlet conditions for the Navier-Stokes equations is not a trivial task. Generally, if there is something interesting happening at an outflow boundary, extend the computational domain to include this phenomenon.

Tip

All of the formulations for the **Outlet** type are also available, possibly slightly modified, in other boundary types as well. This means that there is nothing in the mathematical formulations to prevent a fluid from entering the domain through boundaries where the **Outlet** boundary type is specified.

- Pressure, No Viscous Stress (the default)
- Velocity
- Normal Stress
- Pressure
- No Viscous Stress

Note

The Pressure, No Viscous Stress, Velocity, and Normal Stress boundary conditions are described for the **Inlet** node.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that represent outlets.

BOUNDARY CONDITION

Select a **Boundary condition** for the outlet. **Pressure, no viscous stress** is the default. The boundary conditional available is based on interface.

PRESSURE

This boundary condition prescribes only a Dirichlet condition for the pressure $p = p_0$.

Enter the **Pressure** p_0 (SI unit: Pa) at the boundary. While this boundary condition is flexible and seldom produces artifacts on the boundary (compared to Pressure, No Viscous Stress), it can be numerically unstable. Theoretically, the stability is guaranteed by using streamline diffusion for a flow with a cell Reynolds number $\mathrm{Re}^c = \rho|\mathbf{u}|h/(2\mu) >>1$ (h is the local mesh element size). It does however work well in most other situations as well.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

Symmetry

The **Symmetry** node adds a boundary condition that describes symmetry boundaries in a fluid flow simulation. The boundary condition for symmetry boundaries prescribes no penetration and vanishing shear stresses. The boundary condition is a combination of a Dirichelet condition and a Neumann condition:

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad \left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^{T})-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = \mathbf{0}$$

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad (-p\mathbf{I}+\eta(\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}}))\mathbf{n} = \mathbf{0}$$

for the compressible and the incompressible formulation respectively. The Dirichlet condition takes precedence over the Neumann condition, and the above equations are equivalent to the following equation for both the compressible and incompressible formulation:

$$\mathbf{u}\cdot\mathbf{n} = \mathbf{0}, \qquad \mathbf{K}-(\mathbf{K}\cdot\mathbf{n})\mathbf{n} = \mathbf{0}$$
$$\mathbf{K} = \mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}})\mathbf{n}$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that are symmetry boundaries.

2D Axi

For 2D axial symmetry, a boundary condition does not need to be defined. For the symmetry axis at $r = 0$, the software automatically provides a condition that prescribes $u_r = 0$ and vanishing stresses in the z direction and adds an **Axial Symmetry** feature that implements this condition on the axial symmetry boundaries only.

Open Boundary

The **Open Boundary** node adds boundary conditions that describe boundaries that are open to large volumes of fluid. Fluid can both enter and leave the domain on boundaries with this type of condition.

See Also

- Inlet
- Outlet
- Pressure Point Constraint

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that are open boundaries.

BOUNDARY CONDITIONS

Select a **Boundary condition** for the open boundaries—**Normal stress** (the default) or **No viscous stress**. If **Normal stress** f_0 (SI unit: N/m^2) is selected, enter a value or expression for the boundary condition.

No Viscous Stress

If **No viscous stress** is selected, which is also available for the **Outlet** feature, it prescribes vanishing viscous stress:

$$\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\mathbf{n} = \mathbf{0}$$

$$\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)\mathbf{n} = \mathbf{0}$$

using the compressible and the incompressible formulation respectively.

Tip

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. To make the model numerically stable, combine this boundary condition with a point constraint on the pressure.

Note

This boundary condition is not available if the **Use memory-efficient form** check box is selected (click the **Show** button ( ) and select **Advanced Physics Options>Advanced Settings** to display the section) on any single-phase flow interface settings window.

Boundary Stress

The **Boundary Stress** node adds a boundary condition that represents a very general class of conditions also known as *traction boundary conditions*.

See Also

Inlet

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply boundary stress.

BOUNDARY CONDITION

Select a **Boundary condition** for the boundary stress—**General stress** (the default), **Normal stress**, or **Normal stress, normal flow**.

Normal Stress

If **Normal stress** f_0 (SI unit: N/m^2) is selected, enter a value or expression.

General Stress

If **General stress** is selected, enter the components of the **Stress F** (SI unit: N/m^2).The total stress on the boundary is set equal to a given stress **F**:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = \mathbf{F}$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = \mathbf{F}$$

using the compressible and the incompressible formulation respectively.

This boundary condition implicitly sets a constraint on the pressure that for 2D flows is

$$p = 2\mu\frac{\partial u_n}{\partial n}-\mathbf{n}\cdot\mathbf{F} \qquad (12\text{-}18)$$

If $\partial u_n/\partial n$ is small, Equation 12-18 states that $p \approx -\mathbf{n}\cdot\mathbf{F}$.

Normal Stress, Normal Flow

If **Normal stress, normal flow** is selected, enter the magnitude of the **Normal stress** f_0 (SI unit: N/m^2).

In addition to the stress condition set in the **Normal stress** condition, this condition also prescribes that there must be no tangential velocities on the boundary:

$$\left(-p\mathbf{I}+\left(\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)-\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = -\mathrm{f}_0\mathbf{n}, \qquad \mathbf{t}\cdot\mathbf{u} = 0$$

$$(-p\mathbf{I}+\mu(\nabla\mathbf{u}+(\nabla\mathbf{u})^T))\mathbf{n} = -\mathrm{f}_0\mathbf{n}, \qquad \mathbf{t}\cdot\mathbf{u} = 0$$

using the compressible and the incompressible formulation respectively.

This boundary condition also implicitly sets a constraint on the pressure that for 2D flows is

$$p = 2\eta\frac{\partial u_n}{\partial n}+\mathrm{f}_0 \qquad (12\text{-}19)$$

If $\partial u_n/\partial n$ is small, Equation 12-19 states that $p \approx \mathrm{f}_0$.

Note

This boundary condition is not available if the **Use memory-efficient form** check box is selected (click the **Show** button ( ) and select **Advanced Physics Options>Advanced Settings** to display the section) on any single-phase flow interface settings page.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select **Use weak constraints** as required.

Periodic Flow Condition

The **Periodic Flow Condition** splits its selection in two groups: one source group and one destination group. Fluid that leaves the domain through one of the destination boundaries enters the domain over the corresponding source boundary. This corresponds to a situation where the geometry is a periodic part of a larger geometry.

If the boundaries are not parallel to each other, the velocity vector is automatically transformed.

Tip

If the boundaries are curved, it is recommended to only include two boundaries.

The **Periodic Flow Condition** has no input when the interface property **Compressibility** is set to **Compressible flow (Ma<0.3)**. Typically when a periodic boundary condition is used with a compressible flow the pressure is the same at both boundaries and the flow is driven by a volume force. When **Compressibility** is set to **Incompressible flow**, the boundary condition contains an input field for a **Pressure difference**, $p_{\mathrm{src}} - p_{\mathrm{dst}}$. This pressure difference can, for example, drive the flow in a fully developed channel flow.

To set up a periodic boundary condition select both boundaries in the **Periodic Flow Condition** node. COMSOL automatically assigns one boundary as the source and the other as the destination. To manually set the destination selection, add a **Destination Selection** node to the **Periodic Flow Condition** node. All destination sides must be connected.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select **Use weak constraints** as required.

Note

This boundary condition is not available if the **Use memory-efficient form** check box is selected (click the **Show** button ( ) and select **Advanced Physics Options>Advanced Settings** to display the section) on any single-phase flow interface settings page.

See Also

Periodic Boundary Conditions

Flow Continuity

The **Flow Continuity** node is suitable for pairs where the boundaries match; it prescribes that the flow field is continuous across the pair.

A **Wall** subnode is added by default to the **Flow Continuity** node. The **Wall** feature applies to the parts of the pair boundaries where a source boundary lacks a corresponding destination boundary and vice versa. The **Wall** feature can be overridden by any other boundary condition that applies to exterior boundaries. Right-click the **Flow Continuity** node to add additional subfeatures.

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries for the selected pairs.

PAIR SELECTION

When **Flow Continuity** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Note

This boundary condition is not available if the **Use memory-efficient form** check box is selected (click the **Show** button ( ) and select **Advanced Physics Options>Advanced Settings** to display the section) on any single-phase flow interface settings page.

Pressure Point Constraint

The **Pressure Point Constraint** feature adds a pressure constraint at a point. If it is not possible to specify the pressure level using a boundary condition, the pressure must be set in some other way, for example, by specifying a fixed pressure at a point.

POINT SELECTION

From the **Selection** list, choose the points to use a pressure constraint.

PRESSURE CONSTRAINT

Enter a point constraint for the **Pressure** p_0 (SI unit: Pa).

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Select the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.

13

The Heat Transfer Branch

This chapter describes the different types of Heat Transfer interfaces (Heat Transfer in Solids and Heat Transfer in Fluids), and the Joule Heating interface, all found under the **Heat Transfer** branch (♨) in the **Model Wizard**.

In this chapter:

About the Heat Transfer Interfaces

The Heat Transfer interfaces model heat transfer by conduction and convection. Surface-to-ambient radiation effects around edges and boundaries can also be included. The interfaces are suitable for modeling heat transfer in solids and fluids. The interfaces are available in 1D, 2D, and 3D and for axisymmetric models with cylindrical coordinates in 1D and 2D. The default dependent variable is the temperature, *T*.

There are **Heat Transfer** interfaces displayed in the Model Builder with the same name but with different icons and default models. After selecting a **Heat Transfer** interface in the Model Wizard, default settings are added under the main node. For example, if **Heat Transfer in Solids** ( ) is selected, a **Heat Transfer** node is added with a default Heat Transfer in Solids model. If **Heat Transfer in Fluids** ( ) is selected, a **Heat Transfer in Fluids** model is added instead, but the parent nodes are both called Heat Transfer. Any interface based on the main Heat Transfer feature has the suffix `ht`.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Heat Transfer in Solids and Heat Transfer in Fluids

Use the **Heat Transfer in Solids (ht)** ( ) to model mainly heat transfer in solid materials. A default **Heat Transfer in Solids** model is added, but all functionality for including fluid domains is also available.

Use the **Heat Transfer in Fluids (ht)** ( ) to model mainly heat transfer in fluid materials. A default **Heat Transfer in Fluids** model is added, but all functionality for including solid domains is also available.

Joule Heating

Select **Joule Heating (jh)** ( ), found under the **Electromagnetic Heating** subbranch ( ), to combine all features from the Electric Currents interface with the Heat Transfer interface for modeling Joule heating (also called resistive heating or ohmic heating). See The Joule Heating Interface.

Theory for the Heat Transfer Interfaces

This section reviews the theory about the heat transfer equations. For more detailed discussions of the fundamentals of heat transfer, see Ref. 1 and Ref. 3.

The Heat Transfer Interface theory is described in this section:

- What is Heat Transfer?
- The Heat Equation
- A Note About Heat Flux
- Heat Flux Variables and Heat Sources
- About the Boundary Conditions for the Heat Transfer Interfaces
- Radiative Heat Transfer in Transparent Media
- Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces
- References for the Heat Transfer Interfaces

What is Heat Transfer?

Heat transfer is defined as the movement of energy due to a difference in temperature. It is characterized by the following mechanisms:

- *Conduction*—Heat conduction takes place through different mechanisms in different media. Theoretically it takes place in a gas through collisions of the molecules; in a fluid through oscillations of each molecule in a “cage” formed by its nearest neighbors; in metals mainly by electrons carrying heat and in other solids by molecular motion which in crystals take the form of lattice vibrations known as phonons. Typical for heat conduction is that the heat flux is proportional to the temperature gradient.
- *Convection*—Heat convection (sometimes called heat advection) takes place through the net displacement of a fluid, which transports the heat content in a fluid through the fluid’s own velocity. The term convection (especially convective cooling

and convective heating) also refers to the heat dissipation from a solid surface to a fluid, typically described by a heat transfer coefficient.

- *Radiation*—Heat transfer by radiation takes place through the transport of photons. Participating (or semitransparent) media absorb, emit and scatter photons. Opaque surfaces absorb or reflect them.

The Heat Equation

The fundamental law governing all heat transfer is the first law of thermodynamics, commonly referred to as the principle of conservation of energy. However, internal energy, U, is a rather inconvenient quantity to measure and use in simulations. Therefore, the basic law is usually rewritten in terms of temperature, T. For a fluid, the resulting *heat equation* is:

$$\rho C_p\left(\frac{\partial T}{\partial t}+(\mathbf{u}\cdot\nabla)T\right)=-(\nabla\cdot\mathbf{q})+\tau{:}\mathbf{S}-\frac{T}{\rho}\frac{\partial\rho}{\partial T}\bigg|_p\left(\frac{\partial p}{\partial t}+(\mathbf{u}\cdot\nabla)p\right)+Q \qquad (13\text{-}1)$$

where

- ρ is the density (SI unit: kg/m^3)
- C_p is the specific heat capacity at constant pressure (SI unit: J/(kg·K))
- T is absolute temperature (SI unit: K)
- $\mathbf{u}$ is the velocity vector (SI unit: m/s)
- $\mathbf{q}$ is the heat flux by conduction (SI unit: W/m^2)
- p is pressure (SI unit: Pa)
- τ is the viscous stress tensor (SI unit: Pa)
- $\mathbf{S}$ is the strain-rate tensor (SI unit: 1/s):

$$\mathbf{S}=\frac{1}{2}(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)$$

- Q contains heat sources other than viscous heating (SI unit: W/m^3)

For a detailed discussion of the fundamentals of heat transfer, see Ref. 1.

Note

Specific heat capacity at constant pressure is the amount of energy required to raise one unit of mass of a substance by one degree while maintained at constant pressure. This quantity is also commonly referred to as *specific heat* or *specific heat capacity.*

In deriving Equation 13-1, a number of thermodynamic relations have been used. The equation also assumes that mass is always conserved, which means that density and velocity must be related through:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v}) = 0$$

The heat transfer interfaces use Fourier's law of heat conduction, which states that the conductive heat flux, **q**, is proportional to the temperature gradient:

$$q_i = -k\frac{\partial T}{\partial x_i} \tag{13-2}$$

where k is the thermal conductivity (SI unit: W/(m·K)). In a solid, the thermal conductivity can be anisotropic (that is, it has different values in different directions). Then k becomes a tensor

$$k = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix}$$

and the conductive heat flux is given by

$$q_i = -\sum_j k_{ij}\frac{\partial T}{\partial x_j}$$

The second term on the right of Equation 13-1 represents viscous heating of a fluid. An analogous term arises from the internal viscous damping of a solid. The operation ":" is a contraction and can in this case be written on the following form:

$$\mathbf{a}:\mathbf{b} = \sum_n\sum_m a_{nm}b_{nm}$$

The third term represents pressure work and is responsible for the heating of a fluid under adiabatic compression and for some thermoacoustic effects. It is generally small for low Mach number flows. A similar term can be included to account for thermoelastic effects in solids.

Inserting Equation 13-2 into Equation 13-1, reordering the terms and ignoring viscous heating and pressure work puts the heat equation into a more familiar form:

$$\rho C_p\frac{\partial T}{\partial t}+\rho C_p\mathbf{u}\cdot\nabla T = \nabla\cdot(k\nabla T)+Q$$

The Heat Transfer interface with the Heat Transfer in Fluids feature solves this equation for the temperature, T. If the velocity is set to zero, the equation governing pure conductive heat transfer is obtained:

$$\rho C_p\frac{\partial T}{\partial t}+\nabla\cdot(-k\nabla T) = Q$$

A Note About Heat Flux

The concept of heat flux is not as simple as it might first appear. The reason is that heat is not a conserved property. The conserved property is instead the total energy. There is hence heat flux and energy flux which are similar, but not identical.

This section briefly describes the theory for the variables for *Total heat flux* and *Total energy flux*. The approximations made do not affect the computational results, only variables available for results analysis and visualization.

TOTAL ENERGY FLUX

The total energy flux for a fluid is equal to (Ref. 4, chapter 3.5)

$$\rho\mathbf{u}(H_0+\Psi)-k\nabla T+\tau\cdot\mathbf{u}+q_\mathrm{r} \qquad (13\text{-}3)$$

Above, H_0 is the total enthalpy

$$H_0 = H+\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})$$

where in turn H is the enthalpy. In Equation 13-3 τ is the viscous stress tensor and $q_\mathbf{r}$ is the radiative heat flux. Ψ in Equation 13-3 is the force potential. It can be formulated in some special cases, for example, for gravitational effects (Chapter 1.4 in Ref. 4), but it is in general rather difficult to derive. Potential energy is therefore often excluded and the total energy flux is approximated by

$$\rho\mathbf{u}\left(H+\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\right)-k\nabla T+\tau\cdot\mathbf{u}+q_{\mathrm{r}} \tag{13-4}$$

For a simple compressible fluid, the enthalpy, H, has the form (Ref. 5)

$$H = H_{\mathrm{ref}}+\int_{T_{\mathrm{ref}}}^{T} C_p dT+\int_{p_{\mathrm{ref}}}^{p}\frac{1}{\rho}\left(1+\frac{T}{\rho}\left(\frac{\partial\rho}{\partial T}\right)\bigg|_p\right)dp \tag{13-5}$$

where p is the absolute pressure. The reference enthalpy, H_{ref}, is the enthalpy at reference temperature, T_{ref}, and reference pressure, p_{ref}. In COMSOL, T_{ref} is 298.15 K and p_{ref} is one atmosphere. In theory, any value can be assigned to H_{ref} (Ref. 7), but for practical reasons, it is given a positive value according to the following approximations

- Solid materials and ideal gases: $H_{\mathrm{ref}} = C_{p,\,\mathrm{ref}}T_{\mathrm{ref}}$
- Gas/liquid: $H_{\mathrm{ref}} = (C_{p,\,\mathrm{ref}}/\gamma_{\mathrm{ref}})T_{\mathrm{ref}}+p_{\mathrm{ref}}/\rho_{\mathrm{ref}}$

where subscript "ref" indicates that the property is evaluated at the reference state.

The two integrals in Equation 13-5 are sometimes referred to as the *sensible enthalpy* (Ref. 7). These are evaluated in COMSOL by numerical integration. The second integral is only included for gas/liquid since it is commonly much smaller than the first integral for solids and it is identically zero for ideal gases.

Note

For the evaluation of H to work, it is important that the dependence of C_p, ρ and γ on the temperature are prescribed either via model input or as a function of the temperature variable. If C_p, ρ or γ depend on the pressure, the dependency must be prescribed either via model input or by using the variable pA which is the variable for the absolute pressure.

HEAT FLUX

The total heat flux vector is defined as (Ref. 6):

$$\rho\mathbf{u}U-k\nabla T+q_{\mathrm{r}} \tag{13-6}$$

where U is the internal energy. It is related to the enthalpy via

$$H = U+\frac{p}{\rho} \tag{13-7}$$

What is the difference between Equation 13-4 and Equation 13-7? As an example, consider a channel with fully developed incompressible flow with all properties of the fluid independent of pressure and temperature. The walls are assumed to be insulated. Assume that the viscous heating is neglected. This is a common approximation for low-speed flows.

There will be a pressure drop along the channel that drives the flow. Since there is no viscous heating and the walls are isolated, Equation 13-5 will give that $H_{\mathrm{in}}>H_{\mathrm{out}}$. Since everything else is constant, Equation 13-4 shows that the energy flux into the channel is higher than the energy flux out of the channel. On the other hand $U_{\mathrm{in}}=U_{\mathrm{out}}$, so the heat flux into the channel is equal to the heat flux going out of the channel.

If the viscous heating on the other hand is included, then $H_{\mathrm{in}}=H_{\mathrm{out}}$ (first law of thermodynamics) and $U_{\mathrm{in}}<U_{\mathrm{out}}$ (since work has been converted to heat).

Heat Flux Variables and Heat Sources

This section lists some predefined variables that are available to compute heat fluxes and sources. All the variable names start with the physics interface prefix. By default the Heat Transfer interface prefix is `ht`. As an example, the variable named `tflux` can be analyzed using `ht.tflux` (as long as the physics interface prefix is `ht`).

TABLE 13-1: HEAT FLUX VARIABLES

VARIABLE	NAME	GEOMETRIC ENTITY LEVEL
`tflux`	Total heat flux	domains, boundaries
`dflux`	Conductive heat flux	domains, boundaries
`turbflux`	Turbulent heat flux	domains, boundaries
`aflux`	Convective heat flux	domain, boundaries
`trlflux`	Translation heat flux	domains, boundaries
`teflux`	Total energy flux	domains, boundaries

`ccflux_u`
`ccflux_d`
`ccflux_z`
	Convective out-of-plane heat flux	out-of-plane domains (1D and 2D)

`rflux_u`
`rflux_d`
`rflux_z`
	Radiative out-of-plane heat flux	out-of-plane domains (1D and 2D), boundaries

`q0_u`
`q0_d`
`q0_z`
	Out-of-plane inward heat flux	out-of-plane domains (1D and 2D)

TABLE 13-1: HEAT FLUX VARIABLES

VARIABLE	NAME	GEOMETRIC ENTITY LEVEL
`ntflux`	Total normal heat flux	boundaries
`ndflux`	Normal conductive heat flux	boundaries
`naflux`	Normal convective heat flux	boundaries
`ntrlflux`	Normal translational heat flux	boundaries
`nteflux`	Normal total energy flux	boundaries
`ccflux`	Convective heat flux	boundaries
`Qtot`	Domain heat source	domains
`Qbtot`	Boundary heat source	boundaries
`Ql`	Line heat source	edges
`Qp`	Point heat source	points

DOMAIN HEAT FLUXES

On domains the heat fluxes are vector quantities. Their definition can vary depending on the active features and selected properties.

Total Heat Flux

On domains the total heat flux, `tflux`, corresponds to the conductive and convective heat flux. For accuracy reasons the radiative heat flux is not included.

Tip

See Radiative Heat Flux to evaluate the radiative heat flux.

For solid domains, for example heat transfer in solids and biological tissue domains, the total heat flux is defined by:

$$\mathrm{tflux} = \mathrm{trlflux} + \mathrm{dflux}$$

For fluid domains (for example, heat transfer in fluids), the total heat flux is defined by:

$$\mathrm{tflux} = \mathrm{aflux} + \mathrm{dflux}$$

Conductive Heat Flux

The conductive heat flux variable, `dflux` is evaluated using the temperature gradient and the effective thermal conductivity:

$$\mathrm{dflux} = -k_{\mathrm{eff}}\nabla T$$

When out-of-plane property is activated (1D and 2D only) the conductive heat flux is defined by

$$\text{dflux} = -d_z k_{\text{eff}} \nabla T$$

in 2D (d_z is the domain thickness)

$$\text{dflux} = -A_c k_{\text{eff}} \nabla T$$

in 1D (A_c is the cross-section area)

In the general case k_{eff} is the thermal conductivity, k.

For heat transfer in fluids with turbulent flow $k_{\text{eff}} = k + k_T$ where k_T is the turbulent thermal conductivity.

For heat transfer in porous media, $k_{\text{eff}} = k_{\text{eq}}$ where k_{eq} is the equivalent conductivity defined in the Porous Matrix feature.

Turbulent Heat Flux

The turbulent heat flux variable `turbflux` enables to access the part of the conductive heat flux that is due to the turbulence.

$$\text{turbflux} = -k_T \nabla T$$

Convective Heat Flux

The conductive heat flux variable `aflux` is defined using the internal energy:

$$\text{aflux} = \rho \mathbf{u} E$$

When out-of-plane property is activated (1D and 2D only) the convective heat flux is defined as

$$\text{aflux} = d_z \rho \mathbf{u} E$$

in 2D (d_z is the domain thickness)

$$\text{aflux} = A_c \rho \mathbf{u} E$$

in 1D (A_c is the cross-section area)

E is the internal energy defined by:

- $E = C_p T$ for solid domains,
- $E = C_p T/\gamma$ for ideal gas fluid domains,

- $E = H - p/\rho$ for other fluid domains.

H is the enthalpy defined by:

- $H = C_pT$ for solid domains,
- $H = C_pT + p/\rho$ for ideal gas fluid domains,
- $H = C_pT/\gamma + p/\rho$ for other fluid domains.

Translational Heat Flux

Similar to convective heat flux but defined for solid domains with translation. The variable name is `trlflux`.

Total Energy Flux

The total energy flux, `teflux`, is defined when viscous heating is enabled:

$$\mathrm{teflux} = \rho\mathbf{u}H_0 + \mathrm{dflux} + \tau\cdot\mathbf{u}$$

where the total enthalpy, H_0, is defined as:

$$H_0 = H + \frac{\mathbf{u}\cdot\mathbf{u}}{2}$$

Radiative Heat Flux

In participating media, the radiative heat flux, q_r, is not available for analysis on domains because it is much more accurate to evaluate

$$Q_r = \nabla\cdot q_r$$

the radiative heat source.

OUT-OF-PLANE DOMAIN FLUXES

When out-of-plane property is activated (1D and 2D only), out-of-plane domain fluxes are defined. If there are no out-of-plane features, they are evaluated to zero.

Convective Out-of-Plane Heat Flux

The convective out-of-plane heat flux, `ceflux`, is generated by the **Out-of-Plane Convective Cooling** feature.

In 2D:

$$\text{upside: } \mathrm{ccflux\_u} = h_u(T_{\mathrm{ext},\,u} - T)$$

$$\text{downside: } \mathrm{ccflux\_d} = h_d(T_{\mathrm{ext},\,d} - T)$$

In 1D:

$$\mathrm{ccflux\_z} = h_z(T_{\mathrm{ext},\,z} - T)$$

Radiative Out-of-Plane Heat Flux

The radiative out-of-plane heat flux, `rflux`, is generated by the **Out-of-Plane Radiation** feature.

In 2D:

$$\text{upside: } \mathbf{rflux\_u} = \varepsilon_u\sigma(T^4_{\mathrm{amb},\,u} - T^4)$$

$$\text{downside: } \mathbf{rflux\_d} = \varepsilon_d\sigma(T^4_{\mathrm{amb},\,d} - T^4)$$

In 1D:

$$\mathbf{rflux\_z} = \varepsilon_z\sigma(T^4_{\mathrm{amb},\,z} - T^4)$$

Out-of-Plane Inward Heat Flux

The convective out-of-plane heat flux, `q0`, is generated by the **Out-of-Plane Heat Flux** feature.

In 2D:

$$\text{upside: } q0\_\mathrm{u} = h_u(T_{\mathrm{ext},\,u} - T)$$

$$\text{downside: } q0\_\mathrm{d} = h_d(T_{\mathrm{ext},\,d} - T)$$

In 1D:

$$q0\_\mathrm{z} = h_z(T_{\mathrm{ext},\,z} - T)$$

BOUNDARY HEAT FLUXES

All the domain heat fluxes (vector quantity) are also available as boundary heat fluxes. The boundary heat fluxes are then equal to the mean value of the adjacent domains. In addition normal boundary heat fluxes (scalar quantity) are available on boundaries.

Total Normal Heat Flux

The variable `ntflux` is defined by:

$$\mathrm{ntflux} = \mathrm{mean}(\mathrm{tflux}) \cdot \mathbf{n}$$

Normal Conductive Heat Flux

The variable `ndflux` is defined by:

$$\mathrm{ndflux} = \mathrm{mean}(\mathrm{dflux}) \cdot \mathbf{n}$$

Normal Convective Heat Flux

The variable `naflux` is defined by:

$$\mathrm{naflux} = \mathrm{mean}(\mathrm{aflux}) \cdot \mathbf{n}$$

Normal Translational Heat Flux

The variable `ntrlflux` is defined by:

$$\mathrm{ntrlflux} = \mathrm{mean}(\mathrm{trlflux}) \cdot \mathbf{n}$$

Normal Total Energy Flux

The variable `nteflux` is defined by:

$$\mathrm{nteflux} = \mathrm{mean}(\mathrm{teflux}) \cdot \mathbf{n}$$

Radiative Heat Flux

On boundaries the radiative heat flux, `rflux`, is a scalar quantity defined as:

$$\mathrm{rflux} = \varepsilon\sigma(T_{\mathrm{amb}}^4 - T^4) + \varepsilon\sigma(G - T^4) + q_w$$

where the terms respectively account for surface-to-ambient radiative flux, surface-to-surface radiative flux and radiation in participating net flux.

Convective Heat Flux

Convective heat flux, `ccflux`, is defined as the contribution from **Convective Cooling** boundary condition:

$$\mathrm{ccflux} = h(T_{\mathrm{ext}} - T)$$

When out-of-plane property is activated (1D and 2D only) the convective cooling heat flux is defined as

in 2D (d_z is the domain thickness):

$$\mathrm{ccflux} = d_z h(T_{\mathrm{ext}} - T)$$

in 1D (A_c is the cross section area):

$$\mathrm{ccflux} = A_c h(T_{\mathrm{ext}} - T)$$

DOMAIN HEAT SOURCES

The sum of the domain heat sources added by different features are available in one variable, Q_{tot} (SI unit: W/m^3). This variable `Qtot` is the sum of:

- Q which is the heat source added by **Heat Source** and **Electromagnetic Heat Source** features.
- Q_{met} which is the heat source added by the **Bioheat** feature.

Note

The out-of-plane contributions (convective cooling, heat flux, and radiation), and the blood contribution in **Bioheat** are considered flux so that they are not added to Q_{tot}.

BOUNDARY HEAT SOURCES

The sum of the boundary heat sources added by different boundary conditions is available in one variable, $Q_{\mathrm{b,tot}}$ (SI unit: W/m^2). This variable `Qbtot` is the sum of:

- Q_{b} which is the boundary heat source added by **Boundary heat Source**, **Electrochemical reaction heat flux** and **Reaction heat flux** boundary conditions.
- Q_{sh} which is the boundary heat source added by **Boundary Electromagnetic Heat Source** boundary condition.
- Q_{s}: which is the boundary heat source added by **Layer heat source** subfeature of **Highly conductive layer**.

LINE AND POINT HEAT SOURCES

The sum of the line heat sources is available in a variable called `Ql` (SI unit: W/m).

The sum of the point heat sources is available in a variable called `Qp` (SI unit: W).

2D Axi

In 2D axisymmetric models, the SI unit for the variable `Qp` is W/m.

TEMPERATURE AND HEAT FLUX BOUNDARY CONDITIONS

The heat equation accepts two basic types of boundary conditions: *specified temperature* and *specified heat flux*. The former is of a constraint type and prescribes the temperature at a boundary:

$$T = T_0 \qquad \text{on } \partial\Omega$$

while the latter specifies the inward heat flux

$$-\mathbf{n}\cdot\mathbf{q} = q_0 \qquad \text{on } \partial\Omega$$

where

- $\mathbf{q}$ is the conductive heat flux vector (SI unit: W/m^2) where $\mathbf{q} = -k\nabla T$.
- $\mathbf{n}$ is the normal vector of the boundary.
- q_0 is inward heat flux (SI unit: W/m^2), normal to the boundary.

The inward heat flux, q_0, is often a sum of contributions from different heat transfer processes (for example, radiation and convection). The special case $q_0 = 0$ is called *thermal insulation*.

A common type of heat flux boundary conditions are those where $q_0 = h\cdot(T_{\mathrm{inf}} - T)$, where T_{inf} is the temperature far away from the modeled domain and the heat transfer coefficient, h, represents all the physics occurring between the boundary and “far away.” It can include almost anything, but the most common situation is that h represents the effect of an exterior fluid cooling or heating the surface of solid, a phenomenon often referred to as convective cooling or heating.

OVERRIDING MECHANISM FOR HEAT TRANSFER BOUNDARY CONDITIONS

Many boundary conditions are available in heat transfer. Some of them can be associated (for example, Heat Flux and Highly Conductive Layer). Others cannot be associated (for example, Heat Flux and Thermal Insulation).

Several categories of boundary condition exist in heat transfer. Table 13-2 gives the overriding rules for these groups.

- Temperature, Convective Outflow, Open Boundary, Inflow Heat Flux
- Thermal Insulation, Symmetry, Periodic Heat Condition
- Highly Conductive Layer

- Heat Flux, Convective Cooling, Electrochemical Reaction Heat Flux, Reaction Heat Flux
- Boundary Heat Source, Radiation Group
- Surface-to-Surface Radiation, Re-radiating Surface, Prescribed Radiosity, Surface-to-Ambient Radiation
- Opaque Surface, Incident Intensity, Continuity on interior boundaries
- Thin Thermally Resistive Layers

TABLE 13-2: OVERRIDING RULES FOR HEAT TRANSFER BOUNDARY CONDITIONS

A\B	1	2	3	4	5	6	7	8
1-Temperature	X	X	X					X
2-Thermal Insulation	X	X				X		
3-Highly Conductive Layer	X		X					
4-Heat Flux	X	X						
5-Boundary heat source								
6-Surface-to-surface radiation		X				X		
7-Opaque Surface							X	
8-Thin Thermally Resistive Layer	X							X

When there is a boundary condition *A* above a boundary condition *B* in the model tree and both conditions apply to the same boundary, use Table 13-2 to determine if *A* is overridden by *B* or not:

- Locate the line that corresponds to *A* group (see above the definition of the groups). In the table above only the first member of the group is displayed.
- Locate the column that corresponds to the group of *B*.
- If the corresponding cell is empty *A* and *B* contribute. If it contains an X, *B* overrides A.

Important

Group 4 and group 5 boundary conditions are always contributing. That means that they never override any other boundary condition. But they might be overridden.

Example 1

Consider a boundary where Temperature is applied. Then a Surface-to-surface Radiation boundary condition is applied on the same boundary afterward.

- Temperature belongs to group 1
- Surface-to-surface radiation belongs to group 6.
- The cell on the line of group 1 and the column of group 6 is empty so Temperature and Surface-to-Surface radiation contribute.

Example 2

Consider a boundary where Convective Cooling is applied. Then a Symmetry boundary condition is applied on the same boundary afterward.

- Convective Cooling belongs to group 4.
- Symmetry belongs to group 2
- The cell on the line of group 4 and the column of group 2 contains an X so Convective Cooling is overridden by Symmetry.

Note

In Example 2 above, if Symmetry followed by Convective Cooling is added, the boundary conditions contribute.

Radiative Heat Transfer in Transparent Media

This discussion so far has considered heat transfer by means of conduction and convection. The third mechanism for heat transfer is radiation. Consider an environment with fully transparent or fully opaque objects. Thermal radiation denotes the stream of electromagnetic waves emitted from a body at a certain temperature.

DERIVING THE RADIATIVE HEAT FLUX

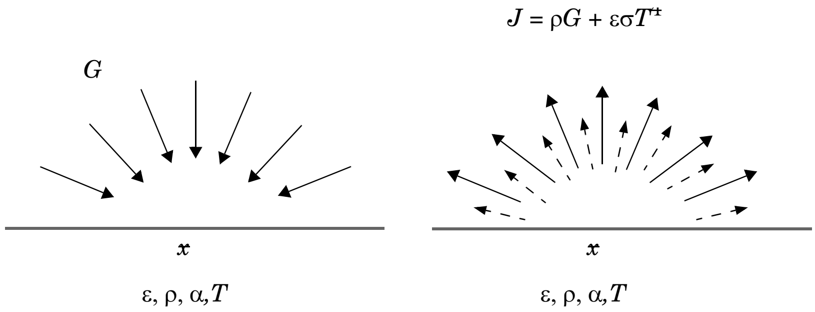

G
$J = \rho G + \varepsilon\sigma T^4$
$\bar{x}$
$\varepsilon, \rho, \alpha, T$
$\bar{x}$
$\varepsilon, \rho, \alpha, T$

Figure 13-1: Arriving irradiation (left), leaving radiosity (right).

Consider Figure 13-1. A point $\bar{x}$ is located on a surface that has an emissivity ε, reflectivity ρ, absorptivity α, and temperature T. Assume the body is opaque, which means that no radiation is transmitted through the body. This is true for most solid bodies.

The total *arriving* radiative flux at $\bar{x}$ is named the irradiation, G. The total outgoing radiative flux $\bar{x}$ is named the radiosity, J. The radiosity is the sum of the reflected radiation and the emitted radiation:

$$J = \rho G + \varepsilon\sigma T^4 \tag{13-8}$$

The net inward radiative heat flux, q, is then given the difference between the irradiation and the radiosity:

$$q = G - J \tag{13-9}$$

Using Equation 13-8 and Equation 13-9 J can be eliminated and a general expression is obtained for the net inward heat flux into the opaque body based on G and T.

$$q = (1-\rho)G - \varepsilon\sigma T^4 \tag{13-10}$$

Most opaque bodies also behave as ideal gray bodies, meaning that the absorptivity and emissivity are equal, and the reflectivity is therefore given from the following relation:

$$\alpha = \varepsilon = 1 - \rho \tag{13-11}$$

Thus, for ideal gray bodies, q is given by:

$$q = \varepsilon(G - \sigma T^4) \tag{13-12}$$

This is the equation used as a radiation boundary condition.

RADIATION TYPES

It is common to differentiate between two types of radiative heat transfer: *surface-to-ambient* radiation and *surface-to-surface* radiation. Equation 13-12 holds for both radiation types, but the irradiation term, G, is different for each of them. The Heat Transfer interface supports radiation.

SURFACE-TO-AMBIENT RADIATION

Surface-to-ambient radiation assumes the following:

- The ambient surroundings in view of the surface have a constant temperature, T_{amb}.
- The ambient surroundings behave as a blackbody. This means that the emissivity and absorptivity are equal to 1, and zero reflectivity.

These assumptions allows the irradiation to be explicitly expressed as

$$G = \sigma T_{\mathrm{amb}}^4 \qquad (13\text{-}13)$$

Inserting Equation 13-13 into Equation 13-12 results in the net inward heat flux for surface-to-ambient radiation

$$q = \varepsilon\sigma(T_{\mathrm{amb}}^4 - T^4) \qquad (13\text{-}14)$$

For boundaries where a surface-to-ambient radiation is specified, COMSOL Multiphysics adds this term to the right-hand side of Equation 13-14.

Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces

Several of the Heat Transfer interfaces have this advanced option to set the stabilization method parameters. Below is some information pertaining to these options.

To display this section, click the **Show** button ( ) and select **Stabilization**.

See Also

- Show Stabilization
- Stabilization Techniques and Numerical Stabilization in the *COMSOL Multiphysics Reference Guide*

CONSISTENT STABILIZATION

This section contains two consistent stabilization methods: streamline diffusion and crosswind diffusion. These are consistent stabilization methods, which means that they do not perturb the original transport equation.

The consistent stabilization methods take effect for fluids and for solids with Translational Motion. A stabilization method is active when the corresponding check box is selected.

Streamline Diffusion

Streamline diffusion is active by default and should remain active for optimal performance for heat transfer in fluids or other applications that include a convective or translational term.

Crosswind Diffusion

The crosswind diffusion provides extra diffusion in the region of sharp gradients. The added diffusion is orthogonal to the streamline diffusion, so streamline diffusion and crosswind diffusion can be used simultaneously.

When **Crosswind diffusion** is selected, enter a **Lower gradient limit** $\mathbf{g}_{\mathrm{lim}}$ (SI unit: K/m). The default is `0.01[K]/jh.helem`. The variable $\mathbf{g}_{\mathrm{lim}}$ is needed because both Equation 13-15 and Equation 13-16 contain terms of the form $1/|\nabla T|$, which become singular if $\nabla T = 0$. Hence, all occurrences of $1/|\nabla T|$ are replaced by $1/\max(|\nabla T|, g_{\mathrm{lim}})$ where g_{lim} is a measure of a small gradient.

The method in the Heat Transfer interfaces adds the following contribution to the weak formulation (see Codina in Ref. 2):

$$-\sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega^e} \frac{1}{2}\max\left(0, C^e - \frac{2k}{h|\beta|}\right)\frac{h|R|}{|\nabla T|}\nabla\hat{T}\left(I - \frac{\mathbf{u}\otimes\mathbf{u}}{|\mathbf{u}|^2}\right)\nabla T d\Omega \qquad (13\text{-}15)$$

where R is the PDE residual, $\hat{T}$ is the test function for T, h is the element size, and β is defined as

$$\beta = \begin{cases} \dfrac{\rho C_p(\mathbf{u}\cdot\nabla T)}{|\nabla T|^2}\nabla T & \text{if } \nabla T \neq 0 \\ 0 & \text{if } \nabla T = 0 \end{cases} \qquad (13\text{-}16)$$

INCONSISTENT STABILIZATION

This section contains one inconsistent stabilization method: isotropic diffusion. Adding isotropic diffusion is equivalent to adding a term to the physical diffusion coefficient. This means that the original problem is not solved, which is why isotropic diffusion is an inconsistent stabilization method. Still, the added diffusion definitely dampens the effects of oscillations, but try to minimize the use of isotropic diffusion.

By default there is no isotropic diffusion. To add isotropic diffusion, select the **Isotropic diffusion** check box. The field for the tuning parameter δ_{id} then becomes available. The default value is 0.25; increase or decrease the value of δ_{id} to increase or decrease the amount of isotropic stabilization.

References for the Heat Transfer Interfaces

1. F.P. Incropera and D.P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed., John Wiley & Sons, 1996.

2. R. Codina, “Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation,” *Comp. Meth.Appl. Mech. Engrg*, vol. 156, pp. 185–210, 1998.

3. A. Bejan, *Heat Transfer*, Wiley, 1993.

4. G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 2000.

5. R.L. Panton, *Incompressible Flow*, 2nd ed., John Wiley & Sons, 1996.

6. M. Kaviany, *Principles of Convective Heat Transfer*, 2nd ed., Springer, 2001.

7. T. Poinsot and D. Veynante, *Theoretical and Numerical Combustion*, Second Edition, Edwards, 2005.

The Heat Transfer Interface

The **Heat Transfer (ht)** interface is available in many forms and each one has the equations, boundary conditions, and sources for modeling conductive and convective heat transfer, and solving for the temperature.

When this interface is added, default nodes are added to the **Model Builder** based on the selection made in the **Model Wizard**—**Heat Transfer in Solids** or **Heat Transfer in Fluids**, **Thermal Insulation** (the default boundary condition), and **Initial Values**.Right-click the **Heat Transfer** node to add other features that implement, for example, boundary conditions and sources.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `ht`.

See Also

About the Heat Transfer Interfaces

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define heat transfer and a temperature field. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

Select a **Default model**— **Heat transfer in solids** or **Heat transfer in fluids**.

DEPENDENT VARIABLES

The **Heat Transfer** interface has a dependent variable for the **Temperature** T. The dependent variable names can be changed. Editing the name of a scalar dependent variable changes both its field name and the dependent variable name. If a new field name coincides with the name of another field of the same type, the fields will share degrees of freedom and dependent variable names. A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

ADVANCED SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. Normally these settings do not need to be changed.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**.

Select a **Temperature**—**Quadratic** (the default), **Linear**, **Cubic**, or **Quartic**. Select an element order for the **Surface Radiosity**—**Linear** (the default), **Quadratic**, **Cubic**, or **Quartic**. Specify the **Value type when using splitting of complex variables**—**Real** (the default) or **Complex**.

CONSISTENT AND INCONSISTENT STABILIZATION

To display this section, click the **Show** button ( ) and select **Stabilization**. The **Streamline diffusion** check box is selected by default and should remain selected for optimal performance for heat transfer in fluids or other applications that include a convective or translational term. **Crosswind diffusion** provides extra diffusion in the region of sharp gradients. The added diffusion is orthogonal to the streamline diffusion, so streamline diffusion and crosswind diffusion can be used simultaneously.

If **Crosswind diffusion** is selected, enter a **Lower gradient limit** g_{lim} (SI unit: K/m). The default is `0.01[K]/ht.helem`.

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Features for the Heat Transfer Interfaces
- Show Stabilization
- Consistent and Inconsistent Stabilization Methods for the Heat Transfer Interfaces
- Theory for the Heat Transfer Interfaces

Domain, Boundary, Edge, Point, and Pair Features for the Heat Transfer Interfaces

The Heat Transfer Interface has these domain, boundary, edge, point, and pair conditions available (listed in alphabetical order):

- Boundary Heat Source
- Continuity on Interior Boundary
- Continuity
- Heat Flux
- Heat Source
- Heat Transfer in Fluids
- Heat Transfer in Solids
- Initial Values
- Line Heat Source
- Outflow
- Pair Boundary Heat Source
- Pair Thin Thermally Resistive Layer
- Periodic Heat Condition
- Point Heat Source
- Reaction Heat Flux
- Surface-to-Ambient Radiation
- Symmetry
- Temperature
- Thermal Insulation (the default boundary condition)
- Thin Thermally Resistive Layer
- Translational Motion

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries into account and automatically adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Heat Transfer in Solids

The **Heat Transfer in Solids** model uses the heat equation version in Equation 13-17 as the mathematical model for heat transfer in solids:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \tag{13-17}$$

For a steady-state problem the temperature does not change with time and the first term disappears. It has these material properties: density ρ, heat capacity C_p, thermal conductivity k (a scalar or a tensor if the thermal conductivity is anisotropic), and Q, which is the heat source (or sink)—one or more heat sources can be added separately.

When parts of the model (for example, a heat source) are moving, right-click the **Heat Transfer in Solids** node to add a Translational Motion feature to take this into account.

DOMAIN SELECTION

Model

2D Heat Transfer Benchmark with Convective Cooling: Model Library path **COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d**

From the **Selection** list, choose the domains to define the heat transfer. The default is to use all domains.

MODEL INPUTS

This section contains fields and values that are inputs to expressions that define material properties. If such user-defined materials are added, the model inputs appear here. Initially, this section is empty.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used for interpreting directions of orthotropic and anisotropic thermal conductivity.

Heat Conduction

The default setting is to use the **Thermal conductivity** k (SI unit: W/(m·K)) **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the thermal conductivity, and enter another value or expression.

The thermal conductivity describes the relationship between the heat flux vector $\mathbf{q}$ and the temperature gradient ∇T as in $\mathbf{q} = -k\nabla T$, which is Fourier's law of heat conduction. Enter this quantity as power per length and temperature.

The components of a thermal conductivity k in the case that it is a tensor (k_{xx}, k_{yy}, and so on) are available as `ht.kxx`, `ht.kyy`, and so on (using the default Heat Transfer interface identifier `ht`). The single scalar mean effective thermal conductivity `ht.kmean` is the mean value of the diagonal elements k_{xx}, k_{yy}, and k_{zz}.

Thermodynamics

The default **Density** ρ (SI unit: kg/m^3) and **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)) use values **From material**. Select **User defined** to enter other values or expressions. The heat capacity at constant pressure describes the amount of heat energy required to produce a unit temperature change in a unit mass.

Model

Axisymmetric Transient Heat Transfer: Model Library path **COMSOL_Multiphysics/Heat_Transfer/heat_transient_axi**

Translational Motion

The **Translational Motion** node provides movement by translation to model heat transfer in solids. It adds the following contribution to the right-hand side of Equation 13-17:

$$-\rho C_\mathrm{p}\mathbf{u} \cdot \nabla T$$

The contribution describes the effect of a moving coordinate system that is required to model, for example, a moving heat source.

Caution

Special care must be taken at boundaries where $\mathbf{n}\cdot\mathbf{u} \neq 0$. The Heat Flux boundary condition does not, for example, work at boundaries where $\mathbf{n}\cdot\mathbf{u} < 0$.

DOMAIN SELECTION

From the **Selection** list, choose the domains to prescribe a translational motion.

Note

By default, the selection is the same as for the **Heat Transfer in Solids** node that it is attached to, but it is possible to use more than one **Heat Translation** subnode, each covering a subset of the **Heat Transfer in Solids** node's selection.

TRANSLATIONAL MOTION

Enter component values for x, y, and z (in 3D) for the **Velocity field** $\mathbf{u}_{\mathrm{trans}}$ (SI unit: m/s).

Heat Transfer in Fluids

The **Heat Transfer in Fluids** model uses the following version of the heat equation as the mathematical model for heat transfer in fluids:

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q \qquad (13\text{-}18)$$

For a steady-state problem the temperature does not change with time and the first term disappears. It has these material properties:

- The density (ρ)
- The fluid heat capacity at constant pressure (C_{p})—describes the amount of heat energy required to produce a unit temperature change in a unit mass
- The fluid thermal conductivity (k)—a scalar or a tensor if the thermal conductivity is anisotropic

- The fluid velocity field ($\mathbf{u}$)—can be an analytic expression or a velocity field from a fluid-flow interface
- The heat source (or sink) (Q)—one or more heat sources can be added separately

Also, the *ratio of specific heats* γ is defined. It is the ratio of heat capacity at constant pressure, C_p, to heat capacity at constant volume, C_v. When using the ideal gas law to describe a fluid, specifying γ is enough to evaluate C_p. For common diatomic gases such as air, $\gamma = 1.4$ is the standard value. Most liquids have $\gamma = 1.1$ while water has $\gamma = 1.0$. γ is used in the streamline stabilization and in the variables for heat fluxes and total energy fluxes.

Model

Heat Transfer by Free Convection: Model Library path **COMSOL_Multiphysics/Multiphysics/free_convection**

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the heat transfer.

MODEL INPUTS

This section has fields and values that are inputs to expressions that define material properties. If such user-defined property groups are added, the model inputs appear here.

There are also two standard model inputs—**Absolute pressure** and **Velocity field**. The absolute pressure is used in some predefined quantities that include the enthalpy (the energy flux, for example).

Absolute Pressure

Enter the **Absolute pressure** p_A (SI unit: Pa). The default is atmosphere pressure, 1 atm (101,325 Pa).

This section controls both the variable as well as any property value (reference pressures) used when solving for pressure. There are usually two ways of calculating the pressure when describing fluid flow, and mass and heat transfer. Solve for the absolute pressure or a pressure (often denoted gauge pressure) that relates back to the absolute pressure through a reference pressure.

Using one or the other usually depends on the system and the equations being solved for. For example, in a straight incompressible flow problem, the pressure drop over the modeled domain is probably many orders of magnitude less than atmospheric pressure, which, if included, reduces the chances for stability and convergence during the solving process for this variable. In other cases, the absolute pressure may be required to be solved for, such as where pressure is a part of an expression for gas volume or diffusion coefficients.

The absolute pressure model input is controlled by both a drop-down list and a check box within this section. Use the **User defined** option to manually define the absolute pressure in a system. This is the default setting.

The pressure variables solved for by a fluid-flow interface can also be used, which is selected from the list as, for example, **Pressure spf/fp**. Selecting a pressure variable also activates a check box for defining the reference pressure, where `1[atm]` (1 atmosphere) is the default value. This makes it possible to use a system-based (gauge) pressure as the pressure variable while automatically including the reference pressure in places where it is required, such as for gas flow governed by the gas law. While this check box maintains control over the pressure variable and instances where absolute pressure is required within this respective physics interface, it may not with physics interfaces that being coupled to. In such models, check the coupling between any interfaces using the same variable.

Velocity Field

From the **Velocity field** list, select an existing velocity field in the model (for example, **Velocity field (spf/fp1)** from a Laminar Flow interface) or select **User defined** to enter values or expressions for the components of the **Velocity field** (SI unit: m/s).

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used for interpreting directions of orthotropic and anisotropic thermal conductivity.

HEAT CONDUCTION

The default **Thermal conductivity** k (SI unit: W/(m·K)) is taken **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the thermal conductivity, and enter another value or expression.

The thermal conductivity describes the relationship between the heat flux vector $\mathbf{q}$ and the temperature gradient ∇T as in $\mathbf{q} = -k\nabla T$ which is Fourier's law of heat conduction. Enter this quantity as power per length and temperature.

THERMODYNAMICS

The default **Density** ρ (SI unit: kg/m^3), **Heat capacity at constant pressure** C_P (SI unit: J/(kg·K)), and **Ratio of specific heats** γ (unitless) for a general gas or liquid use values **From material**. Select **User defined** to enter other values or expressions.

Heat Source

The **Heat Source** describes heat generation within the domain. Express heating and cooling with positive and negative values, respectively. Add one or more nodes as required—all heat sources within a domain contribute to the total heat source. Specify the heat source as the heat per volume in the domain, as linear heat source, or as a total heat source (power).

DOMAIN SELECTION

From the **Selection** list, choose the domains to add the heat source.

HEAT SOURCE

Click the **General source**, **Linear source**, or **Total power** button.

Note

For the **Heat Transfer in Porous Media** interface, and for the Batteries & Fuel Cells Module, Corrosion Module, or Electrodeposition Module, when **General Source** is selected, heat sources from the electrochemical current distribution interfaces are also made available in the list under Heat Source. Choose the appropriate one or **User defined** to defined your own.

- If **General source** is selected, enter a value for the distributed heat source Q (SI unit: W/m^3).
- If **Total power** is selected, enter the total power (total heat source) P_{tot} (SI unit: W).
- If **Linear source** ($Q = q_s \cdot T$) is selected, enter the **Production/absorption coefficient** q_s (SI unit: W/(m^3·K)).

Tip

The advantage of writing the source in this form is that it can be stabilized by the streamline diffusion. The theory covers q_s that is independent of the temperature, but some stability can be gained as long as q_s is only weakly dependent on the temperature.

See Also

Stabilization Techniques in the *COMSOL Multiphysics Reference Guide*

Initial Values

The **Initial Values** node adds an initial value for the temperature that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. Right-click to add additional **Initial Values** nodes.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the **Temperature** T (SI unit: K). The default value is approximately room temperature, 293.15 K (20 °C).

Temperature

Use the **Temperature** node to specify the temperature somewhere in the geometry; for example, on boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a temperature.

TEMPERATURE

The equation for this condition is $T = T_0$ where T_0 is the prescribed temperature on the boundary. Enter the value or expression for the **Temperature** T_0 (SI unit: K). The default is 293.15 K.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options**. By default **Classic constraints** is selected. If required, select the **Use weak constraints** check box.

Use **Discontinuous Galerkin constraints** as an alternative to the **Classic constraints** when it does not work satisfactorily. This option is especially useful to prevent oscillations on inlet boundaries where convection dominates.

Unlike the **Classic constraints**, these constraints do not enforce the temperature on the boundary extremities. This is relevant on fluid inlets where the temperature may not be enforced on the walls at the inlet extremities.

Thermal Insulation

The **Thermal Insulation** node is the default boundary condition for all heat transfer interfaces. This boundary condition means that there is no heat flux across the boundary:

$$\mathbf{n}\cdot(k\nabla T) = 0$$

This condition specifies where the domain is well insulated. Intuitively this equation says that the temperature gradient across the boundary must be zero. For this to be true, the temperature on one side of the boundary must equal the temperature on the other side. Because there is no temperature difference across the boundary, heat cannot transfer across it.

An interesting numerical check for convergence is the numerical evaluation of the thermal insulation condition along the boundary. Another check is to plot the temperature field as a contour plot. Ideally the contour lines are perpendicular to any insulated boundary.

BOUNDARY SELECTION

Note

The default **Thermal Insulation** feature does not require any user input. If required, add more features by right-clicking the **Heat Transfer** node and selecting the boundaries to apply the thermal insulation.

Outflow

The **Outflow** node provides a suitable boundary condition for convection-dominated heat transfer at outlet boundaries. In a model with convective heat transfer, this condition states that the only heat transfer over a boundary is by convection. The temperature gradient in the normal direction is zero, and there is no radiation. This is usually a good approximation of the conditions at an outlet boundary in a heat transfer model with fluid flow.

BOUNDARY SELECTION

The **Outflow** node does not usually require any user input. If required, select the boundaries that are convection-dominated outlet boundaries.

Symmetry

The **Symmetry** node provides a boundary condition for symmetry boundaries. This boundary condition is similar to an insulation condition, and it means that there is no heat flux across the boundary.

BOUNDARY SELECTION

Note

In most cases, the **Symmetry** node does not require any user input. If required, define the symmetry boundaries.

Heat Flux

Use the **Heat Flux** node to add heat flux across boundaries. A positive heat flux adds heat to the domain. This feature is not applicable to inlet boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to add the heat flux contribution.

PAIR SELECTION

If **Heat Flux** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

HEAT FLUX

Click one of the **General inward heat flux**, **Inward heat flux**, or **Total heat flux** buttons.

- If **General inward heat flux** q_0 (SI unit: W/m^2) is selected, it adds to the total flux across the selected boundaries. Enter a value for q_0 to represent a heat flux that enters the domain. For example, any electric heater is well represented by this condition, and its geometry can be omitted.
- If **Inward heat flux** is selected, enter the **Heat transfer coefficient** h (SI unit: W/(m^2·K)). The default is 0. Also enter an **External temperature** T_{ext} (SI unit: K). The

default is 293.15 K. The value depends on the geometry and the ambient flow conditions. Inward heat flux is defined by this equation:

$$q_0 = h(T_{\text{ext}} - T)$$

For a thorough introduction about how to calculate heat transfer coefficients, see Incropera and DeWitt in Ref. 1.

- If **Total heat flux** is selected, enter the total heat flux q_{tot} (SI unit: W) for the total heat flux across the boundaries where the **Heat Flux** node is active.

Surface-to-Ambient Radiation

Use the **Surface-to-Ambient Radiation** boundary condition to add surface-to-ambient radiation to boundaries. The net inward heat flux from surface-to-ambient radiation is

$$q = \varepsilon\sigma(T_{\text{amb}}^4 - T^4)$$

where ε is the surface emissivity, σ is the Stefan-Boltzmann constant (a predefined physical constant), and T_{amb} is the ambient temperature.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to add surface-to-ambient radiation contribution.

SURFACE-TO-AMBIENT RADIATION

The default **Surface emissivity** e (a dimensionless number between 0 and 1) is taken **From material**. An emissivity of 0 means that the surface emits no radiation at all and an emissivity of 1 means that it is a perfect blackbody.

Enter an **Ambient temperature** T_{amb} (SI unit: K). The default is 293.15 K.

Periodic Heat Condition

Use the **Periodic Heat Condition** node to add a periodic heat condition to boundaries. Right-click the node to add a **Destination Selection** feature.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to add a periodic heat condition.

See Also

- Periodic Condition
- Destination Selection
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Boundary Heat Source

The **Boundary Heat Source** node models a heat source (or heat sink) that is embedded in the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply the heat source.

BOUNDARY HEAT SOURCE

Click the **General source** or **Total boundary power** button.

- If **General source** is selected, enter the boundary heat source Q_b (SI unit: W/m^2). A positive Q_b is heating and a negative Q_b is cooling. The default is 0.
- If **Total boundary power** is selected, enter the total power (total heat source) $P_{b,\mathrm{tot}}$ (SI unit: W).

Pair Boundary Heat Source

The **Pair Boundary Heat Source** node models a heat source (or heat sink) that is embedded in the boundary. It also prescribes that the temperature field is continuous across the pair.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply the heat source.

PAIR SELECTION

When **Pair Boundary Heat Source** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

BOUNDARY HEAT SOURCE

Click the **General source** or **Total boundary power** button.

- If **General source** is selected, enter the boundary heat source Q_b (SI unit: W/m^2). A positive Q_b is heating and a negative Q_b is cooling. The default is 0.
- If **Total boundary power** is selected, enter the total power (total heat source) $P_{\mathrm{b,tot}}$ (SI unit: W).

Continuity

The **Continuity** node can be added to pairs. It prescribes that the temperature field is continuous across the pair. **Continuity** is only suitable for pairs where the boundaries match.

BOUNDARY SELECTION

The selection list in this section shows the boundaries for the selected pairs.

PAIR SELECTION

When **Continuity** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Pair Thin Thermally Resistive Layer

Use the **Pair Thin Thermally Resistive Layer** node to define the thickness and thermal conductivity of a resistive material located on boundaries. It can be added to pairs.

The heat flux across the **Pair Thin Thermally Resistive Layer** is defined by

$$-\mathbf{n}_d \cdot (-k_d \nabla T_d) = -k_s \frac{T_u - T_d}{d_s}$$

$$-\mathbf{n}_u \cdot (-k_u \nabla T_u) = -k_s \frac{T_d - T_u}{d_s}$$

where u and d subscript refer respectively to the upside and the downside of the pair.

BOUNDARY SELECTION

The selection list in this section shows the boundaries for the selected pairs.

PAIR SELECTION

When **Pair Thin Thermally Resistive Layer** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

PAIR THIN THERMALLY RESISTIVE LAYER

Enter a value or expression for the **Layer thickness** d_s (SI unit: m).

The default is to use the **Thermal conductivity** k_s (SI unit: W/(m·K)) **From material**. Select **User defined** to enter another value or expression.

Thin Thermally Resistive Layer

Use the **Thin Thermally Resistive Layer** node to define the thickness and thermal conductivity of a resistive material located on boundaries.

The heat flux across the **Thin Thermally Resistive Layer** is defined by

$$-\mathbf{n}_d \cdot (-k_d \nabla T_d) = -k_s \frac{T_u - T_d}{d_s}$$

$$-\mathbf{n}_u \cdot (-k_u \nabla T_u) = -k_s \frac{T_d - T_u}{d_s}$$

where u and d subscript refer respectively to the upside and the downside of the slit.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to add a thermally resistive layer.

MODEL INPUTS

The temperature model input used to evaluate the material properties is equal to the mean temperature on interior boundaries.

THIN THERMALLY RESISTIVE LAYER

Enter a value or expression for the **Layer thickness** d_s (SI unit: m).

The default is to use the **Thermal conductivity** k_s (SI unit: W/(m·K)) **From material**. Select **User defined** to enter another value or expression.

Line Heat Source

The **Line Heat Source** node models a heat source (or sink) that is so thin that it has no thickness in the model geometry. Select this feature from the **Edges** menu.

3D

The **Line Heat** Source node is available in 3D only because a line in 2D is a boundary and a domain in 1D.

In theory, the temperature in a line source in 3D is plus or minus infinity (to compensate for the fact that the heat source does not have any volume). The finite element discretization used in COMSOL returns a finite temperature distribution along the line, but that distribution must be interpreted in a weak sense.

EDGE SELECTION

From the **Selection** list, choose the edges to apply the heat source.

LINE HEAT SOURCE

Click the **General source** or **Total power** button.

- If **General source** is selected, enter a value for the distributed heat source, Q_l, in unit power per unit length (SI unit: W/m). Positive Q_l is heating while a negative Q_l is cooling.
- If **Total power** is selected, enter the total power (total heat source) $P_{\mathrm{l,tot}}$ (SI unit: W)

Point Heat Source

The **Point Heat Source** node models a heat source (or sink) that is so small that it can be considered to have no spatial extension. Select this feature from the **Points** menu.

2D

3D

The **Point Heat Source** is available in 2D and 3D. It is not available in 1D since points are boundaries (possibly internal boundaries) there.

In theory, the temperature in a point source in 2D or 3D is plus or minus infinity (to compensate for the fact that the heat source does not have a spatial extension). The finite element discretization used in COMSOL returns a finite value, but that value must be interpreted in a weak sense.

POINT SELECTION

From the **Selection** list, choose the points to apply the heat source.

POINT HEAT SOURCE

Enter the quantity Q_p in unit power (SI unit: W). Positive Q_p is heating while a negative Q_p is cooling.

Continuity on Interior Boundary

The **Continuity on Interior Boundary** node enables intensity conservation across internal boundaries. It is the default boundary condition for all internal boundaries.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to activate the continuity on interior boundaries.

PAIR SELECTION

If **Continuity on Interior Boundary** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Reaction Heat Flux

Note

The **Reaction Heat Flux** feature is available only with the Plasma Module and is documented in the *Plasma Module User's Guide*.

The Joule Heating Interface

The **Joule Heating** interface ( ), found under the **Heat Transfer>Electromagnetic Heating** branch ( ) in the **Model Wizard**, combines all feature from the Electric Currents interface with the Heat Transfer interface for modeling *Joule heating* (*resistive heating* or *ohmic heating*). The interface is available in 2D, axisymmetric 2D, and 3D and has the equations, boundary conditions, and sources for modeling Joule heating, solving for the temperature and the electric potential. The interaction is coupled in both directions:

- The resistive heating appears as a heat source in the default **Electromagnetic Heat Source** node.
- The default setting is to use the value for the electrical conductivity σ from the material. By selecting **Linearized resistivity** from the σ list, the following temperature-dependent expression describes the electrical conductivity: $1/(\rho_0(1+\alpha(T-T_{\mathrm{ref}})))$, where T is the dependent variable for temperature from the heat transfer part, which automatically appears as a model input. By default, the values for ρ_0 (resistivity at reference temperature), α (temperature coefficient), and T_{ref} (reference temperature) are taken from the material. These settings are built into the Joule Heating Model node, which is the central feature in the Joule Heating interface.

Model

Joule Heating of a Microactuator—Distributed Parameter Version: Model Library path **COMSOL_Multiphysics/Tutorial_Models/thermal_actuator_jh_distributed**

When this interface is added, these default nodes are also added to the **Model Builder**—**Joule Heating Model**, **Electromagnetic Heat Source**, **Boundary Electromagnetic Heat Source, Thermal Insulation** and **Electric Insulation** (the default boundary conditions), and **Initial Values**. Right-click the **Joule Heating** node to add other features that implement boundary conditions and sources.

Note

Except where described in this section, this interface shares it features with The Electric Currents Interface and The Heat Transfer Interface.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `jh`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define heat transfer and a temperature field. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

If your license includes the Heat Transfer Module, you can extend the physical model with the following options:

- Select the **Out-of-plane heat transfer** check box (2D models only) to include heat transfer out of the plane.
- If your license includes the Heat Transfer Module, you can select the **Surface-to-surface radiation** check box to include surface-to-surface radiation as part of the heat transfer. This adds a **Radiation Settings** section. See the **Physical Model** section in The Heat Transfer Interface for details.
- If your license includes the Heat Transfer Module, you can select the **Radiation in participating media** check box to include radiation in participating media as part of the heat transfer. This adds a **Participating media Settings** section. See the **Physical Model** section in The Heat Transfer Interface for details.
- If your license includes the AC/DC Module, MEMS Module, or Plasma Module, you can select the **Porous media and mixtures** to include current conservation in mixtures and Archie's law as part of the physical model.
- If your license includes the Heat Transfer Module, you can select the **Heat Transfer in biological tissue** check box to enable the **Biological Tissue** feature. See the **Physical Model** section in The Bioheat Transfer Interface for details.

Radiation Settings

Note

This section is available when the **Surface-to-surface radiation** check box is selected. See **Radiation Settings** as described for The Heat Transfer Interface.

Participating Media Settings

Note

This section is available when the **Radiation in participating media** check box is selected. See **Participating Media Settings** as described for The Heat Transfer Interface.

Out-of-Plane Thickness

2D

Enter a default value for the **Out-of-plane thickness** d (SI unit: m). The default value of 1 is typically not representative for a thin dielectric medium, for example. Instead it describes a unit thickness that makes the 2D equation identical to the equation used for 3D models. See also Change Thickness (Out-of-Plane) (described for the **Electrostatics** interface).

Terminal Sweep Settings

This section is only available if your license includes the AC/DC Module, MEMS Module, or Plasma Module.

Enter a **Reference impedance** Z_{ref} (SI unit: W). The default is 50 W.

Select the **Activate terminal sweep** check box to switch on the sweep and invoke a parametric sweep over the terminals. Enter a **Sweep parameter name** to assign a specific name to the variable that controls the terminal number solved for during the sweep. The default is `PortName`. The generated lumped parameters are in the form of capacitance matrix elements. The terminal settings must consistently be of either fixed voltage or fixed charge type.

The lumped parameters are subject to **Touchstone file export**. Enter a file path or **Browse** for a file. Select an **Output format** for the Touchstone export—**Magnitude angle** (the default), **Magnitude (dB) angle**, or **Real imaginary**. Select a **Parameter to export**—**Z** (the default), **Y**, or **S**.

DEPENDENT VARIABLES

The interface has dependent variables for the **Temperature** T and **Electric potential** V. If the license includes the Heat Transfer Module, the **Surface radiosity** J is also included as a dependent variable.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select **Linear**, **Quadratic** (the default), **Cubic**, or **Quartic** for the **Temperature**, **Surface radiosity**, and **Electric potential**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex**.

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Features for the Joule Heating Interface

Domain, Boundary, Edge, Point, and Pair Features for the Joule Heating Interface

Because The Joule Heating Interface is a multiphysics interface, most features are shared with, and described for, other interfaces. Below are links to the domain, boundary, edge, pair, and point features as indicated and listed in alphabetical order.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

These are described in this section:

- Boundary Electromagnetic Heat Source
- Electromagnetic Heat Source
- Initial Values
- Joule Heating Model

These features are described for the **Heat Transfer** interface:

- Boundary Heat Source and Pair Boundary Heat Source
- Heat Flux
- Heat Source
- Heat Transfer in Fluids
- Heat Transfer in Solids
- Line Heat Source
- Outflow
- Point Heat Source
- Surface-to-Ambient Radiation
- Symmetry
- Temperature
- Thermal Insulation
- Thin Thermally Resistive Layer and Pair Thin Thermally Resistive Layer

These features are described for the **Electric Currents**, **Electrostatics**, or another interface:

- Boundary Current Source
- Contact Impedance
- Current Conservation
- Current Source
- Distributed Impedance
- Electric Insulation
- Electric Potential
- External Current Density
- Ground
- Line Current Source
- Line Current Source (on Axis)
- Normal Current Density
- Periodic Condition
- Point Current Source

Joule Heating Model

The **Joule Heating Model** node uses the following version of the *heat equation* as the mathematical model for heat transfer in solids:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \qquad (13\text{-}19)$$

with the following material properties:

- ρ is the *density*.
- C_p is the *heat capacity*.
- k is the *thermal conductivity* (a scalar or a tensor if the thermal conductivity is anisotropic).
- Q is the heat source (or sink). For Joule heating, it comes from the electric current and is added in the **Electromagnetic Heat Source** node.

For a steady-state problem the temperature does not change with time and the first term disappears. In addition, an equation for the electric current is also added.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define the Joule heating.

MODEL INPUTS

This section contains fields and values that are inputs to expressions that define material properties. If user-defined property groups are added, the model inputs appear here. By default, this section shows, unavailable, the temperature variable included in the Joule Heating interface.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

CONDUCTION CURRENT

The default **Electrical conductivity** σ (SI unit: S/m) is taken **From material.** Select **User defined** to enter a different value or expression or select **Linearized resistivity** to use the following temperature-dependent expression to describe the electrical conductivity:

$$1/(\rho_0(1+\alpha(T-T_{\mathrm{ref}})))$$

where T is the dependent variable for temperature. By default, the **Reference temperature** T_{ref} (SI unit: K), **Reference resistivity** ρ_0 (SI unit: Ω·m), and **Resistivity temperature coefficient** α (SI unit: 1/K) are taken **From material**. Select **User defined** to enter different values or expressions for these properties.

ELECTRIC FIELD

Specify the **Constitutive relation** that describes the macroscopic properties of the medium (relating the electric displacement $\mathbf{D}$ with the electric field $\mathbf{E}$) and the applicable material properties, such as the relative permittivity.

Select an option (the corresponding equation appears underneath the list)—**Relative permittivity**, **Polarization**, or **Remanent displacement**. Select:

- Select **Relative permittivity** to use the constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E}$ (the default). Then specify ε_r, the relative permittivity (unitless). The default is to take its value from the material, or select **User defined** to specify a value or expression for the relative permittivity.
- Select **Polarization** to use the constitutive relation $\mathbf{D} = \varepsilon_0\mathbf{E} + \mathbf{P}$. Then specify $\mathbf{P}$, the polarization vector (SI unit: C/m^2). Enter its components based on space dimension in the fields under **Polarization**.
- Select **Remanent displacement** to use the constitutive relation $\mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E} + \mathbf{D}_r$, where $\mathbf{D}_r$ (SI unit: C/m^2) is the remanent displacement (the displacement when no electric field is present). In this case, specify ε_r, the relative permittivity, and $\mathbf{D}_r$, the remanent displacement. For the relative permittivity, use the list under **Relative permittivity**: Select **From material** (the default) to use the value from the material or select **User defined** to specify a value or expression for the relative permittivity in the field that appears. For the remanent displacement, enter its components based on space dimension in the fields under **Remanent displacement**.

HEAT CONDUCTION

The default **Thermal conductivity** k (SI unit: W/(m·K)) uses values **From material**. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** based on the characteristics of the thermal conductivity and enter another value or expression in the field or matrix.

The thermal conductivity k describes the relationship between the heat flux vector $\mathbf{q}$ and the temperature gradient ∇T as in

$$\mathbf{q} = -k\nabla T$$

which is *Fourier's law of heat conduction*. Enter this quantity as power per length and temperature.

THERMODYNAMICS

The default **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)) and **Density** ρ (SI unit: kg/m^3) use values **From material**. Select **User defined** to enter other values or expressions.

Electromagnetic Heat Source

The **Electromagnetic Heat Source** feature is added as a default node. This node represents the source term Q (W/m^3) in the heat equation implemented by the Joule Heating Model node; see Equation 13-19. The resistive heating (ohmic heating), due to the electric current is

$$Q = \mathbf{J} \cdot \mathbf{E}$$

where $\mathbf{J}$ is the current density (A/m^2) and $\mathbf{E}$ is the electric field strength (V/m).

DOMAIN SELECTION

No user input is required for the default condition. Right-click the node and add additional **Electromagnetic Heat Source** features as required.

Boundary Electromagnetic Heat Source

The **Boundary Electromagnetic Heat Source** feature maps the electromagnetic surface losses as a heat source on the boundary (SI unit: W/m^2) in the heat transfer part of the model. It appears as a default feature.

BOUNDARY SELECTION

The default includes all boundaries in the model. From the **Selection** list, choose the boundaries to apply the model.

Initial Values

The **Initial Values** node adds initial values for the temperature T and the electric potential V that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver. If more than one set of initial values needs to be defined, right-click to add additional **Initial Values** nodes.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define initial values.

INITIAL VALUES

Enter values or expressions for the initial values of the **Temperature** T and the **Electric potential** V. The default value for the temperature is approximately room temperature, 293.15 K (20° C) and for the electric potential it is 0 V.

14

The Structural Mechanics Branch

This chapter explains how to use the Solid Mechanics interface, found under the **Structural Mechanics** branch () in the Model Wizard, to simulate and analyze applications involving solid mechanics. The interface is used for stress analysis and general solid mechanics simulation.

The optional Structural Mechanics Module contains physics interfaces and models that allow for extended, specialized analyses of structural and solid mechanics problems.

In this chapter:

- Solid Mechanics Geometry and Structural Mechanics Physics Symbols
- The Solid Mechanics Interface
- Theory for the Solid Mechanics Interface

Solid Mechanics Geometry and Structural Mechanics Physics Symbols

The Solid Mechanics interface in the Structural Mechanics Module is available for these space dimensions, which are described in this section:

- 3D Solid Geometry
- 2D Geometry (plane stress and plane strain)
- Axisymmetric Geometry

There are also physics symbols available with structural mechanics features as described in these sections:

- Physics Symbols for Boundary Conditions
- About Coordinate Systems and Physics Symbols
- Displaying Physics Symbols in the Graphics Window—An Example

3D Solid Geometry

The degrees of freedom (dependent variables) in 3D are the global displacements u, v, and w in the global x, y, and z directions, respectively, and the pressure help variable (used only if a nearly incompressible material is selected).

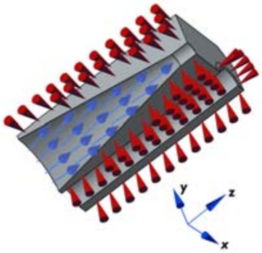


Figure 14-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.

2D Geometry

PLANE STRESS

The plane stress variant of the 2D interface is useful for analyzing thin in-plane loaded plates. For a state of plane stress, the out-of-plane components of the stress tensor are zero.

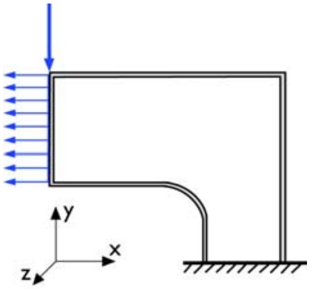


Figure 14-2: Plane stress models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

The 2D interface for plane stress allows loads in the *x* and *y* directions, and it assumes that these are constant throughout the material's thickness, which can vary with *x* and *y*. The plane stress condition prevails in a thin flat plate in the *xy*-plane loaded only in its own plane and without any *z* direction restraint.

PLANE STRAIN

The plane strain variant of the 2D interface that assumes that all out-of-plane strain components of the total strain ε_z, ε_{yz}, and ε_{xz} are zero.

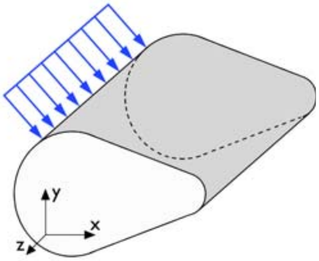


Figure 14-3: A geometry suitable for plane strain analysis.

Loads in the *x* and *y* directions are allowed. The loads are assumed to be constant throughout the thickness of the material, but the thickness can vary with *x* and *y*. The plane strain condition prevails in geometries, whose extent is large in the *z* direction compared to in the *x* and *y* directions, or when the *z* displacement is in some way restricted. One example is a long tunnel along the *z*-axis where it is sufficient to study a unit-depth slice in the *xy*-plane.

Axisymmetric Geometry

The axisymmetric variant of the **Solid Mechanics** interface uses cylindrical coordinates *r*, φ (`phi`), and *z*. Loads are independent of φ, and the axisymmetric variant of the interface allows loads only in the *r* and *z* directions.

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

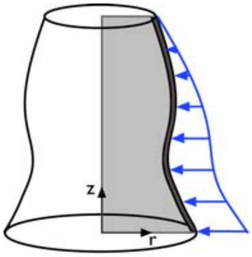


Figure 14-4: Rotating a 2D geometry to recover a 3D solid.

Physics Symbols for Boundary Conditions

To display the physics symbols, select **Options>Preferences>Graphics** from the Main menu and select the **Show physics symbols** check box to display the boundary condition symbols listed in Table 14-1. These symbols are available with the applicable structural mechanics features.

TABLE 14-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY FEATURE	NOTES
	Added Mass[1]	Added Mass	
	Antisymmetry[1]	Antisymmetry	
	Body Load[1]	Body Load	

TABLE 14-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY FEATURE	NOTES
	3D Coordinate System		Green indicates the Y direction, blue indicates the Z direction, and red indicates the X direction.
	2D Coordinate System		Green indicates the Y direction and red indicates the X direction.
	Distributed Force	Boundary Load Face Load Edge Load	Can be displayed together with the Distributed Moment symbol, depending on the values given in the feature.
	Damping[1]	Spring Foundation	Can be displayed together with the Spring symbol, depending on the values given in the feature.
	Distributed Moment[1]	Boundary Load Face Load Edge Load	Can be displayed together with the Distributed Force symbol, depending on the values given in the feature.
	Fixed Constraint	Fixed Constraint	
	No Rotation[1]	No Rotation	
	Pinned[1]	Pinned	
	Point Force	Point Load	Can be displayed together with the Point Moment symbol, depending on the values given in the feature.
	Point Mass[1]	Point Mass	
	Point Moment[1]	Point Load	Can be displayed together with the Point Force symbol, depending on the values given in the feature.
	Prescribed Displacement	Prescribed Displacement	
	Prescribed Velocity[1]	Prescribed Velocity	
	Rigid Connector[1]	Rigid Connector	
	Roller	Roller	

TABLE 14-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY FEATURE	NOTES
	Spring[1]	Spring Foundation Thin Elastic Layer	Can be displayed together with the Damping symbol, depending on the values given in the feature.
	Symmetry	Symmetry	

[1] Requires the Structural Mechanics Module

About Coordinate Systems and Physics Symbols

Physics symbols connected to a feature for which input can be given in different coordinate systems are shown together with a coordinate system symbol. This symbol is either a triad or a single arrow. The triad is shown if data are to be entered using vector components, as for a force. The single arrow is displayed when a scalar value, having an implied direction, is given. An example of the latter case is a pressure.

In both cases, the coordinate directions describe the direction in which a positive value acts. The coordinate direction symbols do not change with the values actually entered for the data.

Physics symbols are displayed even if no data values have been entered in the feature.

Important

For cases when physics symbol display is dependent on values actually given in the feature, it may be necessary to move to another feature before the display is actually updated on the screen.

Displaying Physics Symbols in the Graphics Window—An Example

1 Select **Options>Preferences>Graphics** from the Main menu and select the **Show physics symbols** check box. Click **OK**.

2 Add a physics interface, for example, **Solid Mechanics**, from the **Structural Mechanics** branch of the **Model Wizard**.

Note

The physics symbols also display for any multiphysics interface that includes Structural Mechanics feature nodes.

3 Add any of the feature nodes listed in Table 14-1 to the interface. Availability is based on license and interface.

4 When adding the boundary, edge, or point (a geometric entity) to the **Selection** list in the feature settings window, the symbol displays in the **Graphics** window. See Figure 14-5.

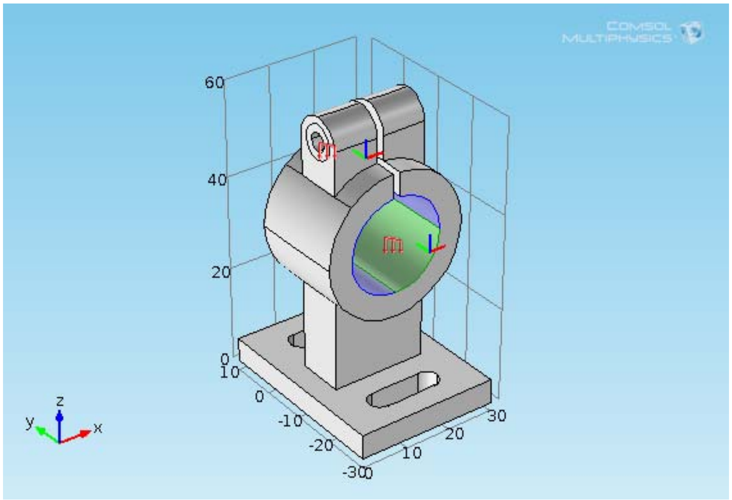


Figure 14-5: Example of the Boundary Load physics symbols as displayed in the COMSOL Multiphysics Model Library model Deformation of a Feeder Clamp.

5 After assigning the boundary condition to a geometric entity, to display the symbol, click the top level physics interface node and view it in the **Graphics** window. See Figure 14-6.

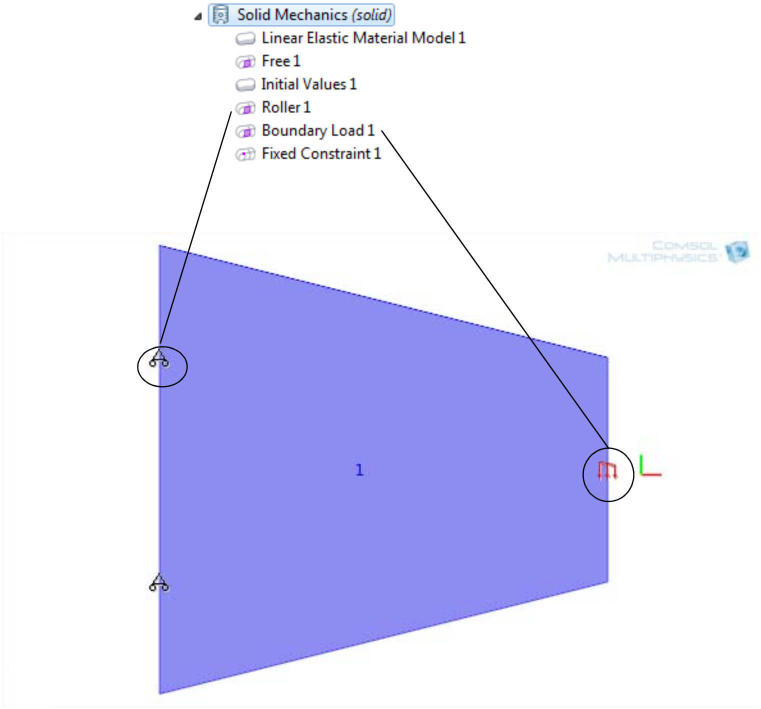


Figure 14-6: Example of Roller and Boundary Load physics symbols as displayed in the COMSOL Multiphysics Model Library model Tapered Membrane End Load.

The Solid Mechanics Interface

The **Solid Mechanics** interface ( ), found under the **Structural Mechanics** branch ( ) in the **Model Wizard**, has the equations and features for stress analysis and general linear solid mechanics, solving for the displacements. The **Linear Elastic Material** is the default material, which adds a linear elastic equation for the displacements and has a settings window to define the elastic material properties.

When this interface is added, these default nodes are also added to the **Model Builder**—**Linear Elastic Material**, **Free** (a boundary condition where boundaries are free, with no loads or constraints), and **Initial Values**. Right-click the **Solid Mechanics** node to add nodes that implement other solid mechanics features.

Model

- Stresses in a Pulley: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/stresses_in_pulley**
- Eigenvalue Analysis of a Crankshaft: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/crankshaft**

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `solid`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the displacements and the equations that describe the solid mechanics. To choose specific domains, select **Manual** from the **Selection** list.

2D APPROXIMATION

2D

From the **2D approximation** list select **Plane stress** or **Plane strain** (the default). For more information see the theory section.

When modeling using plane stress, the **Solid Mechanics** interface solves for the out-of-plane strain, e_z, in addition to the displacement field $\mathbf{u}$.

THICKNESS

2D

Enter a value or expression for the **Thickness** d (SI unit: m).

The default value of 1 m is suitable for plane strain models, where it represents a a unit-depth slice, for example. For plane stress models, enter the actual thickness, which should be small compared to the size of the plate for the plane stress assumption to be valid.

In rare cases, use a Change Thickness node to change thickness in parts of the geometry.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Quasi-static** or **Include inertial terms** to treat the elastic behavior as quasi-static (with no mass effects; that is, no second-order time derivatives) or as a mechanical wave in a time-dependent study. The default is to include the inertial terms to model the mechanical wave.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the coordinates for the **Reference point for moment computation** $\mathbf{x}_{\mathrm{ref}}$ (SI unit: m; COMSOL Multiphysics variable `refpnt`). All summed moments (applied or as reactions) are then computed relative to this reference point.

DEPENDENT VARIABLES

The interface uses the global spatial components of the **Displacement field** u as dependent variables. You can change both the field name and the individual component names. If a new field name coincides with the name of another displacement field, the two fields (and the interfaces which define them) will share degrees of freedom and dependent variable component names. You can use this behavior to connect a **Solid Mechanics** interface to a **Shell** directly attached to the boundaries of the solid domain, or to another **Solid Mechanics** interface sharing a common boundary.

A new field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization**. Select a **Displacement field**—**Linear**, **Quadratic** (the default)**, Cubic**, **Quartic**, or (in 2D) **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

See Also

- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Features for the Solid Mechanics Interface
- About the Body, Boundary, Edge, and Point Loads
- Theory for the Solid Mechanics Interface

Domain, Boundary, Edge, Point, and Pair Features for the Solid Mechanics Interface

The Solid Mechanics Interface has these domain, boundary, edge, point, and pair features listed in alphabetical order. The list also includes subfeatures.

- Antisymmetry
- Body Load
- Boundary Load
- Change Thickness
- Damping
- Edge Load
- Fixed Constraint
- Free
- Initial Values
- Linear Elastic Material
- Periodic Condition

- Point Load
- Prescribed Displacement
- Roller
- Symmetry

Note

If there are subsequent boundary conditions specified on the same geometrical entity, the last one takes precedence.

2D Axi

For 2D axisymmetric models, COMSOL Multiphysics takes the axial symmetry boundaries (at $r = 0$) into account and automatically adds an **Axial Symmetry** feature to the model that is valid on the axial symmetry boundaries only.

See Also

- Continuity on Interior Boundaries
- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Linear Elastic Material

The **Linear Elastic Material** feature adds the equations for a linear elastic solid and an interface for defining the elastic material properties. Right-click to add a Damping subnode.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define a linear elastic solid and compute the displacements, stresses, and strains.

MODEL INPUTS

Define model inputs, for example, the temperature field of the material uses a temperature-dependent material property. If no model inputs are required, this section is empty.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes (except boundary coordinate systems). The coordinate system is used when stresses or strains are presented in a local system.

LINEAR ELASTIC MATERIAL

Define the **Solid model** and the linear elastic material properties.

Solid Model

The solid model is linear isotropic elastic.

Specification of Elastic Properties for Isotropic Materials

If **Isotropic** is selected, from the **Specify** list, select a pair of elastic properties for an isotropic material. Select:

- **Young's modulus and Poisson's ratio** to specify Young's modulus (elastic modulus) E (SI unit: Pa) and Poisson's ratio ν (dimensionless). For an isotropic material Young's modulus 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. Poisson's ratio defines the normal strain in the perpendicular direction, generated from a normal strain in the other direction and follows the equation$\varepsilon_{\perp} = -\upsilon\varepsilon_{||}$.
- **Young's modulus and shear modulus** to specify Young's modulus (elastic modulus) E (SI unit: Pa) and the shear modulus G (SI unit: Pa). For an isotropic material Young's modulus 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. The shear modulus is a measure of the solid's resistance to shear deformations.
- **Bulk modulus and shear modulus** to specify the bulk modulus K (SI unit: Pa) and the shear modulus G (SI unit: Pa). The bulk modulus is a measure of the solid's resistance to volume changes. The shear modulus is a measure of the solid's resistance to shear deformations.

- **Lamé constants** to specify the first and second Lamé constants λ (SI unit: Pa) and μ (SI unit: Pa).
- **Pressure-wave and shear-wave speeds** to specify the pressure-wave speed (longitudinal wave speed) c_p (SI unit: m/s) and the shear-wave speed (transverse wave speed) c_s (SI unit: m/s).

Note

This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.

Tip

Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another (see Table 14-2).

Density

The default **Density** ρ (SI unit: kg/m^3) uses values **From material**. If **User defined** is selected, enter another value or expression.

Change Thickness

2D

The **Change Thickness** feature is available in 2D.

Use the **Change Thickness** feature to model domains with a thickness other than the overall thickness defined in the physics interface's **Thickness** section.

DOMAIN SELECTION

From the **Selection** list, choose the domains use a different thickness.

CHANGE THICKNESS

Enter a value for the **Thickness** d (SI unit: m). This value replaces the overall thickness for the domains selected above.

Damping

Right-click the **Linear Elastic Material** node to add a **Damping** node, which is used in time-dependent, eigenfrequency, and frequency domain studies to model undamped or damped problems. The **Damping** node adds Rayleigh damping by default

Note

The time-stepping algorithms also add numerical damping, which is independent of any explicit damping added.

For the generalized alpha time-stepping algorithm it is possible to control the amount of numerical damping added.

DOMAIN SELECTION

From the **Selection** list, choose the domains to add damping.

Note

By default, this node inherits the selection from its parent node, and only a selection that is a subset of the parent node's selection can be used.

DAMPING SETTINGS

Rayleigh damping is the only available **Damping type**.

Rayleigh Damping

Enter the **Mass damping parameter** α_{dM} (SI unit: 1/s) and the **Stiffness damping parameter** β_{dK} (SI unit: s). The default values are 0 (no damping).

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

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

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass, There is no direct physical interpretation of the mass damping parameter α_{dM} (SI unit: 1/s) and the stiffness damping parameter β_{dM} (SI unit: s).

Initial Values

The **Initial Values** node adds initial values for the displacement field and velocity field that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. Right-click to add additional **Initial Values** nodes.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter values or expressions for the initial values of the **Displacement field** u (SI unit: m) (the displacement components u, v, and w in 3D) and **Velocity field** $\partial u/\partial t$ (SI unit: m/s). The defaults are 0.

About the Body, Boundary, Edge, and Point Loads

Add force loads acting on all levels of the geometry to The Solid Mechanics Interface. Add a:

- Body Load to domains (to model gravity effects, for example).
- Boundary Load to boundaries (a pressure acting on a boundary, for example).
- Edge Load to edges in 3D (a force distributed along an edge, for example).
- Point Load to points (concentrated forces at points).

Body Load

Add a **Body Load** to domains for gravity effects, for example.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define a body load.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

2D

Select a **Load type** for 2D models and then enter values or expressions for the components in the matrix:

- **Load defined as force per unit area** $\mathbf{F}_A$ (SI unit: N/m^2). The body load as force per unit volume is then the value of F divided by the thickness.
- **Load defined as force per unit volume** $\mathbf{F}_V$ (the default) (SI unit: N/m^3)
- **Total force** $\mathbf{F}_{tot}$ (SI unit: N). COMSOL then divides the total force by the volume of the domains where the body load is active.

3D

Select a **Load type** for 3D models and then enter values or expressions for the components in the matrix:

- **Load defined as force per unit volume** $\mathbf{F}_V$ (the default) (SI unit: N/m^3)
- **Total force** $\mathbf{F}_{tot}$ (SI unit: N). COMSOL then divides the total force by the volume of the domains where the load is active.

Boundary Load

Add a **Boundary Load** to boundaries for a pressure acting on a boundary, for example.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a load.

PAIR SELECTION

If **Boundary Load** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the load. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

FORCE

Note

After selecting a **Load type**, the **Load** list normally only contains **User defined**. When combining the **Solid Mechanics** interface with, for example, film damping, it is also possible to choose a predefined load from this list.

2D

Select a **Load type** for 2D models and then enter a value or expression in the matrix:

- **Load defined as force per unit length** $\mathbf{F}_L$ (SI unit: N/m)
- **Load defined as force per unit area** $\mathbf{F}_A$ (SI unit: N/m^2)
- **Total force** $\mathbf{F}_{\mathrm{tot}}$ (SI unit: N). COMSOL then divides the total force by the area of the surfaces where the load is active.
- **Pressure** p (SI unit: Pa), which can represent a pressure or another external pressure. The pressure is positive when directed toward the solid.

3D

Select a **Load type** for 3D models and then enter a value or expression in the matrix:

- **Load defined as force per unit area** $\mathbf{F}_A$ (SI unit: N/m^2)
- **Total force** $\mathbf{F}_{\mathrm{tot}}$ (SI unit: N). COMSOL then divides the total force by the area of the surfaces where the load is active.
- **Pressure** p (SI unit: Pa), which can represent a pressure or another external pressure. The pressure is positive when directed toward the solid.

Edge Load

Add an **Edge Load** to 3D models for a force distributed along an edge, for example.

EDGE SELECTION

From the **Selection** list, choose the edges to define an edge load.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

3D

Select a **Load type** for 3D models and then enter values or expressions for the components in the matrix:

- **Load defined as force per unit length** $\mathbf{F}_\mathrm{L}$ (SI unit: N/m). When combining the **Solid Mechanics** interface with, for example, film damping, it is also possible to choose a predefined load from this list.
- **Total force** $\mathbf{F}_\mathrm{tot}$ (SI unit: N). COMSOL then divides the total force by the volume where the load is active.

Point Load

Add a **Point Load** to points for concentrated forces at points.

POINT SELECTION

From the **Selection** list, choose the points to define a point load.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

FORCE

Enter values or expressions for the components of the **Point load** $\mathbf{F}_\mathrm{p}$ (SI unit: N).

Free

The **Free** feature is the default boundary condition. It means that there are no constraints and no loads acting on the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that are free.

PAIR SELECTION

If **Free** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Fixed Constraint

The **Fixed Constraint** node adds a condition that makes the geometric entity fixed (fully constrained); that is, the displacements are zero in all directions. For domains, this condition is found under **More** on the context menu.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose, the geometric entity (domains, boundaries, edges, or points) that are fixed.

PAIR SELECTION

If **Fixed Constraint** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

Using Weak Constraints

Prescribed Displacement

The **Prescribed Displacement** feature adds a condition where the displacements are prescribed in one or more directions to the geometric entity (domain, boundary, edge, or point).

If a displacement is prescribed in one direction, this leaves the solid free to deform in the other directions. Also define more general displacements as a linear combination of the displacements in each direction.

Note

- If a prescribed displacement is not activated in any direction, this is the same as a **Free** constraint.
- If a zero displacement is applied in all directions, this is the same as a **Fixed Constraint**.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) to prescribe a displacement.

PAIR SELECTION

If **Prescribed Displacement** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes. If you choose another, local coordinate system, the displacement components change accordingly.

PRESCRIBED DISPLACEMENT

Define the prescribed displacements using a **Standard notation** or a **General notation**.

Standard Notation

To define the displacements individually, click the **Standard notation** button (the default).

2D

For 2D models, select one or both of the **Prescribed in x direction** and **Prescribed in y direction** check boxes. Then enter a value or expression for the prescribed displacements u_0 or v_0 (SI unit: m).

2D Axi

For 2D axisymmetric models, select one or both of the **Prescribed in r direction** and **Prescribed in z direction** check boxes. Then enter a value or expression for the prescribed displacements u_0 or w_0 (SI unit: m).

3D

For 3D models, select one or more of the **Prescribed in x direction**, **Prescribed in y direction**, and **Prescribed in z direction** check boxes. Then enter a value or expression for the prescribed displacements u_0, v_0, or w_0 (SI unit: m).

General Notation

To specify the displacements using a **General notation** that includes any linear combination of displacement components, click the **General notation Hu=R** button. For example, for 2D models, use the relationship

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

Enter values in the **H matrix** and **R vector** fields. For the H matrix, also select an **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** matrix and enter values as required. For example, to achieve the condition $u = v$, use the settings

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

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

CONSTRAINT SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

Using Weak Constraints

Symmetry

The **Symmetry** feature adds a boundary condition that represents symmetry in the geometry and in the loads. A symmetry condition is free in the plane and fixed in the out-of-plane direction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that are symmetry boundaries.

PAIR SELECTION

If **Symmetry** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

Using Weak Constraints

Antisymmetry

The **Antisymmetry** feature adds a boundary condition for an antisymmetry boundary, which must exist in both the geometry and in the loads. An antisymmetry condition is fixed in the plane and free in the out-of-plane direction.

Caution

In a geometrically nonlinear analysis, large rotations must not occur at the antisymmetry plane because this causes artificial straining.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that are antisymmetry boundaries.

PAIR SELECTION

If **Antisymmetry** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

Using Weak Constraints

Roller

The **Roller** node adds a roller constraint as the boundary condition; that is, the displacement is zero in the direction perpendicular (normal) to the boundary, but the boundary is free to move in the tangential direction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries that have roller constraints.

PAIR SELECTION

If **Roller** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

CONSTRAINT SETTINGS

To display this section, select click the **Show** button ( ) and select **Advanced Physics Options**. Select a **Constraint type**—**Bidirectional, symmetric** or **Unidirectional**. If required, select the **Use weak constraints** check box.

See Also

Using Weak Constraints

Periodic Condition

The **Periodic Condition** feature adds a periodic boundary condition. This periodicity makes $u_i(x_0) = u_i(x_1)$ for a displacement u_i. Control the direction that the periodic condition applies to. Right-click the **Periodic Condition** node to add a **Destination Selection** boundary condition. If the source and destination boundaries are rotated with respect to each other, this transformation is automatically performed, so that corresponding displacement components are connected.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a periodic boundary condition. The software automatically identifies the boundaries as either source boundaries or destination boundaries.

Note	This works fine for cases like opposing parallel boundaries. In other cases use a **Destination Selection** subnode to control the destination. By default it contains the selection that COMSOL Multiphysics has identified. In cases where the periodic boundary is split into several boundaries within the geometry, it may be necessary to apply separate periodic conditions to each pair of geometry boundaries.

PERIODICITY SETTINGS

Select a **Type of periodicity**—**Continuity** (the default), **Antiperiodicity** or **User defined**. If **User defined** is selected, select the **Periodic in u**, **Periodic in v** (for 3D and 2D models), and **Periodic in w** (for 3D and 2D axisymmetric models) check boxes as required. Then for each selection, choose the **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

See Also	• Periodic Condition and Destination Selection • Using Periodic Boundary Conditions • Periodic Boundary Condition Example

Perfectly Matched Layers

Note	For information about this feature, see About Infinite Element Domains and Perfectly Matched Layers.

Theory for the Solid Mechanics Interface

The Solid Mechanics Interface theory is described in this section:

- Material and Spatial Coordinates
- Coordinate Systems
- Lagrangian Formulation
- About Linear Elastic Materials
- Strain-Displacement Relationship
- Stress-Strain Relationship
- Plane Strain and Plane Stress Cases
- Axial Symmetry
- Loads
- Pressure Loads
- Equation Implementation
- Setting up Equations for Different Studies
- Damping Models

Material and Spatial Coordinates

The Solid Mechanics interface, through its equations, describes the motion and deformation of solid objects in a 2- or 3-dimensional space. In COMSOL's terminology, this physical space is known as the *spatial frame* and positions in the physical space are identified by lowercase *spatial coordinate* variables x, y, and z (or r, φ , and z in axisymmetric models).

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

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

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \qquad (14\text{-}1)$$

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

By default, the Solid Mechanics interface uses the calculated displacement and Equation 14-1 to define the difference between spatial coordinates **x** and material coordinates **X**. This means the material coordinates relate to the original geometry, while the spatial coordinates are solution dependent.

Material coordinate variables X, Y, and Z must be used in coordinate-dependent expressions that refer to positions in the original geometry, for example, for material properties that are supposed to follow the material during deformation. On the other hand, quantities that have a coordinate dependence in physical space, for example, a spatially varying electromagnetic field acting as a force on the solid, must be described using spatial coordinate variables x, y, and z. Any use of the spatial variables will be a source of nonlinearity if a geometrically nonlinear study is performed.

Coordinate Systems

Force vectors, stress and strain tensors, as well as various material tensors are represented by their components in a specified coordinate system. By default, material properties use the canonical system in the material frame. This is the system whose basis vectors coincide with the X, Y, and Z axes. When the solid deforms, these vectors rotate with the material.

Loads and constraints, on the other hand, are applied in spatial directions, by default in the canonical spatial coordinate system. This system has basis vectors in the x, y, and z directions, which are forever fixed in space. Both the material and spatial default coordinate system are referred to as the *Global coordinate system* in the user interface and are denoted by (G) in the following theory section.

Vector and tensor quantities defined in the global coordinate system on either frame use the frame's coordinate variable names as indices in the tensor component variable names. For example, `SXY` is the material frame *XY*-plane shear stress, also known as a second Piola-Kirchhoff stress, while `sxy` is the corresponding spatial frame stress, or Cauchy stress. There are also a few mixed tensors, most notably the deformation gradient `FdxY`, which has one spatial and one material index because it is used in converting quantities between the material and spatial frames.

It is possible to define any number of user coordinate systems on the material and spatial frames. Most types of coordinate systems are specified only as a rotation of the basis with respect to the canonical basis in an underlying frame. This means that they can be used both in contexts requiring a material system and in contexts requiring a spatial one.

The coordinate system can be selected separately for each added material model, load, and constraint. This is convenient if, for example, an anisotropic material with different orientation in different domains is required. The currently selected coordinate system is known as the *local coordinate system* and is denoted by (L) throughout this text.

Lagrangian Formulation

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

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

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

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

About Linear Elastic Materials

The total strain tensor is written in terms of the displacement gradient

$$\varepsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$$

or in components as

$$\varepsilon_{mn} = \frac{1}{2}\left(\frac{\partial u_m}{\partial x_n} + \frac{\partial u_n}{\partial x_m}\right) \tag{14-2}$$

The Duhamel-Hooke's law relates the stress tensor to the strain tensor and temperature:

$$s = s_0 + \mathsf{C} : (\varepsilon - \varepsilon_0 - \alpha\theta)$$

where C is the 4th order *elasticity tensor*, ":" stands for the double-dot tensor product (or double contraction), s_0 and ε_0 are initial stresses and strains, $\theta = T - T_{\mathrm{ref}}$, and α is the thermal expansion tensor.

The elastic energy is

$$W_{\mathrm{s}} = \frac{1}{2}(\varepsilon - \varepsilon_0 - \alpha\theta) : \mathsf{C} : (\varepsilon - \varepsilon_0 - \alpha\theta) \tag{14-3}$$

or using the tensor components:

$$W_{\mathrm{s}} = \sum_{i, j, m, n} \frac{1}{2}\mathsf{C}^{ijmn}(\varepsilon_{ij} - \varepsilon^0_{ij} - \alpha_{ij}\theta)(\varepsilon_{mn} - \varepsilon^0_{mn} - \alpha_{mn}\theta)$$

TENSOR VS. MATRIX FORMULATIONS

Because of the symmetry, the strain tensor can be written as the following matrix:

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

Similar representation applies to the stress and the thermal expansion tensors:

$$\begin{bmatrix} s_x & s_{xy} & s_{xz} \\ s_{xy} & s_y & s_{yz} \\ s_{xz} & s_{yz} & s_z \end{bmatrix}, \begin{bmatrix} \alpha_x & \alpha_{xy} & \alpha_{xz} \\ \alpha_{xy} & \alpha_y & \alpha_{yz} \\ \alpha_{xz} & \alpha_{yz} & \alpha_z \end{bmatrix}$$

Due to the symmetry, the elasticity tensor can be completely represented by a symmetric 6-by-6 matrix as:

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

which is the *elasticity matrix.*

ISOTROPIC MATERIAL AND ELASTIC MODULI

In this case, the elasticity matrix becomes

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

and the thermal expansion matrix is:

$$\begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix}$$

Different pairs of elastic moduli can be used, and as long as two moduli are defined, the others can be computed according to Table 14-2.

TABLE 14-2: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E,\nu)$	$D(K,G)$	$D(\lambda,\mu)$
Young's modulus	E		$\frac{9KG}{3K+G}$	$\mu\frac{3\lambda+2\mu}{\lambda+\mu}$
Poisson's ratio	ν		$\frac{1}{2}\left(1-\frac{3G}{3K+G}\right)$	$\frac{\lambda}{2(\lambda+\mu)}$
Bulk modulus	K	$\frac{E}{3(1-2\nu)}$		$\lambda+\frac{2\mu}{3}$
Shear modulus	G	$\frac{E}{2(1+\nu)}$		μ
Lamé constant λ	λ	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$K-\frac{2G}{3}$	
Lamé constant μ	μ	$\frac{E}{2(1+\nu)}$	G	
Pressure-wave speed	c_p		$\sqrt{\frac{K+4G/3}{\rho}}$	
Shear-wave speed	c_s		$\sqrt{G/\rho}$	

According to Table 14-2, the elasticity matrix D for isotropic materials is written in terms of Lamé parameters λ and μ,

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

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

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

ENTROPY AND THERMOELASTICITY

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

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

where $W_s(\varepsilon, T)$ is given by Equation 14-3. Hence, the stress can be found as

$$s = \left(\frac{\partial F}{\partial \varepsilon}\right)_T = \left(\frac{\partial W}{\partial \varepsilon}\right)_T = \mathsf{C} : (\varepsilon - \varepsilon_0 - \alpha\theta)$$

and the entropy per unit volume can be calculated as

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

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

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

For an isotropic material, it simplifies into

$$S_\mathrm{elast} = \alpha(s_x + s_y + s_z)$$

The heat balance equation can be written as

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

where k are the thermal conductivity matrix, and

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

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

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

$$\rho C_p\frac{\partial T}{\partial t}+\sum_{m,n}T\alpha_{mn}\frac{\partial}{\partial t}s_{mn} = \nabla\cdot(k\nabla T)+Q_h \tag{14-4}$$

In many cases, the second term can be neglected in the left-hand side of Equation 14-4 because all $T\alpha_{mn}$ are small. The resulting approximation is often called uncoupled thermoelasticity.

Strain-Displacement Relationship

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:

$$\begin{aligned}
\varepsilon_x&=\frac{\partial u}{\partial x} & \qquad \varepsilon_{xy}&=\frac{\gamma_{xy}}{2}=\frac{1}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\\
\varepsilon_y&=\frac{\partial v}{\partial y} & \qquad \varepsilon_{yz}&=\frac{\gamma_{yz}}{2}=\frac{1}{2}\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)\\
\varepsilon_z&=\frac{\partial w}{\partial z} & \qquad \varepsilon_{xz}&=\frac{\gamma_{xz}}{2}=\frac{1}{2}\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right).
\end{aligned} \tag{14-5}$$

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 DEFORMATIONS

As a start, consider a certain physical particle, initially located at the coordinate **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 **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*, F, defined by

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

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

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

A deformation state where $J = 1$ is often called incompressible. From the deformation gradient, it is possible to define the right Cauchy-Green tensor as

$$C = F^T F$$

As can be shown by simple insertion, a finite rigid body rotation causes nonzero values of the engineering strain defined by Equation 14-5. 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-Lagrange strains*, ε, is defined as

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

Using the displacements, they be also written as

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \cdot \frac{\partial u_k}{\partial X_j}\right) \tag{14-6}$$

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

The deformation state characterized by finite (or large displacements) but small to moderate strains is sometimes referred to as *geometric nonlinearity* or *nonlinear geometry.* This typically occurs when the main part of the deformations presents a finite rigid body rotation

STRAIN RATE AND SPIN

The spatial velocity gradient is defined in components as

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

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

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

where the material time derivative is used.

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

$$L = L_d + L_w$$

where

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

is called the *rate of strain tensor*, and

and

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

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

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

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

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

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

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 two-point tensor.
- 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

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

Plane Strain and Plane Stress Cases

For a general anisotropic linear elastic material in case of plane stress, COMSOL solves three equations $s_{i3} = 0$ for ε_{i3} with $i = 1, 2, 3$, and uses the solution instead of Equation 14-2 for these three strain components. Thus, three components ε_{i3} are treated as extra degrees of freedom. For isotropy and orthotrophy, only with an extra degree of freedom, ε_{33},is used since all out of plane shear components of both stress and strain are zero. The remaining three strain components are computed as in 3D case according to Equation 14-2.

Note

For an isotropic material, only the normal out-of-plane component ε_{33} needs to be solved for.

In case of plane strain, set $\varepsilon_{i3} = 0$ for $i = 1, 2, 3$. The out-of-plane stress components s_{i3} are results and analysis variables.

Axial Symmetry

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

The metric tensor is

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

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

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

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

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

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

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

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

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

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

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

For any tensor, the physical components are defined as

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

where no summation is done over repeated indices.

MIXED COMPONENTS AND PRINCIPAL INVARIANTS

The mixed strain components are given by

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

The principal invariants are

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

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

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

DISPLACEMENTS AND AXIAL SYMMETRY ASSUMPTIONS

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

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

$$s^{ij} = s_0^{ij} + \mathsf{C}^{ijkl}(\varepsilon_{kl} - \alpha_{kl}\theta - \varepsilon_{0kl})$$

where $\theta = T - T_{\mathrm{ref}}$.

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

$$\begin{bmatrix} s_r \\ s_\varphi \\ s_{z\varphi} \\ s_{r\varphi} \\ s_{\varphi z} \\ s_{rz} \end{bmatrix} = \begin{bmatrix} s_r \\ s_\varphi \\ s_{z\varphi} \\ s_{r\varphi} \\ s_{\varphi z} \\ s_{rz} \end{bmatrix}_0 + D\left(\begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_{z\varphi} \\ 2\varepsilon_{r\varphi} \\ 2\varepsilon_{\varphi z} \\ 2\varepsilon_{rz} \end{bmatrix} - \begin{bmatrix} \varepsilon_r \\ \varepsilon_\varphi \\ \varepsilon_{z\varphi} \\ 2\varepsilon_{r\varphi} \\ 2\varepsilon_{\varphi z} \\ 2\varepsilon_{rz} \end{bmatrix}_0 - \theta \begin{bmatrix} \alpha_r \\ \alpha_\varphi \\ \alpha_{z\varphi} \\ 2\alpha_{r\varphi} \\ 2\alpha_{\varphi z} \\ 2\alpha_{rz} \end{bmatrix}\right)$$

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

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

and the corresponding tensor components are computed internally according to:

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

For an isotropic material:

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

where λ and μ are the first and second Lamé elastic constants.

Loads

Specify loads as

- Distributed loads. The load is a distributed force in a volume, on a face, or along an edge.
- Total force. The specification of the load is as the total force. The software then divides this value with the area or the volume where the force acts.
- Pressure (boundaries only).

2D

For 2D models choose how to specify the distributed boundary load as a load defined as force per unit area or a load defined as force per unit length acting on boundaries.

In the same way, choose between defining the load as force per unit volume or force per unit area for body loads acting in a domain. Also define a total force (SI unit: N) as required.

For 2D and axisymmetric models, the boundary loads apply on edges (boundaries).

2D Axi

For 2D and axisymmetric models, the boundary loads apply on edges (boundaries).

3D

For 3D solids, the boundary loads apply on faces (boundaries).

Table 14-3 shows how to define distributed loads on different geometric entity levels; the entries show the SI unit in parentheses.

TABLE 14-3: DISTRIBUTED LOADS

GEOMETRIC ENTITY	POINT	EDGE	FACE	DOMAIN
2D	force (N)	force/area (N/m^2) or force/length (N/m)	Not available	force/volume (N/m^3) or force/area (N/m^2)
Axial symmetry	total force along the circumferential (N)	force/area (N/m^2)	Not available	force/volume (N/m^3)
3D	force (N)	force/length (N/m)	force/area (N/m^2)	force/volume (N/m^3)

Pressure Loads

A pressure load is directed inward along the normal of boundary on which it is acting. This load type acts as a source of nonlinearity, since its direction depends on the current direction of the boundary normal. In a linearized context, for example in the frequency domain, the pressure is equivalent to a specified normal stress.

Note

For general cases, if the problem is linear in all other respects, the solution can be made more efficient by forcing the solver to treat the problem as linear. See Stationary Solver in the *COMSOL Multiphysics Reference Guide*.

Equation Implementation

The COMSOL Multiphysics implementation of the equations in the Solid Mechanics interface is based on the principle of virtual work.

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:

$$W = \int_V (-\varepsilon : s + \mathbf{u} \cdot \mathbf{F}_V) dv$$

$$+ \int_S (\mathbf{u} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}^t \cdot \mathbf{F}_p)$$

The principle of virtual work states that $\delta W = 0$ which leads to

$$\int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt}) dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p)$$

Setting up Equations for Different Studies

The Solid Mechanics interface supports stationary (static), eigenfrequency, time-dependent (transient), frequency domain, and modal solver study types.

STATIONARY STUDIES

COMSOL Multiphysics uses an implementation based on the stress and strain variables. The normal and shear strain variables depend on the displacement derivatives.

Using the tensor strain, stress, and displacement variables, the principle of virtual work is expressed as:

$$\delta W = \int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V) dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p)$$

TIME-DEPENDENT STUDIES

$$\int_V (-\varepsilon_{\text{test}} : (s + \beta_{dM} s_t) + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt} - \alpha_{dM} \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_t) dv$$

$$+ \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}^t_{\text{test}} \cdot \mathbf{F}_p) \qquad (14\text{-}7)$$

where the terms proportional to α_{dM} and β_{dK} appear if the *Rayleigh damping* is used.

For more information about the equation form in case of geometric nonlinearity see

See Also

FREQUENCY-DOMAIN STUDIES

In the frequency domain the frequency response is studied when applying harmonic loads. Harmonic loads are specified using two components:

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

To derive the equations for the linear response from harmonic excitation loads

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

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

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

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

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

Also describe this relationship using complex notation

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

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

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

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 STUDIES

The eigenfrequency equations are derived by assuming a harmonic displacement field, similar as for the frequency response formulation. The difference is that this study type uses a new variable $j\omega$ explicitly expressed in the eigenvalue $j\omega = -\lambda$. The eigenfrequency f is then derived from $j\omega$ as

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

Damped eigenfrequencies can be studied by adding viscous damping terms to the equation. In addition to the eigenfrequency the quality factor, Q, and decay factor, δ, for the model can be examined:

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

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

Damping Models

The Solid Mechanics interface offers two predefined damping models: Rayleigh damping and loss factor damping.

RAYLEIGH DAMPING

To model damping effects within the material, COMSOL Multiphysics uses *Rayleigh damping*, where two damping coefficients are specified.

The weak contribution due to the alpha-damping is always accounted for as shown in Equation 14-7. The contribution from the beta-damping that shown in Equation 14-7 corresponds to the case of small strains. In case of geometric nonlinearity, it becomes

$$\int_V (-\beta_{dM} \nabla \mathbf{u}_{\text{test}} : P_t) dv$$

where P is the first Piola-Kirchhoff stress tensor.

To further clarify the use of the Rayleigh damping, consider a system with a single degree of freedom. The equation of motion for such a system with viscous damping is

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

In the Rayleigh damping model the damping coefficient c can be expressed in terms of the mass m and the stiffness k as

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

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

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

It is possible to transform relative damping to Rayleigh damping parameters. The relative damping, ξ, for a specified pair of Rayleigh parameters, α_{dM} and β_{dK}, at a frequency, f, is

$$\xi = \frac{1}{2}\left(\frac{\alpha_{dM}}{2\pi f} + \beta_{dK} 2\pi f\right)$$

Using this relationship at two frequencies, f_1 and f_2, with different relative damping, ξ_1 and ξ_2, results in an equation system that can be solved for α_{dM} and β_{dK}:

$$\begin{bmatrix} \frac{1}{4\pi f_1} & \pi f_1 \\ \frac{1}{4\pi f_2} & \pi f_2 \end{bmatrix} \begin{bmatrix} \alpha_{dM} \\ \beta_{dK} \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

Using the same relative damping, $\xi_1 = \xi_2$, does not result in a constant damping factor inside the interval $f_1 < f < f_2$. It can be shown that the damping factor is lower inside the interval, as Figure 14-7 shows.

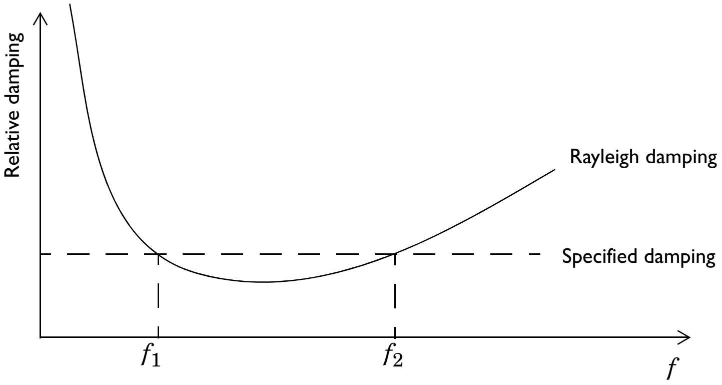


Figure 14-7: An example of Rayleigh damping.

LOSS FACTOR DAMPING

Loss factor damping (sometimes referred to as material or structural damping) can be applied in the frequency domain.

In COMSOL Multiphysics, the loss information appears as a multiplier of the elastic stress in the stress-strain relationship:

$$s = s_0 + (1 + j\eta_s)\mathsf{C} : [\varepsilon - \varepsilon_0 - \alpha\theta]$$

where η_s is the loss factor, and j is the imaginary unit.

Choose between these loss damping types:

- Isotropic loss damping η_s
- Orthotropic loss damping with components of η_E, the loss factor for an orthotropic Young's modulus, and η_G, the loss factor for an orthotropic shear modulus.
- Anisotropic loss damping with an isotropic or symmetric loss damping η_D for the elasticity matrix.

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

$$S_{\mathrm{elast}} = \alpha : (s - j\eta_s(\mathrm{C} : \varepsilon))$$

This is because the entropy is a function of state and thus independent of the strain rate, while the damping represents the rate-dependent effects in the material (for example, viscous or viscoelastic effects). The internal work of such inelastic forces averaged over the time period 2π/ω can be computed as:

$$Q_h = \frac{1}{2}\omega\eta_s\mathrm{Real}(\varepsilon : \mathrm{Conj}(\mathrm{C} : \varepsilon)) \tag{14-8}$$

Equation 14-8 can be used as a heat source for modeling of the heat generation in vibrating structures, when coupled with the frequency-domain analysis for the stresses and strains.

15

Equation-Based Modeling

This chapter describes the use of the mathematics interfaces, found under the **Mathematics** branch (Δu) in the **Model Wizard**, which are used for equation-based modeling. With those interfaces you can solve various types of PDEs using different formulations. You can also solve ODEs and other global equations.

In this chapter:

The Mathematics Interfaces

The *mathematics interfaces* are physics interfaces for equation-based modeling. They support several PDE formulation as well as general ways of adding ODEs, algebraic equations, and other global (space-independent) equations. The following mathematics interfaces are found under the Model Wizard's **Mathematics** branch:

- *PDE Interfaces*. These are interfaces for solving PDEs in different forms:
  - *Coefficient form* for linear or almost linear PDEs, explained in detail in Coefficient Form PDE Interface (c).
  - General form for nonlinear PDEs, explained in detail in General Form PDE Interface (g).
  - *Weak form* using the weak formulation of the PDE for maximum flexibility. See Weak Form Modeling.

  All of these PDE interfaces are available in domains and also on boundaries and edges and at points).
- *ODE and DAE Interfaces*. Use these interfaces to solve space-independent equations that include ordinary differential equations (ODEs), differential-algebraic equations (DAEs), algebraic equations, and transcendental equations, either as global equations or as distributed ODEs/DAEs (on domains, boundaries, edges, or at points). For more information about global equations and ODEs, see Solving ODEs and DAEs.
- *Events Interface*. This is a specialized interface to create solver events. For more information about using events, see The Events Interface.
- *Optimization*. This interface requires an Optimization Module license. See the *Optimization Module User's Guide* for details.
- *Sensitivity*. This interface is for Sensitivity Analysis.
- *Classical PDEs*. These specialized instances of the coefficient form PDE provide interfaces for a number of Classical PDEs.
- *Moving Interface*. The Level Set interface and the Phase Field interface provide the possibility to track fluid-fluid interfaces or other moving interfaces for multiphase applications, for example. See the Microfluidics Module or the Chemical Reaction Engineering Module documentation as moving interfaces require these modules.
- *Deformed Mesh*. The Deformed Geometry Interface makes it possible to model prescribed deformation of the mesh that represents the model domain. The Moving

Mesh Interface is vital for modeling moving meshes using the ALE (arbitrary Lagrangian-Eulerian) technique in applications such as fluid-structure interaction (FSI).

- *Wall Distance*. This interface, which computes the distance to the boundary (wall), is primarily intended for use in connection to turbulence modeling for fluid flow. See The Wall Distance Interface.

Table 15-1 lists the available interfaces for PDEs, ODEs, and DAEs with their default interface identifiers:

TABLE 15-1: DEFAULT INTERFACE IDENTIFIERS FOR THE MATHEMATICS INTERFACES

MATHEMATICS INTERFACE	INTERFACE IDENTIFIER
PDE INTERFACES	
Coefficient Form PDE	c
Coefficient Form Boundary PDE	cd
Coefficient Form Edge PDE	ce
Coefficient Form Point PDE	cp
General Form PDE	g
General Form Boundary PDE	gb
General Form Edge PDE	ge
General Form Point PDE	gp
Weak Form PDE	w
Weak Form Boundary PDE	wb
Weak Form Edge PDE	we
Weak Form Point PDE	wp
CLASSICAL PDES	
Convection-Diffusion Equation	cdeq
Heat Equation	hteq
Helmholtz Equation	hzeq
Laplace’s Equation	lpeq
Poisson’s Equation	poeq
Wave Equation	waeq
ODE AND DAE INTERFACES	
Global ODEs and DAEs	ge
Domain ODEs and DAEs	dode
Boundary ODEs and DAEs	bode

TABLE 15-1: DEFAULT INTERFACE IDENTIFIERS FOR THE MATHEMATICS INTERFACES

MATHEMATICS INTERFACE	INTERFACE IDENTIFIER
Edge ODEs and DAEs	eode
Point ODEs and DAEs	pode

See Also

Physics List in COMSOL Multiphysics

PDE Interfaces

In this section:

Starting a Model Using a PDE Interface

To create a new model using one of the equation interfaces:

1 In the **Model Wizard**, select a space dimension on the **Select Space Dimension** page. Click the **Next** button (⇨).

2 Expand the **Mathematics>PDE Interfaces** node in the list of physics interfaces and select one of the PDE interface in the list. For PDEs on other geometric entities than domains, expand the **Lower Dimensions** node.

3 Double-click the interface or click the **Add Selected** button (✚) underneath the list to add the selected PDE interface to the model. The interface is added to **Selected physics**.

4 Click the **Next** button (⇨) in the upper-right corner of the **Model Wizard** window.

5 Optionally, choose a **Study Type**.

6 Click the **Finish** button (🏁) in the upper-right corner of the **Model Wizard** window.

SPECIFYING A SYSTEM OF EQUATIONS

COMSOL Multiphysics allows the creation of equations with more than one dependent variable. To do so, after Step 4 above, under **Dependent variables** type the number of dependent variables in the **Number of dependent variables** field. COMSOL Multiphysics then automatically assigns variable names, typically `u1`, `u2`, `u3`, and so on. Also enter the name of your choice (as long as they are valid and unique) in **Variable names** fields. Several scalar PDEs can also be coupled using a multiphysics approach

Tip	Regardless if you start with a PDE interface in coefficient form or general form, additional equation nodes can be added in coefficient form, general form, or weak form to the same PDE interface.

Notational Conventions

COMSOL Multiphysics PDE interfaces and this chapter describing them use a slightly different notation, compared to other physics interfaces and their documentation. The difference lies in the meaning of the symbol ∇, pronounced *nabla* or *del*. In equations on display in the physics interfaces, the ∇ symbol applied to a scalar or vector variable denotes the following coordinate system-independent *gradient*, *divergence*, and *curl* operations:

$$\nabla u = \mathrm{grad(u)}$$
$$\nabla \cdot \mathbf{u} = \mathrm{div}(\mathbf{u})$$
$$\nabla \times \mathbf{u} = \mathrm{curl}(\mathbf{u})$$

In the PDE interfaces, *nabla/del* is interpreted as the vector of partial derivatives:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right)$$

The spatial coordinates are here denoted $x_1, \ldots, x_n$, where n represents the number of space dimensions. When applied to a scalar or vector in a Cartesian coordinate system, this definition leads to an expression that is identical in form to the gradient, divergence, or curl in the same coordinate system. The same does not apply, however, in curvilinear systems such as the one implied in an axisymmetric geometry.

For example, the divergence of a vector $\mathbf{u} = [u_r, u_z]$ in an axisymmetric cylindrical system is

$$\mathrm{div}(\mathbf{u}) = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} = \frac{1}{r}\frac{\partial}{\partial r}(r u_r) + \frac{\partial u_z}{\partial z}$$

while the PDE interface interpretation of *nabla/del* is

$$\nabla \cdot \mathbf{u} = \frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z}$$

In practice, this means that to correctly implement equations containing the gradient, divergence, or curl in an axisymmetric geometry, you need to compensate for the missing terms related to the curvature of the coordinate system.

The following related examples follow the same principle:

- The symbol Δ is the Laplace operator

$$\frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$$

- $\nabla \cdot (c \nabla u)$ means

$$\frac{\partial}{\partial x_1}\left(c\frac{\partial u}{\partial x_1}\right) + \ldots + \frac{\partial}{\partial x_n}\left(c\frac{\partial u}{\partial x_n}\right)$$

- $\beta \cdot \nabla u$ means

$$\beta_1\frac{\partial u}{\partial x_1} + \ldots + \beta_n\frac{\partial u}{\partial x_n}$$

where $\beta_1, \ldots, \beta_n$ are the components of the vector β.

Note

The axisymmetric versions of physics interfaces take the cylindrical system into account, and no compensation is therefore needed.

Coefficient Form PDE Interface (c)

The PDE interface (Δu) in coefficient form, **Coefficient Form PDE**, found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, covers many well-known PDEs. This section covers the formulation and settings pertaining to the coefficient form, as well as the general PDE terminology used in COMSOL Multiphysics.

When this interface is added, these default nodes are also added to the **Model Builder**—**Coefficient Form PDE**, **Zero Flux**, and **Initial Values**. Right-click the **PDE** node to add other nodes that implement other boundary conditions, for example.

The following sections provides information about all nodes available in the Coefficient Form PDE interface. The Coefficient Form PDE interface is also available as Coefficient Form Boundary PDE, Coefficient Form Edge PDE, and Coefficient Form Point PDE for PDEs in coefficient form on boundaries and edges and at points.

THE SCALAR COEFFICIENT FORM EQUATION

A single dependent variable u is an unknown function on the computational domain. COMSOL Multiphysics determines it by solving the PDE problem specified. In coefficient form, the PDE problem reads

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu & \text{on } \partial\Omega \\ u = r & \text{on } \partial\Omega \end{cases} \tag{15-1}$$

where

- Ω is the computational domain—the union of all domains
- $\partial\Omega$ is the domain boundary
- $\mathbf{n}$ is the outward unit normal vector on $\partial\Omega$

The first equation in the list above is the PDE, which must be satisfied in Ω. The second and third equations are the boundary conditions, which must hold on ∂Ω. The second equation is a generalization of a *Neumann* boundary condition, whereas the third equation is a *Dirichlet* boundary condition. In finite element terminology, the boundary condition corresponding to a Neumann boundary condition is called a *natural boundary conditions*, because it does not occur explicitly in the weak form of the PDE problem. In the PDE interfaces, the corresponding condition is called a *flux* or *source*, because it specifies the value of the numerical flux at the boundary.

Dirichlet conditions are also known as *essential boundary conditions* in finite element theory, because they impose a restriction on the trial space which is not part of the main equation. In the PDE interfaces, a distinction is made between Dirichlet boundary conditions and *constraints*. The former directly specify the value of the dependent variable at the boundary, while the latter force an arbitrary function of the dependent variables to equal zero on the boundary. This makes the constraint a more general feature.

Within COMSOL Multiphysics, the coefficients c, α, γ, β, a, and h, and the terms f, g, and r are specified. They can all be functions of the spatial coordinates.

- A PDE is *linear* when the coefficients depend only on the spatial coordinates (or are constants).
- A PDE is *nonlinear* if the coefficients depend on u or its derivatives (for example, the components of ∇u).
- All the coefficients in the above equation are scalars except α, β, and γ, which are vectors with n components. The coefficient c can alternatively be an n-by-n matrix to model anisotropic materials.

The e_a coefficient in Equation 14-1 is a scalar or a matrix for time-dependent systems called the *mass matrix* (or mass coefficient). The d_a coefficient represents a damping term (however, if $e_a = 0$, then d_a is often called the mass coefficient).

See Also

- Modeling Anisotropic Materials
- Solving Time-Dependent Problems

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `c` (in domains), `cb`, `ce`, or `cp` (on boundaries, edges, and at points, respectively).

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define the PDE. To choose specific domains, select **Manual** from the **Selection** list.

UNITS

By default, the PDE interfaces are unitless, but you can define units for the dependent variable and the source term (that is, the overall left and right side of the equation). The units for these quantities—in combination with the units for length and time—fully define the units for all other terms in the equations. To select the units you select the corresponding quantity from a list of physical quantities or enter the unit directly.

Select the quantity that defines the unit for the dependent variable u from the **Dependent variable quantity** list. The default is **Dimensionless [1]**. Select **None** to enter a unit (for example, `K`, `m/s`, or `mol/m^3`) in the **Unit** field that appears.

Select the quantity that defines the unit for the source term f (the unit for the right—and left—side of the PDE) from the **Source term quantity** list. Select **None** (the default) to enter a unit (for example, `W/m^3` or `A/m^3`) in the **Unit** field that appears. The default unit is `m^-2`, which is a consistent with a dimensionless dependent variable.

Model

For an example of the use of units in a PDE interface, see Shell Diffusion: Model Library path **COMSOL_Multiphysics/Equation-Based_Models/shell_diffusion**

DEPENDENT VARIABLES

Choose the number of dependent variables and set the field and dependent variable names. The default field and variable name for a single scalar PDE variable is u. If the field name coincides with the name of another field of the same unit and number of components, the two fields (and the interfaces which define them) will share degrees of freedom and dependent variable names. A field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**.

Select from a number of different shape function types (finite element types) from the **Shape function type** list and an associated element order (order of the shape function for the element) from the **Element order** list. The default is to use quadratic Lagrange elements.

The **Value type** (**Real** or **Complex)** of dependent variables needs to be specified when **Split complex variables in real and imaginary parts** setting is activated in the **Compile Equations** node of any solver sequence used. The default is complex value type. If split complex variables setting is not active the value type is ignored.

See Also

- Element Types and Discretization
- Show More Physics Options

Coefficient Form PDE

This is the default equation for a **Coefficient Form PDE** interface. Specify the coefficients for a coefficient form PDE (see Equation 14-1), with the following equation coefficients:

$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au\ =f$$

- e_a is the *mass coefficient*.
- d_a is a *damping coefficient* or a *mass coefficient*.
- c is the *diffusion coefficient*.
- α is the *conservative flux convection coefficient*.
- β is the *convection coefficient*.
- a is the *absorption coefficient*.

- γ is the *conservative flux source* term.
- f is the *source term*.

See Also

Interpreting Boundary Conditions

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the coefficient form PDE is active. To choose specific domains, select **Manual** from the **Selection** list.

DIFFUSION COEFFICIENT

Enter a value or expression for the **Diffusion coefficient** c, which is a tensor. Select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter a c coefficient on various forms in 2D and 3D. If there are multiple dependent variables, there is a matrix of c component inputs.

ABSORPTION COEFFICIENT

Enter a value or expression for the absorption coefficient a. If there are multiple dependent variables, there is a matrix of a component inputs.

SOURCE TERM

Enter a value or expression for the source term f. If there are multiple dependent variables, there is a vector of f component inputs.

MASS COEFFICIENT

Enter a value or expression for the mass coefficient e_a. If there are multiple dependent variables, there is a matrix of e_a component inputs.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping (or mass) coefficient d_a. If there are multiple dependent variables, there is a matrix of d_a component inputs.

CONSERVATIVE FLUX CONVECTION COEFFICIENT

Enter values or expressions for the conservative flux convection coefficient α vector's components. If there are multiple dependent variables, there is a matrix of α vector component inputs.

CONVECTION COEFFICIENT

Enter values or expressions for the convection coefficient β vector's components. If there are multiple dependent variables, there is a matrix of β vector component inputs.

CONSERVATIVE FLUX SOURCE

Enter values or expressions for the conservative flux source term γ vector's components. If there are multiple dependent variables, there is a vector of γ vector component inputs.

General Form PDE Interface (g)

The PDE interface (Δu) in general form, **General Form PDE**, found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, provides a flexible way of specifying PDEs in a general form. This section covers the formulation and settings pertaining to the general form.

When this interface is added, these default nodes are also added to the **Model Builder**—**General Form PDE**, **Zero Flux**, and **Initial Values**. Right-click **General Form PDE** to add other nodes that implement other boundary conditions, for example.

The following sections provides information about all nodes available in the General Form PDE interface. The General Form PDE interface is also available as General Form Boundary PDE, General Form Edge PDE, and General Form Point PDE for PDEs in general form on boundaries and edges and at points.

THE SCALAR GENERAL FORM EQUATION

Use the *general form* for nonlinear PDEs. Assuming that you are working with a single dependent variable u, then a transient problem in general form reads

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot\Gamma = f & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = G + \left(\dfrac{\partial R}{\partial u}\right)^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

The first equation is the PDE. The second and third equations are the Neumann and Dirichlet boundary conditions, respectively.

See Also

For information on the time-dependent general form equation, see Solving Time-Dependent Problems.

The terms Γ, f, G, and R are coefficients. They can be functions of the spatial coordinates, the solution u, and the space derivatives of u. The coefficients f, G, and R are scalar, whereas Γ is the *flux vector*. The superscript “T” in the Neumann boundary condition denotes the transpose. The variable μ is a Lagrange multiplier.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `g` (in domains), `gb`, `ge`, or `gp` (on boundaries, edges, and at points, respectively).

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the PDE is active. To choose specific domains, select **Manual** from the **Selection** list.

UNITS

By default, the PDE interfaces are unitless, but you can define units for the dependent variable and the source term (that is, the overall left and right side of the equation). The units for these quantities fully define the units for all other terms in the equations. To select the units you select the corresponding quantity from a list of physical quantities or enter the unit directly.

Select the quantity that defines the unit for the dependent variable u from the **Dependent variable quantity** list. The default is **Dimensionless [1]**. Select **None** to enter a unit (for example, `K`, `m/s`, or `mol/m^3`) in the **Unit** field that appears.

Select the quantity that defines the unit for the source term f from the **Source term quantity** list. Select **None** (the default) to enter a unit (for example, `W/m^3` or `A/m^3`) in the **Unit** field that appears. The default unit is `m^-2`, which is a consistent with a dimensionless dependent variable.

DEPENDENT VARIABLES

Choose the number of dependent variables and set the field and dependent variable names. The default field and variable name for a single scalar PDE variable is u. If the field name coincides with the name of another field of the same unit and number of components, the two fields (and the interfaces which define them) will share degrees of freedom and dependent variable names. A field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**.

Select from a number of different shape function types (finite element types) from the **Shape function type** list and an associated element order (order of the shape function for the element) from the **Element order** list. The default is to use quadratic Lagrange elements.

The **Value type** (**Real** or **Complex)** of dependent variables needs to be specified when **Split complex variables in real and imaginary parts** setting is activated in the Compile Equations node of any solver sequence used. The default is complex value type. If split complex variables setting is not active the value type is ignored.

See Also

- Element Types and Discretization
- Show More Physics Options

General Form PDE

This is the default equation for a **General Form PDE** interface. Specify the coefficients for a general form PDE, with the following equation coefficients:

$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot\Gamma = f$$

- e_a is the *mass coefficient.*
- d_a is a *damping coefficient* or a *mass coefficient.*
- Γ is the conservative *flux vector.*
- *f* is the *source term.*

The interface is also available as General Form Boundary PDE, General Form Edge PDE, and General Form Point PDE for PDEs in general form on boundaries and edges and at points.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the general form PDE is active. To choose specific domains, select **Manual** from the **Selection** list.

CONSERVATIVE FLUX

Enter a values or expressions for the components of the conservative flux vector Γ. The default values `-ux`, `-uy`, and `-uz` (in 3D) represent the negative gradient of *u* and makes the left-hand side equal to the Laplace operator. If there are multiple dependent variables, there is one Γ vector for each variable.

SOURCE TERM

Enter a value or expression for the source term *f*. If there are multiple dependent variables, there is a vector of *f* component inputs.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping (or mass) coefficient d_{a}. If there are multiple dependent variables, there is a matrix of d_{a} component inputs.

MASS COEFFICIENT

Enter a value or expression for the mass coefficient e_{a}. If there are multiple dependent variables, there is a matrix of e_{a} component inputs.

Initial Values

The **Initial Values** node adds initial values for the dependent variables that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear solver.

If you need to specify more than one set of initial values, you can add additional **Initial Values** nodes.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the dependent variables.

Adding Extra Source Terms

You can add additional source term nodes on different geometry levels: **Source** on domains, **Edge Source** on edges in 3D models and **Point Source** at points.

DOMAIN, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) to define a source.

SOURCE TERM

Enter a value or expression for the source term f.

Boundary Conditions for the PDE Interfaces

The PDE interfaces include the following boundary condition types:

- Zero Flux
- Periodic Condition
- Destination Selection
- Dirichlet Boundary Condition
- Constraint
- Flux/Source

Dirichlet Boundary Condition

The **Dirichlet Boundary Condition** specifies a value of u on the boundary of the domain: $u = r$. This is the default boundary condition.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a Dirichlet boundary condition.

DIRICHLET BOUNDARY CONDITION

Enter a value or expression for the dependent variables on the boundary. To activate the Dirichlet boundary condition for a dependent variable (for example, u_2), select the corresponding check box (**Prescribed values for u2**, for example). Then enter a value or expression for the prescribed value in the associated field.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder**.

This section contains settings for specifying the type of constraint and whether to use a pointwise or weak constraint:

- From the **Constraint type** list, select **Bidirectional, symmetric** to use a bidirectional constraint or select **Unidirectional** (the default) to use a unidirectional constraint.
- Select the **Use weak constraints** check box to use weak constraints instead of the default pointwise constraints.

See Also

For an overview of the differences between the constraint types and cases where a unidirectional constraint is required, see Bidirectional vs. Unidirectional Weak Constraints

Constraint

The **Constraint** boundary condition specifies a value of R in the constraint $R = 0$. This is a bidirectional constraint. Use an expression for the constraint that, for example, implements a *Robin boundary condition* by including a term on the form qu, where u is the dependent variable.

See Also

Bidirectional vs. Unidirectional Weak Constraints

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a constraint.

BIDIRECTIONAL CONSTRAINT, R = 0

Enter a value or expression for the value of R in the constraint $R = 0$. For example, to constrain u to 2, type `2-u` in the field for R.

CONSTRAINT SETTINGS

To display this section, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder**. Normally these settings do not need to be changed.

This section contains settings for specifying the type of constraint and whether to use a pointwise or weak constraint:

- From the **Constraint type** list, select **Bidirectional, symmetric** to use a bidirectional constraint (the default) or select **Unidirectional** to use a unidirectional constraint.
- Select the **Use weak constraints** check box to use weak constraints instead of the default pointwise constraints.

See Also

For an overview of the differences between the constraint types and cases where a unidirectional constraint is required, see Bidirectional vs. Unidirectional Weak Constraints

Flux/Source

The **Flux/Source** boundary condition adds a flux or source g on the boundary:

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma) = -g+qu$$

or:

$$\mathbf{n}\cdot\Gamma = -g+qu$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a flux or source.

BOUNDARY FLUX/SOURCE

Enter a value or expression for the value of the boundary flux or source g in the corresponding field. The default value is 0.

BOUNDARY ABSORPTION/IMPEDANCE TERM

Enter a value or expression for the value of the coefficient q in the corresponding field. The default value is 0. It adds a term qu to the boundary flux or source, which can represent absorption or impedance at the boundary.

Zero Flux

The **Zero Flux** boundary condition is the default boundary condition and prescribes a zero flux (insulation) across the boundary:

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)=0$$

or:

$$\mathbf{n}\cdot\Gamma=0$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a flux or source.

Periodic Condition

The **Periodic Condition** node adds a *periodic boundary condition*. This periodicity can be continuous (the default) so that $u(x_0) = u(x_1)$ or anti-periodic so that $u(x_0) = -u(x_1)$ and can control which of the dependent variables that the periodic condition applies to.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define a periodic boundary condition. The software automatically identifies the boundaries as either source boundaries or destination boundaries. This works fine for cases like opposing parallel boundaries. In other cases use the Destination Selection subnode to control the destination (right-click **Periodic Condition** in the Model Tree and choose **Destination Selection**). By default it contains the selection that COMSOL Multiphysics has identified. Also, under **Model>Definitions**, an **Explicit** selection node ( ), which is "read only," shows the selected destination boundaries.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** (the default) to make the dependent variables equal, or **Antiperiodicity** to make them antiperiodic: $u(x_0) = -u(x_1)$.

For each dependent variable in the PDE, choose to apply the periodic condition by selecting, for example, the **Apply condition on variable u1** check box. By default, the periodic condition applies to all dependent variables.

See Also

- Destination Selection
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Destination Selection

Use the **Destination Selection** node to change the selection for the destination. The selection that COMSOL Multiphysics makes appears as the default selection in the **Selection** list (as **Selection 1**, for example).

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the destination for the periodic boundary condition. By default it contains the selection that COMSOL Multiphysics has identified.

Note

The software usually automatically identifies the boundaries as either source boundaries or destination boundaries. By default it contains the selection that COMSOL Multiphysics has identified. An **Explicit** selection node (⧉), which is “read only,” under **Model>Definitions** shows the selected destination boundaries.

See Also

- Periodic Condition
- Using Periodic Boundary Conditions
- Periodic Boundary Condition Example

Interpreting Boundary Conditions

The formulation of the boundary conditions imposes both Dirichlet and Neumann conditions. This combination is possible because of a new dependent variable μ, which is defined only on the boundary. The unknown variable μ is called a *Lagrange multiplier*. Often boundary conditions can be reformulated without Lagrange multipliers. In structural mechanics problems the Lagrange multiplier equals the reaction forces on the boundary. The factor h^T in the Neumann boundary condition is the transpose of h. If h is a scalar, then $h^T = h$.

How do the Lagrange multipliers relate to the conventional formulation with either a Dirichlet or a Neumann boundary condition?

- First, assume that $h = 1$. Then the Dirichlet condition is $u = r$. The Neumann condition becomes:

  $$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g - \mu$$

  The Lagrange multiplier, μ, adjusts so as to satisfy the requested Dirichlet condition. Specifying a nonzero g changes the value of the Lagrange multiplier but does not affect the actual solution u. Therefore, this equation can usually be ignored, leaving a pure Dirichlet condition.
- Second, assume that $h = 0$ and $r = 0$. Then the Dirichlet condition reads $0 = 0$, and the Neumann condition is

  $$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g$$

  This is the generalized Neumann condition without a Lagrange multiplier. For more on the Lagrange multiplier formulation, see System for Two Variables in the General Form.

The vector $\Gamma = -c\nabla u - \alpha u + \gamma$ is the *flux vector*. In transport equations its first term describes diffusion, the second term describes convection with a velocity $-\alpha$, and the third term γ is a source term.

In certain applications Γ can be discontinuous across interior boundaries. There can be a jump in the normal component of Γ across such a boundary. For instance, denote the two adjacent domains 1 and 2, and let Γ_i and $\mathbf{n}_i$ be the values of Γ and $\mathbf{n}$ from the two domains. Then state the jump condition as the Neumann condition

$$-\mathbf{n}_1 \cdot \Gamma_1 - \mathbf{n}_2 \cdot \Gamma_2 + qu = g - h^T\mu$$

where $\mathbf{n}_1$ is the outward normal from domain 1, and $\mathbf{n}_2$ is the outward normal from domain 2 so that $\mathbf{n}_1 = -\mathbf{n}_2$. At the same time there is a Dirichlet condition $u = r$.

The rest of this section handles stationary problems where $\partial u/\partial t = 0$ and transient effects have vanished.

Note

The c, α, and γ coefficients in the equations for the boundary conditions come from the PDE specification.

Multiple Dependent Variables—Equation Systems

THE COEFFICIENT FORM EQUATION SYSTEM

With two dependent variables u_1 and u_2, the stationary PDE problem in coefficient form results in the following equation system:

$$e_a\frac{\partial^2 u}{\partial t^2} + d_a\frac{\partial u}{\partial t} - \nabla\cdot(c\nabla u + \alpha u - \gamma) + \beta\cdot\nabla u + au = f$$

where $u = (u_1, u_2)$. The mass term is defined as

$$e_a\frac{\partial^2 u}{\partial t^2} = \begin{bmatrix} e_{a11} & e_{a12} \\ e_{a21} & e_{a22} \end{bmatrix}\begin{bmatrix} \frac{\partial^2 u_1}{\partial t^2} \\ \frac{\partial^2 u_2}{\partial t^2} \end{bmatrix} = \begin{bmatrix} e_{a11}\frac{\partial^2 u_1}{\partial t^2} + e_{a12}\frac{\partial^2 u_2}{\partial t^2} \\ e_{a21}\frac{\partial^2 u_1}{\partial t^2} + e_{a22}\frac{\partial^2 u_2}{\partial t^2} \end{bmatrix}$$

Similarly, the damping term is

$$d_a\frac{\partial u}{\partial t} = \begin{bmatrix} d_{a11} & d_{a12} \\ d_{a21} & d_{a22} \end{bmatrix}\begin{bmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial u_2}{\partial t} \end{bmatrix} = \begin{bmatrix} d_{a11}\frac{\partial u_1}{\partial t} + d_{a12}\frac{\partial u_2}{\partial t} \\ d_{a21}\frac{\partial u_1}{\partial t} + d_{a22}\frac{\partial u_2}{\partial t} \end{bmatrix}$$

However, if $e_a = 0$, then d_a is often called the mass coefficient.

The diffusive flux is defined as

$$c\nabla u = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \nabla \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \nabla u_1 \\ \nabla u_2 \end{bmatrix} = \begin{bmatrix} c_{11}\nabla u_1 + c_{12}\nabla u_2 \\ c_{21}\nabla u_1 + c_{22}\nabla u_2 \end{bmatrix}$$

$$= \begin{bmatrix} c_{11}\begin{bmatrix} \dfrac{\partial u_1}{\partial x} \\ \dfrac{\partial u_1}{\partial y} \end{bmatrix} + c_{12}\begin{bmatrix} \dfrac{\partial u_2}{\partial x} \\ \dfrac{\partial u_2}{\partial y} \end{bmatrix} \\ c_{21}\begin{bmatrix} \dfrac{\partial u_1}{\partial x} \\ \dfrac{\partial u_1}{\partial y} \end{bmatrix} + c_{22}\begin{bmatrix} \dfrac{\partial u_2}{\partial x} \\ \dfrac{\partial u_2}{\partial y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_{11}\dfrac{\partial u_1}{\partial x} + c_{12}\dfrac{\partial u_2}{\partial x} \\ c_{11}\dfrac{\partial u_1}{\partial y} + c_{12}\dfrac{\partial u_2}{\partial y} \end{bmatrix} \\ \begin{bmatrix} c_{21}\dfrac{\partial u_1}{\partial x} + c_{22}\dfrac{\partial u_2}{\partial x} \\ c_{21}\dfrac{\partial u_1}{\partial y} + c_{22}\dfrac{\partial u_2}{\partial y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathsf{cu1x} \\ \mathsf{cu1y} \end{bmatrix} \\ \begin{bmatrix} \mathsf{cu2x} \\ \mathsf{cu2y} \end{bmatrix} \end{bmatrix}$$

where ∇u_1 and ∇u_2 are column vectors. The flux matrix or flux tensor is a column vector in this presentation. For anisotropic materials, the components c_{11}, c_{12}, c_{21}, and c_{22} can be matrices as described above for the one-variable coefficient form PDE. In this case, the diffusive flux reads

$$c\nabla u = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \nabla u_1 \\ \nabla u_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c_{1111} & c_{1112} \\ c_{1121} & c_{1122} \end{bmatrix} & \begin{bmatrix} c_{1211} & c_{1212} \\ c_{1221} & c_{1222} \end{bmatrix} \\ \begin{bmatrix} c_{2111} & c_{2112} \\ c_{2121} & c_{2122} \end{bmatrix} & \begin{bmatrix} c_{2211} & c_{2212} \\ c_{2221} & c_{2222} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \nabla u_1 \\ \nabla u_2 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} c_{1111} & c_{1112} \\ c_{1121} & c_{1122} \end{bmatrix} \nabla u_1 + \begin{bmatrix} c_{1211} & c_{1212} \\ c_{1221} & c_{1222} \end{bmatrix} \nabla u_2 \\ \begin{bmatrix} c_{2111} & c_{2112} \\ c_{2121} & c_{2122} \end{bmatrix} \nabla u_1 + \begin{bmatrix} c_{2211} & c_{2212} \\ c_{2221} & c_{2222} \end{bmatrix} \nabla u_2 \end{bmatrix}$$

The conservative convective flux is defined as

$$\alpha u = \alpha\begin{bmatrix}u_1\\u_2\end{bmatrix} = \begin{bmatrix}\alpha_{11} & \alpha_{12}\\\alpha_{21} & \alpha_{22}\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}\alpha_{111}\\\alpha_{112}\end{bmatrix} & \begin{bmatrix}\alpha_{121}\\\alpha_{122}\end{bmatrix}\\\begin{bmatrix}\alpha_{211}\\\alpha_{212}\end{bmatrix} & \begin{bmatrix}\alpha_{221}\\\alpha_{222}\end{bmatrix}\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}\alpha_{111}\\\alpha_{112}\end{bmatrix}u_1 + \begin{bmatrix}\alpha_{121}\\\alpha_{122}\end{bmatrix}u_2\\\begin{bmatrix}\alpha_{211}\\\alpha_{212}\end{bmatrix}u_1 + \begin{bmatrix}\alpha_{221}\\\alpha_{222}\end{bmatrix}u_2\end{bmatrix}$$

Here the third index, k, of α_{ijk} corresponds to the spatial coordinate suffixes x and y.

The conservative flux source is defined as

$$\gamma = \begin{bmatrix}\begin{bmatrix}\gamma_{11}\\\gamma_{12}\end{bmatrix}\\\begin{bmatrix}\gamma_{21}\\\gamma_{22}\end{bmatrix}\end{bmatrix}$$

Here the second index, j, of γ_{ij} denotes the spatial coordinate suffixes for x and y.

For the flux terms the divergence operator works on each row separately. To illustrate this, consider the divergence of the conservative flux source

$$\nabla\cdot\gamma = \nabla\cdot\begin{bmatrix}\begin{bmatrix}\gamma_{11}\\\gamma_{12}\end{bmatrix}\\\begin{bmatrix}\gamma_{21}\\\gamma_{22}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\nabla\cdot\begin{bmatrix}\gamma_{11}\\\gamma_{12}\end{bmatrix}\\\nabla\cdot\begin{bmatrix}\gamma_{21}\\\gamma_{22}\end{bmatrix}\end{bmatrix}$$

The convection term is defined as

$$\beta\cdot\nabla u = \begin{bmatrix}\beta_{11} & \beta_{12}\\\beta_{21} & \beta_{22}\end{bmatrix}\cdot\begin{bmatrix}\nabla u_1\\\nabla u_2\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}\beta_{111}\\\beta_{112}\end{bmatrix} & \begin{bmatrix}\beta_{121}\\\beta_{122}\end{bmatrix}\\\begin{bmatrix}\beta_{211}\\\beta_{212}\end{bmatrix} & \begin{bmatrix}\beta_{221}\\\beta_{222}\end{bmatrix}\end{bmatrix}\cdot\begin{bmatrix}\nabla u_1\\\nabla u_2\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}\beta_{111}\\\beta_{112}\end{bmatrix}\cdot\nabla u_1 + \begin{bmatrix}\beta_{121}\\\beta_{122}\end{bmatrix}\cdot\nabla u_2\\\begin{bmatrix}\beta_{211}\\\beta_{212}\end{bmatrix}\cdot\nabla u_1 + \begin{bmatrix}\beta_{221}\\\beta_{222}\end{bmatrix}\cdot\nabla u_2\end{bmatrix}$$

The variable names for these components are beu1 and beu2.

The absorption term is defined as

$$au = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_{11}u_1 + a_{12}u_2 \\ a_{21}u_1 + a_{22}u_2 \end{bmatrix}$$

The variable names for these components are `au1` and `au2`.

The source term is defined as

$$f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

The variable names for these components are `f1` and `f2`.

The Boundary Condition Terms

The Dirichlet boundary condition, in expanded form, reads

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

If the Dirichlet condition is chosen, the generalized Neumann boundary condition also results, which reads

$$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g - h^T \mu$$

The normal vector $\mathbf{n} = (n_x, n_y)$ operates on the flux vector in the same way as the divergence operator as explained earlier. If h has full rank (as in the default identity matrix, for example) only the constraints from the Dirichlet condition are active.

If the Neumann condition is chosen, you get only the boundary condition

$$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + qu = g$$

The normal component of the diffusive flux is defined as

$$\mathbf{n} \cdot c\nabla u = \mathbf{n} \cdot \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \nabla u_1 \\ \nabla u_2 \end{bmatrix} = \begin{bmatrix} \mathbf{n} \cdot (c_{11}\nabla u_1 + c_{12}\nabla u_2) \\ \mathbf{n} \cdot (c_{21}\nabla u_1 + c_{22}\nabla u_2) \end{bmatrix}$$

The normal component of the conservative convective flux is defined as

$$\mathbf{n}\cdot\alpha u = \mathbf{n}\cdot\begin{bmatrix}\begin{bmatrix}\alpha_{111}\\ \alpha_{112}\end{bmatrix}u_1+\begin{bmatrix}\alpha_{121}\\ \alpha_{122}\end{bmatrix}u_2\\ \begin{bmatrix}\alpha_{211}\\ \alpha_{212}\end{bmatrix}u_1+\begin{bmatrix}\alpha_{221}\\ \alpha_{222}\end{bmatrix}u_2\end{bmatrix} = \begin{bmatrix}(n_x,n_y)\cdot\left(\begin{bmatrix}\alpha_{111}\\ \alpha_{112}\end{bmatrix}u_1+\begin{bmatrix}\alpha_{121}\\ \alpha_{122}\end{bmatrix}u_2\right)\\ (n_x,n_y)\cdot\left(\begin{bmatrix}\alpha_{211}\\ \alpha_{212}\end{bmatrix}u_1+\begin{bmatrix}\alpha_{221}\\ \alpha_{222}\end{bmatrix}u_2\right)\end{bmatrix}$$

The normal component of the conservative flux source is defined as

$$\mathbf{n}\cdot\gamma = (n_x,n_y)\cdot\begin{bmatrix}\begin{bmatrix}\gamma_{11}\\ \gamma_{12}\end{bmatrix}\\ \begin{bmatrix}\gamma_{21}\\ \gamma_{22}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}(n_x,n_y)\cdot\begin{bmatrix}\gamma_{11}\\ \gamma_{12}\end{bmatrix}\\ (n_x,n_y)\cdot\begin{bmatrix}\gamma_{21}\\ \gamma_{22}\end{bmatrix}\end{bmatrix}$$

The boundary absorption term is defined as

$$qu = \begin{bmatrix}q_{11} & q_{12}\\ q_{21} & q_{22}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} = \begin{bmatrix}q_{11}u_1+q_{12}u_2\\ q_{21}u_1+q_{22}u_2\end{bmatrix}$$

The boundary source term is defined as

$$g = \begin{bmatrix}g_1\\ g_2\end{bmatrix}$$

THE GENERAL FORM EQUATION SYSTEM

In the case of several dependent variables u_1, u_2, …, u_N, the following system of equations represents a stationary problem in the general form:

$$\begin{cases}\nabla\cdot\Gamma_l = F_l & \text{in }\Omega\\ -\mathbf{n}\cdot\Gamma_l = G_l+\dfrac{\partial R_m}{\partial u_l}\mu_m & \text{on }\partial\Omega\\ 0 = R_m & \text{on }\partial\Omega\end{cases}$$

The equation index l ranges from 1 to N, while the constraint index m ranges from 1 to M. This discussion uses the summation convention. F_l, G_l, and R_m are scalars, whereas Γ_l is a vector. In this case there are several Lagrange multipliers: $\mu_1, \mu_2, \ldots, \mu_M$.

For a more compact form, let u be a vector with components u_k, let Γ be a vector with components Γ_l, and so on. Then the system of equations takes on the same form as given above for a single dependent variable.

It is possible to rewrite the system to introduce the components Γ_{lj} of the vector Γ_l and the components n_j of the normal vector $\mathbf{n}$. Then the system of equations becomes

$$\begin{cases} \dfrac{\partial\Gamma_{lj}}{\partial x_j} = F_l & \text{in } \Omega \\ -n_j\Gamma_{lj} = G_l + \dfrac{\partial R_m}{\partial u_l}\mu_m & \text{on } \partial\Omega \\ 0 = R_m & \text{on } \partial\Omega \end{cases}$$

System for Two Variables in the General Form

The following example of a PDE in the general form is a stationary system for $N = 2$ solution components in $n = 2$ space dimensions with $M = 2$ constraints:

$$\begin{cases} \nabla\cdot\Gamma_1 = F_1 & \text{in } \Omega \\ \nabla\cdot\Gamma_2 = F_2 & \text{in } \Omega \end{cases}$$

with the Neumann boundary conditions

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 + \dfrac{\partial R_1}{\partial u_1}\mu_1 + \dfrac{\partial R_2}{\partial u_1}\mu_2 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 + \dfrac{\partial R_1}{\partial u_2}\mu_1 + \dfrac{\partial R_2}{\partial u_2}\mu_2 & \text{on } \partial\Omega \end{cases}$$

and the Dirichlet boundary conditions

$$\begin{cases} 0 = R_1 & \text{on } \partial\Omega \\ 0 = R_2 & \text{on } \partial\Omega \end{cases}$$

Specifying and Interpreting Boundary Conditions

The same set of boundary conditions are accessible in all PDE interfaces. To illustrate the flexibility of the Constraint boundary condition $R = 0$, consider five cases:

1 Let $R_1 = R_2 = 0$. Then the Dirichlet boundary conditions give $0 = 0$. In addition, the terms containing the Lagrange multipliers disappear from the Neumann boundary condition. Thus you have only the Neumann boundary conditions

$$\begin{cases} -\mathbf{n} \cdot \Gamma_1 = G_1 & \text{on } \partial\Omega \\ -\mathbf{n} \cdot \Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

2 Let $R_1 = r_1 - u_1$ and $R_2 = r_2 - u_2$. Then the Dirichlet conditions are the usual $u_1 = r_1$ and $u_2 = r_2$. The Neumann boundary conditions become

$$\begin{cases} -\mathbf{n} \cdot \Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n} \cdot \Gamma_2 = G_2 - \mu_2 & \text{on } \partial\Omega \end{cases}$$

These equations impose no restrictions on u_1 or u_2, because the Lagrange multipliers μ_1 and μ_2 always adjust so as to fulfill the Dirichlet conditions. In this case, ignore the Neumann boundary conditions.

3 Let $R_1 = r_1 - u_1$ and $R_2 = 0$. Then the Dirichlet conditions are

$$\begin{cases} 0 = r_1 - u_1 & \text{on } \partial\Omega \\ 0 = 0 & \text{on } \partial\Omega \end{cases}$$

and the Neumann conditions are

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

The first Neumann condition can be ignored because it imposes no restriction on u_1 or u_2. You effectively have only the Dirichlet condition on u_1 together with the second Neumann condition.

4 The same as Case 3 above but with the two PDEs interchanged (Γ_1 and Γ_2 as well as F_1 and F_2). Then the PDEs are

$$\begin{cases} \nabla\cdot\Gamma_2 = F_2 & \text{in } \Omega \\ \nabla\cdot\Gamma_1 = F_1 & \text{in } \Omega \end{cases}$$

The Dirichlet condition is similar to that in Case 3: $u_1 = r_1$. The Neumann conditions then become

$$\begin{cases} -\mathbf{n}\cdot\Gamma_2 = G_2 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_1 = G_1 & \text{on } \partial\Omega \end{cases}$$

Effectively, you have only the Neumann condition $-\mathbf{n}\cdot\Gamma_1 = G_1$. In comparison with Case 3 above, the PDEs and the Dirichlet conditions are identical, while the Neumann conditions are different. Thus, when mixing Dirichlet and Neumann conditions, the ordering of the equations and the dependent variables are important. However, the ordering of the Dirichlet conditions does not matter because the different Lagrange multipliers are for all practical purposes indistinguishable from each other.

5 Finally, let $R_1 = u_2 - u_1$ and $R_2 = 0$. Then the Dirichlet conditions are

$$\begin{cases} 0 = u_2 - u_1 & \text{on } \partial\Omega \\ 0 = 0 & \text{on } \partial\Omega \end{cases}$$

and the Neumann conditions are

$$\begin{cases} -\mathbf{n}\cdot\Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 + \mu_1 & \text{on } \partial\Omega \end{cases}$$

The same Lagrange multiplier now appears in both Neumann conditions, which can have different definitions of Γ and G. Therefore, contrary to Cases 2 and 3 above, the Neumann conditions cannot be ignored. Instead, adding the two conditions, it becomes apparent that the solution and flux on the boundary must fulfill

$$\begin{cases} 0 = u_2 - u_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 - \mathbf{n}\cdot\Gamma_2 = G_1 + G_2 & \text{on } \partial\Omega \end{cases}$$

In these examples, the values of the Lagrange multipliers do not matter. However, they often have a physical significance. In structural mechanics, the term

$$\left(\frac{\partial R}{\partial u}\right)^T \mu$$

in the Neumann condition is the reaction force necessary to satisfy the kinematic constraints described by the Dirichlet conditions. This term has a special form because of the variational principles that give rise to it.

COEFFICIENT FORM VS. GENERAL FORM

The following substitutions show that the coefficient form and the general form are equivalent:

$$\Gamma = -c\nabla u - \alpha u + \gamma, \quad F = f - \beta\nabla u - au, \quad G = g, \quad R = r - u$$

This duality lets you choose the representation that best suits a particular PDE.

Solving Time-Dependent Problems

To get the equation for a time-dependent PDE in COMSOL Multiphysics, add terms containing time derivatives to the left-hand side of the stationary equation. The time derivatives must appear linearly, and the Dirichlet conditions must be linear. A time-dependent problem in the coefficient form reads

$$
\begin{cases}
e_a\dfrac{\partial^2 u}{\partial t^2}+d_a\dfrac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au=f & \text{in } \Omega\\
\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)+qu=g-h^T\mu & \text{on } \partial\Omega\\
u=r & \text{on } \partial\Omega
\end{cases}
$$

If u is a vector of dependent variables then the *mass coefficient* e_a is a matrix. All coefficients can depend on time. The name *mass matrix* or mass coefficient stems from the fact that in many physics applications e_a contains the mass density. The d_a coefficient represents damping for wave-type equations. However, if $e_a = 0$, then d_a is often called the mass coefficient. The default settings are $e_a = 0$ and $d_a = 1$, representing a time-dependent PDE such as the heat equation. Using $e_a = 1$ and $d_a = 0$ represents an undamped wave equation.

For a stationary model, COMSOL Multiphysics ignores any values or expressions that are entered in the field for the d_a and e_a coefficient. To activate the d_a and e_a coefficients and convert the model into a time-dependent model, select a time-dependent study.

The time-dependent problem in the general form is

$$
\begin{cases}
e_a\dfrac{\partial^2 u}{\partial t^2}+d_a\dfrac{\partial u}{\partial t}+\nabla\cdot\Gamma=F & \text{in } \Omega\\
-\mathbf{n}\cdot\Gamma=G+\left(\dfrac{\partial R}{\partial u}\right)^T\mu & \text{on } \partial\Omega\\
0=R & \text{on } \partial\Omega
\end{cases}
$$

The flux vector Γ and the scalar coefficients F, G, and R can be functions of the spatial coordinates, the solution u, and the space and time derivatives of u. The superscript T in the Neumann boundary condition denotes the transpose. The variable μ is the Lagrange multiplier.

For the weak form, the time-dependent problem (in 2D) looks like

$$\int_{\Omega} W^{(2t)}dA + \int_{B} W^{(1t)}ds + \sum_{P} W^{(0t)} = \int_{\Omega} W^{(2)}dA + \int_{B} W^{(1)}ds + \sum_{P} W^{(0)} +$$

$$+ \int_{\Omega} v_l \frac{\partial R_m^{(2)}}{\partial u_l}\, \mu_m^{(2)} dA + \int_{B} v_l \frac{\partial R_m^{(1)}}{\partial u_l}\, \mu_m^{(1)} ds + \sum_{P} v_l \frac{\partial R_m^{(0)}}{\partial u_l}\, \mu_m^{(0)}$$

$$0 = R^{(2)} \quad \text{on } \Omega$$

$$0 = R^{(1)} \quad \text{on B}$$

$$0 = R^{(0)} \quad \text{on P}$$

where the integrands $W^{(it)}$ depend bilinearly on the test functions v_l and on the first and second time derivatives of the dependent variables and their space derivatives. For example,

$$W^{(2t)} = v_1 \frac{\partial u_2}{\partial t} - \frac{\partial v_1}{\partial x} \cos(3u_1) \frac{\partial u_1}{\partial t} + v_2 \frac{\partial^2 u_2}{\partial t \partial y}$$

Time-Dependent Systems

For time-dependent systems of PDEs the d_a and e_a coefficients are matrices. For example, for the system

$$e_{11} \frac{\partial^2 u}{\partial t^2} + e_{12} \frac{\partial^2 u_2}{\partial t^2} + d_{11} \frac{\partial u_1}{\partial t} + d_{12} \frac{\partial u_2}{\partial t} + \nabla \cdot (-c_{11} \nabla u_1 - c_{12} \nabla u_2) + a_{11} u_1 + a_{12} u_2 = f_1$$

$$e_{21} \frac{\partial^2 u}{\partial t^2} + e_{22} \frac{\partial^2 u_2}{\partial t^2} + d_{21} \frac{\partial u_1}{\partial t} + d_{22} \frac{\partial u_2}{\partial t} + \nabla \cdot (-c_{21} \nabla u_1 - c_{22} \nabla u_2) + a_{21} u_1 + a_{22} u_2 = f_2$$

the coefficient matrices are

$$e_a = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \qquad d_a = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

$$c = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \qquad a = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

(where each element c_{ij} can be an n-by-n matrix). Many interesting problems have a singular d_a matrix (with $e_a = 0$), or a singular nonzero e_a matrix. Such problems are called *differential-algebraic equation* (DAE) systems. The COMSOL Multiphysics solvers for time-dependent problems handle DAEs.

Solving Eigenvalue Problems

COMSOL Multiphysics handles scalar eigenvalue problems for all PDE forms. These eigenvalue problems are related to time-dependent problems via the correspondence $\partial/\partial t \leftrightarrow -\lambda$, linking the time derivative to the eigenvalue λ (with the eigenvalue name `lambda`). An eigenvalue problem in the coefficient form reads

$$\begin{cases} (\lambda-\lambda_0)^2 e_a u - (\lambda-\lambda_0) d_a u + \nabla\cdot(-c\nabla u - \alpha u) + \beta\cdot\nabla u + au = f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u + \alpha u) + qu = -h^T\mu & \text{on } \partial\Omega \\ u = r & \text{on } \partial\Omega \end{cases}$$

where λ_0 is the linearization point for the eigenvalue. The source terms are included if they depend on the solution components. If the coefficients depend on u or the eigenvalue λ, COMSOL Multiphysics performs a linearization of the problem about the linearization point $u = u_0$, $\lambda = \lambda_0$.

To specify the linearization point u_0 and the linearization point λ_0, use the settings in the **Linearization Point** section in the **Eigenvalue** node's settings window.

See Also

Solver Studies and Study Types

Interpreting PDE Coefficients

The COMSOL PDE formulations can model a variety of problems, but this documentation uses coefficient names that fall within the realm of continuum mechanics and mass transfer. For the coefficient form PDE:

- e_a is the *mass coefficient.*
- d_a is a *damping coefficient* or a *mass coefficient.*
- c is the *diffusion coefficient.*
- α is the *conservative flux convection coefficient.*

- β is the *convection coefficient.*
- a is the *absorption coefficient.*
- γ is the *conservative flux source term.*
- f is the *source term.*

$$\overset{\text{Mass}}{e_a\frac{\partial^2 u}{\partial t^2}} + \overset{\text{Damping/ Mass}}{d_a\frac{\partial u}{\partial t}} - \nabla\cdot\underbrace{(\overset{\text{Diffusion}}{c\nabla u} + \overset{\text{Convection}}{\alpha u} - \overset{\text{Source}}{\gamma})}_{\text{Conservative Flux}} + \overset{\text{Convection}}{\beta\cdot\nabla u} + \underset{\text{Absorption}}{a u} = \overset{\text{Source}}{f}$$

There are many interesting PDE problems to which this interpretation does not apply. For instance, a time-harmonic PDE such as the Helmholtz equation represents a time-dependent phenomenon transformed into the frequency domain.

For the Neumann boundary condition of the coefficient form

$$\mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu$$

- q is the b*oundary absorption coefficient.*
- g is the *boundary source term.*

Classical PDEs

Many classical PDEs are instances of the coefficient form PDE. The classical PDEs in this section have their own interfaces and are found under the **Mathematics>Classical PDEs** branch (∇²) in the **Model Wizard**. Table 14-2 shows the available classical PDEs using two notations: the compact notation of vector analysis (used in this documentation) and an expanded mathematical notation.

TABLE 15-2: CLASSICAL PDES IN COMPACT AND STANDARD NOTATION

EQUATION	COMPACT NOTATION	STANDARD NOTATION (2D)
Laplace's equation	$-\nabla\cdot(\nabla u) = 0$	$-\frac{\partial}{\partial x}\frac{\partial u}{\partial x}-\frac{\partial}{\partial y}\frac{\partial u}{\partial y} = 0$
Poisson's equation	$-\nabla\cdot(c\nabla u) = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Helmholtz equation	$-\nabla\cdot(c\nabla u)+au = f$	$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)+au = f$
Heat equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u) = f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Wave equation	$e_a\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u) = f$	$e_a\frac{\partial^2 u}{\partial t^2}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right) = f$
Convection-diffusion equation	$d_a\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+\beta\cdot\nabla u = f$	$d_a\frac{\partial u}{\partial t}-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)$ $+\beta_x\frac{\partial u}{\partial x}+\beta_y\frac{\partial u}{\partial y} = f$

The default values are 1 for f and c and −1 for a, so the default Helmholtz equation, for example, is $-\Delta u - u = 1$.

Note

The Classical PDEs have the same boundary conditions and other settings as the Coefficient Form PDE Interface (c).

Bidirectional and Unidirectional Constraints

Constraints formulated through the coefficients h and r in the Coefficient Form PDE interface and through the coefficient R in the General Form PDE interface give rise to constraints called *bidirectional*. A bidirectional constraint dictates exactly how the flux conditions (or Neumann boundary conditions) are influenced by the constraint force. For the coefficient form, the flux condition is

$$\mathbf{n}\cdot(c\nabla u+\alpha u)+qu = g{-}h^T\mu$$

and for the general form, the flux condition is

$$-\mathbf{n}\cdot\Gamma = G+\left(\frac{\partial R}{\partial u}\right)^T\mu$$

The last term on the right-hand side in both expressions is the bidirectional constraint force. Thus, with bidirectional constraints a flux condition cannot be enforced independently of the constraints. In mathematics as well as in multiphysics modeling it is often necessary to enforce Neumann conditions and Dirichlet conditions more freely than what is possible through bidirectional constraints. As an example, consider again the general form and assume that you want to enforce the boundary conditions

$$\begin{cases} 0 = r_1 - u_1 & \text{on } \partial\Omega \\ -\mathbf{n}\cdot\Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}.$$

If $r_1 = r_1(u_2)$, the first condition is fulfilled but not the second. This is because the bidirectional constraint force is not zero:

$$-\mathbf{n}\cdot\Gamma_2 = G_2+\frac{\partial R_1}{\partial u_2}\mu_1+\frac{\partial R_2}{\partial u_2}\mu_2 = G_2+\frac{\partial r_1}{\partial u_2}\mu_1\neq G_2\,.$$

To remedy this limitation with bidirectional constraints use *unidirectional constraints*, by which the constraint force is specified independently of the constraints. The term unidirectional refers to the fact that the constraint force differs from $-h^T\mu$ for coefficient form and from $\left(\frac{\partial R}{\partial u}\right)^T\mu$ for general form. In multiphysics modeling, unidirectional constraints are, for example, necessary for the following boundary conditions:

- Normal-direction constraints on a moving mesh, where the mesh motion is part of the problem. These conditions are of the type $\mathbf{n}\cdot\mathbf{u}-r = 0$ where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the boundary normal, $\mathbf{u}$ is a vector field (displacements or velocity), and $\mathbf{x}$ is the mesh coordinate vector. Bidirectional constraints give constraint forces not only on the equations for $\mathbf{u}$ but also on the equations for $\mathbf{x}$, which typically are not wanted.
- Constraints on time derivatives, such as $\frac{\partial u}{\partial t} = 1$ on the boundary (typing `1-ut` using COMSOL syntax for R in the constraint $R = 0$. The bidirectional constraint attempts to apply the test function on the time derivative of u, which is not

supported. The unidirectional constraint must also be a weak constraint because pointwise constraints for time derivatives are not supported.

- Wall boundary conditions for turbulent fluid flow. For the k-ε turbulence model this condition is of the type $k - r(\varepsilon)$, $-\mathbf{n} \cdot \nabla\varepsilon$, where r is a given function. Bidirectional constraints for the first relation imply that the second relation cannot hold.

Unidirectional constraints can be enforced in COMSOL Multiphysics both in a pointwise sense and in a weak sense.

See Also

For descriptions of how to use pointwise and weak unidirectional constraints, see Constraint Forces for PDEs and The Weak Constraint Node, respectively.

PDE Interface Variables

The following list shows the variables that are available for results evaluation and visualization in addition to the other variables for PDEs:

NAME	TYPE	DESCRIPTION	EXPRESSION
`ui`	S, B	The solution variable (dependent variable)	u_i
`uixj`	S, B	The derivative of the solution variable u_i with respect to the spatial coordinate x_j, for example, `uy`	$\frac{\partial u_i}{\partial x_i}$
`uixjxk`	S, B	The second derivative of the solution variable u_i with respect to the spatial coordinates x_j and x_k, for example, `uxx`, `uxy`	$\frac{\partial^2 u_i}{\partial x_j \partial x_k}$
`uit`	S, B	The derivative of the solution variable u_i with respect to time	$\frac{\partial u_i}{\partial t}$
`uitt`	S, B	The second derivative of the solution variable u_i with respect to time	$\frac{\partial^2 u_i}{\partial t^2}$
`uixjt`	S, B	The mixed derivative of the solution variable u_i with respect to time and the spatial coordinate x_j	$\frac{\partial^2 u_i}{\partial x_j \partial t}$

PDE Coefficients and Boundary Conditions with Time Derivatives

The d coefficients relate to 1st-order time derivatives of the solution u. The only directly available d coefficient is d_a, the coefficient in front of $\partial u/\partial t$; the subscript a is used because d_a is similar to the a coefficient in the absorption term except that it multiplies $\partial u/\partial t$ instead of u. In contrast, the coefficients d_c, $\mathbf{d}_\alpha$, and $\mathbf{d}_\beta$ for $\partial u/\partial t$—analogous to the coefficients c, α, and β for u, respectively—in the extended PDE formulation

$$-\nabla\cdot\left(d_c\nabla\frac{\partial u}{\partial t}+\mathbf{d}_\alpha\frac{\partial u}{\partial t}\right)+\mathbf{d}_\beta\cdot\nabla\frac{\partial u}{\partial t}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u+\ldots)$$

are not directly available in the general or coefficient form PDE models. Instead, enter them in the existing γ and f terms:

1D

In 1D, add `-d_c*uxt-d_al*ut` to the γ term, and add `-d_be*uxt` to the f term.

2D

In 2D, add `-d_c*uxt-d_al1*ut` to the first γ component, and add `-d_c*uyt-d_al2*ut` to the second γ component. Add `-d_be1*uxt-d_be2*uyt` to the f term.

Replace the variables `d_c`, `d_al`, `d_be`, `d_al1`, `d_al2`, `d_be1`, and `d_be2` with appropriate expressions.

Enter 2nd-order time derivative terms in an analogous manner.

To specify a boundary condition containing time-derivative terms as in

$$\mathbf{n}\cdot(c\nabla u+\ldots)+e_q\frac{\partial^2 u}{\partial t^2}+d_q\frac{\partial u}{\partial t}+qu\;=\;g-h^T\mu\,,$$

add the terms `-e_q*utt-d_q*ut` to the g term, and provide appropriate values or expressions for the coefficients e_q and d_q in, for instance, a **Global Expressions** node's settings window.

Element Types and Discretization

The PDE and weak form interfaces provide a selection of different shape functions and associated element order (order of the shape functions). Select the shape function type from the **Shape function type** list and the order from the **Element order** list as, in most cases, **Constant**, **Linear**, **Quadratic**, **Cubic**, **Quartic**, or **Quintic** (order 0–5, respectively). Not all shape functions are available for all space dimensions and types of equations, and not all shape functions support all orders. The following table provides an overview of the available shape function types and the element orders they support:

TABLE 15-3: SHAPE FUNCTION TYPES

NAME	ORDER	COMMENTS
Lagrange	1–5 (1D and 2D); 1–4 (3D). Default: 2	The default type
Hermite	3–5 (1D and 2D); 3–4 (3D). Default: 3	
Argyris	Order 5 only.	2D only
Discontinuous Lagrange	0–5 (1D and 2D); 0–4 (3D). Default: 2	
Nodal discontinuous Lagrange	0–10	Special shape functions for wave equations
Discontinuous scalar density	0–5 (1D and 2D); 0–4 (3D). Default: 2	Not available on boundaries, edges, or points
Bubble	2 (1D); 3 (2D); 4 (3D)	Lower order on boundaries, edges, and points
Gauss point data	0, 2, 4, 8, or 10. Default: 4	Discrete values associated with the quadrature points in an integration rule of the given order
Divergence	1–4 (1D and 2D); 1–3 (3D). Default: 2	For vector fields only (1, 2, and 3 dependent variables in 1D, 2D, and 3D, respectively
Curl	1–4 (2D); 1–3 (3D). Default: 2	2D and 3D only. For vector fields only (1, 2, and 3 dependent variables in 1D, 2D, and 3D, respectively)

See Also

For more information about each shape function type, see Finite Elements in the *COMSOL Multiphysics Reference Guide*.

Weak Form Modeling

In this section:

About Weak Form Modeling

Do not be misled by the term “weak;” the weak form is very powerful and flexible. The term *weak form* is borrowed from mathematics, but in this context it has a slightly different meaning; this implementation incorporates capabilities in addition to those defined in the mathematical weak form. Moreover, knowledge of the mathematical weak form is not a prerequisite to using the COMSOL Multiphysics implementation.

The distinguishing characteristics of the weak form in COMSOL Multiphysics are that it makes it possible to:

- Enter certain equations that can be derived from an energy principle in a very compact and convenient form. Such equations, for example, arise in structural mechanics.
- Add and modify nonstandard constraints, such as various contact and friction models.
- Build models with extra equations on boundaries, edges, and points.
- Use the *test operator* to conveniently work with problems in variational calculus and parametric optimization.

COMSOL Multiphysics provides the possibility to add weak form contributions to any physics interface in the model. In addition, add *weak constraints*, which, for example, provide accurate fluxes and reaction forces.

By default, COMSOL Multiphysics converts all models to the *weak form* before solving. Although this conversion is a part of the solution technique rather than part of the modeling process, it nonetheless belongs in this discussion because it is based on the weak form implementation.

The discussion in this section reviews the ways to work with the weak form by adding weak form contributions.

Note

For a thorough mathematical and physics-related background to the weak formulation in COMSOL Multiphysics, see Weak Form Theory Background in the *COMSOL Multiphysics Reference Guide*.

The Weak Form Interfaces

In the **Model Wizard**, add weak form interfaces on all domain levels: **Weak Form PDE** on domains, **Weak Form Boundary PDE** on boundaries, **Weak Form Edge PDE** on edges (in 3D models), and **Weak Form Point** for points. All the interfaces are found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**. In all of them, weak expressions can be added, which COMSOL Multiphysics adds to the overall equation. Adding one of these interfaces creates a **PDE** node (∫du) for PDE modeling using a weak formulation. Also add a Weak Form PDE on the domain level to any other PDE interface.

When this interface is added, these default nodes are also added to the **Model Builder**—**Weak Form PDE**, **Zero Flux** (for a Weak Form PDE on the domain level only), and **Initial Values**. Right-click the main **PDE** node to add other nodes that implement other boundary conditions, for example. On the domain level, edge level, and boundary levels the same boundary conditions can be used as for the Coefficient Form PDE and General Form PDE.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `w`, `wb`, `we`, or `wp`, depending on the geometry level.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity level (domains, boundaries, edges, or points) to define the weak-form PDE. The default setting include **All domains**, **All boundaries**, **All edges**, or **All Points** in the model.

UNITS

By default, the weak form interfaces are unitless, but you can define units for the dependent variable and the source term. The units for these quantities fully define the units for all other terms in the equations. To select the units you select the corresponding quantity from a list of physical quantities or enter the unit directly.

Select the quantity that defines the unit for the dependent variable u from the **Dependent variable quantity** list. The default is **Dimensionless [1]**. Select **None** to enter a unit (for example, `K`, `m/s`, or `mol/m^3`) in the **Unit** field that appears.

Select the quantity that defines the unit for the source term f from the **Source term quantity** list. Select **None** (the default) to enter a unit (for example, `W/m^3` or `A/m^3`) in the **Unit** field that appears. The default unit is `m^-2`, which is a consistent with a dimensionless dependent variable.

DEPENDENT VARIABLES

Choose the number of dependent variables and set the field and dependent variable names. The default field and variable name for a single scalar PDE variable is u. If the field name coincides with the name of another field of the same unit and number of components, the two fields (and the interfaces which define them) will share degrees of freedom and dependent variable names. A field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**.

Select from a number of different shape function types (finite element types) from the **Shape function type** list and an associated element order (order of the shape function for the element) from the **Element order** list. The default is to use quadratic Lagrange elements.

The **Value type** (**Real** or **Complex)** of dependent variables needs to be specified when **Split complex variables in real and imaginary parts** setting is activated in the Compile

Equations node of any solver sequence used. The default is complex value type. If split complex variables setting is not active the value type is ignored.

See Also

- Element Types and Discretization
- Show More Physics Options

Weak Form PDE

Use **Weak Form PDE**, found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, to specify the PDE using a weak formulation.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity level (domains, boundaries, edges, or points) to define the weak form PDE. The default includes **All domains**, **All boundaries**, **All edges**, or **All points** in the model.

WEAK EXPRESSION

Enter the weak expression that COMSOL Multiphysics (together with any other weak expressions on the same domain) sets equal to 0 in the **weak** field. On the domain level, the default expression is `-test(ux)*ux-test(uy)*uy+test(u)`, which is the weak formulation of Poisson's equation with $f = 1$. On other geometric entity levels, the default weak expression is 0.

The Weak Contribution Node

The **Weak Contribution** node is available in all interfaces and for all geometry levels, including the global level.

To add a **Weak Contribution** node to the main PDE or other physics interface, in the **Model Builder**, right-click the main interface node and then, depending on the geometric entity level, select **More>Weak Contribution**, **Edges>Weak Contribution** or **Points>Weak Contribution** from the context menu. Or select **Global>Weak Contribution** option to apply it globally. To access these nodes, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder**.

To create an independent weak form equation rather than a weak form contribution, add extra states (dependent variables) using the **Auxiliary Dependent Variables** subnode, by right-clicking the **Weak Contribution** node. You can then use the auxiliary dependent variables in the weak-form expression.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity level (domains, boundaries, edges, or points) to define the weak contribution.

WEAK CONTRIBUTION

Enter the weak form contribution in the **Weak expression** field. For example, `-ux*test(ux)-uy*test(uy)+1*test(u)` is the weak formulation of a Poisson's equation with u as the dependent variable 1 as the source term on the right-hand side. To add a time derivative as in the time-dependent coefficient form PDE, add `-ut*test(u)` (notice the sign and the syntax `ut` for the time derivative of u).

QUADRATURE SETTINGS

These settings affect the numerical integration, and you do normally not need to change them. The **Use automatic quadrature settings** check box is selected by default. If the check box is cleared, the following settings become available:

Integration Order

The integration order is typically twice the order of the shape function. For example, the default integration order for second-order Lagrange elements is 4. The integration order is a positive integer that is entered in the **Integration order** field.

Multiplication by 2πr

This settings is only applicable for axisymmetric models. By default, the **Multiply by 2πr** check box is selected to represent, for example, the flux per area rather than by length and full revolution. If the check box is cleared, the value of the Lagrange multiplier is not multiplied by $2\pi r$.

Note

When other weak constraints are activated in Dirichlet-type boundary conditions, there is no automatic multiplication by $2\pi r$ for axial symmetry.

Integrate on Frame

This setting determines which frame to base the integration on: the spatial frame, the material frame, the mesh frame, or the geometry frame. The default is to use the default frame for the physics interface. Select the frame from the **Integrate on frame** list.

VARIATIONAL CALCULUS

Use the `test` operator to work with problems in variational calculus and optimization theory.

The Weak Contributions on Mesh Boundaries Node

To display this option, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder** and then right-click the main PDE node and select **More>Weak Contribution on Mesh Boundaries** at the domain level.

This node is similar to The Weak Contribution Node but is active on mesh boundaries. The settings are the same.

Auxiliary Dependent Variable

Note

Enable the option by first clicking the **Show** button ( ) and selecting **Advanced Physics Options**.

If a complete equation is specified on weak form in a **Weak Contribution** node and a new dependent variable that is not part of the physics interface is used, right-click the **Weak Contribution** node to add an **Auxiliary Dependent Variable**.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

Select the geometric entity to add the auxiliary dependent variable. The default geometric entity is the same as for the parent **Weak Contribution** node (domains, boundaries, edges in 3D models, or points in 2D and 3D models). In rare cases, more than one **Auxiliary Dependent Variable** node can be used on subsets of the **Weak Contribution** node's selection (to use different initial values, for example).

AUXILIARY DEPENDENT VARIABLE

Define the name and properties of the auxiliary dependent variable.

- Enter the **Field variable name** (the dependent variable).
- Enter the **Initial value** for the dependent variable. The default is 0.
- Enter the **Initial time derivative** for the dependent variable. The default is 0.

DISCRETIZATION

Select from a number of different shape function types (finite element types) from the **Shape function type** list and an associated element order (order of the shape function for the element) from the **Element order** list. The default is to use quadratic Lagrange elements.

See Also

Element Types and Discretization

The **Value type** (**Real** or **Complex)** of dependent variables needs to be specified when **Split complex variables in real and imaginary parts** setting is activated in the Compile Equations node of any solver sequence used. The default is complex value type. If split complex variables setting is not active the value type is ignored.

Modeling with PDEs on Boundaries, Edges, and Points

All PDE formulations—Coefficient Form PDE, General Form PDE, and Weak Form PDE—are available also on boundaries and edges and at points in the geometry.

Extra weak equations can be added by adding auxiliary dependent variables to a **Weak Contribution** node. Use such weak form equations as a way of handling thin layers; COMSOL Multiphysics then solves the problem by modeling rather than meshing. This approach reduces the solution time.

Model

See Transport and Adsorption (Model Library path **COMSOL_Multiphysics/ Chemical_Engineering/transport_and_adsorption**) to learn how to use the weak form boundary mode to model a thin adsorption layer with diffusion as a PDE on the boundary of a convection-diffusion problem.

See Shell Diffusion (Model Library path **COMSOL_Multiphysics/ Equation-Based_Models/shell_diffusion**) for an example of tangential derivative variables.

Specifying PDEs in Weak Form

In addition to shape function variables, use the test operator in the **Weak expression** fields in the **Weak Contribution** nodes and the **Constraint expression** fields in the **Weak**

Constraint nodes. The test operator can operate on the shape function variables: for example, `test(u1)`.

The test functions always occur linearly in the weak terms. Use these weak terms either alone to define the problem (in the Weak Form PDE interface or by adding an auxiliary dependent variable), or in addition to the other equations defined by the physics interface.

Note

For more information about the test operator and other operators, see Special Operators.

Using Weak Constraints

In this section:

- About Weak Constraints
- The Weak Constraint Node
- The Pointwise Constraint Node
- Bidirectional and Unidirectional Constraints
- Constraint Forces for PDEs
- Limitations of Weak Constraints

About Weak Constraints

The *weak constraints* in COMSOL Multiphysics implement constraints by using finite elements on the constrained domain for the Lagrange multipliers, and by solving for the Lagrange multipliers along with the original problem. The weak constraints have a number of distinct advantages. They can:

- Provide consistent flux or force values at a Dirichlet boundary, for use as input to other physics. The Lagrange multipliers along the constraints optimally balance the finite element projection of the applied loads (reaction forces). Direct evaluation of flux or force variables—which are based on derivatives of the dependent variables—is not as accurate and may lead to considerable violation of global conservation criteria. In such cases, using the Lagrange multiplier from a weak constraint as a replacement is a good option.

  Note that the Lagrange multipliers are accurate only in the integral sense required by the finite element method, and not pointwise. However, the reaction force operator (`reacf`) is a better alternative than weak constraints for calculating integrals of reaction forces or fluxes. The reaction force operator does not have the drawbacks discussed below.

See Also

- Calculating Accurate Fluxes
- The Reacf Operator

- Handle nonlinear constraints—the nonlinear solver does not store the Lagrange multipliers arising from standard constraints from one step to the next, which affects the convergence when constraints are nonlinear. Weak constraints can handle general nonlinearities because the Lagrange multipliers are updated as a part of the solution vector and give correct contributions to the stiffness matrix.
- Implement general nonsymmetric constraints, for example including nontangential derivatives—which are otherwise ill-posed. Note that unidirectional standard constraints are usually a better alternative for this type problems.

Note

Boundary conditions involving the normal component of the derivative can almost always be reformulated as a Neumann condition, which is usually preferable.

Weak constraints also come with the following drawbacks:

- Because extra unknowns are introduced for the Lagrange multipliers the problem size increases compared to when eliminating the constraints.
- The formulation normally implies that a saddle-point problem is introduced with zeros on the main diagonal of the Jacobian of the discretized equations. This class of problems is often more difficult to solve. Because iterative methods for linear systems are sensitive to the eigenvalue distribution of the system matrix, a weak constraint formulation can be much more difficult to solve than the corresponding pointwise constraint formulation.
- Lagrange multipliers are in some cases difficult to interpret. For example, Lagrange multipliers from Dirichlet conditions in axial symmetry are not equal to the reaction flux per area but rather per length and full revolution. For separate Weak Constraints nodes in axial symmetry, the default quadrature settings include a multiplication by $2\pi r$, making the Lagrange multiplier represent flux per area.
- Discontinuous constraints result in (theoretically) infinite Lagrange multipliers. In practice large oscillations result.

The Weak Constraint Node

To add a **Weak Constraint** node on the domain, boundary, edge, point, or global level, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder** and then, depending on the geometric entity level, select **More>Weak Constraint**,

Edges>Weak Constraint, or **Points>Weak Constraint** from the context menu. There is no global weak constraint option.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) to add a weak constraint—to domains, boundaries, edges in 3D models, and points in 2D and 3D models.

The default setting is to include all geometric entities in the model.

WEAK CONSTRAINT

Select a **Constraint type**—**Bidirectional, symmetric** or **User defined**. Enter a **Constraint expression**, which COMSOL constrains to 0. For example, entering `2-u` constrains u to the value 2. If **User defined** is selected, enter a **Constraint force expression**.

See Also

- Bidirectional and Unidirectional Constraints
- Constraint Forces for PDEs

QUADRATURE SETTINGS

These settings affect the numerical integration, and these do not normally need to be changed. The **Use automatic quadrature settings** check box is selected by default. If the check box is cleared, the following settings become available:

Integration Order

The integration order is typically twice the order of the shape function. For example, the default integration order for second-order Lagrange elements is 4. The integration order is a positive integer entered in the **Integration order** field.

Multiplication by $2\pi r$

This settings is only applicable for axisymmetric models. By default, the **Multiply by $2\pi r$** check box is selected to make the Lagrange multiplier represent, for example, the flux per area rather than by length and full revolution. If the check box is cleared, the

Lagrange multiplier is not multiplied by $2\pi r$ where it is used in the constraint equation, and therefore represent flux per length and full revolution.

Note

When weak constraints are activated in other Dirichlet-type boundary conditions, there is no automatic multiplication by $2\pi r$ for axial symmetry and the Lagrange multiplier therefore represents flux or force per length and full revolution rather than per area.

Integrate on Frame

This setting determines which frame to base the integration on: the spatial frame, the material frame, the mesh frame, or the geometry frame. The default is to use the default frame for the physics interface. Select the frame from the **Integrate on frame** list.

LAGRANGE MULTIPLIER

Enter a **Lagrange multiplier variable** and an **Initial value**. The variable names using weak constraints, such as `lm`, correspond to the Lagrange multipliers. Change the name if required. The weak constraints supersede the original constraints and implement them using a weak formulation.

Select a **Shape function**—**Linear Lagrange**, **Quadratic Lagrange** (the default), **Cubic Lagrange**, **Constant discontinuous**, **Linear discontinuous**, or **Quadratic discontinuous**.

From the **Define derivatives with respect to frame** list, select **Spatial**, **Material**, or **Mesh**.

The Pointwise Constraint Node

To add a **Pointwise Constraint** node on the domain, boundary, edge, or point level, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder** and then, depending on the geometric entity level, select **More>Weak Constraint** at the domain or boundary level, **Edges>Weak Constraint**, or **Points>Weak Constraint** from the context menu. There is no global pointwise constraint option.

Also add pointwise (strong) constraints, which do not create Lagrange multipliers. The settings in the **Pointwise Constraint** section are similar to those for the weak constraints.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity level (domains, boundaries, edges, or points) to add a pointwise constraint. The default includes **All domains**, **All boundaries**, **All edges**, or **All points** in the model.

POINTWISE CONSTRAINT

Select a **Constraint type**—**Bidirectional, symmetric** or **User defined**. Enter a **Constraint expression**, which COMSOL constrains to 0. For example, entering `2-u` constrains u to the value 2. If **User defined** is selected, enter a **Constraint force expression**.

See Also

- Bidirectional and Unidirectional Constraints
- Constraint Forces for PDEs

SHAPE FUNCTION

Select a **Shape function**—**Linear Lagrange**, **Quadratic Lagrange** (the default), **Cubic Lagrange**, **Constant discontinuous**, **Linear discontinuous**, or **Quadratic discontinuous**.

From the **Define derivatives with respect to frame** list, select **Spatial**, **Material**, or **Mesh**.

Bidirectional and Unidirectional Constraints

The default constraint type is *bidirectional*, which means that COMSOL Multiphysics derives the constraint force directly from the constraint at a boundary, edge, or point. Also use *unidirectional* constraints, which for constraints written as $0 = R$ on $\partial\Omega$, change the constraint force according to

$$\left(\frac{\partial R}{\partial u}\right)^T \mu \to -I_{\partial\Omega}\mu$$

where $I_{\partial\Omega}$ is an identity operator for points on the boundary $\partial\Omega$. For example, using two variables u_1 and u_2 and the constraints

$$\begin{cases} 0 = R_1(u_1, u_2) & \text{on } \partial\Omega \\ 0 = R_2(u_1, u_2) & \text{on } \partial\Omega \end{cases},$$

then the unidirectional constraint force is $-\mu$ acting on both equations for the boundary $\partial\Omega$.

BIDIRECTIONAL VS. UNIDIRECTIONAL WEAK CONSTRAINTS

Bidirectional weak constraints in theory implement the same boundary conditions as the standard, pointwise, constraints. In some cases, the result may differ slightly, but this is only due to the different discretization. Using unidirectional constraints, on the other hand, modifies the way boundary conditions are interpreted. Switching from pointwise constraints to unidirectional weak constraints therefore can modify the *physics* of the model, while using bidirectional weak constraints only affects the *numerics*.

The difference between bidirectional and unidirectional constraints is the way the Lagrange multipliers—which can be interpreted as generalized reaction forces—are applied. In a bidirectional constraint, the Lagrange multipliers are applied symmetrically on all dependent variables involved in the constraint so as to keep symmetric problems symmetric. In a unidirectional constraint, the reaction forces are applied only on the physics interface's dependent variable at the boundary where the constraint is specified.

Imagine a model with two physics interfaces A and B, with dependent variables u and v, respectively. If physics interface A specifies a standard constraint as $R = 2u - 3v = 0$, COMSOL Multiphysics implements the bidirectional weak boundary constraint by adding the weak term

$$\int_B \hat{\mu} \cdot (2u - 3v)ds + \int_B \mu \cdot (2\hat{u} - 3\hat{v})ds$$

where μ and $\hat{\mu}$ are the Lagrange multiplier and corresponding test function, and $\hat{u}$ and $\hat{v}$ are the test functions corresponding to u and v, respectively. Because μ multiplies both test functions in the second integral, both u and v are affected by the constraint, which is therefore *bidirectional*.

The corresponding unidirectional weak constraint adds the weak form contribution

$$\int_B \hat{\mu} \cdot (2u - 3v)ds + \int_B \mu \cdot \hat{u} ds$$

Here, the Lagrange multiplier μ only multiplies the test function $\hat{u}$. All reaction forces therefore apply only to u, while v is left unaffected by the constraint. This makes the constraint *unidirectional*. Note also that the unidirectional constraint does not consider the factor 2 in front of u when applying the forces. Even if the constraint would contain spatial derivatives, the Lagrange multiplier would still be applied only directly on the dependent variable itself.

Constraint Forces for PDEs

In the **Weak Constraint** nodes' settings windows, the constraint type to use can be specified. The default constraint type is *bidirectional*, which means that the Coefficient Form PDE and General Form PDE interfaces derive the constraint force directly from the constraint that is specified in the **Constraint expression** field.

To use the *unidirectional* constraints, the constraint force from a constraint written as **constr** $= R = 0$ on $\partial\Omega$, changes according to

$$\left(\frac{\partial R}{\partial u}\right)^T \mu \rightarrow -I_{\partial\Omega}\mu$$

where $I_{\partial\Omega}$ is an identity operator for points on the boundary $\partial\Omega$. If, for example, a two-variable problem for variables u_1 and u_2 specifies the constraints (using the **Constraint expression** field)

$$\begin{cases} 0 = R_1(u_1, u_2) & \text{on } \partial\Omega \\ 0 = R_2(u_1, u_2) & \text{on } \partial\Omega \end{cases},$$

then the default unidirectional constraint force is $-\mu_1$ acting on u_1 and $-\mu_2$ acting on u_2. The corresponding bidirectional constraint would give reaction forces

$$\left(\frac{\partial R_1}{\partial u_1}\right)^T \mu_1 + \left(\frac{\partial R_2}{\partial u_1}\right)^T \mu_2$$

acting on u_1 and

$$\left(\frac{\partial R_1}{\partial u_2}\right)^T \mu_1 + \left(\frac{\partial R_2}{\partial u_2}\right)^T \mu_2$$

acting on u_1.

If you want a unidirectional constraint or a constraint force different from any of these two choices, a separate expression must be specified for the constraint force. Select **User-defined** from the **Constraint type** list and enter the expression in the **Constraint force expression** field. In this field specify the constraint force Jacobian using the `test` operator. For example, the expression corresponding to the default unidirectional constraint in the example above is

$$\begin{cases}\texttt{-test(u1)}\\ \texttt{-test(u2)}\end{cases}$$

Note

The bidirectional constraint type normally generates a symmetric constraint Jacobian matrix and constraint force Jacobian matrix coupling for the discretized equations, while unidirectional constraints generate an unsymmetrical coupling. See also Solver Studies and Study Types.

Limitations of Weak Constraints

Weak constraints are more difficult to use than conventional pointwise constraints. Note these tips when using them:

- Pointwise and weak constraints on the same set of variables on adjacent boundaries (boundaries that share common node points in the mesh) do not work. This means that if all boundaries must be constrained on a solid and want to use a weak constraint on one boundary segment (one face), the weak constraint must be used on the entire boundary of the solid (if the boundary is connected).
- There must always be a nontrivial constraint (that is, a Dirichlet condition) on the boundaries where the weak constraint is enabled and for the constraint to take effect.
- Problems with the scaling of linear systems sometimes arise in conjunction with weak constraints. The Lagrange multiplier variables always have a different unit than the main system variables, and can therefore be of a completely different order of magnitude. Usually the automatic variable scaling in the solvers is sufficient, but there are cases when manual scaling might need to be considered.
- Always use the same shape function type for the weak constraint as for the variables that are constrained, possibly with lower-order elements for the weak constraint. In some cases (for example, when constraining derivatives) the system of equations can become singular. The reason is usually that there are redundant Lagrange multiplier degrees of freedom in the model. Try to lower the order of the Lagrange multiplier variables or use constraints on the Lagrange multiplier to remove some degrees of freedom.
- To use iterative methods for the linear system and if the weak constraint formulation gives zeros on the diagonal of the system matrix, the SOR class of preconditioners

and smoothers do not work. Instead try use the Vanka algorithm with the Lagrange multipliers as the Vanka variables or use the incomplete LU factorization algorithm.

Solving ODEs and DAEs

In this section:

Adding ODEs, DAEs, and Other Global Equations

When working on complex models, single named degrees of freedom, or *states*, might need to be introduced, which are not logically related to any particular point in space, and their corresponding equations. In particular, such situations arise when modeling physics in interaction with an external system, for example, a controller or an electrical circuit built from standard components. It is often possible to describe such external systems by a system of ordinary differential equations, ODEs, with a limited number of degrees of freedom.

The Global Equations and ODEs interface has a **Global Equations** node that is designed for implementing this type of external equation. Such equations are often tightly coupled to a model in a physical domain. The Global Equations node is also available from any of the physics interfaces. Access it by right-clicking the main node and then choosing **Global>Global Equations**. Use it for ODEs, differential algebraic equations, purely algebraic equations and conditions, and transcendental equations, or to add degrees of freedom to a model using the introduced states. Possible uses include:

- Controllers
- Rigid-body mechanics
- Nonlinear eigenvalue problems
- Continuation
- Integral constraints
- Augmented or generalized equations

The ODEs and DAEs Interface

The **ODEs and DAEs** interface ($\frac{d}{dt}$), found under the **Mathematics** branch (Δu) in the **Model Wizard**, has the possibility to add ODEs or other global equations to a model. The main node is the **Global Equations** node, used to define the global equations including the names of the variables (states), the required initial values, and an optional description. When the **Global Equations and ODEs** interface is added, it adds a default **Global Equations** node. Right-click the **Global Equations and ODEs** node to add additional **Global Equations** node or a **Weak Contribution** node.

The Global Equations Node

Right-click the **ODEs and DAEs** node (first click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder**) Then select **Global>Global Equations** to add global (space-independent) equations such as ODEs. These are added in the **Global Equations** node's settings window:

GLOBAL EQUATIONS

The global equations that you can solve have the following form:

$$f(u, \dot{u}, \ddot{u}, t) = 0$$

with the initial conditions $u(t_0) = u_0$ and $u_t(t_0) = u_{t,0}$ (where the subscript t indicates the time derivative). Several equations can be added and the equation can be coupled.

The first time derivative of u is specified as `ut`, and the second time derivative of u is `utt`. With time derivatives, this equation is an ODE (ordinary differential equation). With no time derivatives, the equation is an algebraic equation or a transcendental equation. The initial conditions are only necessary for ODEs.

In the table, each row corresponds to a named state; that is, it defines a single degree of freedom and one equation. Fill in the name of the state variable in the **Name** column. This also defines time-derivative variables. If a state variable is called `u`, its first and second time derivatives are `ut` and `utt`, respectively. These variables become available in all geometries. Therefore the names must be unique.

Use the **f(u,ut,utt,t)** column to specify the left side of the equation that is to be set equal to zero. The software then adds this global equation to the system of equations. When solving the model, the value of the state variable `u` is adapted in such a way that the associated global equation is satisfied. All state variables and their time derivatives can be used as well as any parameters, global variables, and coupling operators with a scalar output and global destination in the **f(u,ut,utt,t)** column. The variables can be functions of the state variables in the global equations. Setting an equation for a state is optional. The default value of `0` means that the software does not add any additional condition to the model.

If the time derivative of a state variable appears somewhere in the model during a time-dependent solution, the state variable needs an initial condition. Models that contain second time derivatives also require an initial value for the first time derivatives of the state variables. Set these conditions in the third (**Initial value (u_0)**) and fourth (**Initial value (u_t0)**) columns. Enter comments about the state or the equation in the last column, **Description**.

As required, edit the selected row in the table of global equations using the **Name**, **f(u,ut,utt,t)**, **Initial value (u_0)**, **Initial value (u_t0)**, and **Description** fields underneath the table.

To add another equation, make additional entries in the second row.

Move equation rows up and down using the **Move Up** (⇧) and **Move Down** (⇩) buttons. To remove an equation, select some part of that equation's row in the table and click the **Delete** button (✖). Save the definitions of the global equations to a text file by clicking the **Save to File** button (💾) and using the **Save to File** dialog box that appears. To load a text file with global equation definitions, use the **Load from File** button (📂) and using the **Load from File** dialog box that appears. Data must be separated by spaces or tabs.

DISCRETIZATION

To display this section, click the **Show** button (👁) and select **Discretization**.

The **Discretization** section for **Global Equations** is used to specify the **Value type** (**Real** or **Complex**) of the variables defined by global equations for the case of using activated

Split complex variables in real and imaginary parts setting in the **Compile Equations** node of any solver sequence. The default for split complex variables setting is to be not active and in that case you do not need to specify the value type for global equations variables (the value type specified would be ignored in such case). The value type for all the variables defined by this **Global Equations** node is selected in the **Value type when using splitting of complex variables** selection. The default value type is **Complex**.

The Global Constraint Node

Right-click the **ODEs and DAEs** node or click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder** and then right-click the main node for any other physics interface and select **Global>Global Constraint** to add a global (space-independent) constraint. With such a constraint you can, for example, make the sum of some global variables equal to a fixed number. You add the constraint in the **Global Constraint** node's settings window, which contains the following section:

GLOBAL CONSTRAINT

Select a **Constraint type**—**Bidirectional, symmetric** or **User defined**. Enter a **Constraint expression**, which COMSOL constrains to 0. For example, entering `2-u+v` constrains $u+v$ to the value 2. If **User defined** is selected, enter a **Constraint force expression**. See Bidirectional and Unidirectional Constraints and Constraint Forces for PDEs for more information.

The Weak Contribution (ODEs and DAEs) Node

Another option is to enter equations in the weak form using the **Weak Contribution** node. This can be convenient in advanced modeling because it controls the test variables multiplying the equations. Wherever a test function of a state variable appears (in the **Global Equations** node or elsewhere in the model), whatever it multiplies ends up in the same equation in the discrete system.

Note

There can be zero or more weak expressions, regardless of the number of states.

WEAK CONTRIBUTION

Enter the expression that contains the weak formulation in the **Weak expression** field.

Note

See The Wall Distance Interface for an example of how to write an ODE using a weak formulation.

The Discretization Node

To display the option, click the **Show** button ( ) and select **Advanced Physics Options** from the **Model Builder** then right-click the main physics interface node and select **Global>Discretization** to add extra **Discretization** nodes for use in, for example, multigrid hierarchy.

Select which discretization to use in the **Multigrid Level** subnodes for the study steps.

Note

The **Discretization** page usually has one section and it is the same as the **Discretization** section found when it is selected in the main physics interface node's settings window.

See Also

- Show Advanced Physics Options—Context Menu Equation-Based Feature Nodes
- Element Types and Discretization
- Show Advanced Physics Options

In the *COMSOL Multiphysics Reference Guide*:

- Discretization of the Equations
- Finite Elements
- For information about how to specify the splitting of complex variables, see Compile Equations.

Presenting Results for Global Equations

The state variables are scalar values and they are available globally. To view the results for an ODE, use the Line Graph, Point Graph, and Global plot types and Global Evaluation for displaying the numerical solution.

Solving ODEs

As an example of ODEs, the following equations are the *Lotka-Volterra equations*:

$$\dot{r} = ar - brf$$
$$\dot{f} = -cf + drf$$

where r is the rabbit population, and f is the population of foxes. This is an example of a system of two coupled ODEs.

Enter these equations as `rt-a*r+b*r*f` and `ft+c*f-d*r*f`, where a, b, c, and d are scalar values defined using the **Parameters** node. For this first-order ODE, specify initial values for r and f.

Solving Algebraic and Transcendental Equations

As an example of an algebraic equation, consider the equation $f(u) = 0$ for

$$f(u) = u^3 + u - 2$$

This equation has a single root at $u = 1$. To enter it into the **Global Equations** window, type `u` in the **Name** column and `u^3+u-2` in the **Equation f(u,ut,utt,t)** column (both entries on the same row). Then solve this using a stationary solver. To display the solution value, right-click **Numerical Results** in the **Results** branch and select **Global Evaluation**. In the settings window, select **u** from the list of predefined quantities (click the down arrow in the **Expression** section and select **ODEs and DAEs>(u)**).Click the Evaluate button, and the value of the root appears in the **Results** section.

As an example of a transcendental equation, consider the equation $f(u) = 0$ for

$$f(u) = e^{-u} - u$$

A root to this equation is approximately 0.56714. To enter it into the **Global Equations** window, type `u` in the **Name** column and `exp(-u)-u` in the **f(u,ut,utt,t)** column (both entries on the same row). Then compute the solution. To display the solution value, right-click **Numerical Results** in the **Results** branch and select **Global Evaluation**. In the settings window, select **u** from the list of predefined quantities (click the down arrow in the **Expression** section and select **ODEs and DAEs>(u)**). Click the **Evaluate** button, and the value of the root (rounded to 0.567) appears in the **Results** section.

Distributed ODEs and DAEs

For ODEs and DAEs in domains, on boundaries and edges, and at points, the Distributed ODEs and DAEs interface is available for solving the following ODE (or DAE):

$$e_{\mathrm{a}}\frac{\partial^2 u}{\partial t^2}+d_{\mathrm{a}}\frac{\partial u}{\partial t} = f$$

Because it is defined on a geometry but is space independent, it solves the ODE or DAE as a distributed ODE or DAE, with a solution that is defined as a field but with no space variation.

The Distributed ODEs and DAEs Interfaces

The interfaces for distributed ODEs and DAEs—**Domain ODEs and DAEs**, **Boundary ODEs and DAEs**, **Edge ODEs and DAEs**, and **Point ODEs and DAEs** ($\frac{d}{dt}$)—found under the **Mathematics>ODE and DAE Interfaces** branch ($\frac{d}{dt}$) in the **Model Wizard**, provides the possibility to solve distributed ODEs and DAE in domains, on boundaries and edges, and at points.

When this interface is added, these default nodes are also added to the **Model Builder**—**Distributed ODE** and **Initial Values**. Right-click **Distributed ODEs and DAEs** to add other nodes that implement other algebraic equations, for example.

The following sections provides information about nodes available in the distributed ODEs and DAEs interfaces.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `dode`, `bode`, `eode`, or `pode` (in domains, on boundaries and edges, and at points, respectively).

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the distributed ODEs or DAEs are active. To choose specific domains, select **Manual** from the **Selection** list.

DEPENDENT VARIABLES

Choose the number of dependent variables and set the field and dependent variable names. The default field and variable name for a single scalar PDE variable is u. If the field name coincides with the name of another field of the same unit and number of components, the two fields (and the interfaces which define them) will share degrees of freedom and dependent variable names. A field name must not coincide with the name of a field of another type, or with a component name belonging to some other field. Component names must be unique within a model except when two interfaces share a common field name.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**.

Select from a number of different shape function types (finite element types) from the **Shape function type** list and an associated element order (order of the shape function for the element) from the **Element order** list. The default is to use quadratic Lagrange elements.

See Also

- Element Types and Discretization
- Show More Physics Options

The **Value type** (**Real** or **Complex)** of dependent variables needs to be specified when **Split complex variables in real and imaginary parts** setting is activated in the Compile Equations node of any solver sequence used. The default is complex value type. If split complex variables setting is not active the value type is ignored.

Distributed ODE

This is the default equation for a **Distributed ODE** in an interface for distributed ODEs and DAEs. Specify the coefficients for an ODE with the following equation coefficients:

$$e_{\mathrm{a}}\frac{\partial^2 u}{\partial t^2}+d_{\mathrm{a}}\frac{\partial u}{\partial t} = f$$

- e_{a} is the *mass coefficient*.
- d_{a} is a *damping coefficient* or a *mass coefficient*.
- f is the *source term*.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the distributed ODEs are active. To choose specific domains, select **Manual** from the **Selection** list.

SOURCE TERM

Enter a value or expression for the source term f. If there are multiple dependent variables, there is a vector of f component inputs.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping (or mass) coefficient d_{a}. If there are multiple dependent variables, there is a matrix of d_{a} component inputs.

MASS COEFFICIENT

Enter a value or expression for the mass coefficient e_{a}. If there are multiple dependent variables, there is a matrix of e_{a} component inputs.

Algebraic Equation

This node adds an **Algebraic Equation** in an interface for distributed ODEs and DAEs. Specify an algebraic equation as

$$f = 0 \qquad (15\text{-}2)$$

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the algebraic equation is active. To choose specific domains, select **Manual** from the **Selection** list.

ALGEBRAIC EQUATION

Enter an expression *f* for the equation $f = 0$ that defines the algebraic equation in the **f** field. If there are multiple dependent variables, there is a vector of algebraic equations for each variable.

The Events Interface

Use the **Events** interface ( ) to create solver events. An event can be explicit or implicit, and the difference is that you for explicit events must specify the exact time when the event occurs. When an event occurs, the solver stops and provides a possibility to reinitialize the values of states and dependent variables.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `ev`.

Discrete States

Use the **Discrete States** feature ( ) to define discrete state variables that do not have to be continuous in time. They are often used as logical help variables that control what expression to use in a weak expression or constraint.

DISCRETE STATES

Enter values or expressions in the table for the **Name**, **Initial value (u0)**, and enter a **Description**.

Indicator States

Use the **Indicator States** feature ( ) to define state variables that the solver uses to trigger implicit events, which occurs at a zero crossing of an indicator state variable.

INDICATOR VARIABLES

Multiple state variable can be defined in the table together with corresponding g-coefficients and initial values. A state variable, u, gets its value by solving the following equation:

$$g(u, \dot{u}, \ddot{u}, t) - u = 0$$

with the initial conditions $u(t_0) = u_0$ (where the dot notation indicates time derivatives). You use the state variables in the implicit event condition that you specify for the Implicit Event feature.

Enter values or expressions in the table for the **Name**, g-coefficients **g(u,ut,utt,t)**, **Initial value (u0)**, and enter a **Description**.

Explicit Event

An **Explicit Event** feature ( ) specify an event that occurs at predefined times entered in the **Event Timings** section. When an event occurs, it is possible to specify reinitialization of global states in the **Reinitialization** section. Right-click an **Explicit Event** node to add the Reinitialization on Domains node.

EVENT TIMINGS

Write the first time an explicit event occur in the **Start of event** edit field. It is also possible to repeatedly trigger the event by entering the period in the **Period of event** field.

REINITIALIZATION

In the table, enter information in the **Variable** and **Expression** columns for the global state variables that the event reinitializes to the given expression.

Implicit Event

Use the **Implicit Event** feature ( ) to specify an event that occurs when an condition involving an indicator state is fulfilled. When an event occurs, it is possible to specify reinitialization of global state variables. Right-click an **Implicit Event** node to add a Reinitialization on Domains node.

EVENT CONDITIONS

Enter the condition in the **Condition** field. When the condition changes its state from false to true, this triggers the implicit event.

REINITIALIZATION

In the table, enter information in the **Variable** and **Expression** columns for the global state variables that the event reinitializes to the given expression.

Reinitialization on Domains

The **Reinitialization on Domains** feature can reinitialize dependent variables defined on domains when an event occurs. Right-click an Explicit Event or Implicit Event node to add this node.

REINITIALIZATION

In the table under **Variables** enter the dependent variables. In the **Expression** column, enter the corresponding information for each variable as defined on domains that the event reinitializes.

The Wall Distance Interface

The **Wall Distance** interface ( ), found under the **Mathematics** branch (Δu) in the **Model Wizard**, has the equations and boundary conditions for calculating the distance to walls in fluid-flow simulation using a modified Eikonal equation, solving for a dependent variable G that is related to the wall distance. The main node is the **Distance Equation** node, which adds the distance equation (modified Eikonal equation) and provides an interface for defining the reference length scale.

When this interface is added, these default nodes are also added to the **Model Builder**—**Wall Distance**, **Distance Equation**, and **Initial Values**. Default boundary conditions are also added—a homogeneous Neumann condition that does not appear in the Model Builder.

Right-click the **Wall Distance** node to add boundary conditions for walls. Interface Identifier

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `wd`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model to define where the distance equation is active. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

Enter a **Smoothing parameter** σ_w. in Equation 15-5. The default value is 0.1.

DEPENDENT VARIABLES

The dependent variable is the **Reciprocal wall distance** G. You can change its name, which changes both the field name and the variable name. If the new name coincides with the name of another reciprocal distance field in the model, the interfaces will share degrees of freedom. The new name must not coincide with the name of a field of another type, or with a component name belonging to some other field.

DISCRETIZATION

To display this section, click the **Show** button ( ) and select **Discretization** from the **Model Builder**. Select **Quadratic** (the default), **Linear, Cubic**, or **Quartic** for the **Reciprocal wall distance**.

See Also

Show More Physics Options

Distance Equation

The **Distance Equation** node adds to Equation 15-5, and the **Distance Equation** form contains the following sections for defining the length scale.

DOMAIN SELECTION

From the **Selection** list, choose the domains to use the distance equation with the length scale defined in this **Distance Equation** node.

DISTANCE EQUATION

Specify the length scale using the reference length l_{ref}. From the **Length scale specification**, choose from the following to compute the reference length:

- Select **Automatic** to use Equation 15-6 to compute l_{ref}. This is the default setting.
- Select **Manual** to specify the reference length (SI unit: m) manually in the l_{ref} field. The default value is 1.

Initial Values

The **Initial Values** node adds an initial value for the reciprocal wall distance that can serve as initial guess for a nonlinear solver.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define an initial value.

INITIAL VALUES

Enter a value or expression for the initial value of the reciprocal wall distance G (SI unit: 1/m) in the **Reciprocal wall distance** field. The default value is $G_0 = 2/l_{\mathrm{ref}}$.

Boundary Conditions for the Wall Distance Interface

The following boundary conditions are available:

- The default boundary condition is a homogeneous Neumann condition, which applies to all boundaries that are not walls:

  $$\nabla G \cdot \mathbf{n} = 0$$

  This boundary condition does appear as a node in the model tree.
- The Wall boundary condition (see below) for boundaries that are walls.

1D Axi

2D Axi

For axisymmetric models, COMSOL Multiphysics automatically takes the axial symmetry boundaries (at $r = 0$) into account, and prohibits the wall boundary node from being defined here.

Wall

The **Wall** node implements the following boundary condition for walls:

$$G = G_0 = \frac{2}{l_{\mathrm{ref}}}$$

BOUNDARY SELECTION

Select the boundaries that represent walls.

Theory for the Wall Distance Interface

The Wall Distance Interface theory is described in this section:

- The Eikonal Equation
- Modified Eikonal Equation
- Reference for the Wall Distance Interface

The Eikonal Equation

Turbulence models often use the distance to the closest wall to approximate the mixing length or for regularization purposes. One way to compute the wall distance is to solve the *Eikonal equation*:

$$|\nabla D| = 1 \tag{15-3}$$

with $D = 0$ on solid walls and $\nabla D \cdot \mathbf{n} = 0$ on other boundaries. The Eikonal equation can be tricky to solve, and produces the exact distance to the closest wall. The modified equation described below is less computationally expensive to solve. It also uses a reference length to put more emphasis on solid objects larger than the reference length and reduce the emphasis on smaller objects. The introduction of a reference length produces a more relevant wall distance in the following instances:

- In convex regions of small dimensions, the wall distance is reduced to reflect the close proximity of several walls.
- When a small object, such as a thin wire for example, is present in free flow, the wall distance is affected only in a very small region around it.

Modified Eikonal Equation

COMSOL Multiphysics uses a modified Eikonal equation based on the approach in Ref. 1. This modification changes the dependent variable from D to $G = 1/D$. Equation 15-3 then transforms to

$$\nabla G \cdot \nabla G = G^4 \tag{15-4}$$

Additionally, the modification adds some diffusion and multiplies G^4 by a factor to compensate for the diffusion. The result is the following equation, which the Wall Distance interface uses:

$$\nabla G \cdot \nabla G + \sigma_w G(\nabla \cdot \nabla G) = (1 + 2\sigma_w)G^4 \tag{15-5}$$

where σ is a small constant. If σ_w is less than 0.5, the maximum error falls of exponentially when σ_w tends to zero. The default value of 0.1 is a good choice for both linear and quadratic elements.

The boundary conditions for Equation 15-5 is $G = G_0 = C/l_{\text{ref}}$ on solid walls and homogeneous Neumann conditions on other boundaries. The effect of C is that the solution becomes less smeared the higher the value of C. The error tends asymptotically to $0.2l_{\text{ref}}$ as C tends to infinity, but making it very large destabilizes Equation 15-5. C is 2 in the Wall Distance interface.

For a channel, l_{ref} should typically be set to the channel width or there about. l_{ref} has a lower bound in that it must be larger than all cells adjacent to any boundary where the boundary condition $G = G_0$ is applied; otherwise, the solution displays oscillations. l_{ref} is the only parameter in the model, and the default value is half the shortest side of the geometry bounding box. If the geometry consists of several very slender entities, or if the geometry contains very fine details, this measure can be too large. Then define l_{ref} manually.

The initial value is by default defined as $G_0 = 2/l_{\text{ref}}$, in correspondence with the boundary conditions.

The wall distance $D_w = 1/G - 1/G_0$ is a predefined variable that is used for analysis. You have also access to a vector-valued variable that represents the direction toward the nearest wall, which is defined as

$$\frac{\nabla G}{\sqrt{\max(\nabla G^2, \text{eps})}} \tag{15-6}$$

Reference for the Wall Distance Interface

1. E. Fares and W. Schröder, “A Differential Equation for Approximate Wall Distance,” *International Journal for Numerical Methods in Fluids*, vol. 39, pp. 743–762, 2002.

16

Sensitivity Analysis

This chapter describes how to perform sensitivity analysis using the Sensitivity interface, which is found under the **Mathematics>Optimization and Sensitivity** (🎯) branch in the **Model Wizard**.

In this chapter:

Theory for the Sensitivity Interface

The Sensitivity Interface theory is described in this section:

- About Sensitivity Analysis
- Sensitivity Problem Formulation
- Theory for Stationary Sensitivity Analysis
- Specification of the Objective Function
- Choosing a Sensitivity Method
- Issues to Consider Regarding the Control Variable
- Issues to Consider Regarding the Objective Function

About Sensitivity Analysis

The Sensitivity interface ( ) is special in the sense that it does not contain any physics of its own. Instead, it is a tool that makes it possible to evaluate the sensitivity of a model with respect to almost any variable.

Simulation is a powerful tool in science and engineering for predicting the behavior of physical systems, particularly those that are governed by partial differential equations. However, often a single simulation (or just a few) is not enough to provide sufficient understanding of a system. Hence, a more exploratory process might be needed, such as *sensitivity analysis*, where one is interested in the sensitivity of a specific quantity with respect to variations in certain parameters included in the model. Such an analysis can for example be used for estimating modeling errors caused by uncertainties in material properties, or for predicting the effect of a geometrical change.

Many times it is possible to reformulate problems of the above type as the problem of calculating derivatives, so differentiation plays a central role in solving such problems. The Sensitivity interface can calculate derivatives of a scalar *objective function* with respect to a specified set of *control variables*. The objective function is in general a function of the solution to a multiphysics problem, which is in turn parameterized by the control variables.

Sensitivity Problem Formulation

Because the Sensitivity interface does not contain any physics, it is not intended for use on its own. When the interface is added to a multiphysics model, no new equations are introduced, and the set of solution variables remains the same. Instead, an objective function and a control variable can be specified. The physics interface can perform these distinct tasks:

- Select a control variables and set their values
- Define a scalar objective function
- Compute the sensitivities efficiently using the sensitivity solver

Note

The control variables are independent variables whose values are not affected by the solution process, but they are also degrees of freedom (DOFs) stored in the solution vector. When defining a control variable, its *initial value* must be supplied. The initial value is used to initialize the control variable DOFs, which remain fixed during the solution step.

Theory for Stationary Sensitivity Analysis

Evaluating the sensitivity of a scalar-valued objective function $\boldsymbol{Q}(\xi)$ with respect to the control variables, ξ, at a specific point, ξ_0, can be rephrased as the problem of calculating the derivative $\partial\boldsymbol{Q}/\partial\xi$ at $\xi = \xi_0$. In the context of a multiphysics model, $\boldsymbol{Q}$ is usually not an explicit expression in the control variables ξ alone. Rather, $\boldsymbol{Q}(u(\xi), \xi)$ is also a function of the solution variables u, which are in turn implicitly functions of ξ.

The multiphysics problem is a PDE, which after discretization is represented as a system of equations $\boldsymbol{L}(u(\xi), \xi) = 0$. If the PDE has a unique solution $u = \boldsymbol{L}^{-1}(\xi)$, the sensitivity problem can be informally rewritten using the chain rule as that of finding

$$\frac{d}{d\xi}\boldsymbol{Q}(u(\xi), \xi) = \frac{\partial\boldsymbol{Q}}{\partial\xi} + \frac{\partial\boldsymbol{Q}}{\partial u}\cdot\frac{\partial u}{\partial\boldsymbol{L}}\cdot\frac{\partial\boldsymbol{L}}{\partial\xi}$$

The first term, which is an explicit partial derivative of the objective function with respect to the control variables is easy to compute using symbolic differentiation. The second term is more difficult. Assuming that the PDE solution has N degrees of freedom and that there are n control variables ξ_i, $\partial\boldsymbol{Q}/\partial u$ is an N-by-1 matrix, $\partial u/\partial\boldsymbol{L}$ is an N-by-N matrix (since $\boldsymbol{L}^{-1}$ is unique) and $\partial\boldsymbol{L}/\partial\xi$ is an N-by-n matrix.

The first and last factors, $\partial Q/\partial u$ and $\partial L/\partial \xi$ can be computed directly using symbolic differentiation. The key to evaluating the complete expression lies in noting that the middle factor can be computed as $\partial u/\partial L = (\partial L/\partial u)^{-1}$ and that $\partial L/\partial u$ is the PDE Jacobian at the solution point:

$$\frac{d}{d\xi}Q(u(\xi), \xi) = \frac{\partial Q}{\partial \xi} + \frac{\partial Q}{\partial u} \cdot \left(\frac{\partial L}{\partial u}\right)^{-1} \cdot \frac{\partial L}{\partial \xi} \qquad (16\text{-}1)$$

However, actually evaluating the inverse of the N-by-N Jacobian matrix is too expensive. In order to avoid that step, an auxiliary linear problem can be introduced. This can be done in two different ways, each requiring at least one additional linear solution step.

FORWARD SENSITIVITY METHOD

To use the *forward sensitivity* method, introduce the N-by-n matrix of solution sensitivities

$$\frac{\partial u}{\partial \xi} = \left(\frac{\partial L}{\partial u}\right)^{-1} \cdot \frac{\partial L}{\partial \xi}$$

These can be evaluated by solving n linear systems of equations

$$\frac{\partial L}{\partial u} \cdot \frac{\partial u}{\partial \xi_i} = \frac{\partial L}{\partial \xi_i}$$

using the same Jacobian $\partial L/\partial u$, evaluated at $u(\xi_0)$. Inserting the result into Equation 16-1, the desired sensitivities can be easily computed as

$$\frac{d}{d\xi}Q(u(\xi), \xi) = \frac{\partial Q}{\partial \xi} + \frac{\partial Q}{\partial u} \cdot \frac{\partial u}{\partial \xi}$$

ADJOINT SENSITIVITY METHOD

To use the *adjoint sensitivity* method, introduce instead the N-by-1 adjoint solution u^*, which is defined as

$$u^* = \frac{\partial Q}{\partial u} \cdot \left(\frac{\partial L}{\partial u}\right)^{-1}$$

Multiplying this relation from the right with the PDE Jacobian $\partial L/\partial u$ and transposing leads to a single linear system of equations

$$\frac{\partial L}{\partial u}^T \cdot u^* = \frac{\partial Q}{\partial u}$$

using the transpose of the original PDE Jacobian.

Specification of the Objective Function

The objective function may in general be a sum of a number of terms:

$$Q(u, \xi) = Q_{\text{global}}(u, \xi) + Q_{\text{probe}}(u, \xi) + \sum_{k=0}^{n} Q_{\text{int}, k}(u, \xi)$$

where n is the space dimension of the multiphysics model and the different contributions in the sum above are defined as follows:

- Q_{global} is the *global contribution* to the objective function Q. It is given as one or more general global expressions.
- Q_{probe} is a *probe contribution* to the objective function Q. It is a *probe objective* so its definition is restricted to a point on a given geometrical entity The *probe point* used for the point evaluation is a point given by the user and has to be contained in the domain.
- $Q_{\text{int},k}$ is an *integral contribution* to the objective function Q. It is an *integral objective* so its definition is restricted to a specific set of geometrical entities of the same dimension. For integral contributions on points, the integration reduces to a summation.

Several global, probe, and integral contributions can be defined. In such cases, the total global, probe, and integral contribution is given as the sum of the aforementioned global, probe, and integral contributions that are actively selected in the solver feature settings for the optimization.

Choosing a Sensitivity Method

To evaluate sensitivities as part of a multiphysics problem solution, an auxiliary linear problem must be solved, in addition to the original equation. Choose between these methods:

- Select the *forward sensitivity* method to evaluate the derivatives of all solution variables and an optional objective function.
- Select the *adjoint sensitivity* method to look only at derivatives of a scalar objective function.

FORWARD SENSITIVITY

Use the forward sensitivity method to solve for the derivatives of all dependent variables, plus an optional scalar objective function, with respect to a small number of control variables. The forward method requires one extra linear system solution for each control variable.

The linear system that must be solved is the same as the last linearization needed for solving the forward model. Thus, when using a direct solver (for example, PARDISO) the extra work amounts only to one back-substitution per control variable DOF. The iterative linear and segregated solvers can reuse preconditioners and other data but must otherwise perform a complete solution each time.

ADJOINT SENSITIVITY

The adjoint method solves for the derivatives of a single scalar objective function with respect to any number of control variables, requiring only one single additional linear system solution. In addition to the objective function gradient, the discrete adjoint solution is computed. This quantity represents the sensitivity of the objective function with respect to an additional generalized force applied as a nodal force to the corresponding solution component.

The auxiliary linear system is in this case the transpose of the last linearization needed for solving the forward model. The MUMPS and PARDISO linear solvers can solve the transposed problem at the cost of a back-substitution, while the SPOOLES linear solver needs to do a new factorization if the problem is not symmetric or Hermitian.

The iterative solvers can reuse most preconditioning information as can the segregated solver, which, however, loops over the segregated steps in reversed order.

Note

Sensitivity analysis can be used together with all stationary and parametric standard solvers and with the BDF solver for transient studies. The available solvers are described in the section Solvers and Study Types. For technical details about the solution procedure, see the *COMSOL Multiphysics Reference Guide*.

Issues to Consider Regarding the Control Variable

THE EFFECT OF DISCRETIZATION

The sensitivity analysis is always performed on the discretized system of equations. As already mentioned, the control variable can be a scalar/vector or an element in some infinite-dimensional function space. In the latter case it is represented on the finite element mesh, just like the solution variables, or global scalar quantities. When using a control variable field represented on the finite element mesh, the sensitivities are therefore associated with individual control variable degrees of freedom rather than with the field value at each point. This makes it difficult to interpret the result. For example, if a domain control variable is set up using a first-order Lagrange shape function representation to control the material density in your model, the solution contains the sensitivity of the objective function with respect to the discrete density value at each node point in the mesh. Because each node influences the density in a small surrounding region, the size of which varies from node to node, the individual sensitivities are not directly comparable to each other.

Displaying such domain control variables results in a plot that is not smooth due to the varying element size. It must therefore not be used to draw any conclusions about the physics and the effect of changing the physical field represented by the control variable. Some insight may, however, be gained by looking at the sensitivities divided by the mesh volume scale factor `dvol`. This makes the sensitivities in the plot comparable between different parts of the surface but still not mathematically well defined. In particular, using discontinuous constant shape functions together with the division by `dvol` results in a plot that is proportional to the true pointwise sensitivity.

Note

If the plan is to use the sensitivities in an automatic optimization procedure, as is done through the Optimization interface available with the optional Optimization Module, the discrete nature of the sensitivities causes no additional complication. The optimization solver searches for optimum values of the discrete control variables using the discrete gradient provided by the sensitivity analysis.

GEOMETRICAL SENSITIVITY

Use the control variables directly to parameterize any aspect of the physics which is controlled by an expression. This applies for example to material properties, boundary conditions, loads and sources. However, the shape, size, and position of parts of the geometry cannot be changed as easily at solution time, and therefore require special attention.

Control variables cannot be used directly in the geometry description. Instead, the model must be set up using *ALE* and tie all physics to an *ALE frame* controlled by a Deformed Geometry physics interface (in 2D only) or a Moving Mesh (ALE) physics interface. Then use control variables to control the mesh movement, effectively parameterizing the geometry.

Note

See The Deformed Geometry and Moving Mesh Interfaces for details about these physics interfaces and ALE in general.

Issues to Consider Regarding the Objective Function

THE PRINCIPLE OF VIRTUAL WORK

Potential energy has a special status among scalar objective functions, because its derivatives with respect to scalar control variables can in many cases be interpreted as (true or generalized) forces.

COMPLEX-VALUED OBJECTIVE FUNCTIONS

Sensitivity analysis can be applied only when the objective function is a real-valued differentiable function of the control variables. This is usually not a very severe constraint, even for frequency-domain models where the PDE solution variables are complex valued. The reason is that physical quantities of interest to the analyst are always real valued, and if complex-valued control variables are required, it is possible to treat the real and imaginary parts separately.

Many common quantities of interest are time averages which can be written in the form $Q = \mathrm{real}(a \cdot \mathrm{conj}(b))$, where a and b are complex-valued linear functions of the solution variables and therefore implicit functions of the control variables. The problem with this expression is that while Q is indeed a real-valued differentiable function of the control variables, it is not an analytical function of a and b. This complicates matters slightly because the sensitivity solver relies on partial differentiation and the chain rule.

While the partial derivatives of Q with respect to a and b are, strictly speaking, undefined, it can be proven that if they are chosen such that

$$Q(a + \delta a, b + \delta b) \approx Q(a, b) + \mathrm{real}\left(\frac{\partial Q}{\partial a}\delta a + \frac{\partial Q}{\partial b}\delta b\right) \qquad (16\text{-}2)$$

for any small complex increments δa and δb, the final sensitivities are evaluated correctly. The special function `realdot(a,b)` is identical to `real(a*conj(b))` when evaluated but implements partial derivatives according to Equation 16-2. For that reason, use it in the definition of any time-average quantity set as objective function in a sensitivity analysis.

The Sensitivity Interface

The **Sensitivity** interface ( ), found under the **Mathematics>Optimization and Sensitivity** ( ) branch in the **Model Wizard**, has the tools for defining and solving sensitivity problems.

See Also

For a more extensive introduction to the mathematics implemented by this interface, see the Theory for the Sensitivity Interface.

The **Sensitivity** interface includes the tools for defining and solving sensitivity problems. The main purpose of the interface is its ability to solve stationary PDE-constrained sensitivity problems.

The objective functions and constraints are defined in terms of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model) and restrict these to specific domains or make them globally available. This flexibility is reflected in the user interface by grouping these settings according to the dimension of the domain to which they apply. In such a group of settings, the following settings can be specified, to each of which corresponds a separate feature and its settings window:

- Integral Objective
- Probe Objective
- Control Variable Field

The main Sensitivity interface node's settings window contains the following section:

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `sens`.

See Also

- Global Objective
- Global Control Variables
- Adding a Sensitivity Interface to a Model

Integral Objective

An **Integral Objective** is defined as the integral of a closed form expression of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model) that are either global or available in the domain in question. Hence, its definition is restricted to a set of geometric entities of the same dimension. For integral objectives on points, the integration reduces to a summation.

DOMAIN, EDGE, BOUNDARY, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) used in the integration for the integral objective.

OBJECTIVE

Enter an **Objective expression** that is integrated over the geometric entity level in the integral objective.

PAIR SELECTION

If **Integral Objective** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

QUADRATURE SETTINGS

Specify the settings for the **Quadrature** used to numerically evaluate the integral in the integral objective: the integration order (default: 4) in the **Integration order** field and the frame to integrate on (default: the spatial frame), which is selected from the **Integrate on frame** list.

Probe Objective

A **Probe Objective** is defined as a point evaluation of a closed form expression of control and solution variables (the latter are given as the solution to the differential equations defined by the multiphysics model) that are either global or available in the domain in question. The point used for the point evaluation has to be contained in the domain.

DOMAIN SELECTION

From the **Selection** list, choose the domain containing the point used for the point evaluation.

OBJECTIVE

Enter an **Objective expression** that is evaluated at the point in the domain.

PROBE COORDINATES

Specify the **Probe coordinates** for the point in the domain where the expression for the objective is evaluated. After specifying the probe coordinates, select an option from the **Evaluate in frame**—**Spatial** (the default), **Material**, or **Mesh**.

Control Variable Field

Specify the **Control Variable Field** specific to the geometric entity level (domain, edge, boundary, or point) in question.

DOMAIN, EDGE, BOUNDARY, OR POINT SELECTION

From the **Selection** list, choose the geometric entity (domains, boundaries, edges, or points) where the control variable field is defined.

PAIR SELECTION

If **Control Variable Field** is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

See Also

- Identity and Contact Pairs
- Specifying Boundary Conditions for Identity Pairs

CONTROL VARIABLE

Enter a **Control variable name** and **Initial value**.

DISCRETIZATION

This section contains settings for the element used to discretize the control variable. Select a **Shape function type**: **Lagrange** (the default) or **Discontinuous Lagrange**. Also select an **Element order**: **Linear**, **Quadratic** (the default), **Cubic**, **Quartic**, or **Quintic**.

Global Objective

Specify the **Global Objective** contribution to the function.

OBJECTIVE

Enter an **Objective expression** that defines the contribution to the objective function. It can be an expression of those components of the control and solution variable (the solution variable is given as the solution to the differential equations defined by the multiphysics model) that are globally available.

Global Control Variables

Specify any globally available control variables.

CONTROL VARIABLES

In the table, enter **Variable** names and **Initial values** of the control variables that are globally available.

To add a control variable to the table, click the **Add** button (✚). To remove a control variable and its values from the table, click the **Delete** button (✖).

Adding a Sensitivity Interface to a Model

You can add a **Sensitivity** interface when creating a new model or at any time during modeling. For a new model, select physics interfaces as the second step in the **Model Wizard** (after specifying the space dimension). In an active model, right-click the model node in the **Model Builder** and select **Add Physics**. In both cases, the **Add Physics** page appears with a list of interfaces.

1 Expand the **Mathematics>Optimization and Sensitivity** node in the list of physics interfaces.

2 Select **Sensitivity** (▮).

3 Click the **Add Selected** button (✚) underneath the list to add the selected physics interface to the model. The physics interface then appears in the list under **Selected physics**.

4 When you are ready click the **Next** button (⇨) in the upper-right corner of the **Model Wizard** window.

5 Optionally, choose a study type for the sensitivity analysis on the **Select Study Type** page.

6 Click the **Finish** button (🏁) in the upper-right corner of the **Model Wizard** window.

You also need to add a **Sensitivity** subnode to the **Solver** node in the solver configuration to fully define the sensitivity analysis.

17

Multiphysics Modeling

This chapter discusses Creating Multiphysics Models using COMSOL Multiphysics for multiphysics modeling and coupled-field analyses. It describes various ways to build multiphysics models and includes an overview of the available predefined multiphysics interfaces in COMSOL Multiphysics and the add-on modules.

Creating Multiphysics Models

In this section:

- Multiphysics Modeling Approaches
- Using Predefined Multiphysics Interfaces
- Acoustics
- Fluid Flow
- Heat Transfer
- Plasma
- Structural Mechanics
- Adding Physics Sequentially
- Building a Multiphysics Model Directly
- Adding Multiphysics Couplings
- Deleting Physics Interfaces

Multiphysics Modeling Approaches

The ability to create multiphysics models—those with more than one type of physics or equation such as coupled-field problems—is one of COMSOL Multiphysics' most powerful capabilities. In such a model, the software can solve all the equations, taken from various areas of physics, as one fully coupled system.

Within COMSOL Multiphysics you can choose from several ways to approach multiphysics modeling and coupled-field analysis.

Model

See Marangoni Convection for an example of combining the Laminar Flow and Heat Transfer in Fluids interfaces (Model Library path **COMSOL_Multiphysics/Multiphysics/marangoni_convection**).

Using Predefined Multiphysics Interfaces

COMSOL Multiphysics and the add-on modules provide a number of predefined multiphysics interfaces. They appear in the **Model Wizard**'s list of physics interfaces, in folders with the names in the Category/Folder column in Table 17-1. The multiphysics interfaces includes the functionality of two or more physics interfaces with additional features for the multiphysics couplings. Default settings provide the typical field couplings for the multiphysics application. The available multiphysics couplings vary depending on the installed modules (see Table 17-1 for details).

TABLE 17-1: PREDEFINED MULTIPHYSICS COUPLINGS

CATEGORY/ FOLDER	NAME	SPACE DIMENSION	REQUIRED MODULES	OPTIONAL MODULES
Acoustics	Acoustic-Solid Interaction	2D, 2D Axi, 3D	Acoustics	Structural Mechanics, MEMS
Acoustics	Acoustic-Shell Interaction	2D and 3D	Acoustics and Structural Mechanics	
Acoustics	Acoustic-Piezoelectric Interaction	2D, 2D Axi, 3D	Acoustics	Structural Mechanics, MEMS
Acoustics	Elastic Waves and Poroelastic Waves	2D, 2D Axi, 3D	Acoustics	
Acoustics	Aeroacoustics with Flow	All	Acoustics	
Fluid Flow	Fluid-Structure Interaction	2D, 2D Axi, 3D	Structural Mechanics or MEMS	CFD
Fluid Flow	Non-Isothermal Flow/ Conjugate Heat Transfer	2D, 2D Axi, 3D	CFD or Heat Transfer	
Fluid Flow	Free and Porous Media Flow	2D, 2D Axi, 3D	Batteries & Fuel Cells, CFD, Chemical Reaction Engineering, Subsurface Flow, or Microfluidics	
Heat Transfer	Joule Heating	2D, 2D Axi, 3D	None	AC/DC, Heat Transfer, MEMS
Heat Transfer	Induction Heating	2D, 2D Axi, 3D	AC/DC	Heat Transfer
Heat Transfer	Microwave Heating	2D, 2D Axi, 3D	RF	Heat Transfer
Heat Transfer	Heat Transfer with Surface-to-Surface Radiation	2D and 3D	Heat Transfer	

TABLE 17-1: PREDEFINED MULTIPHYSICS COUPLINGS

CATEGORY/ FOLDER	NAME	SPACE DIMENSION	REQUIRED MODULES	OPTIONAL MODULES
Heat Transfer	Heat Transfer with Radiation in Participating Media	2D and 3D	Heat Transfer	
Plasma	Capacitively Coupled Plasma	All	Plasma	
Plasma	DC Discharge	All	Plasma	
Plasma	Inductively Coupled Plasma	2D, 2D Axi, 3D	Plasma, AC/DC	
Plasma	Microwave Plasma	2D, 2D Axi, 3D	Plasma, RF	
Structural Mechanics	Electromechanics	2D, 2D Axi, 3D	MEMS	AC/DC
Structural Mechanics	Thermal Stress	2D, 2D Axi, 3D	Structural Mechanics or MEMS	Heat Transfer
Structural Mechanics	Joule Heating and Thermal Expansion	2D, 2D Axi, 3D	Structural Mechanics or MEMS	Heat Transfer, AC/DC
Structural Mechanics	Piezoelectric Devices	2D, 2D Axi, 3D	Structural Mechanics, MEMS, or Acoustics	
Structural Mechanics	Poroelasticity	2D, 2D Axi, 3D	Subsurface Flow	

The predefined multiphysics interfaces include all features in the optional modules if your license includes them; otherwise they include the corresponding but more limited physics interface in COMSOL Multiphysics or the required module. The couplings are identical in both cases, but the optional modules typically offer additional functionality in other areas. These predefined multiphysics interfaces are quick entry points for common multiphysics applications. It is possible to create the same couplings using any of the other methods for multiphysics modeling, and you can continue to add, modify, disable, and remove physics interfaces in a model that you start using one of the predefined multiphysics interfaces. If you want to add additional physics interfaces in the **Model Builder**, right-click the model node, choose **Add Physics** and then use the **Model Wizard** that opens.

Acoustics

The **Acoustics** branch ( ) contains a variety of multiphysics interfaces all requiring the Acoustics Module. For more information see the *Acoustics Module User's Guide*.

Tip

For more information all the available interfaces with the Acoustics Module, see the specification chart on the COMSOL website: www.comsol.com/products/specifications/acoustics.

ACOUSTIC-SOLID INTERACTION

The **Acoustic-Solid Interaction, Frequency Domain** interface ( ) for acoustic-structural interaction is available with the Acoustics Module. This multiphysics interface combines pressure acoustics and solid mechanics and connects the acoustics pressure in a fluid domain with the structural deformation in a solid domain. Special interface conditions define the fluid loads on the solid domain and the structural acceleration's effect on the fluid. There is also a **Acoustic-Solid Interaction, Transient** ( ) version of the interface available

ACOUSTIC-SHELL INTERACTION

The **Acoustic-Shell Interaction, Frequency Domain** ( ) interface for acoustic-structural interaction also requires the Structural Mechanics Module license. This multiphysics interface combines Pressure Acoustics and the Shell interface to connect the pressure acoustics in a fluid domain with the structural deformation of shell boundary. There is also a **Acoustic-Shell Interaction, Transient** ( ) version of the interface available.

ACOUSTIC-PIEZOELECTRIC INTERACTION

The **Acoustic-Piezoelectric Interaction, Frequency Domain** ( ) interface for acoustic-piezoelectric interaction in available with the Acoustics Module. This multiphysics interface combine Pressure Acoustics and the Piezoelectric Devices interface to connect the pressure acoustics in a fluid domain with a piezoelectric solid. There is also a **Acoustic-Piezoelectric Interaction, Transient** ( ) version of the interface available.

ELASTIC WAVES AND POROELASTIC WAVES

The **Elastic Waves** multiphysics interface ( ) combines Pressure Acoustics and Solid Mechanics to connect the fluid pressure with the structural deformation in solids. The **Poroelastic Waves** ( ) version of the interface models linear elastic waves in porous elastic materials damped by a pore fluid.

AEROACOUSTICS WITH FLOW

The **Aeroacoustics with Flow, Frequency Domain** interface ( ) combines Aeroacoustics and Compressible Potential Flow and connects the result from compressible potential flow simulation with the Aeroacoustic model. It is where calculations of density and mean flow velocity potential in an ideal barotropic, irrotational fluid at constant entropy is combined with aeroacoustic modeling. There is also a **Aeroacoustics with Flow, Transient** ( ) version of the interface available.

Fluid Flow

Tip	For more information about all the available fluid flow interfaces, see the specification chart on the COMSOL website: www.comsol.com/products/specifications/fluid-flow

Various subbranches of the **Fluid Flow** branch ( ) have multiphysics interfaces available.

FLUID-STRUCTURE INTERACTION (FSI)

The **Fluid-Structure Interaction** interface ( ) for fluid-structure interaction (FSI) combines fluid flow with structural mechanics, using a moving mesh to capture the fluid movement. The interface provides a predefined interface condition for the solid-fluid boundary. For more information and example models, see the *Structural Mechanics Module User's Guide* or the *MEMS Module User's Guide*.

FLOW WITH VARIABLE DENSITY—NON-ISOTHERMAL FLOW

The CFD Module and the Heat Transfer Module include the laminar and turbulent Non-Isothermal Flow/Conjugate Heat Transfer predefined multiphysics interface. In the CFD Module you select both laminar and turbulent non-isothermal flow in the **Non-Isothermal Flow** folder ( ); in the Heat Transfer Module, you select a laminar or turbulent non-isothermal flow in the **Conjugate Heat Transfer** folder ( ). For more information about these predefined couplings, see the *CFD Module User's Guide* or the *Heat Transfer Module User's Guide*.

FREE AND POROUS MEDIA FLOW

The **Free and Porous Media Flow** interface ( ) combines the Laminar Flow interface and the Brinkman Equations interface to model fluid flow in systems with both free and porous media flow and solving for the velocity field and pressure.

Tip

For more information about all the available heat transfer interfaces, see the specification chart on the COMSOL website:

www.comsol.com/products/specifications/heat-transfer

The multiphysics interfaces found under the **Heat Transfer>Electromagnetic Heating** branch ( ) couple electromagnetic fields and heat transfer:

There is also the Heat Transfer with Radiation in Participating Media multiphysics interface available under the **Heat Transfer>Radiation** branch ( ), and the **Conjugate Heat Transfer** interfaces (described in Fluid Flow).

JOULE HEATING

The Joule Heating Interface ( ), combines all feature from the Electric Currents interface with the Heat Transfer interface for modeling of *Joule heating* (*resistive heating* or *ohmic heating*). The interaction is coupled in both directions:

- The resistive heating appears as a heat source in the default **Electromagnetic Heat Source** node.
- The default setting is to use the value for the electrical conductivity σ from the material. By selecting **Linearized resistivity** from the σ list, the following temperature-dependent expression describes the electrical conductivity: $1/(\rho_0(1+\alpha(T-T_{\mathrm{ref}})))$, where T is the dependent variable for temperature from the heat transfer part, which automatically appears as a model input. By default, the values for ρ_0 (resistivity at reference temperature), α (temperature coefficient), and T_{ref} (reference temperature) are taken from the material. These settings are built into the Joule Heating Model node, which is the central feature in the Joule Heating interface.

You can use a stationary or time-dependent study.

Model

Joule Heating of a Microactuator: Model Library path **COMSOL_Multiphysics/Multiphysics/thermal_actuator_jh**

INDUCTION HEATING

The **Induction Heating** interface ( ) combines the features of the Induction Currents interface with those of the Heat Transfer interface. The predefined interaction adds the resistive heating from the induction currents as a heat source.

For more information, see the *AC/DC Module User's Guide*.

MICROWAVE HEATING

The **Microwave Heatin**g interface ( ) combines the features of an Electromagnetic Waves interface from the RF Module with those of the Heat Transfer interface. The predefined interaction adds the resistive heating from the electromagnetic waves as a heat source.

For more information, see the *RF Module User's Guide*.

HEAT TRANSFER WITH SURFACE-TO-SURFACE RADIATION

The **Heat Transfer with Surface-to-Surface Radiation** interface ( ) combines the Surface-to-Surface Radiation and Heat Transfer interfaces to model surface-to-surface radiation combined with heat transfer in solids and fluids. Solves for radiative intensity and temperature fields.

HEAT TRANSFER WITH RADIATION IN PARTICIPATING MEDIA

The **Heat Transfer with Radiation in Participating Media** interface ( ) combines the Radiation in Participating Media and Heat Transfer interfaces to model radiative heat transfer inside a participating medium combined with heat transfer in solids and fluids. Solves for radiative intensity and temperature fields.

Plasma

The multiphysics interfaces found under the **Plasma** branch ( ) all require the Plasma Module. For more information see the *Plasma Module User's Guide*.

Tip

For more information all the available interfaces with the Plasma Module, see the specification chart on the COMSOL website: www.comsol.com/products/specifications/plasma.

CAPACITIVELY COUPLED PLASMA

The **Capacitively Coupled Plasma** ( ) couples the Drift Diffusion, Heavy Species Transport, and Electrostatics interfaces into an integrated multiphysics interface to study discharges that are sustained by a time-varying electrostatic field, solving for the electric potential and electron density.

DC DISCHARGE

The **DC Discharge** ( ) interface couples the Drift Diffusion, Heavy Species Transport, and Electrostatics interfaces. Used to study discharges sustained by a static electric field and solving for the electric potential and electron density. Townsend coefficients can be used instead of rate coefficients for electron source terms.

INDUCTIVELY COUPLED PLASMA

The **Inductively Coupled Plasma** ( ) couples the Drift Diffusion, Heavy Species Transport, Electrostatics, and Magnetic Fields interfaces, and used to study discharges sustained by induction currents and solving for the magnetic vector potential, electric potential, and electron density. The induction currents are solved for in the frequency domain. This interface also requires the AC/DC Module.

MICROWAVE PLASMA

The **Microwave Plasma** ( ) interface couples the Drift Diffusion, Heavy Species Transport, Electrostatics and Electromagnetic Waves interfaces and studies discharges sustained by electromagnetic waves (wave-heated discharge). The electromagnetic waves are solved for in the frequency domain and solving for the electric field, electric potential, and electron density. This interface also requires the RF Module.

Structural Mechanics

Tip	For more information about all the available structural mechanics interfaces, see the specification chart on the COMSOL website: www.comsol.com/products/specifications/structural-mechanics

The **Structural Mechanics** branch ( ) has a variety of multiphysics interfaces, which all require additional module licenses.

THERMAL STRESS

The **Thermal Stress** interface ( ) combines solid mechanics with heat transfer for modeling of thermal expansion. It combines the features of a Solid Mechanics interface with those of a Heat Transfer interface.

The interaction is a one-way coupling using the temperature from the heat transfer interface to define the strain temperature.

You can use a steady-state or time-dependent study. For the time-dependent study, the software by default treats the structural part as static (quasi-static transient behavior).

By default, COMSOL Multiphysics solves for the temperature and displacements simultaneously. For large problems, you can take advantage of the one-way coupling and solve the problem sequentially (unless there are thermal properties that depend on the displacements): first solve for temperature and then perform the stress-strain analysis using the computed temperature field from the heat transfer equation using a segregated solution scheme.

ELECTROMECHANICS

The **Electromechanics** interface ( ) combines solid mechanics and electrostatics with a moving mesh to model the deformation of electrostatically actuated structures. It is also compatible with piezoelectric materials. This interface requires the MEMS Module. It combines the features of an Electrostatics interface and a Moving Mesh interface with those of a Solid Mechanics interface from the MEMS Module.

The interactions include the force contribution that results from the electromechanical (or Maxwell) stress tensor in the adjacent free space domains, which is added as a surface force at the external interfaces of an electromechanical solid.

You can select from several study types, including frequency domain, stationary, and time dependent. By default, COMSOL Multiphysics solves for the electric potential and displacements simultaneously.

JOULE HEATING AND THERMAL EXPANSION

The **Joule Heating and Thermal Expansion** interface ( ) combines Joule heating and thermal expansion for modeling, for example, thermoelectromechanical (TEM) effects. This interface requires the Structural Mechanics Module or the MEMS Module. It combines the features of a Heat Transfer interface and an Electric Currents interface with those of a Solid Mechanics interface from the Structural Mechanics Module or the MEMS Module.

The interactions include Joule heating, which is fully coupled in both directions (see Joule Heating), and thermal expansion, which is a one-way coupling (see Thermal Stress).

You can select a steady-state or transient analysis (where the conductive media and solid mechanics equations are stationary). By default, COMSOL Multiphysics solves for the temperature, electric potential, and displacements simultaneously.

PIEZOELECTRIC DEVICES

The **Piezoelectric Devices** interface ( ) combines Solid Mechanics and Electrostatics for modeling on piezoelectric devices where all or some of the domains contain a piezoelectric material. The piezoelectric coupling can be on stress-charge or strain-charge form. You also have access to all solid mechanics and electrostatics functionality.

POROELASTICITY

The **Poroelasticity** interface ( ) is available in the Subsurface Flow Module and combines the Darcy's Law physics interface from that module with a structural mechanics physics interface from the Structural Mechanics Module, which is required in addition to the Subsurface Flow Module. The Poroelasticity predefined multiphysics coupling is useful for modeling poroelastic phenomena such as diffusion in elastic solids or elasticity of fluid-infiltrated porous solids. For more information, see the *Subsurface Flow Module User's Guide* or the *Geomechanics Module User's Guide*.

Adding Physics Sequentially

With this approach you can verify that each type of physics or equation gives the expected results before adding more complexity to the model by adding another physics or coupling fields.

Use these steps to add one physics at the time:

1. Start a new model using the **Model Wizard**.
2. Draw the geometry.
3. Define the physics settings.
4. Solve and visualize the results.

   Then add another physics interface:

5. Right-click the model node in the **Model Builder** and choose **Add Physics**.
6. Add one or more new physics interfaces using the **Model Wizard**.
7. On the **Select Study Type** page, you can add a new study or replace the current study. Select (the default) or clear the **Deactivate added physics in other studies** check box

to control whether the new physics that you add should be active in other studies or not.

8. Add and modify feature nodes in the physics interfaces, including couplings between the physics.

Building a Multiphysics Model Directly

Another approach is to include multiple physics interfaces when you start creating the model in the Model Wizard:

1. On the **Add Physics** page, select an physics interface for the model.
2. Click the **Add Selected** button (✚) underneath the list of available physics interfaces. The physics interface then appears in the list under **Selected Physics**.
3. If needed, the names of the dependent variables can be changed. For some physics interfaces and mathematical interfaces you can also adjust the number of dependent variables.
4. Continue selecting physics interfaces and adding them to the list under **Selected Physics** by clicking the **Add Selected** button.
5. When done, continue in the Model Wizard by clicking the **Next** button (⇨) at the top of the **Add Physics** page.

Adding Multiphysics Couplings

Model inputs appear in the equation model node's **Model Inputs** section. Model inputs are fields such as temperature and velocities that act as inputs for material models and model equations. They appear in the **Model Inputs** section if a material is defined so that a material property becomes a function of the temperature, for example. The program automatically connects the model input to an existing field (dependent variable) within the physics interface (but not to available fields in other physics interfaces).

With more than one physics interface in the model, coupling of the fields is easy: all applicable fields that can serve as inputs in another physics interface automatically appears in the other physics interface's settings window's **Model Inputs** section. For example, with a Heat Transfer (ht) interface for heat transfer in fluids and a Laminar Flow (spf) interface, you can select **Velocity field (spf/fp1)**, which the **Fluid Properties 1 {fp1}** node in the **Laminar Flow** branch defines, from the **Velocity field** list in the **Model Inputs** section of the **Heat Transfer in Fluids** node's settings window. For a coupling in the other direction (if you use temperature-dependent fluid properties, for example), you can select **Temperature (ht/fluid1)**, which the **Heat Transfer in Fluids 1 {fluid1}** node in the **Heat Transfer** branch defines, from the **Temperature** list in the **Model Inputs** section of the **Fluid Properties** node's settings window.

A list in the **Model Inputs** section becomes grayed out if the physics interface itself defines the field because it is then automatically connected to that field. For example, with a Heat Transfer (ht) interface for heat transfer in fluids the **Temperature** list is grayed out in the **Heat Transfer in Fluids 1 {fluid1}** node. This automatic connection selects the **Temperature (ht/fluid1)** field because it is defined by the Heat Transfer interface. As long as the list is grayed out, you cannot change it. If you want to use another temperature field or an expression, you first make the list editable by clicking the **Make All Model Inputs Editable** button ().

In the case that you want to use another expression for these model inputs, select **User defined** to enter a user-defined value or expression in the associated text field.

Deleting Physics Interfaces

To delete a physics interface from a model:

1. Select the main feature node for the physics interface under the model node in the **Model Builder**.
2. Right-click and choose **Delete** from the context menu or press the Delete key.
3. Click **OK**.

18

The Deformed Geometry and Moving Mesh Interfaces

This chapter explains how to use the modeling interfaces that control mesh deformation and are found under the **Mathematics>Deformed Mesh** branch ( ) in the **Model Wizard**. It also contains fundamentals about deformed meshes and information about the Eulerian and Lagrangian formulations of the physics, the frame types that support these formulations, and the arbitrary Lagrangian-Eulerian (ALE) method.

In this chapter:

- Deformed Mesh Fundamentals
- The Moving Mesh Interface
- The Deformed Geometry Interface

Deformed Mesh Fundamentals

In this section:

About Deformed Meshes

A deformed mesh can be useful if the boundaries of your computational domain are moving in time or deform as a function of some parameter. The deformation can also be physics induced—for example, depend on computed velocities or solid deformation. The point is that a new mesh need not be generated for each configuration of the boundaries—instead, the software perturbs the mesh nodes so they conform with the moved boundaries.

In COMSOL Multiphysics, control the movement of the interior nodes in these ways:

- By propagating the moving boundary displacement throughout the domain to obtain a smooth mesh deformation everywhere. This is done by solving PDEs for the mesh displacements (a Laplace, Winslow, or hyperelastic smoothing PDE, or one borrowed from continuum mechanics) with boundary conditions given by the movement of the boundaries.

- By specifying an explicit formula for the mesh deformation. The formula can make use of other dependent variables, such as the displacement components of structural mechanics.
- By leaving the control of the mesh displacement to a Solid Mechanics interface, which has built-in deformed mesh functionality, or to a multiphysics interface of which Solid Mechanics is part.

Deformed Geometry vs. Moving Mesh

There are two physics interfaces implementing different types of deformed meshes: the Deformed Geometry interface ( ) and the Moving Mesh interface ( ). Both are found under the **Mathematics>Deformed Mesh** branch ( ) in the **Model Wizard**.

In the Deformed Geometry interface, the material does not follow the change in shape. Deformation of the geometry boundaries therefore corresponds to addition or removal of material. In the Moving Mesh interface, solid materials follow the mesh deformation and deform in the same way as the mesh. Fluids and gases, on the other hand, are added or removed so as to always fill the current shape of the domain—any effects of compression or expansion must be introduced explicitly into the equations.

- Use a Deformed Geometry interface to study the behavior of different shapes of an original object. In a Deformed Geometry interface the material never follows a perturbation of the shape. The total mass of the first shape is not the same as the mass for the second, perturbed, geometry. Any deformation can be regarded as removal or addition of material.
- Use a Moving Mesh interface to study how a solid object deforms as the results of physical load, and how fluids in adjacent domains react to displacement of the domain boundaries—for example, how a tank impeller moves a fluid, or how a MEMS switch moves under the influence of an electric field. Using the Moving Mesh interface, a solid material follows the mesh deformation. A movement of a boundary can therefore be regarded as bending or punching the original object. Undeformed and deformed solid objects have the same mass, but the total amount of fluid in an a domain whose boundaries deform may change.

Arbitrary Lagrangian-Eulerian Formulation (ALE)

The partial differential equations of physics are usually formulated either in a *spatial* coordinate system, with coordinate axes fixed in space, or in a *material* coordinate system, fixed to the material in its reference configuration and following the material as it deforms. The former is often referred to as an *Eulerian* formulation, while the latter is a *Lagrangian* formulation.

Structural mechanics and other fields of physics dealing with a possibly anisotropic, solid, material are most conveniently simulated using material coordinates. The Lagrangian formulation makes the anisotropic material properties independent of the current spatial orientation of the material.

If, on the other hand, the focus is on simulating the physical state at fixed points in space, an Eulerian formulation is usually more convenient. In particular, when liquids and gases are involved, it is often unreasonable to follow the state of individual material particles. Rather, the quantities of interest are pressure, temperature, concentration, etc., at fixed positions in space.

An inherent problem with the pure Eulerian formulation is that it cannot handle moving domain boundaries, since physical quantities are referred to fixed points in space, while the set of spatial points currently inside the domain boundaries changes with time. Therefore, to allow moving boundaries, the Eulerian equations must be rewritten so as to describe all physical quantities as functions of some coordinate system in which the domain boundaries are fixed. The finite element mesh offers one such system: the *mesh* coordinates.

In the mesh coordinate system, the domain is fixed, and there is a one-to-one map from the mesh coordinates to the current spatial configuration of the domain. Otherwise, the mesh coordinate system can be defined freely and separately from both the spatial and material systems. The natural choice is to let the mesh coordinate system, at least initially, coincide with the *geometry* coordinates. This follows immediately from the way meshes are created, and means that points in the domain are identified by their position in the original geometry.

As the domain—and mesh—deforms, the map from mesh coordinates to spatial coordinates can become increasingly ill-conditioned. Before the degradation of the mesh mapping goes too far you can, through a remeshing operation, stop the simulation, create a new mesh in the current configuration of the domain, and map all quantities to the new mesh. When the simulation is restarted, points in the domain are internally identified by their new mesh coordinates, which coincide with the spatial coordinates at the state where the simulation was stopped. Therefore, the geometry and mesh coordinates of a given point will differ after remeshing the deformed geometry.

Rewriting physics equations in this way, on a freely moving mesh, results in an *arbitrary Lagrangian-Eulerian* (ALE) method. In the special case when the map from mesh coordinates to spatial coordinates follows the material deformation, a Lagrangian method is recovered. Similarly, when the map is an identity map, the ALE method becomes entirely Eulerian.

The ALE method is therefore an intermediate between the Lagrangian and Eulerian methods, and it combines the best features of both—it allows moving boundaries without the need for the mesh movement to follow the material.

Frames

COMSOL Multiphysics refers to the spatial, material/reference, geometry and mesh coordinate systems described above as *spatial frame*, *material frame* (reference frame), *geometry frame*, and *mesh frame*, respectively. Physics can be formulated on the spatial frame or on the material frame, depending on whether it is more convenient to interpret the equations as Eulerian or Lagrangian, respectively. It is not possible to use the geometry and mesh frames and their associated coordinates to formulate physics because they are neither connected to the material nor to the true Euclidean space.

Conceptually, all four frames always exist, but all or some of them may point to the same actual coordinate system. It is the actual coordinate system that decides the names of the independent variables (the coordinate names like `x`, `y`, `z` or `r`, `phi`, `z`). Before adding a Moving Mesh or Deformed Geometry interface to a model, all four frames coincide and use the spatial coordinate names. Also all interfaces based on solid mechanics include moving mesh functionality and by default behave much in the same way as a Moving Mesh interface.

When a Moving Mesh or Solid Mechanics interface is added, the spatial frame is separated from the material frame, which is given a new set of independent variable names (by default capital `X`, `Y`, `Z` or `R`, `PHI`, `Z`). From this point, Eulerian and Lagrangian formulations behave differently since they, among other things, define derivatives with respect to different sets of independent variables.

The geometry frame and the material frame share coordinate system until a Deformed Geometry interface is added. At that point, a new geometry coordinate system is created and given a new set of independent variable names (by default `Xg`, `Yg`, `Zg` or `Rg`, `PHIg`, `Zg`). The new geometry frame refers to the geometry as it is represented by the Geometry Sequence. By inserting a nontrivial transformation from geometry coordinates to material coordinates, the shape of the geometry can be effectively changed without having to create a new mesh. This can in some situations be useful as a means of parameterizing the geometry, for example before performing optimization or sensitivity analysis.

Using Deformed Geometry affects both Eulerian and Lagrangian physics in the same way. The reason is that the Deformed Geometry interface controls the material frame in relation to the geometry frame. Unless there is also a Moving Mesh or Solid Mechanics interface present, the material frame and the spatial frame still refer to the same coordinate system. The three frames refer to three different sets of coordinates only when there is both a Deformed Geometry and some Moving Mesh interface active in the model.

The geometry frame and the mesh frame coincide until a manual or automatic remeshing operation is performed. At that point, a new mesh is created in the original geometry together with a new set of coordinates (independent variable typically `Xm`, `Ym`, `Zm` or `Rm`, `PHIm`, `Zm`). The original geometry coordinates are mapped and stored together with the new mesh such that any Deformed Geometry interface can still define the material frame relative to the original geometry frame.

To avoid confusion, note that:

- The *spatial frame* is the usual, fixed, global, Euclidean coordinate system with the *spatial coordinates* (x, y). In the ALE context, the spatial coordinate system as such is fixed while the spatial coordinates (x, y) of each material point and mesh node can be functions of time. Therefore, it is correct to refer to the model as having a *moving mesh*.
- The *material frame* is a coordinate system which identifies material points by their spatial coordinates (X, Y) in some—actual or imagined—reference configuration. Think of the material coordinate system as having been printed on the material in the reference configuration such that it follows it during deformation. It is therefore in general curvilinear and cannot be used directly to measure true distances and angles. See also Figure 18-1 and Figure 18-2
- The *geometry frame* is a coordinate system which identifies points by their spatial coordinates (X_g, Y_g) in the original geometry. It is often natural to use the original geometry also as reference state to define material coordinates. Therefore, the geometry frame and material frame usually coincide. The only exception is when a Deformed Geometry interface is used to deform or parameterize the original geometry.
- The *mesh frame* is a coordinate system used internally by the finite element method. It identifies mesh points by their spatial coordinates (X_m, Y_m) at the time the mesh was created. The original mesh is always created based on the original geometry.

Therefore, the mesh frame coincides with the geometry frame until a new mesh is created in the—then current—deformed configuration.

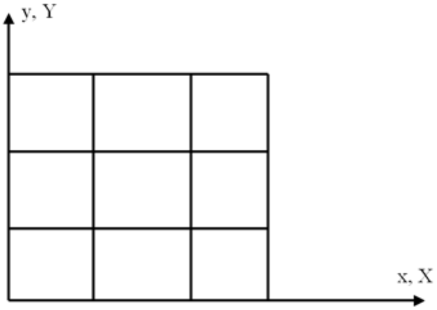


Figure 18-1: An undeformed mesh. In the reference configuration, which may be the actual configuration at a reference time or a hypothetical state, the spatial frame (x, y) and the material frame (X, Υ) coincide.

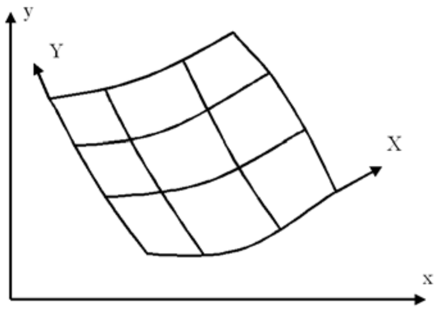


Figure 18-2: After deformation of the material, the spatial frame (x, y) remains the same, while the material coordinate system (X, Υ) has been deformed, following the material. Meanwhile, the material coordinates of each material point remains the same, but its spatial coordinates have changed.

Mathematical Description of the Mesh Movement

Though moving meshes are also possible in 3D, consider a 2D geometry for simplicity, where the spatial and material frame coordinates are called (x, y) and (X, Y), respectively. Let (X_0, Y_0) be the spatial coordinates of a mesh node in the initial material configuration. The spatial coordinates (x_0, y_0) of the same mesh node at some other time, t, are then given by the functions

$$x_0 = x(X_0, Y_0, t), \qquad y_0 = y(X_0, Y_0, t) \tag{18-1}$$

These functions can be either explicit transformations (expressions) or be the solution to a mesh smoothing equation. The mesh node's material coordinates (X_0, Y_0) may in turn bee seen as functions of an underlying system of geometry coordinates (X_g, Y_g) and a parameter, p, such that

$$X_0 = X(X_g, Y_g, p), \qquad Y_0 = Y(X_g, Y_g, p) \tag{18-2}$$

with similar options for the transformations. Obviously, the transformations can also be chained such that (x_0, y_0) are seen as functions of (X_g, Y_g), t, and p.

Introducing a vector notation for the coordinates:

- Spatial coordinates $\mathbf{x} = [x, y, z]$
- Material coordinates $\mathbf{X} = [X, Y, Z]$
- Geometry coordinates $\mathbf{X}_g = [X_g, Y_g, Z_g]$
- Mesh coordinates $\mathbf{X}_m = [X_m, Y_m, Z_m]$

the general relation between the frames can be written

$$\mathbf{x} = \mathbf{f}(\mathbf{X}, t) = \mathbf{f}(\mathbf{g}(\mathbf{X}_g, p), t)$$
$$\mathbf{X} = \mathbf{g}(\mathbf{X}_g, p)$$
$$\mathbf{X}_m = \mathbf{h}(\mathbf{X}_g, i)$$

where $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ are vector valued functions, t is time, p is some set of parameters controlling a Deformed Geometry interface, and i is number of times the geometry has been remeshed. From the physics point of view, the domain is fixed in the geometry frame coordinates $\mathbf{X}_g$, which are therefore seen as constant in the above formulas.

From the finite elements' point of view, it is instead the mesh frame coordinates $\mathbf{X}_m$ which are constant and $\mathbf{X}_g = \mathbf{h}^{-1}(\mathbf{X}_m, i)$. Therefore when assembling the finite-element matrices, the relation actually used is

$$\mathbf{x} = \mathbf{f}(\mathbf{g}(\mathbf{h}^{-1}(\mathbf{X}_m, i), p), t)$$

where $\mathbf{f}$ is a unit map if the spatial and material frames coincide, $\mathbf{g}$ is a unit map if the material and geometry frames coincide, and the inverse mapping $\mathbf{h}^{-1}(\mathbf{X}_m, i)$ is initially a unit map and then updated by interpolation after each remeshing operation.

In addition to the different sets of coordinate variables, some other geometric variables that the software defines are available for both the spatial and the material frames (see Geometric Variables).

Derivatives of Dependent Variables

When solving for some physical quantity, u, COMSOL Multiphysics always stores the solution for a fixed set of mesh nodes. That is, the dependent variable u is treated internally as a function of the mesh coordinates, $u(X_m, Y_m, t)$. The essence of the ALE system is that it allows treating the physical quantities as functions of the material or spatial coordinates, $u(X, Y, t)$ or $u(x, y, t)$, instead. This transformation is possible only if the mappings given by Equation 18-1 and Equation 18-2 are invertible.

SPATIAL DERIVATIVES

With respect to spatial differentiation, each dependent variable is treated as a function of one or more of the frames present in the model. Most physics interfaces are based on a formulation which is either Eulerian or a Lagrangian. They therefore lock their dependent variables to the spatial or the material frame, respectively. A few physics interfaces can formulate their equations in either material or spatial frame, as set by the **Frame type** setting found under **Discretization** in the Physics interface settings.

For a dependent variable `u`, there is typically two possibilities:

- The variable is defined on the spatial frame and its derivatives with respect to the spatial coordinates are denoted `ux` and `uy` in the software.
- The variable is defined on the material frame and its derivatives with respect to the material coordinates are denoted `uX` and `uY` in the software.

In a few cases, both sets of derivatives exist, but normally in COMSOL Multiphysics it is only possible to use one of these types of derivatives of each dependent variable.

TIME DERIVATIVES

When using ALE, the software defines two kinds of time derivatives:

- The common *frame time derivative*, valid for a fixed point in the frame on which the variable is defined. This derivative is always denoted `ut` in the software. For example, for `u` defined on the spatial frame:

$$u_t(x_0, y_0) = \left.\frac{\partial u}{\partial t}\right|_{x_0, y_0}$$

- The *mesh time derivative*, which is taken for a fixed point in the mesh:

$$u_{\mathrm{TIME}}(X_m, Y_m) = \left.\frac{\partial u}{\partial t}\right|_{X_m, Y_m}$$

  This derivative is denoted `uTIME` in the software. Since internally, everything is formulated on the mesh frame, the mesh time derivative is the one computed by the solvers and stored in the solution vector.

The two derivatives are related by the chain rule:

$$u_t = u_{\mathrm{TIME}} - u_x x_{\mathrm{TIME}} - u_y y_{\mathrm{TIME}}$$

where $(x_{\mathrm{TIME}}, y_{\mathrm{TIME}})$ is the mesh velocity. The mesh time derivative is often less important from the user point of view because its value depends on the mesh movement, which in itself has no physical significance. However, for the special case when the mesh follows the material's motion, the mesh time derivative is physically significant and is also called the *material time derivative*.

Smoothing Methods

In the domains with free displacement, the Moving Mesh interface solves an equation for the mesh displacement. This equation smoothly deforms the mesh given the constraints placed on the boundaries. Choose between *Laplace smoothing*, *Winslow smoothing*, and *hyperelastic smoothing*. To specify the smoothing methods, use the **Mesh smoothing type** list in the **Free Deformation Settings** section of the **Moving Mesh** or **Deformed Geometry** node. To see how these smoothing methods differ, let x and y be the spatial coordinates of the spatial frame, and let X and Y be the reference coordinates of the material frame.

- If Laplace smoothing is selected, the software introduces deformed mesh positions x and y as degrees of freedom in the model. In the static case, it solves the equation

$$\frac{\partial^2 x}{\partial X^2} + \frac{\partial^2 x}{\partial Y^2} = 0$$

  and in the transient case it solves the equation

$$\frac{\partial^2}{\partial X^2}\frac{\partial x}{\partial t}+\frac{\partial^2}{\partial Y^2}\frac{\partial x}{\partial t} = 0$$

Similar equations hold for the y coordinate.

- If Winslow smoothing is selected, the software solves the equation

$$\frac{\partial^2 X}{\partial x^2}+\frac{\partial^2 X}{\partial y^2} = 0$$

  and does the same for Y. Equivalently, X and Y satisfy Laplace equations as functions of the x and y coordinates.

- The hyperelastic smoothing method searches for a minimum of a mesh deformation energy inspired by neo-Hookean materials:

$$W = \int_{\Omega}\frac{\mu}{2}(I_1-3)+\frac{\kappa}{2}(J-1)^2 dV$$

  where μ and κ are artificial shear and bulk moduli, respectively, and the invariants J and I_1 are given by

$$J = \mathrm{det}(\nabla_X x)$$
$$I_1 = J^{-2/3}\mathrm{tr}((\nabla_X x)^T\nabla_X x)$$

Like the Winslow smoothing equation, hyperelastic smoothing is nonlinear. Both are therefore more expensive than the Laplace smoothing, but at the same time more robust. Hyperelastic smoothing can give a smoother result than Winslow, in particular in regions where the mesh is stretched.

Limitations of the ALE Method

The following limitations apply to the ALE method in general and therefore to the Moving Mesh and Deformed Geometry interfaces:

- The connectivity of the mesh remains unchanged during the mesh deformation, which means that topological changes in the geometry are not allowed.
- When the mesh deformation becomes large, the quality of the mesh created by the smoothing equations can deteriorate. If this happens, the solver runs into convergence problems. Sometimes the warning **Inverted mesh element** displays in the **Progress** window for the solver, which means that a mesh element has (partially)

warped inside-out. Sometimes, introducing extra boundaries with explicit deformation inside the domains can help. A new mesh can be generated for the region covered by the deformed mesh and let the solver continue by deforming the new mesh; see the section Remeshing a Deformed Mesh. See also Tips for Modeling Using Deformed Meshes below.

- When using **Geometry shape order** larger than 1 in the Moving Mesh and Deformed Geometry interfaces, the mesh moving techniques often produce elements with distorted shapes. If there are warnings about inverted mesh elements, consider reducing the geometry shape order to 1. This, however, makes the geometry representation polygonal, which might affect accuracy. Note also that the measure of mesh quality does not capture these distorted shapes because it is computed from the positions of the corners of the mesh element (ignoring midside nodes, for instance).

Tips for Modeling Using Deformed Meshes

When working with a deformed mesh to move things around, the computational mesh gets deformed. If the deformations become too large, some of the mesh elements might get inverted. This means that the accuracy of the solution deteriorates and eventually the solvers diverge due to an ill-conditioned system. Here are some tips how to keep the mesh under control:

- Try a different mesh. It is often preferable to start from a reasonably uniform mesh. One way to achieve this is to first select a coarse mesh in the predefined mesh size settings and then set a small maximum element size. Also, quad meshes and mapped meshes tend to perform better than triangles.
- Try a different smoothing type. Winslow smoothing is slightly slower, more memory consuming, and usually, but not always, more stable than Laplace smoothing. Hyperelastic smoothing sometimes works better than the other methods (in some fluid-structure interaction problems, for example). See Smoothing Methods for more information.
- If solving a time-dependent problem, try to solve the equations more accurately. This is achieved by reducing the absolute and the relative tolerances for the time-dependent solver.
- Help the mesh deformation by sliding the boundary elements along with the general movement of the mesh. This can be achieved by adding a prescribed deformation on the boundary that moves the boundary elements according to the deformation of some point in the model. Define a coupling operator under the

Definitions node and use it to couple the deformation from the point to the boundary mesh.

Remeshing a Deformed Mesh

When the mesh deformation has become so large that the quality of the mesh is too bad, generate a new mesh for the deformed configuration and then continue the solution process. To do so, follow these steps:

1. Add a stop condition.
2. View the deformed mesh.
3. Copy the solution.
4. Create a Deformed Configuration ( ).
5. Remesh the deformed configuration.
6. Continue solving with the new mesh.

Note

To make the Deformed Configuration mesh represent the deformed geometry you must use a Moving Mesh (ALE) interface to model the mesh deformation or, for structural mechanics models, use the option in the study to include geometric nonlinearity (requires the Structural Mechanics Module, MEMS Module, or Acoustics Module).

See Also

- Deformed Configuration in the *COMSOL Multiphysics Reference Guide*
- Solution Data Sets

The following sections contain details about these steps and some additional information.

ADDING A STOP CONDITION

Add a stop condition in the solver to make it stop when the mesh quality becomes too bad. If the time-dependent solver is used, do this by right-clicking, for example, **Study 1>Solver Configurations>Solver 1>Time Dependent Solver 1** and selecting **Stop Condition** from the context menu. If the parametric stationary solver is used, right-click for example **Study 1>Solver Configurations>Solver 1>Stationary Solver 1>Parametric 1** and select **Stop Condition** from the context menu. In the **Stop expression** field enter a scalar expression. For example, enter `mod1.ale.relVolMin-0.01` to stop before the minimum relative element volume becomes less than 0.01. The solver stops when the stop expression becomes negative.

The following variables are useful for defining a stop condition and to monitor the mesh deformation:

- The minimum relative element volume, `ale.relVolMin`, must be > 0, otherwise the mesh elements are inverted. A suitable stop criterion is that minimum relative element volume must be larger than a small positive number.
- An alternative measurement is the minimum mesh quality, `ale.minqual`, must also be > 0; an acceptable mesh quality is typically larger than 0.1 (where the quality measure is a number between 0 and 1).
- The local relative element volume, `ale.relVol`, is a quantity that is plotted using a surface plot, for example, to show how the mesh elements become distorted in parts of the geometry. By plotting this quantity while solving, monitor how the mesh deforms and where it might become too distorted.

VIEWING THE DEFORMED MESH

View the deformed mesh in a surface plot or volume plot by selecting the **Wireframe** check box under **Coloring and Style**, and selecting **No refinement** in **Resolution** under **Quality**. In 3D, it is instructive to use an **Element scale factor** less than 1 under **Shrink Elements**.

COPYING THE SOLUTION

To keep the first solution, right-click **Study 1>Solver Configurations>Solver 1** and select **Solution>Copy**. The copied solution appears as a new solver **Copy 2** under **Solver Configurations**.

CREATING A DEFORMED CONFIGURATION

Create a deformed configuration by right-clicking, for example, **Results>Data Sets>Solution 2** and selecting **Remesh Deformed Configuration** ( ). The deformed configuration appears as a new **Deformed Configuration 1** node ( ) under **Meshes**. The deformed configuration works as a new geometry but with restricted functionality. For example, its context menu only contains **Measurements**, **Export to File**, **Mesh**, **Delete**, **Rename**, **Properties**, and **Dynamic Help**. The settings window of the deformed configuration indicates which solution it was constructed from. Also change the parameter or time for which the deformed configuration was generated. If this is done, click the **Update** button to see the corresponding deformed configuration in the graphics.

REMESH THE DEFORMED CONFIGURATION

Expanding the **Deformed Configuration 1** node ( ) shows that a new mesh sequence has been added beneath it. This mesh sequence only contains a **Size** node ( ) and a **Reference 1** node ( ). The reference node refers to the original mesh sequence. This means that the new mesh sequence uses the same features as the original mesh sequence. Build the new mesh sequence by selecting **Build All** from its context menu or pressing F8. To make changes to the new mesh sequence before building it, right-click **Reference 1** and select **Expand** ( ). Then the features from the original mesh sequence are copied to the new mesh sequence. Also add and remove features in the new mesh sequence. It several meshes are needed on the deformed configuration (for the multigrid solver, for example), add an additional mesh sequence by right-clicking **Deformed Configuration 1** ( ) and selecting **Mesh** ( ).

CONTINUE SOLVING WITH THE NEW MESH

1 In the settings window for the study step (for example, **Study 1>Step1: Time Dependent**), use the **Mesh** list under **Mesh Selection** to select the new mesh sequence (**Mesh 2**, for example).

2 Change the **Times** list or the **Parameter values** list under **Study Settings** to include only the time or parameter corresponding to the deformed configuration plus the remaining times or parameters.

3 Change the initial value to be the last time or parameter of the previous solution. Do this by selecting **Study 1>Solver Configurations>Solver 1>Dependent Variables**. Under **Initial Values of Variables Solved For**, change **Method** to **Solution** and **Solution** to **Solver 1** (for example), and select the appropriate time or parameter value in the list **Time** or **Parameter value**. Usually the **Automatic** alternative, which selects the last time or parameter value, suffices. Change the settings under **Values of Variables Not Solved For** similarly.

4 Solve for the remaining times or parameters by right-clicking **Study 1** and selecting **Compute**.

REMESHING SEVERAL TIMES

Remesh several times by iterating the above steps. For each of the solver runs you get a copy of the solution (**Copy 2**, **Copy 3**, **Copy 4**, and so on) and a corresponding data set (**Solution 2**, **Solution 3**, **Solution 4**, and so on). In the plot group, select one of these data sets for results analysis and visualization.

ALTERNATIVE PROCEDURE USING SEVERAL STUDIES

The above procedure uses a single study that is modified for each solver run. To recompute the whole sequence of runs, an alternative that uses one study for each run is better. To do that, add a new study after meshing each deformed configuration. The copy solution step is not needed in this case. If the settings are changed in the study or its solver sequence, make these changes also in the new study. For example, the stop condition has to be added.

Remeshing a Deformed Geometry—A Quick Example

This section describes the remeshing technique by building a simple model where a force bends a steel beam. This causes the mesh to become distorted. You must then remesh using the deformed geometry to be able to continue the simulation. To build this example model, use the following steps:

1 In the **Model Wizard**, select **2D** on the **Select Space Dimension** page. Click **Next** ( ).

2 On the **Add Physics** page, select **Structural Mechanics>Solid Mechanics** ( ). Click **Add Selected** ( ).

3 Select **Mathematics>Deformed Mesh>Moving Mesh** ( ). Click **Add Selected** ( ). Then click **Next** ( ).

4 On the **Select Study Type** page, select **Preset Studies for Selected Physics> Time Dependent** ( ) for a time-dependent study.

5 Click **Finish** ( ).

Geometry Modeling

1 Right-click **Geometry 1** in the **Model Builder** and select **Rectangle** to create a rectangle with a **Width** of `1.6`, **Height** of `1.2`, and with its lower-left corner coordinates (x, y) at (`-0.6`, `-0.4`).

2 Create a second rectangle with a **Width** of `1.4`, **Height** of `0.2`, and its lower-left corner coordinates (x, y) at (`-0.6`, `-0.2`). Click **Build All** ( ).

Materials

Use structural steel as the material for the beam. No material is required for the surrounding domain where only the moving mesh is active.To add the structural steel material:

1 Right-click **Materials** and select **Open Material Browser** ( ).

2 In the **Material Browser**, right-click **Built-In>Structural steel** and select **Add Material to Model**.

3. In the settings window for the **Structural steel** node, select domain `1` in the list under **Geometric Scope** and then click the **Remove from Selection** button (▭) to remove the material from domain 1 (the air) and keep it in domain 2 (the beam) only.

Physics Settings—Solid Mechanics

1. Click the **Solid Mechanics** node. In its settings window, select domain 1 in the list under **Domain Selection** and then click the **Remove from Selection** button (▭) to remove the solid mechanics physics from domain 1 (the air) and keep it in domain 2 (the beam) only. Also select **Plane stress** from the **2D approximation** list and type `0.01` in the field for the thickness d.
2. Click the **Linear Elastic Material** node. It uses values from the structural steel material no settings need to be changed.
3. Add some damping to stabilize the transient solution. To do so, right-click the **Linear Elastic Material** node and select **Damping**.
4. Verify that **Rayleigh damping** is selected in the **Damping type** list (the default damping type). Type `1` in the $\alpha_{\mathbf{dM}}$ field for the mass damping parameter and `0.001` in the $\beta_{\mathbf{dK}}$ field for the stiffness damping parameter.

For the boundary conditions, the left side of the beam is fixed whereas the other boundaries are subject to a boundary load that increases with time.

1. Right-click the **Solid Mechanics** node and select **Fixed Constraint**. Select boundary 3 (the leftmost boundary and right-click to add it to the selection of boundaries for the **Fixed Constraint** node.
2. Right-click the **Solid Mechanics** node and select **Boundary Load**. Select boundaries 4, 6, and 8; then right-click to add them to the selection of boundaries for the **Boundary Load** node. Select **Load defined as force per unit length** from the **Load type** list and then enter a force that acts in the negative y direction (downward) and increases with time by setting the y component of $\mathbf{F_L}$ to `-1[MN/m]*t[1/s]`, where the unit syntax `[1/s]` makes the time variable t dimensionless. Leave the x component of $\mathbf{F_L}$ at zero.

Tip

To obtain a correct solution, use the support for geometric nonlinearity (large deformations). This functionality is available only in the extended Solid Mechanics interface available in the Structural Mechanics Module and MEMS Module.

Physics Settings—Moving Mesh

1. Click the **Moving Mesh** node. In its settings window, select domain `2` in the list under **Domain Selection** and then click the **Remove from Selection** button (−) to remove the moving mesh from domain 2 (the beam) and keep it in domain 1 (the air) only.
2. In the air domain, the moving mesh is free to deform. Right-click the **Moving Mesh** node and select **Free Deformation**. Select domain 1 (the air) and right-click to add it to the selection of domains for the **Free Deformation** node.
3. The default boundary condition is that the mesh does not move at the boundaries: a **Prescribed Mesh Displacement** node with 0 as the prescribed mesh displacement in all directions.
4. Right-click the **Moving Mesh** node and select **Prescribed Mesh Displacement** to add another **Prescribed Mesh Displacement** node. Select boundaries `4`, `6`, and `8` (the moving boundaries of the beam) and right-click to add them to the selection of boundaries for the **Prescribed Mesh Displacement** node. Enter `u` in the **dx** field and `v` in the **dy** field to make the mesh displacement follow the displacements in the solid. The variables `u` and `v` are the *x*-direction and *y*-direction displacements in the Solid Mechanics interface, respectively.

Mesh Generation

1. Click the **Mesh** node, and then select **Extra fine** from the **Element size** list in the **Mesh** node's settings window.
2. Click the **Build All** button (▦) in the **Mesh** node's settings window toolbar to create the mesh.

Computing the Solution

To make the solver stop when the mesh elements become poor, first enable a stop condition. Also specify the time steps for the time dependent solution.

1. Click **Study 1>Step 1: Time Dependent**. In the study step's settings window, type `range(0,0.01,0.3)` in the **Times** field. This means that the simulation starts at 0 s and runs until 0.3 s with steps of 0.01 s in the solution data.
2. Right-click the **Study 1** node ( ) and select **Show Default Solver** (⟳). This creates a **Solver** node with a **Time Dependent Solver** subnode (and a couple of other subnodes).
3. Right-click the **Time-Dependent Solver** node ( ) and select **Stop Condition** (STOP).
4. In the **Stop Condition** node's settings window, type `mod1.ale.relVolMin-0.01` in the **Stop expression** field. The solver then stops when the minimum relative element volume becomes less than 0.01.

5 Right-click the **Results** node ( ) and select **2D Plot Group** ( ) to create a surface plot of the local relative element volume that the program displays. For the surface plot, right-click the **2D Plot Group 1** node and select **Surface** ( ). In the **Surface 1** node's settings window, under **Expression**, select **Moving Mesh>Local relative element volume (ale.relVol)** from the **Replace Expression** list ( ).

6 Click the **Step 1: Time Dependent** node ( ) and, in its settings window's **Results While Solving** section, select the **Plot** check box to plot the local relative element volume for each result that the time dependent solver stores in the solution.

7 Right-click the **Study 1** node ( ) and select **Compute** ( ) to start the solver.

The solver stops somewhere around $t = 0.08$. The plot of the local relative element volume shows that mesh elements are stretched above the beam and compressed beneath it.

> *Tip*
>
> You can use automatic remeshing instead of the stop condition. The software then creates new meshes when the mesh quality drops below the specified level. To do so, click the **Step 1: Time Dependent** node ( ) and then select the **Automatic remeshing** check box in the settings window's **Study Extensions** section. To use the same condition as for the stop condition above, type `mod1.ale.relVolMin` in the **Mesh quality expression** edit field and `0.01` in the **Minimal mesh quality** edit field in the settings window for the **Automatic Remeshing** ( ) node. The following steps are then not needed, and the results while solving plots are not supported.

Geometry Modeling—Deformed Mesh

Next, create a new geometry and mesh from the deformed mesh. To do this, first copy the solution and then remesh the deformed configuration:

1 Right-click the **Study 1>Solver Configurations>Solver 1** node ( ) and select **Solution>Copy** ( ). This creates a copy of the solver in a **Copy 2** node in the **Study** branch and an associated **Solution 2** data set ( ) under **Results>Data Sets** ( ).

2 Right-click the **Solution 2** data set node and select **Remesh Deformed Configuration** ( ). This adds a **Deformed Configuration** node ( ) under the **Meshes** node ( ), which contains a geometry based on the deformed mesh at the last time step.

3 Click the default **Mesh 2** subnode ( ) and then click the **Build All** button ( ) in the **Mesh 2** node's settings window toolbar to create a mesh for the deformed geometry.

Computing the Solution

1 Click **Study 1>Step 1: Time Dependent**. In the study step's settings window, type `range(0.08,0.01,0.3)` in the **Times** field to start from the last time of the current solution. Generally when solving time-dependent problems this is not absolutely necessary, but in this case it is important because the force in the Solid Mechanics interface depends on time explicitly.

2 Also select **Mesh 2** from the **Mesh** list in the **Mesh Selection** section to restart using the new mesh based on the deformed geometry.

3 When you continue solving the model, it is important to use the solution from the last time step as the new initial value when restarting the solver. In the **Study 1>Solver Configurations>Solver 1>Dependent Variables** node's settings window, select **Solution** from the **Method** list and **Solver 1** from the **Solution** list in both the **Initial Values of Variables Solved For** and the **Values of Variables Not Solved For** sections to use the previously computed solution as the initial value.

4 Right-click the **Study 1** node ( ) and select **Compute** (=) to start the solver again.

The solver continues solving and stops when the mesh elements again becomes poor. Repeat the same procedure to remesh and continue the solution process from the time where the solver stopped.

The Moving Mesh Interface

With the **Moving Mesh** interface (), found under the **Mathematics>Deformed Mesh** branch () in the **Model Wizard**, models can be created where the geometry (actually the mesh) changes shape due to some physics in the model. Study both stationary states and time dependent deformations where the geometry changes its shape due to the dynamics of the problem. In this section:

- Moving Mesh
- Fixed Mesh
- Free Deformation
- Prescribed Deformation
- Prescribed Mesh Displacement
- Prescribed Mesh Velocity
- Prescribed Normal Mesh Velocity
- Zero Normal Mesh Velocity
- Zero Normal Mesh Displacement

Model

- Electrochemical Polishing: Model Library path **COMSOL_Multiphysics/Electromagnetics/electrochemical_polishing**
- Sloshing Tank: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/sloshing_tank**

PREDEFINED VARIABLES

The **Moving Mesh** interface defines the following predefined variables, which can be of interest, for example, for monitoring the quality of the mesh and defining a stop criterion for remeshing (see Adding a Stop Condition):

- The local relative element volume, `ale.relVol`, is a quantity that shows how the mesh elements become distorted in parts of the geometry.
- The minimum relative element volume, `ale.relVolMin`, must be > 0, otherwise the mesh elements are inverted.

- The maximum relative element volume, `ale.relVolMax`, is a positive scalar number that represents the maximum value of the relative element volume.
- The minimum mesh quality, `ale.minqual`, must be > 0; an acceptable mesh quality is typically larger than 0.1 (where the quality measure is a number between 0 and 1).

Moving Mesh

Use the **Moving Mesh** interface to specify stationary or transient deformation of the simulation domain.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `ale`.

DOMAIN SELECTION

From the **Selection** list, choose the domains (extent) for the interface.

Tip	The Moving Mesh interface does not need to be applied in domains for which the displacements appear among the dependent variables, for example, when Solid Mechanics is active.

FRAME SETTINGS

Specify the names of the spatial coordinates of the base frame for the interface—the material frame—in the **Material frame coordinates** fields. The defaults are the coordinates of the spatial frame in uppercase letters (**X**, **Y**, and **Z**). You can change the names in the fields for the **First**, **Second**, and **Third** coordinate. The field labels include the default spatial coordinate names in parentheses.

The **Geometry shape order** setting controls the order of polynomials—**1** (linear), **2** (quadratic—the default), **3** (cubic), **4** (quartic), or **5** (quintic, 2D only)—used for representing the geometry shape in the spatial frame. The same order is used for Lagrange shape functions defining the mesh position in domains where **Free displacement** has been activated.

FREE DEFORMATION SETTINGS

Select the smoothing type for freely deformed domains from the **Mesh smoothing type** list. Choose between Lagrange, Winslow, and Hyperelastic smoothing. The default is Laplace smoothing.

See Also

- Smoothing Methods
- Show More Physics Options

Fixed Mesh

Use the **Fixed Mesh** feature to specify that the selected domains remain at their reference material shape and do not move. This is the default.

DOMAIN SELECTION

From the **Selection** list, choose the domains to specify no displacement.

Free Deformation

The **Free Deformation** feature constrains the mesh displacement only by the boundary conditions on the surrounding boundaries. The displacement in the domain is obtained by solving a PDE.

See Also

Smoothing Methods

DOMAIN SELECTION

From the **Selection** list, choose the domains to specify free displacement.

INITIAL DEFORMATION

Give initial values for mesh displacements, relative to the material frame, in the **Initial mesh displacement** fields.

Prescribed Deformation

Use the **Prescribed Deformation** feature to define the deformation explicitly using expressions, or want the spatial coordinates to follow a deformation computed by, for example, a Solid Mechanics interface. (You can also achieve the latter effect by excluding the domains where the Solid Mechanics interface is defined from the domains where the Moving Mesh interface is active.)

DOMAIN SELECTION

From the **Selection** list, choose the domains to prescribe displacement.

PRESCRIBED MESH DISPLACEMENT

Specify expressions that define the deformation in the **Prescribed mesh displacement** fields (dx, dy, dz for 3D models, for example) (SI unit: m). The default is for no mesh displacement.

Prescribed Mesh Displacement

Use the **Prescribed Mesh Displacement** feature on the boundary of domains with free deformation. The spatial frame in the adjacent domain moves in accordance with the displacement.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to prescribe displacement.

PRESCRIBED MESH DISPLACEMENT

Select the check box for each coordinate direction to prescribe the displacement. The default settings provide a fixed boundary (zero displacements in all directions).

For boundaries adjacent to domains where displacement variables are defined, for example domains where a Solid Mechanics interface is active, let these variables drive the mesh displacement by typing the component field names in the corresponding fields (for example, setting **dx** to u, **dy** to v, and **dz** to w in 3D).

CONSTRAINT SETTINGS

Click the **Show** button ( ) and select **Advanced Physics Options** to display this section. Then select the **Use weak constraints** check box to use weak constraints instead of the default strong constraints.

Prescribed Mesh Velocity

On the boundary of domains with free displacement, specify the velocity of the boundary. Right-click the **Free Deformation** node to add a **Prescribed Mesh Velocity** node. The spatial frame in the adjacent domains moves in accordance with the velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to prescribe a velocity.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED MESH VELOCITY

Select the check box for each coordinate directory to prescribe a velocity. The default settings provide zero velocities in all directions.

CONSTRAINT SETTINGS

Click the **Show** button ( ) and select **Advanced Physics Options** to display this section. Then select the **Use weak constraints** check box to use weak constraints instead of the default strong constraints.

Prescribed Normal Mesh Velocity

Use the **Prescribed Normal Mesh Velocity** node, available under the **Moving Mesh** node, to specify the normal velocity of the boundary. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

Simulations using moving meshes, with a boundary moving in the normal direction, can sometimes need a stabilizing term to suppress the formation of local boundary segments of high curvature. This can be of particular importance in an automatic remeshing sequence, where the remeshing step might amplify local curvature artifacts.

The **Moving Boundary Smoothing** option smooths the normal mesh velocity of the **Prescribed Normal Mesh Velocity** feature according to the following equation:

$$\frac{\partial \mathbf{X}}{\partial t}\cdot\mathbf{n} = v_0 + v_{\mathrm{mbs}}$$

where v_0 is the desired normal mesh velocity, and v_{mbs} is a smoothing velocity according to:

$$v_{\mathrm{mbs}} = \delta_{\mathrm{mbs}}|v_0|hH$$

Here δ_{mbs} is a moving boundary smoothing tuning parameter (unitless), h is the mesh element size (SI unit: m), and H the mean surface curvature (SI unit: 1/m), defined as:

$$H = -\frac{1}{2}\nabla_T\cdot\mathbf{n}$$

where ∇_T is the surface gradient operator.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the prescribed mesh velocity.

NORMAL MESH VELOCITY

Enter a value or expression for the **Normal mesh velocity** v_{n} (SI unit: m/s).

MOVING BOUNDARY SMOOTHING

By default, the **Enable moving boundary smoothing** check box is not selected. To enable boundary smoothing, click to select the check box, and then enter a value or expression for the **Moving boundary smoothing tuning parameter**, δ_{mbs} (unitless). The default value is 0.5.

Zero Normal Mesh Velocity

Use the **Zero Normal Mesh Velocity** node, which is added from the **Moving Mesh** submenu, to set the normal velocity of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set on the tangential velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the zero normal mesh velocity.

Zero Normal Mesh Displacement

Use the **Zero Normal Mesh Displacement** node to set the displacement in the normal direction of the boundary to zero. The node can be used on the boundary of domains with free displacement. No constraints are set in the tangential direction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define the zero normal mesh displacement.

The Deformed Geometry Interface

Using the **Deformed Geometry** interface ( ), found under the **Mathematics>Deformed Mesh** branch ( ) in the **Model Wizard**, study how the physics changes when the geometry changes as a function of a parameter. The only real difference between the Deformed Geometry interface and the Moving Mesh interface is that the former defines a deformation of the material frame relative to the geometry frame, while the latter defines a displacement of the spatial frame relative to the material frame.

Model

Electrochemical Polishing: Model Library path **COMSOL_Multiphysics/Electromagnetics/electrochemical_polishing**

In this section:

- Deformed Geometry
- Free Deformation
- Free Deformation
- Prescribed Deformation
- Fixed Mesh
- Prescribed Mesh Displacement
- Prescribed Mesh Velocity

See Also

For detailed information about selecting geometric entities (domains, boundaries, edges, and points), see Selecting and Deselecting Geometric Entities

For more information about deformed meshes, see The Moving Mesh Interface.

Deformed Geometry

Use the **Deformed Geometry** feature to specify the domains and properties of the deformed geometry interface.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics interface if appropriate. Such situations could occur when coupling this interface to another physics interface, or when trying to identify and use variables defined by this physics interface, which is used to reach the fields and variables in expressions, for example. It can be changed to any unique string in the **Identifier** field.

The default identifier (for the first interface in the model) is `dg`.

DOMAIN SELECTION

From the **Selection** list, choose the domains (extent) for the interface.

Important

The Deformed Geometry interface requires a domain selection which covers all domains in which some physics is active.

FRAME SETTINGS

Specify the spatial coordinates of the base frame for the interface— the geometry frame—in the **Geometry frame coordinates** fields. The default is upper case versions of the spatial coordinates followed by a lowercase g (for example, **Yg**). You can change the names in the fields for the **First**, **Second**, and **Third** coordinate. The field labels include the default spatial coordinate names in parentheses.

The **Geometry shape order** setting controls the order of polynomials used for representing the geometry shape in the material frame. The same order is used for Lagrange shape functions defining the mesh position in domains where **Free displacement** has been activated.

FREE DEFORMATION SETTINGS

Select the smoothing type for freely deformed domains. Choose between Lagrange, Winslow, and Hyperelastic smoothing.

See Also

- Smoothing Methods
- Show More Physics Options

Free Deformation

The **Free Deformation** feature constrains the mesh displacement only by the boundary conditions on the surrounding boundaries. The material frame displacement in the domain is obtained by solving a PDE.

See Also

Smoothing Methods

DOMAIN SELECTION

From the **Selection** list, choose the domains to prescribe free displacement.

INITIAL DEFORMATION

Give initial values for mesh displacements, relative to the material frame, in the **Initial mesh displacement** fields.

Prescribed Deformation

Use the **Prescribed Deformation** feature to define the deformation of the material frame explicitly using expressions.

DOMAIN SELECTION

From the **Selection** list, choose the domains to prescribe displacement.

PRESCRIBED MESH DISPLACEMENT

Specify expressions that define the deformation in the **Prescribed mesh displacement** fields. Select the check box to enable the prescribed mesh displacement in the directions to use such a displacement condition. Use one expression per spatial coordinate.

Fixed Mesh

Use the **Fixed Mesh** feature to specify that the selected domains retain their original shape as defined by the geometry and original mesh. This is the default.

DOMAIN SELECTION

From the **Selection** list, choose the domains to specify no displacement.

Prescribed Mesh Displacement

Add the **Prescribed Mesh Displacement** feature on the boundaries of domains with free deformation. The material frame in the adjacent domain moves in accordance with the displacement.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to prescribe the mesh displacement.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED MESH DISPLACEMENT

Select the check box for each coordinate direction where you want to prescribe the displacement. The default settings provide a fixed boundary (zero displacements in all directions).

CONSTRAINT SETTINGS

Click the **Show** button ( ) and select **Advanced Physics Options** to display this section. Select the **Use weak constraints** check box to use weak constraints instead of the default strong constraints.

Prescribed Mesh Velocity

Use the **Prescribed Mesh Velocity** feature on the boundary of domains with free displacement. The material frame in the adjacent domains moves in accordance with the velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to prescribe a velocity.

COORDINATE SYSTEM SELECTION

Specify the coordinate system to use for specifying the mesh displacement. From the **Coordinate system** list select from:

- **Global coordinate system** (the default)
- **Boundary System** (a predefined normal-tangential coordinate system)
- Any additional user-defined coordinate systems

PRESCRIBED VELOCITY

Select the check box for each coordinate directory to prescribe a velocity. The default settings provide zero velocities in all directions.

CONSTRAINT SETTINGS

Click the **Show** button ( ) and select **Advanced Physics Options** to display this section. Then select the **Use weak constraints** check box to use weak constraints instead of the default strong constraints.

19

Meshing

This chapter summarizes how to create and control your mesh for 1D, 2D, and 3D geometries in COMSOL Multiphysics.

In this chapter:

Creating Meshes

The **Mesh** ( ) features enable the discretization of the geometry model into small units of simple shapes, referred to as *mesh elements*.

In this section:

- About Mesh Elements for 1D, 2D, and 3D Geometries
- The Meshing Toolbar and Features
- Meshing Techniques

See Also

- Meshing Features in the *COMSOL Multiphysics Reference Guide*
- Mesh in the *COMSOL Java API Reference Guide*

About Mesh Elements for 1D, 2D, and 3D Geometries

ELEMENTS FOR 1D GEOMETRIES

The mesh generator discretizes the domains (intervals) into smaller intervals (or mesh elements). The endpoints of the mesh elements are called *mesh vertices*.

The boundaries (or vertices) defined in the geometry are represented in the mesh by *boundary elements* (or *vertex elements*).

ELEMENTS FOR 2D GEOMETRIES

The mesh generator discretizes the domains into *triangular* or *quadrilateral* mesh elements. If the boundary is curved, these elements represent only an approximation of the original geometry. The sides of the triangles and quadrilaterals are called *mesh edges*, and their corners are *mesh vertices*. A mesh edge must not contain mesh vertices in its interior.

The boundaries defined in the geometry are discretized (approximately) into mesh edges, referred to as boundary elements (or *edge elements*), which must conform with the mesh elements of the adjacent domains.

The geometry vertices are represented by vertex elements.

ELEMENTS FOR 3D GEOMETRIES

The mesh generator discretizes the domains into *tetrahedral*, *hexahedral*, *prism*, or *pyramid* mesh elements whose faces, edges, and corners are called *mesh faces*, *mesh edges*, and *mesh vertices*, respectively.

The boundaries in the geometry are discretized into triangular or quadrilateral boundary elements. The geometry edges are discretized into edge elements.

Similar to 2D, the geometry vertices are represented by vertex elements.

Pyramid Elements

Pyramid elements appear in the mesh in these situations. If you:

- Import a NASTRAN file containing pyramid elements
- Create a swept mesh where the source and destination faces share an edge and the source face contains a triangular mesh
- Convert a quad mesh to a triangular mesh for a face adjacent to a domain that contains a mesh
- Create a boundary layer mesh in 3D for a geometry with sharp edges

The Meshing Toolbar and Features

A meshing toolbar is available as well as access to features when you right-click nodes and select items from the context menu (Table 19-2). Buttons on the toolbar also have predefined size element features available from the lists (Table 19-2) for each mesh type. These fine tuning features are also available on the **Size** (△) settings window.

See Also

For more examples and detail, see Meshing Features in the *COMSOL Multiphysics Reference Guide.*

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Size	△	This default node is added to all meshes. It provides access to predefined and custom element size settings.
Free Tetrahedral	△	A meshing technique generating an unstructured mesh with tetrahedral elements for 3D models.

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Swept		A meshing technique generating a structured mesh (at least in the direction of the sweep) with prism or hexahedral elements for 3D models.
Boundary Layers		A meshing technique generating structured layers of elements along specific boundaries integrating into an existing structured or unstructured mesh for 2D or 3D models.
Boundary Layer Properties		To specify the location of the boundary layers and the properties, such as the number and thickness of the boundary layers. Right-click the Boundary Layers node to add. For 2D or 3D models. See Boundary Layer Properties in the *COMSOL Multiphysics Reference Guide* for information.
Free Triangular		A free meshing technique generating an unstructured mesh with triangular elements for 2D or 3D models.
Free Quad		A free meshing technique generating an unstructured mesh with quadrilateral elements for 2D or 3D models.
Mapped		A meshing technique generating a structured mesh with quadrilateral elements for 2D or 3D models. Also right-click to add Edge Groups to this node.
Edge Groups		To specify the four groups of edges around a boundary (3D) /domain (2D) that is used to determine the mapped mesh of the boundary/domain. See Edge Groups in the *COMSOL Multiphysics Reference Guide* for information.
Convert		A meshing technique used to create a simplex mesh. For example, create a structured triangular mesh for 2D models by using the convert operation, which introduces a diagonal edge to each quadrilateral element. See Convert in the *COMSOL Multiphysics Reference Guide* for information.
Copy Face		To copy mesh between faces in a 3D geometry. For example, mesh Face 1 of a block and copy the mesh to the opposite Face 6. Right-click to add an Edge Map, One-Point Map, or Two-Point Map node. See Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D) in the *COMSOL Multiphysics Reference Guide* for information.

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Edge Map		Right-click the Copy Face or the Copy Domain node to add. Specify an edge map for a face copy or a domain copy operation. Use the feature if you need to control how the source and destination faces/domains are matched in a copy mesh operation. For example, create a block and mesh Face 1 with a fine mesh on Edge 1. Copy this mesh to face 6 and ensure that the fine mesh of Edge 1 ends up on Edge 12. See Edge Map in the *COMSOL Multiphysics Reference Guide* for information.
One Point Map		Right-click the Copy Face or the Copy Domain node to add. Use this if you need to control how the source and destination faces/domains are matched in a copy mesh operation. For example, create a block and mesh face 4 with a fine mesh near Point 8. Copy this mesh onto Face 3 and ensure that the fine mesh near Point 8 ends up near Point 3. See One-Point Map in the *COMSOL Multiphysics Reference Guide* for information.
Two Point Map		Right-click the Copy Face or the Copy Domain node to add. Use this feature if you need to control how the source and destination faces/domains are matched in a copy mesh operation. For example, create a block and mesh face 2. Copy this mesh onto the opposite Face 5 and ensure that Point 6 is mapped to Point 4 and Point 5 is mapped to Point 8. See Two-Point Map in the *COMSOL Multiphysics Reference Guide* for information.
Copy Edge	or	To copy an edge mesh from one edge to another. You can use the Copy Edge mesh operation to define periodic boundary conditions, where the solution on the two periodic boundaries has to be identical. Right-click to add Edge Map, One-Point Map, or Two-Point Map nodes. See Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D) in the *COMSOL Multiphysics Reference Guide* for information.
Edge		Use the Edge operation to create and fine tune edge mesh to give you control over the number of elements along edges. See The faces around the source and destination domains in 3D are also allowed to be partitioned differently with exactly that same limitation (source to destination face mapping must be many-to-one). in the *COMSOL Multiphysics Reference Guide* for information.

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Copy Domain	or	To copy mesh between domains in a 2D or 3D geometry. For example, mesh block 1 and copy the mesh to the block 2. Right-click to add an Edge Map, One-Point Map, or Two-Point Map node. See Copy Domain (2D, 3D), Copy Face (3D) and Copy Edge (2D, 3D) in the *COMSOL Multiphysics Reference Guide* for information.
Refine		To refine elements of the mesh. By default, all elements are refined once. The default refinement method in 2D is regular refinement, where all of the specified triangles are divided into four triangles of the same shape. In 3D, the default refinement method is longest. The regular refinement method is only implemented for uniform refinements (that is, when all elements are refined). In 1D, regular refinement, where each element is divided into two elements of the same shape, is always used. See Refine in the *COMSOL Multiphysics Reference Guide* for information.
Reference		To refer to another meshing sequence. For example, create a mixed mesh with quads and triangles on a geometry. Create a second meshing sequence with a scale feature and a reference to the first meshing sequence. The result is a coarser version of the first mesh. Right-click to add a Scale or Expand node. See Reference in the *COMSOL Multiphysics Reference Guide* for information.
Expand		Right-click the Reference node to add and to replace the reference with a copy of the referred sequence, where the attributes have been scaled with the scale attribute features of the reference. See Reference in the *COMSOL Multiphysics Reference Guide* for information.
Scale		To scale size properties defined in a sequence. For example, create a block and mesh it with 10-by-10-by-10 hexahedra. Setting scale to 2 gives you a block with 5-by-5-by-5 hexahedra and setting the scale to 0.5 gives you a block with 20-by-20-by-20 hexahedra. Right-click the Reference node to add. See Scale in the *COMSOL Multiphysics Reference Guide* for information.

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Distribution		Mesh element distribution properties. You can specify a mesh element distribution by specifying the number of elements only, by specifying the number of elements together with properties determining the distribution of the elements, or by specifying the element distribution explicitly. This feature can be assigned to edges in 2D and 3D, domains in 3D or the entire geometry. See Distribution in the *COMSOL Multiphysics Reference Guide* for information.
Plot		Create a Mesh plot and view it in the Graphics window. See Mesh Plots for information.
Import		To import a mesh into a sequence without a corresponding geometry. It is only possible to use this feature when the geometry sequence is empty. If the sequence already contains a mesh, the imported mesh is added to the existing mesh. See Import in the *COMSOL Multiphysics Reference Guide* for more information.
Ball		To split geometric entities in an imported mesh by an element set defined by a ball. See Ball in the *COMSOL Multiphysics Reference Guide* for more information.
Box		To split geometric entities in an imported mesh by an element set defined by a box. See Box in the *COMSOL Multiphysics Reference Guide* for more information.
Create Vertex		To create an additional vertex in an imported mesh. See Create Vertex in the *COMSOL Multiphysics Reference Guide* for more information.
Delete Entities		To delete geometric entities from an imported mesh. See Delete Entities in the *COMSOL Multiphysics Reference Guide* for more information.
Join Entities		To join adjacent geometric entities in an imported mesh. See Join Entities in the *COMSOL Multiphysics Reference Guide* for more information.
Export to File		Export a mesh to a file. See Exporting a Mesh to File in the *COMSOL Multiphysics Reference Guide* for information.
Measure		To measure the volume, area, or length of a selected domain, face, or edge. The result displays in the Messages window. It is also possible to view the coordinates of a vertex, the distance between two vertices, or the number of entities and the geometry representation (if you have license for the CAD Import Module) of a geometry object.

TABLE 19-1: MESHING TOOLBAR BUTTONS AND CONTEXT MENU FEATURES

NAME	BUTTON	USE AND DESCRIPTION
Statistics		To gather some information about the mesh, including the number of mesh elements of different types and the quality of elements. See The Statistics Page in the *COMSOL Multiphysics Reference Guide* for information.
Build Selected		To build all features from the first up to the selected feature.
Build All		To build all features in the sequence.

TABLE 19-2: PREDEFINED ELEMENT SIZE BUTTONS

BUTTON	NAME	BUTTON	NAME
	Extremely Fine		Normal
	Extra Fine		Coarse
	Finer		Coarser
	Fine		Extra Coarse
			Extremely Coarse

Meshing Techniques

PHYSICS-CONTROLLED MESHING

The default is to use a mesh that is controlled by the physics. The mesh is then adapted to the current physics settings in the model. For example, for a fluid-flow model you get a somewhat finer mesh than the default with a boundary layer mesh along the no-slip boundaries. If you want to modify the overall element size of the physics-induced mesh you select another element size in the settings window of the main mesh node and rebuild the mesh. If you change the physics settings in the model and rebuild the meshing sequence, COMSOL Multiphysics creates a new mesh adapted to the new physics settings. To edit a physics-induced meshing sequence, or to see the errors and warnings of a failing mesh build, right-click the Mesh node and select **Edit Physics-Induced Sequence** ( ). The program then adds the nodes under the main **Mesh** node that together form the physics-controlled mesh.

Note	For more information on problems caused by **Boundary Layers** nodes see Boundary Layer Meshing Failures. For general tips on resolving meshing issues, see Troubleshooting Free Tetrahedral Mesh Generation.

USER-CONTROLLED MESHING

Alternatively, you can use a user-controlled mesh. It is then possible to manually build and edit the meshing sequence using the meshing techniques described below for creating a 2D and 3D meshes.

Note	If you select **User-controlled mesh** from the **Sequence type** list in the main **Mesh** node's settings window, the program adds a **Size** node and a node for the default mesher (**Free Triangular** in 2D, for example). If the **Sequence type** list is set to **Physics-controlled** mesh and you add a **Size** node, for example, the mesh sequence switches to a user-controlled mesh, but no default mesher is added.

See Also	Meshing Features in the *COMSOL Multiphysics Reference Guide*

2D MESHING OPTIONS

Unstructured Meshes (Free Meshing)

Free meshing generates an unstructured mesh with triangular or quadrilateral elements.

Structured Meshes

- Create a structured triangular mesh by using the convert operation to introduce a diagonal edge to quadrilateral elements.

- Mapped meshing generates a structured mesh with quadrilateral elements.

  Compared to an unstructured mesh the interior mesh vertices in a structured mesh are adjacent to the same number of elements. If you want to use mapped meshing on a geometry, you must build the geometry so that the domains are reasonably “regular” in shape and do not contain holes.

See Also

- Free Meshing
- Generating a 2D Free Mesh

3D MESHING OPTIONS

Note

Platforms handle floating-point operations differently, which sometimes results in slight differences between identical model files that are generated on two different computers.

Unstructured Meshes (Free Meshing)

Free meshing generates an unstructured mesh with tetrahedral elements.

Structured Meshes

- Swept meshing generates a structured mesh (at least in the direction of the sweep) with prism or hexahedral elements.
- Boundary layer meshing generates structured layers of elements along specific boundaries integrating into an existing structured or unstructured mesh.

See Also

- Free Meshing
- Generating a 3D Free Tetrahedral Mesh
- Generating a 3D Swept Mesh
- Generating a Boundary Layer Mesh

Free Meshing

In this section:

- About Free Meshing
- Generating a 2D Free Mesh
- Generating a 3D Free Tetrahedral Mesh
- Troubleshooting Free Tetrahedral Mesh Generation

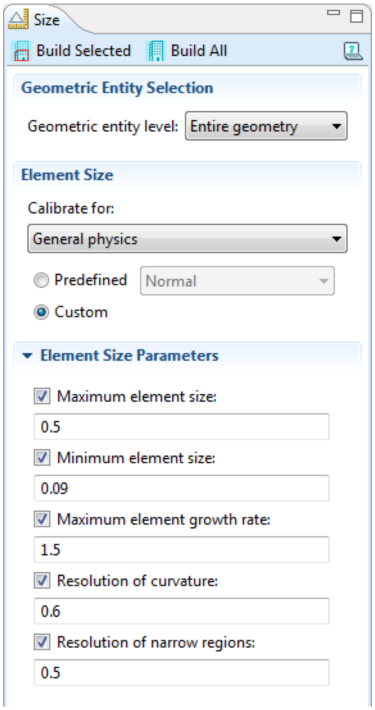
About Free Meshing

The free mesher is available in all dimensions, and you can use it for all types of geometries regardless of their topology or shape. If you have not defined or generated a mesh, the free mesher automatically creates an unstructured mesh and adds a corresponding node to the Model Builder when you compute a study.

When you use the free mesher:

- The number of mesh elements is determined by the shape of the geometry and various mesh parameters.
- You control mesh parameters for the free mesher by Size and Distribution features in the meshing sequences.

You can also control the size of the mesh generated by a specific Free Triangular, Free Quadrilateral, or Free Tetrahedral node by adding **Size** () or **Distribution** () subnodes.



Size

Build Selected | Build All

Geometric Entity Selection

Geometric entity level: Entire geometry

Element Size

Calibrate for:

General physics

○ Predefined Normal

● Custom

Element Size Parameters

☑ Maximum element size:

0.5

☑ Minimum element size:

0.09

☑ Maximum element growth rate:

1.5

☑ Resolution of curvature:

0.6

☑ Resolution of narrow regions:

0.5

Figure 19-1: An example of custom element mesh sizes. You can also select Predefined element sizes.

See Also

- Generating a 2D Free Mesh
- Generating a 3D Free Tetrahedral Mesh

Generating a 2D Free Mesh

Use the free mesher in 2D to create an unstructured mesh with triangular elements or quadrilateral elements. You can combine triangular and quadrilateral meshes by adding domains to the Domain list in the corresponding mesh operation's settings. From here, you can define specific meshing operations to each domain in your model. Figure 19-2 shows the resulting mesh when combining the Free Triangular and Free

Quad features to create a meshing sequence that generates the mesh. The model represents a cross section of a heat sink for high power applications.

Model

For a 2D axisymmetric free triangular mesh example, see Terminal Falling Velocity of a Sand Grain: Model Library path **COMSOL_Multiphysics/ Fluid_Dynamics/falling_sand**

For a 2D free triangular mesh example, see Fluid Valve: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve**

See Also

- Free Triangular and Free Quad in the *COMSOL Multiphysics Reference Guide*
- About Free Meshing
- Generating a 3D Free Tetrahedral Mesh

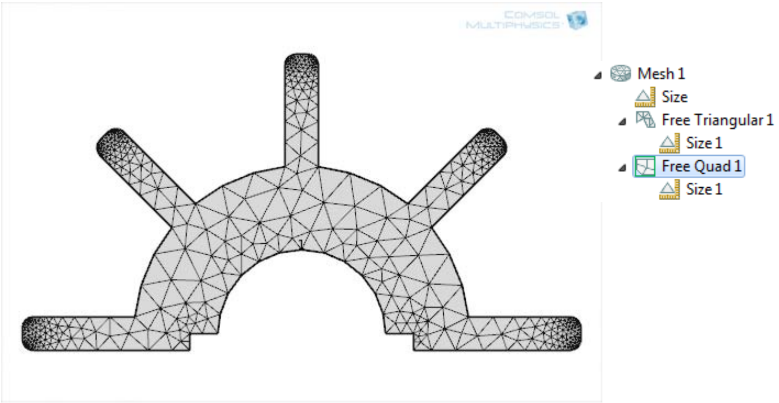


Figure 19-2: A combined triangular and quadrilateral mesh.

Adding a Free Triangular or a Free Quad Mesh to a 2D Model

1 Open a 2D file or create a 2D object.

Tip

If you saved a copy of the geometry from Creating a 2D Geometry Model, then open this file.

2 On the mesh toolbar, click the **Free Triangular** ( ) or **Free Quad** ( ) button.

3 In the **Model Builder**, click the **Free Triangular** ( ) or **Free Quad** ( ) node. In the settings window, under **Domain Selection**, select **Domain** as the **Geometric entity level**.

4 In the **Graphics** window, left-click and then right-click any or all of the domains (in this example, there is only one domain) to add to the **Selection** list.

5 If the geometry requires it, you can now control the number of mesh elements along an edge. In this case, when **Free Quad** ( ) is selected as the mesh, the following example applies.

Controlling the Number of Elements on Edges

In this example, when **Free Quad** is selected as the mesh, Figure 19-3 shows that the two cooling fins at 45 degrees to the horizontal are only meshed with one mesh element at their attachment to the base. Use the **Edge** feature to fine tune the mesh and to give you control over the number of elements along the edge separating the fins and the base.

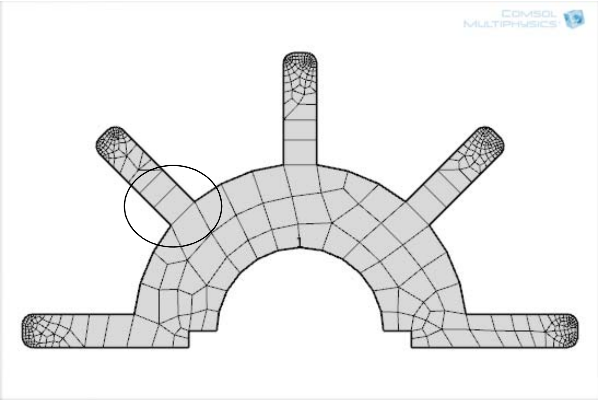


Figure 19-3: A Free Quad mesh with only one mesh element at the attachment to the base

Insert an **Edge** operation before the **Free Quad** operation to control the mesh at the interface between the base and the fin.

1 Right-click the first **Size** node ( ) (under **Mesh 1**) and select **Build Selected** ( ).

2 Right-click **Mesh 1** and select **More Operations>Edge** ( ).

Tip

When inserting a node to the sequence it appears after the current feature. The current feature is indicated by a quadratic frame around its icon.

To make a feature become the current feature, select it, right-click, and select **Build Selected** ( ). You can also move mesh nodes up and down in the sequence by right-clicking and selecting **Move Up** (⇧) or **Move Down** (⇩) or using the equivalent Keyboard Shortcuts.

3 In the **Graphics** window, left-click and then right-click the edges to add to the **Selection** list. Or click the **Paste** button ( ) and enter the edge numbers (for this example: `4,6`) in the **Paste Selection** window.

4 Right-click **Edge** ( ) and select **Distribution** ( ). The **Distribution** node inherits the same list of edges from the **Edge** node.

5 Click the **Distribution** node ( ) and under Distribution, change the Number of elements from 5 (the default) to 30.

6 Click **Build Selected** ( ). In this example, the mesh elements are generated along the selected edges (Figure 19-4).

7 Click the **Free Quad** node ( ) and select **Remaining** from the **Geometric entity level** list. Click **Build Selected** ( ).

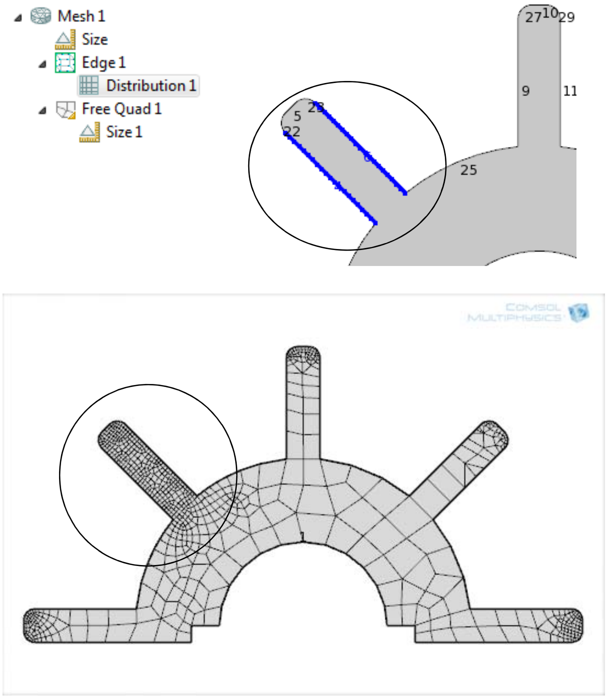


Figure 19-4: Adding a distribution attribute to two edges and rebuilding the mesh defines a finer mesh.

Generating a 3D Free Tetrahedral Mesh

See Also

Free Tetrahedral in the *COMSOL Multiphysics Reference Guide*

Model

For a 3D free tetrahedral mesh example see the Busbar model: Model Library path **COMSOL_Multiphysics/Multiphysics/busbar**

ADDING A FREE TETRAHEDRAL FEATURE

1 Open a 3D model or create a 3D model.

Tip

If you saved a copy of the geometry from Creating a 3D Geometry Model, then open this file.

---

2 On the mesh toolbar, click the **Free Tetrahedral** button ( ).

3 In the **Model Builder**, click the **Free Tetrahedral** node ( ).

4 In the settings window, under **Domain Selection**, select **Domain** as the **Geometric entity level**.

5 In the **Graphics** window, left-click and then right-click any or all of the domains to add to the **Selection** list, or use any of the selection methods described in Working with Geometry and User-Defined Selections.

SPECIFYING THE MESH SIZE

Note

To adapt the mesh for the selected domains, add a **Size** feature to the **Free Tetrahedral** node. Otherwise the Size node directly following the main Mesh node governs the mesh-element size for all Free Tetrahedral nodes.

Figure 19-5 shows a meshing sequence consisting of two Free Tetrahedral mesh operations, one for each meshed domain that correspond to pillars in a heat sink geometry. The mesh sizes are assigned as follows:

- The first **Size** node (under **Mesh1**) is defined as **Extra Coarse**, which is selected from the **Predefined Element Size** list or from the mesh toolbar. In this example, the left pillar is selected to have the extra coarse mesh size and this is generated by the first **Free Tetrahedral 1** operation.
- The mesh size of the second pillar is defined under the **Free Tetrahedral 2** operation, it has its own **Size 1** attribute (**Coarser**), a finer mesh that is also selected from the **Predefined Element Size** list or the mesh toolbar.

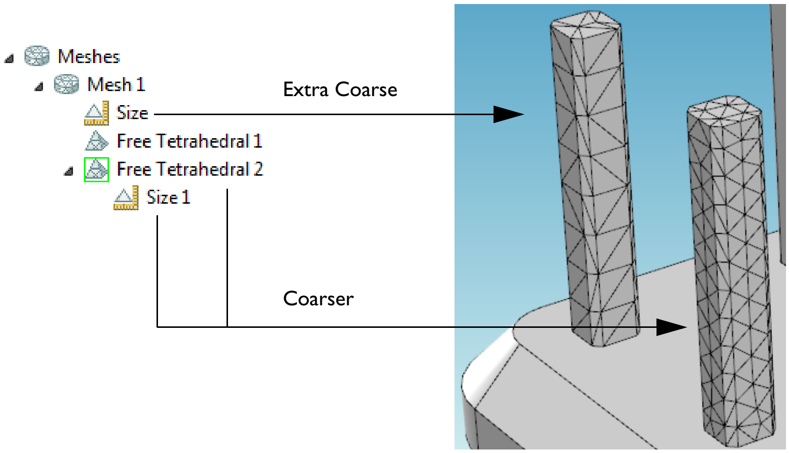


Figure 19-5: An example of two free tetrahedral meshing sizes.

The difference between the two meshes is based on the following.

SPECIFYING A CUSTOM ELEMENT SIZE

Tip

To create a predefined element size, on the mesh toolbar, click the **Mesh 1** node (to display the toolbar) and then click the small arrow next to the **Free Tetrahedral** button ( ) to select one of the predefined mesh sizes as described in Table 19-2.

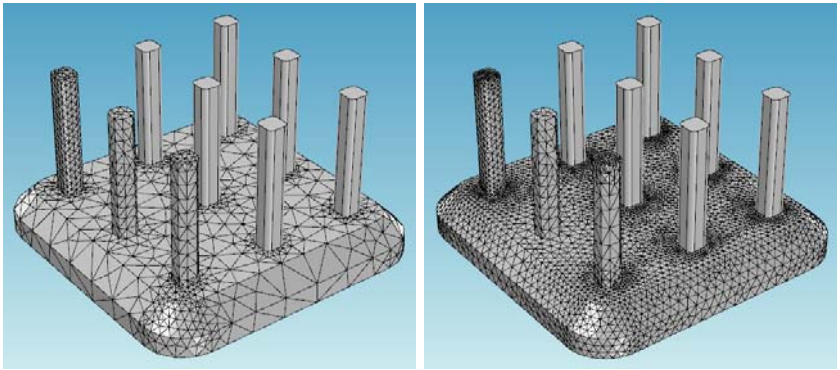
1 In the **Model Builder**, click **Size** ( ) to display the settings window. See Figure 19-1.

2 Under **Element Size**, click **Custom**. Under **Element Size Parameters**, define the mesh by entering values in the fields as required.

- Enter the **Maximum element size** allowed (SI unit: m).
- Enter the **Minimum element size** allowed (SI unit: m). You can use this value to, for example, prevent the generation of many elements around small parts of the geometry.
- Enter the **Maximum element growth rate** at which the element size can grow from a region with small elements to a region with larger elements. The value must be greater or equal to one. For example, with a maximum element growth rate of 1.5, the element size can grow by at most 50% (approximately) from one element

to another. Figure 19-6 shows the base of the heat sink meshed using different two different element growth rates for otherwise identical settings.

- Enter the **Resolution of curvature** to determine the size of boundary elements compared to the curvature of the geometric boundary. The curvature radius multiplied by the resolution of curvature, which must be a positive scalar, gives the maximum allowed element size along the boundary. A lower value gives a finer mesh along curved boundaries.
- Enter the **Resolution of narrow regions** to control the number of layers of elements that are created in narrow regions (approximately). The value must be a non-negative scalar. If the value of this parameter is less than one, the mesh generator might create elements that are anisotropic in size in narrow regions.



Element growth rate = 1.9

Element growth rate = 1.1

Figure 19-6: A comparison of mesh element growth rates in a heat sink.

SPECIFYING MESH DISTRIBUTION

In addition to customizing the mesh size, you can also add a Distribution attribute to a Free Tetrahedral feature. In the settings for this Distribution, constrain the number of elements for the selected edges. Figure 19-7 shows three heat sink pillars with different mesh sizes and distribution.

Note

The settings in the **Distribution** attribute overrule the settings in the **Size** node. See Distribution in the *COMSOL Multiphysics Reference Guide*.

.

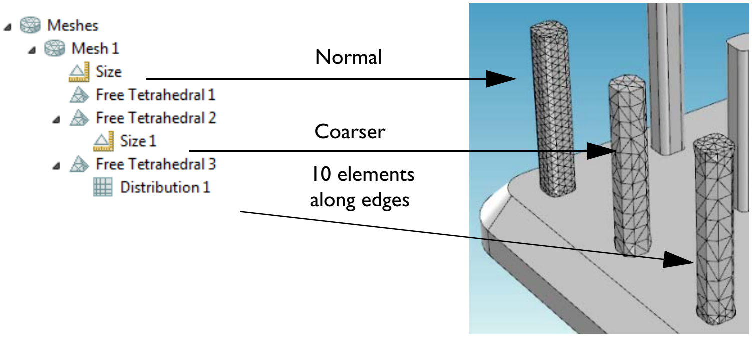


Figure 19-7: Specifying element distribution in heat sink pillars.

See Also

- About Free Meshing
- Generating a 2D Free Mesh

Troubleshooting Free Tetrahedral Mesh Generation

This section gives you some suggestions how to solve problems you might encounter when creating tetrahedral meshes.

BUILD A FINER MESH

As a general rule, it is easier to construct a mesh with smaller elements than a mesh with larger elements. If you get errors or low-quality elements when you try to mesh certain domains, try to decrease the element size using appropriate **Size** () attributes on these domains.

USE AN APPROPRIATE MINIMAL ELEMENT SIZE

If your geometry contains details that are very small compared to the total volume of the mesh, you must ensure that the Minimum element size parameter in the corresponding **Size** () attribute is at least as small as the smallest detail you wish to resolve. If this parameter is too large, you will get warnings when building the feature. For example, the warning “Edge is much shorter than the specified minimum element size” indicates that there are edges significantly shorter than the specified minimum element size. The resulting mesh gets badly shaped elements.

Tip

To locate small details, such as short edges and sliver faces, you can add and build a Free Tetrahedral node with normal size settings. Then you will get warnings with selections that points you to the corresponding small entities. You can also inspect the mesh visually to locate unexpected small elements.

REMOVE UNWANTED GEOMETRY DETAILS

Sometimes, the geometry contains small features, like sliver faces and short edges, which you do not wish to resolve at all. Then you can use Virtual Operations in the Geometry sequence to ignore disturbing details of the geometry (see Using Virtual Geometry Operations). If you have a license for the CAD Import Module, you can also use CAD defeaturing operations to simplify the geometry.

PARTITION THE GEOMETRY INTO SIMPLE DOMAINS

If the geometry has very complex domains or very complex faces that you have trouble meshing, you can try to partition you geometry into less complex entities. On a philosophical level, this method could be classified as a divide and conquer strategy. It is often possible to use a work plane in the geometry to embed a face that splits a complex domain into two domains. You can use a Mesh Control Faces node to make this partitioning only when building the mesh (see Mesh Control Entities).

Tip

To split a solid geometry object into parts using a Work Plane, place the work plane where you want to cut the domain. Then draw a rectangle (or another 2D primitive) in the work plane, making it sufficiently large to cover the cut. Finally, add a Convert to Solid node with the object you want to split and the surface rectangle as input. COMSOL then calculates the exact shape of the splitting face for you.

Structured Meshes

In this section:

- About 2D Mapped Meshing
- About 2D and 3D Boundary Layer Meshes
- Generating a Boundary Layer Mesh
- About 3D Swept Meshes
- Generating a 3D Swept Mesh

About 2D Mapped Meshing

See Also

Mapped in the *COMSOL Multiphysics Reference Guide*

GEOMETRY REQUIREMENTS

For the 2D mapped meshing technique to work properly, the geometry must be reasonably regular. The following conditions must be satisfied:

- Each domain must be bounded by at least four boundary segments.
- Each domain must be bounded by only one connected boundary component (that is, no holes are allowed).
- The domains must not contain isolated or embedded vertices or boundary segments.
- The shape of each domain must not differ significantly from a rectangle.

For a geometry model that does not initially meet these criteria, it is usually possible to modify it so that a mapped mesh is generated, usually this is done by splitting it into simpler domains.

Model

For an example of a 2D mapped mesh, see Tubular Reactor: Model Library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**

About 2D and 3D Boundary Layer Meshes

A boundary layer mesh is a mesh with dense element distribution in the normal direction along specific boundaries. This type of mesh is typically used for fluid flow problems in order to resolve the thin boundary layers along the no-slip boundaries.

- In 2D, a layered quadrilateral mesh is used along the specified no-slip boundaries.
- In 3D, the boundary layer mesh is a layered prism mesh or a hexahedral mesh, depending on whether the corresponding boundary-layer boundaries contain a triangular or quadrilateral mesh.

Boundary layer meshes can also be used to resolve large temperature gradients close to heated surfaces subjected to sudden changes over time. In the heat sink model, you can see the introduction of a boundary layer mesh at the surfaces of the inner half-circle arc.

BOUNDARY LAYER MESHING FAILURES

The boundary layer meshing algorithm is very sensitive to the topology of the model geometry. If you get an error when trying to build a **Boundary Layers** node you should try the following:

- Remove unnecessary interior boundaries such as boundaries (resulting from Boolean operations of geometry objects) that do not separate materials or physics. An efficient way to do this is to mark these boundaries use Mesh Control Entities. Mark these boundaries as mesh control entities in the Geometry Sequence. Then you mesh the domains, using Free Tetrahedral or Swept Mesh. When the domains are meshed, the control boundaries are automatically removed, and you can insert boundary layers, ignoring the interfering boundaries.

- Use boundary layer mesh trimming instead of splitting. By default, the boundary layer mesher creates a boundary layer split at each sharp corner in 2D and along each sharp edge in 3D.

Model

If you have the:

- Batteries & Fuel Cells Module, see Soluble Lead-Acid Redox Flow Battery (2D): Model Library path **Batteries_and_Fuel_Cells_Module/Batteries/pb_flow_battery**
- CFD Module, see Turbulent Flow Over a Backward Facing Step (2D): Model Library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**
- Heat Transfer Module, see Turbulent Flow Over a Backward Facing Step: Model Library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**

See Also

In the *COMSOL Multiphysics Reference Guide* see:

- Removing Interior Boundaries in Union Operations
- To turn off boundary layer splits, see Boundary Layers

Generating a Boundary Layer Mesh

1. In the **Model Builder**, right-click **Mesh** ( ) and select **Boundary Layers** ( ).
2. Click the **Boundary Layer Properties** node ( ) and in the settings window, select the boundaries where boundary layers should be applied. See Figure 19-8.

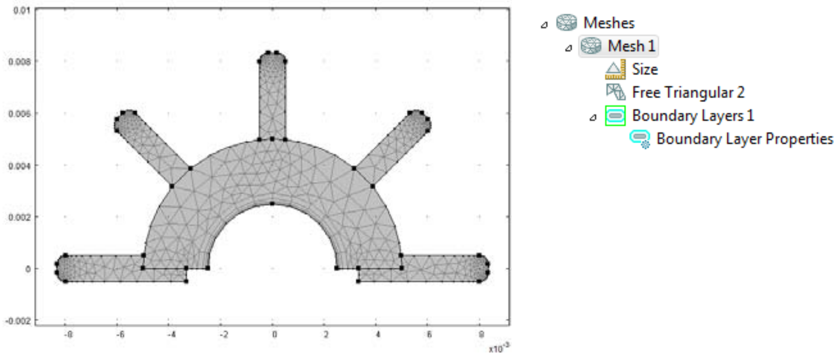


Figure 19-8: Adding boundary layer elements.

About 3D Swept Meshes

The swept mesher operates on a 3D domain by meshing a source face and then sweeping the resulting face mesh along the domain to an opposite destination face. A swept mesh is structured in the sweep direction and can be either structured or unstructured orthogonally to the sweep direction.

You can use several connected faces as source faces. Also the destination may consist of several faces, as long as each destination face correspond exactly to one or more source faces; the partitioning of the source side into faces must be a refinement of the partitioning on the destination side. Each face about a domain that is to be operated on by the swept mesher is classified as either a source face, a destination face, or a linking face. The linking faces are the faces linking the source and destination faces (see Figure 19-9). The swept mesher can handle domains with multiple linking faces in the sweep direction.

The linking edges are the edges, or the chains of edges, connecting the source and destination faces. For a domain to be possible to sweep, there must be at least one linking edge or chain of edges.

Model

- Thin-Layer Diffusion: Model Library path **COMSOL_Multiphysics/Diffusion/thin_layer_diffusion**
- Joule Heating of a Microactuator: Model Library path **COMSOL_Multiphysics/Multiphysics/thermal_actuator_jh**

See Also

Swept in the *COMSOL Multiphysics Reference Guide*

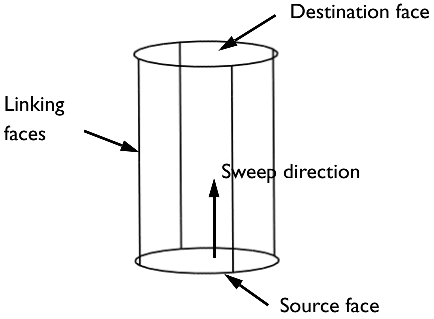


Figure 19-9: Classification of the boundaries about a domain used for swept meshing.

Generating a 3D Swept Mesh

Figure 19-10 shows the 3D swept mesh for a simple geometry but with a layered structure typical for printed circuit boards or MEMS geometries. In such cases, the swept mesh generation presents an alternative to using a free tetrahedral meshing.

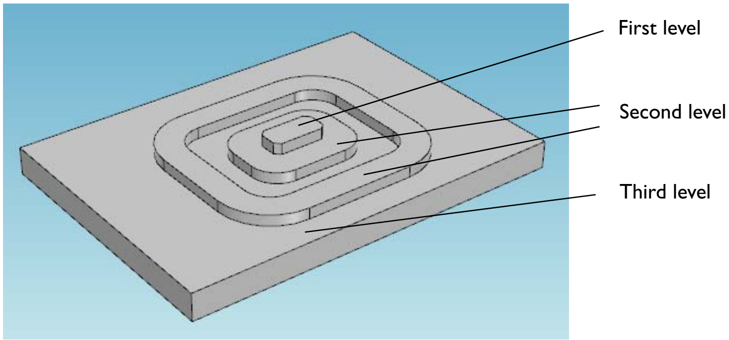
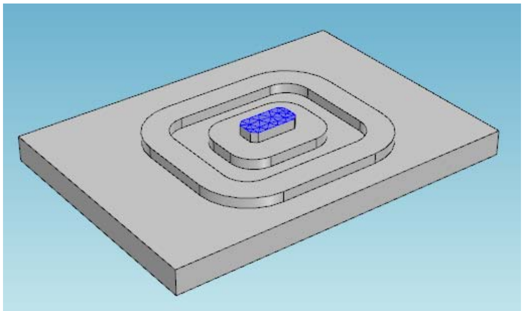


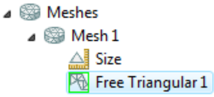
Figure 19-10: An example of the layered geometry used for creating a swept mesh.

1 Right-click **Mesh** ( ) and select **More Operations>Free Triangular** ( ).

2 Add the first level boundary to the selection list (see Figure 19-10 for an example of a suitable geometry).

3 Click **Build Selected** ( ). The mesh below displays.





- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1

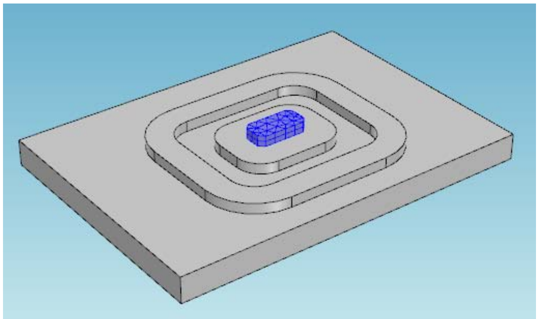
4 Right-click **Mesh** ( ) and select **Swept** ( ).

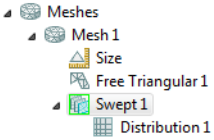
5 Select the domain in the first level.

6 Right-click **Swept 1** and select **Distribution** ( ).

7 Enter the **Number of elements** in the field (for example, 2).

8 Click **Build Selected** ( ).

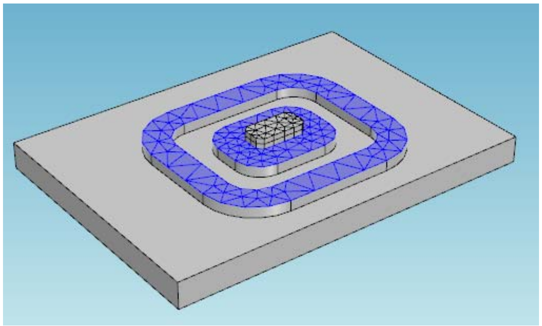


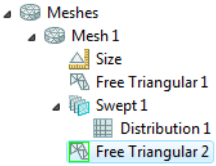


- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1
    - Swept 1
      - Distribution 1

9 Right-click **Mesh** ( ) and select **More Operations>Free Triangular** ( ).

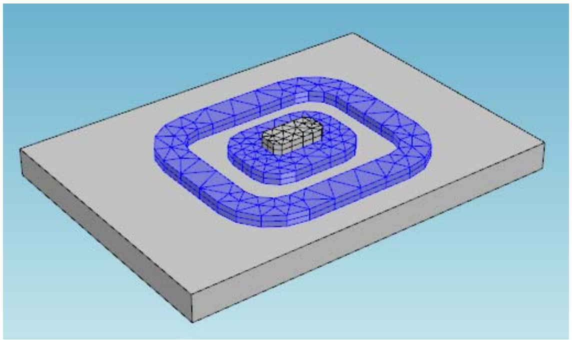
10 Select the boundaries at the second level and click the **Build Selected** button( ).

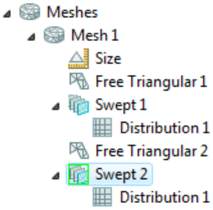




- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1
    - Swept 1
      - Distribution 1
    - Free Triangular 2

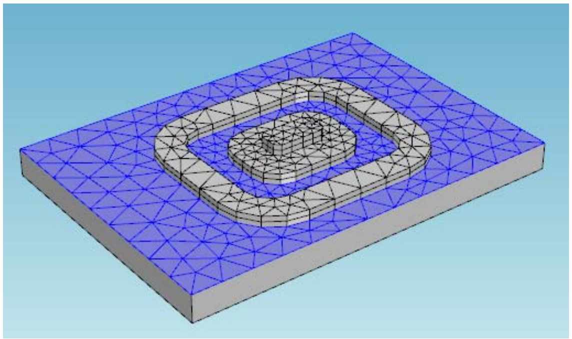
11 Repeat the same swept operations for the first level domains but now for the second level.

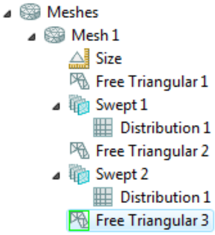




- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1
    - Swept 1
      - Distribution 1
    - Free Triangular 2
    - Swept 2
      - Distribution 1

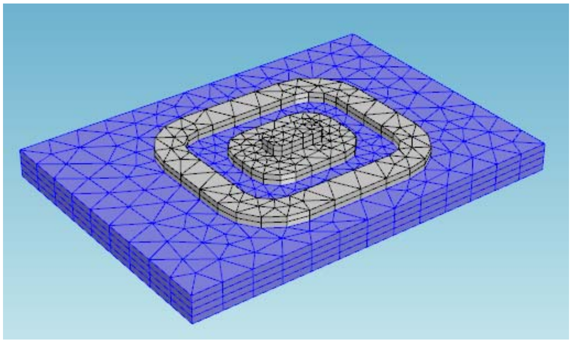
12 Mesh the third level boundaries using the **Free Triangular** mesh operation.

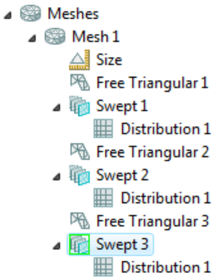




- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1
    - Swept 1
      - Distribution 1
    - Free Triangular 2
    - Swept 2
      - Distribution 1
    - Free Triangular 3

13 Mesh the third level domain. Use the **Swept** mesh operation and enter 4 for the **Number of elements** in the corresponding **Distribution** attribute.





- Meshes
  - Mesh 1
    - Size
    - Free Triangular 1
    - Swept 1
      - Distribution 1
    - Free Triangular 2
    - Swept 2
      - Distribution 1
    - Free Triangular 3
    - Swept 3
      - Distribution 1

The meshing sequence displayed in the **Model Builder** makes it possible to return to your attribute settings and change mesh sizes and distributions. After making any changes, click the **Build All** button ( ) or press F8 to rebuild the entire meshing sequence.

Mesh Control Entities

In this section:

- About Mesh Control Entities
- Using Mesh Control Entities to Control Element Size
- Using Structured and Unstructured Mesh with Boundary Layers

About Mesh Control Entities

Sometimes it is desirable to use certain geometric entities only to control the mesh. For example, you can add a curve inside a domain to control mesh element size there. If you mark this curve as a mesh control entity, it is not included in the geometry used when defining the physics. An advantage is that the final mesh need not respect this curve exactly; it is used only to control element size.

Another situation where you may want to use mesh control entities is when you need very precise control of mesh in certain regions of the geometry. In these regions you typically use a structured mesh with distribution features to control the mesh. In other regions of the geometry you may have to use free mesh.

Suppose that you also want to insert boundary layers. If the boundaries separating the domains with structured and free mesh are ordinary geometry boundaries, the boundary layers have to respect them. This can lead to various problems, including low-quality elements or even meshing failures. If you instead mark such boundaries as mesh control entities, the boundary layer mesh algorithm has more freedom to move mesh nodes and to construct a better mesh.

Using Mesh Control Entities to Control Element Size

Figure 19-11 shows a 2D geometry with two holes, and a Bézier Polygon that is intended not to be a part of the model but is included only to control mesh size inside the domain.

1 Right-click **Geometry** ( ) and select **Virtual Operations>Mesh Control Edges** ( ). Select the edges of the Bézier Polygon in the **Edges to include** selection.

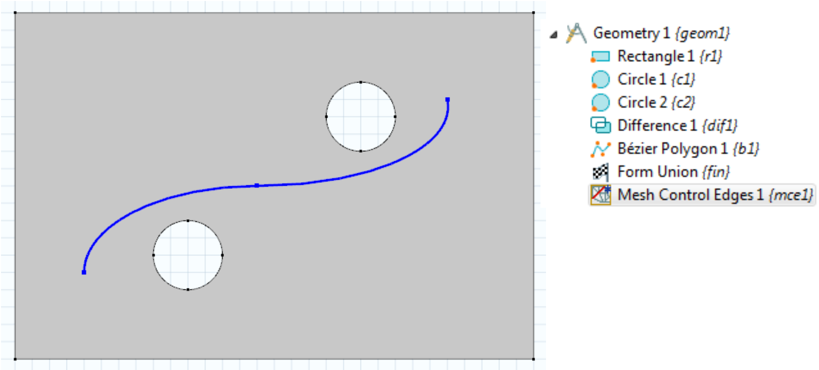


Figure 19-11: A geometry with a Bézier Polygon used to define mesh size inside the domain.

2 Click **Build Selected** ( ). Note that the selected edges are removed.

3 Right-click **Mesh** ( ) and select **Free Triangular** ( ). Note that the edges removed in the previous step are now visible again.

4 Right-click **Free Triangular 1** and select **Size** ( ).

5 Select Boundary as Geometric entity level, and select the edges of the Bézier Polygon.

6 Select the **Extra fine** as **Predefined** element size.

7 Click the **Build All** button (▦) or press F8 to build the entire mesh. Note that the edges of the Bézier Polygon are now removed (Figure 19-12) and that the only trace of them is the fine mesh size inside the domain.

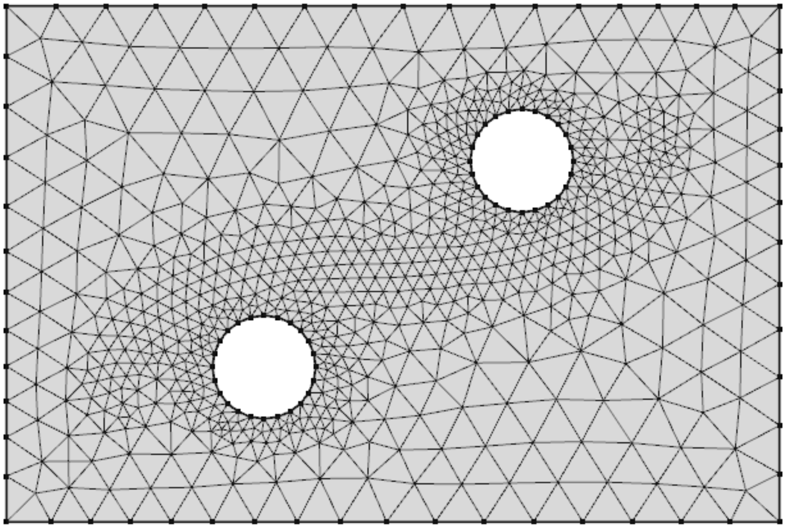
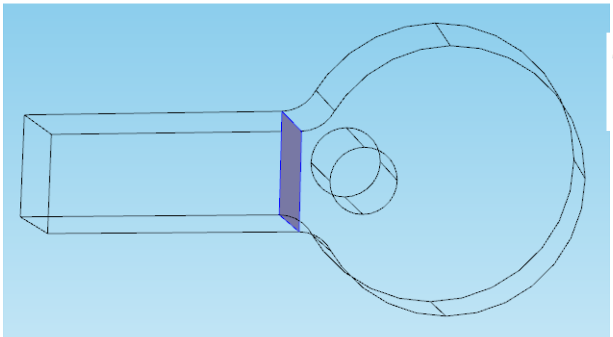


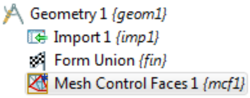
Figure 19-12: Fine mesh inside the domain.

Using Structured and Unstructured Mesh with Boundary Layers

This example demonstrates a geometry where free tetrahedral mesh is used in one domain, swept mesh is used in an other domain. The domains are separated by a mesh control face, which is automatically removed once the domains on both sides are meshed. Finally boundary layers are added, without the need to respect the (now removed) mesh control face.

1 Right-click **Geometry** ( ) and select **Virtual Operations>Mesh Control Faces** ( ). Select the face separating the domains in the **Faces to include** selection.





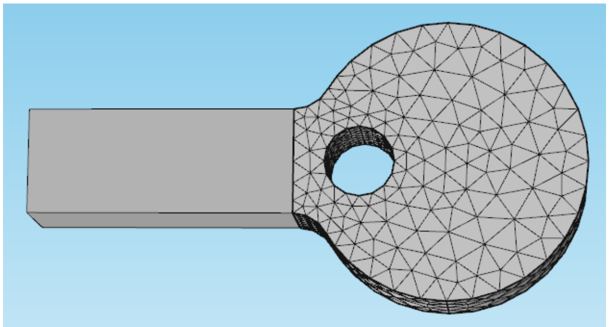
- Geometry 1 *{geom1}*
  - Import 1 *{imp1}*
  - Form Union *{fin}*
  - Mesh Control Faces 1 *{mcf1}*

2 Click **Build Selected** ( ). Note that the face is removed. There is now only one domain.

3 Right-click **Mesh** ( ) and select **Free Tetrahedral** ( ). Note that the face has reappeared, and that there are two domains.

4 Add the cylinder shaped domain with a hole to the selection list.

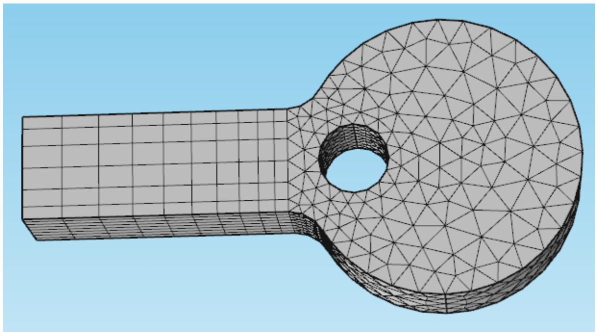
5 Click on the **Size** ( ) node, and select the **Finer** as **Predefined** element size. Build the mesh ( ).



6 Right-click **Mesh** ( ) and select **Swept** ( ).

7 Right-click **Swept 1** and select **Distribution** ( ).

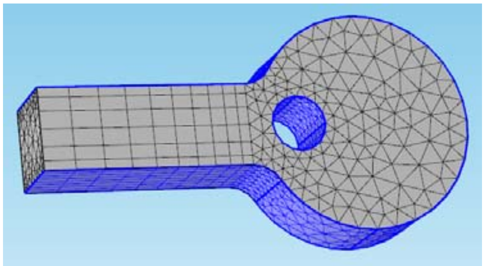
8 Select **Predefined distribution type** and enter 10 in the **Number of elements** field and 3 in the **Element ratio** field. Build the mesh ( ).

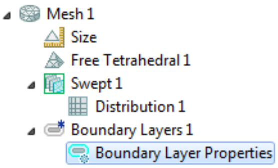


9 Right-click **Mesh** ( ) and select **Boundary Layers** ( ).

10 Click the **Boundary Layer Properties** ( ) node under **Boundary Layers 1**.

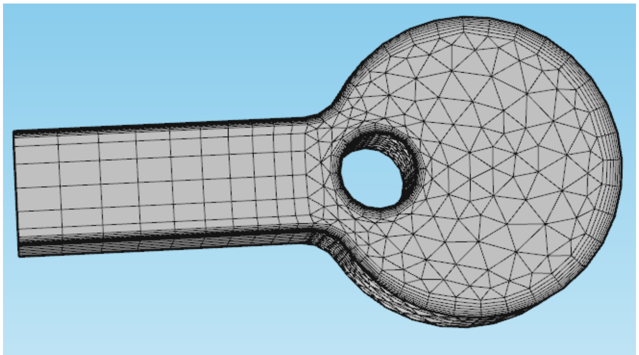
11 Add (for example) the sides of the geometry to the selection list.





- Mesh 1
  - Size
  - Free Tetrahedral 1
  - Swept 1
    - Distribution 1
  - Boundary Layers 1
    - Boundary Layer Properties

12 Click the **Build All** button (🏢) or press F8 to build the entire mesh. Note that the mesh control face is now removed, and that the boundary layer mesh nodes are not located where the boundary was.



Clearing Meshes and Deleting Meshing Sequences

Use a **Clear** function to keep the features and be able to recreate the mesh by building the sequence again. Use a **Delete** function to completely remove a mesh.

Important

Remember, **Undo** is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

CLEARING ALL MESHES FOR ALL GEOMETRIES

If you have a model geometry with several meshes you can clear all meshes at the same time. From the main menu, select **Edit>Clear All Meshes** ( ).

CLEARING A MESH FOR A SPECIFIC GEOMETRY

Under the **Geometry** node where you want to clear the mesh, right-click the **Mesh** node and select **Clear Mesh** ( ) from the context menu.

DELETING A MESHING SEQUENCE

To delete a meshing sequence completely, right-click the **Mesh** node in the **Model Builder** and select **Delete Sequence** ( ). Click **Yes** or **No** to confirm the deletion.

Displaying Mesh Element Quality and Size

Introduction

The mesh resolution and mesh element quality are important aspects to consider when validating a model. Low mesh resolution—in relation to the variations in the solution and the geometry—can lead to inaccurate results, and a low mesh element quality—which measures the regularity of the mesh elements' shapes—can lead to inverted mesh elements (see Avoiding Inverted Mesh Elements) and convergence issues.

COMSOL includes predefined variables for these mesh quantities:

- `h`, the local mesh size
- `qual`, the mesh element quality, which is a dimensionless quantity between 0 and 1, where 1 represents a perfectly regular element, and 0 represents a degenerated element.

Displaying Mesh Element Quality and the Mesh Element Size

You can display the mesh element quality and the mesh element size using, for example, a surface plot in 2D or a volume plot in 3D. You can always use a Mesh data set to display these quantities as soon as you have created a mesh. If you have a solution, you can also use a Solution data set. For a model with a mesh, do the following steps to display the mesh element quality or mesh element size:

1. Right-click the **Mesh** node and select **Plot** ( ). Doing so creates a plot group with a **Mesh** plot node ( ).
2. By default, this plot shows the mesh element quality. In the **Mesh** node's settings, select **Size** instead of **Quality** from the **Element color** list to plot the mesh element size instead.

Alternatively, you can access the variables for mesh element quality and size in a surface plot, for example:

1. Under **Results** ( ), right-click **Data Sets** ( ) and select **Mesh** ( ) to add a Mesh data set.

2 Add a 2D or 3D Plot Group using the Mesh data set as the group’s data set, and then add a Surface or Volume plot. For example, in a **2D Plot Group>Surface** node ( ), select **Mesh element size (h)** or **Mesh element quality (qual)** from the predefined quantities (under **Geometry and Mesh**).

3 Click the **Plot** button ( ) to display a plot of the selected mesh quantity.

For statistical information about the mesh element quality, right-click the **Mesh** node ( ) and select **Statistics** ( ). The **Statistics** window includes information about the minimum and average mesh element quality and a mesh element quality histogram, which shows the relative frequency of mesh elements with different quality values.

Avoiding Inverted Mesh Elements

In this section:

- Inverted Mesh Elements
- Using Linear Geometry Shape Order
- Modifying the Geometry or Mesh
- Visualizing Inverted Mesh Elements

Inverted Mesh Elements

If you have a mesh that is coarse along a curved boundary you might get problems with inverted mesh elements. This means that a mesh element is wrapped inside-out or has zero area (in 2D) or volume (in 3D). More precisely, there is some coordinate for which the Jacobian matrix for the mapping from local to global coordinates has a negative or zero determinant. In most cases, the linear (straight) mesh elements that you see in a mesh plot are not inverted, but the higher-order curved mesh elements used for computing the solution might be. Studying the minimum element quality therefore does not reveal the presence of inverted mesh elements in most cases.

Inverted mesh elements in themselves do not pose any immediate threat to the overall accuracy of your solution. However, if you are using an iterative solver, it might fail to converge. If you reach convergence and the solution looks good, it likely is. It is worth bearing in mind that the faces along which there are inverted elements are less than perfectly resolved. If these faces are important for your results, you might want to pursue a mesh without inverted elements or at least make sure that the mesh resolution is sufficiently fine to guarantee an accurate solution. The easiest way to get an idea of the accuracy is to try a few different meshes and see how the solution changes. If the variation does not exceed your limits of acceptance, you are fine.

The solver prints warnings about inverted mesh elements to the **Log** window (▤) if they appear. Warning nodes (⚠) also appear in the solver sequence where the inverted mesh elements appear. It is often possible to avoid problems with inverted mesh elements by reducing the geometry shape order or by modifying the geometry or the mesh.

Using Linear Geometry Shape Order

When solving a model, the solver ensures that no inverted mesh elements are created. This is done by reducing the geometry shape order for the corresponding elements to first order. By default, the solver does this automatically. Alternatively, you can avoid problems with inverted mesh elements by using linear geometry shape order for all elements. You do this by choosing **Linear** from the **Geometry shape order** list in the **Model Settings** section of the settings window for the main Model node.

Modifying the Geometry or Mesh

If you do not want to use linear geometry shape order to avoid problems with inverted mesh elements you can try any of the following:

- Create an extruded, revolved, or swept 3D mesh instead of using the free mesher.
- Avoid small curved boundaries such as fillets unless they are important for the result.

Visualizing Inverted Mesh Elements

You can visualize inverted mesh elements using the `reldetjacmin` variable, which is the minimum of the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates. A minimum value less than zero for an element means that the element is wrapped inside-out; that is, it is an inverted mesh element.

A typical visualization uses `reldetjacmin` as the quantity to plot as a volume plot. To display only the inverted elements, add a **Filter** subnode using the logical expression `reldetjacmin<0` to include only the inverted elements.

20

Solvers and Study Types

This chapter lists the Solver Studies and Study Types in COMSOL Multiphysics and explains the study steps and solver configurations. It also describes the major solvers and settings as well as batch jobs, parametric sweeps, and cluster computing. See also the *COMSOL Multiphysics Reference Guide* and the *Optimization Module User's Guide* for other supplementary information.

In this chapter:

- Solver Studies and Study Types
- Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Solver Studies and Study Types

In this section:

See Also

In the *COMSOL Multiphysics Reference Guide:*

- Solver Features
- Advanced Solver Topics

In the *COMSOL Java API Reference Guide:*

- Solver

Introduction to Solvers and Studies

The process of solving a problem in COMSOL Multiphysics is divided into a hierarchy of varying details. On the coarsest level, containing the least amount of detail, is the

Study node (). Selecting any of the predefined *study types* adds a Study node to the **Model Builder** tree () and also a corresponding *study step* and, in some cases, additional study steps. The study steps represent the next level of detail. Most study steps are used to control the form of the equations, which physics interfaces are included and which meshes are used. These study steps correspond to a part of a solver sequence, which is the next level of detail (for example, there are also study steps that are used for cluster computing, which correspond to a part of a job sequence). A solver sequence contains nodes that define variables to solve for, the solvers and their settings, and additional sequence nodes for storing the solution, for example. The solvers also contain nodes which allow you to control the solver settings at varying level of detail.

Solving a problem can amount to simply right-clicking the **Study** node for a predefined study type and selecting **Compute** ( ), which generates the a default solver sequence for the corresponding study steps and computes the solution. But you may also control the settings at any level of detail. There might not be any predefined study type which corresponds to the simulations you are interested in doing, in which case you can add individual study steps. Also, by changing the settings in the solver sequence, you can, for example, control the desired tolerance for the error in the solution or which time-integration method or linear solver to use. If you want to continue an interrupted parametric or time dependent solution, for example, right-click the **Study** node for a predefined study type and select **Continue** ().

Adding a Study

There are two ways to add a study to a model:

- You can add a study when creating a model using the **Model Wizard** (), which includes a **Select Study Type** page (the last page), where you can select from preset studies for the physics interfaces solved for and other, custom studies.
- You can also add a study by right-clicking the root node in the **Model Builder** and selecting **Add Study**. This opens the **Select Study Type** page in the **Model Wizard**. The list of preset studies is then based on the existing physics interfaces you choose to solve for in the new study, using the **Solve for** column in the **Selected physics** table. From the list under **Studies**, select **Add study** (the default) to add the selected study to the model (and keep all existing studies), or select, for example, **Replace Study 1** to replace an existing study.

You can have more than one study (each generating one or more solutions) for different scenarios using the same geometry and physics.

Available Study Types

The following study types are available in the Preset Studies branch of the **Studies** list on the **Select Study Type** page. Some of the study types are only available with certain add-on modules:

TABLE 20-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
AC Impedance Stationary		Creates two study steps, where the first solves for a stationary problem, and the second step solves for a harmonic perturbation in the frequency domain of the stationary solution. See AC Impedance Stationary in the *COMSOL Multiphysics Reference Guide*.
AC Impedance Time Dependent		Creates two study steps, where the first solves for a time-dependent problem, and the second step solves for a harmonic perturbation in the frequency domain of the solution at the last time step. See AC Impedance Time Dependent in the *COMSOL Multiphysics Reference Guide*.
Boundary Mode Analysis		A boundary mode analysis combines a mode analysis on a port (boundary) (which can represent, for example, a cross section of a waveguide) with a frequency domain study for the full geometry. Adds a Boundary Mode Analysis study step followed by a Frequency Domain study step to the Study node. See Boundary Mode Analysis in the *COMSOL Multiphysics Reference Guide*.
Coil Current Calculation		This study is available with the AC/DC Module using the Magnetic Fields interface coil features. See Coil Current Calculation Study in the *COMSOL Multiphysics Reference Guide*.
Eigenfrequency		Similar to an Eigenvalue study but computes the eigenfrequencies instead of the eigenvalues. Adds an Eigenfrequency study step to the Study node. See Eigenfrequency in the *COMSOL Multiphysics Reference Guide*.
Eigenvalue		This study uses a formulation to compute eigenvalues and eigenmodes using an eigenvalue solver. Adds an Eigenvalue study step to the Study node. See Eigenvalue in the *COMSOL Multiphysics Reference Guide*.
Empty Study		An empty study that initially does not contain any study steps.

TABLE 20-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Frequency Domain		For a study in the frequency domain such as solutions to a wave equation or frequency response. Adds a Frequency Domain study step to the Study node. See Frequency Domain in the *COMSOL Multiphysics Reference Guide*.
Frequency-Domain, Perturbation		This study is available with a variety of interfaces. See Frequency-Domain, Perturbation in the *COMSOL Multiphysics Reference Guide*.
Frequency-Domain Modal		For analyzing wave problems in the frequency domain using a modal solver, the Frequency Domain Modal study adds an Eigenfrequency study step followed by a Frequency Domain Modal study step to the Study node. See Frequency Domain Modal in the *COMSOL Multiphysics Reference Guide*.
Frequency-Stationary		This study is available with the Induction Heating and Microwave Heating interfaces and the settings are the same as for the Stationary study. See Frequency-Stationary in the *COMSOL Multiphysics Reference Guide*.
Frequency-Transient		This study is available with the Induction Heating, Microwave Heating, Inductively Coupled Plasma, and Microwave Plasma interfaces. The settings are the same as for the Time-Dependent study. See Frequency-Transient in the *COMSOL Multiphysics Reference Guide*.
Linear Buckling		A linear buckling study for a structural model includes a Stationary study step followed by a Linear Buckling study step, which solves for the critical load factor using an eigenvalue solver. See Linear Buckling in the *COMSOL Multiphysics Reference Guide*.
Mean Energies		The Mean Energies study is available with the Boltzmann Equation, Two-Term Approximation interface in the Plasma Module. See Mean Energies in the *COMSOL Multiphysics Reference Guide*.
Mode Analysis		A mode analysis computes the modes for an acoustics or electromagnetic wave using an eigenvalue solver. Adds a Mode Analysis study step to the Study node. See Mode Analysis in the *COMSOL Multiphysics Reference Guide*.
Particle Trajectories		Available with the Particle Tracing Module. See Particle Trajectories Study in the *COMSOL Multiphysics Reference Guide*.

TABLE 20-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Prestressed Analysis, Eigenfrequency		For computing eigenfrequencies which are influenced by a prior static load. It is necessary to have geometric nonlinearities active in order to perform this type of analysis. Adds a Stationary and an Eigenfrequency study step to the Study node. See Prestressed Analysis, Eigenfrequency in the *COMSOL Multiphysics Reference Guide*.
Prestressed Analysis, Frequency Domain		For computing the response to harmonic loads fluctuating around a static preload. It is necessary to have geometric nonlinearities active in order to take the influence on the stiffness from the preload into account. Adds a Stationary and a Frequency Domain study step to the Study node. See Prestressed Analysis, Frequency Domain in the *COMSOL Multiphysics Reference Guide*.
Reduced Electric Fields		The Reduced Electric Fields study is available with the Boltzmann Equation, Two-Term Approximation interface in the Plasma Module. See Reduced Electric Fields in the *COMSOL Multiphysics Reference Guide*.
Small-Signal Analysis, Frequency Domain		For perturbed frequency domain studies of small oscillation about a bias solution. The study creates two study steps, a Stationary study step solves for a stationary problem, and a Frequency Domain, Perturbation step solves for a harmonic perturbation in the frequency domain of the stationary solution. See Small-Signal Analysis, Frequency Domain in the *COMSOL Multiphysics Reference Guide*.
Stationary		For a stationary or steady-state situation where you can use a stationary solver and an equation formulation where all time derivatives vanish. This study type is also used for optimization problems which are constrained with a stationary PDE problem. Adds a Stationary study step to the Study node. See Stationary in the *COMSOL Multiphysics Reference Guide*.
Stationary with Initialization		For stationary two-phase flow models that require an initialization of a level set function or phase field function. Adds a Wall Distance Initialization study step followed by a Stationary study step to the Study node. See Stationary with Initialization in the *COMSOL Multiphysics Reference Guide*.

TABLE 20-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Stationary Plug Flow		For use with predefined Plug flow reactor type in the Reaction Engineering interface, available with the Chemical Reaction Engineering Module. Adds a Stationary Plug Flow step to the Study node that solves for the molar flow rate as function of reactor volume. See Stationary Plug Flow in the *COMSOL Multiphysics Reference Guide*.
Time Dependent		For a time dependent or transient simulation using a time dependent solver for computing the solution over time. This study type is also used for optimization problems which are constrained with a time dependent PDE problem. Adds a Time Dependent study step to the Study node. See Time Dependent and Time Discrete in the *COMSOL Multiphysics Reference Guide*.
Time-Dependent with Initialization, Fixed Geometry and Time Dependent with Initialization		Use this to exclude the deformation/ALE (X, Y, Z) variables from the variables that are actually solved for. Useful, for example, to look at time-dependent electrochemistry in cases were the mesh deformation is small. It adds a Current Distribution Initialization study step and Time-Dependent, Fixed Geometry study step to the Study node. See Time-Dependent with Initialization, Fixed Geometry and Time-Dependent with Initialization in the *COMSOL Multiphysics Reference Guide*.
Time-Dependent Modal		For analyzing time-dependent wave problems using a modal solver, the Time-Dependent Modal study adds an Eigenfrequency study step followed by a Time-Dependent Modal study step to the Study node. See Time-Dependent Modal in the *COMSOL Multiphysics Reference Guide*.
Transient with Initialization		For time dependent two-phase flow models that require an initialization of a level set function or phase field function. Adds a Wall Distance Initialization study step followed by a Time Dependent study step to the Study node. See Transient with Initialization in the *COMSOL Multiphysics Reference Guide*.

See Also

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Additionally, you can add *adaptive mesh refinement* (adaptive remeshing) to a Stationary, Eigenvalue, Eigenfrequency, or Time Dependent study by selecting the **Adaptive mesh refinement** check box in the **Study Extensions** section of the study step's settings window. The solver then minimizes the error according to a specified error norm by adaptively refining the mesh.

For a parametric stationary PDE problems that arise when you vary some parameter of interest in the equations, constraints, or expressions, you can activate the *parametric continuation solver* by selecting the **Continuation** check box and specifying the continuation parameter and its values in the Stationary study step's settings window. To make a parametric study where the parameter varies the geometry, for example, use a parametric sweep.

Adding Study Steps

The predefined study types correspond to the most commonly performed simulations for the different physics interfaces. However, sometimes you might want to do other investigations. For example, you may first want to solve a stationary problem for a physical quantity and use that solution as input to a time dependent simulation for another physical quantity. You can do this by adding (by right-clicking the study node) a **Stationary** study step () followed by a **Time Dependent** study step () and selecting in each study step which physics interfaces to include.

See Also

See Study Types in the *COMSOL Multiphysics Reference Guide* for details about each of the following features.

Most study steps correspond to a part of a solver sequence which includes a solver for the specific problem:

- **Stationary**: Generates equations without time derivatives. Corresponds to a **Stationary solver**.

Note

If the **Optimization** check box is selected, then the Stationary study step corresponds to an **Optimization** solver. This solver requires an Optimization Module license.

- **Time Dependent**, **Time Discrete**: Generates equations for transient (time-dependent) simulations. Corresponds to a **Time Dependent Solver, Time Discrete Solver**, or **Time Explicit Solver**.
- **Eigenvalue**: Generates equations formulated for computing eigenvalues and eigenfunctions. Corresponds to an **Eigenvalue Solver**.
- **Eigenfrequency**: Similar to an **Eigenvalue** study step. Corresponds to an **Eigenvalue Solve**r which is set to transform eigenvalues to eigenfrequencies.
- **Frequency Domain**: Generates stationary equations which are used for a frequency sweep. Corresponds to a stationary parametric solver that is preset to linearize the equations. By selecting the **Use asymptotic waveform evaluation** check box, this study step corresponds to an **AWE Solver**.
- **Time-Dependent Modal:** Generates equations for time-dependent modal analysis. Corresponds to a **Modal Solver** with **Study type** set to **Time dependent**.
- **Frequency Domain Modal**: Generates equations for modal analysis in the frequency domain. Corresponds to a **Modal Solver** with **Study type** set to **Frequency domain**.

There are some study steps that do not generate equations and which can only be used in combination with other study steps. These study steps do not correspond to any part of a solver sequence. Instead, they correspond to a part of a job sequence. The available study steps of this type are:

- **Parametric Sweep** (): Used to formulate a sequence of problems that arise when you vary some parameters in the model. The problem at a fixed parameter value is defined by the rest of the study steps in the study. Generates a **Parametric** job sequence, unless the problem and parameters are such that the parametric sweep can be realized through a **Stationary Solver** with a **Parametric** node (), in which case such a solver is generated in the solver sequence. The parametric sweep can include multiple independent parameters directly, but you can also add more than one

Parametric Sweep node to create nested parametric sweeps. In the **Study** branch, indentations of the node names indicate that the parametric sweeps are nested.

- **Batch** ( ): Creates a job that can be run without the graphical user interface and which stores the solution on disk. Generates a **Batch** job sequence.
- **Batch Sweep**: Used to formulate a sequence of problems that arise when you vary some parameter in the model. Each parameter tuple generates a batch job that runs the model with the given tuple. The results are stored on file and updated into the model. Generates a **Batch** job sequence and a **Parametric** job sequence.
- **Cluster Computing** ( ): Used to solve the problem on a distributed-memory computer architecture. Generates a **Cluster Computing** job sequence and a **Batch** job sequence.
- **Cluster Sweep**: Used to formulate a sequence of problems that arise when you vary some parameter in the model. Each parameter tuple will solve the problem on a distributed-memory computer architecture. The results are stored on file and updated into the model. Generates a **Cluster Computing** job sequence and a **Batch** job sequence and a **Parametric** job sequence.

Working With Study Steps

The study steps form a solver sequence that computes the solutions for the study. The study steps contain the following sections—in addition to specific study settings for each type of study step—in the **Setting** windows that can be used to control the study.

RESULTS WHILE SOLVING

Specify which plot to display while solving and also which probes to include.

MESH SELECTION

Specify—for each geometry—which mesh to use for the study step.

PHYSICS AND VARIABLES SELECTION

Control and specify different cases where the physics to solve for is varied, or, for various analysis cases, which variables and physics features (for example, boundary conditions and sources) to use. The default is to solve for all physics interfaces that are compatible with the study type.

STUDY EXTENSIONS

These are extensions to the study's main solver such as adaptive mesh refinement and automatic remeshing. The options vary depending on the study type.

VALUES OF DEPENDENT VARIABLES

Specify the source of the values of variables that you do not solve for. It can be the specified initial values or the values from an existing solution.

See Also

In the *COMSOL Multiphysics Reference Guide:*

- Selecting Physics and Variables in the Study Steps
- Individual study steps described for the Study Types

Generating Solver Configurations and Job Configurations

When you have added the study steps to the study, a solver sequence (and maybe a job sequence) is generated after right-clicking the study and selecting **Compute** (☰).

The generated solver sequence is displayed under the **Solver Configurations** node ( ) and if applicable, under a **Job Configurations** node ( ). The **Job Configurations** node ( ) is only visible if you click the **Show** button ( ) and select **Advanced Study Options**. The node automatically displays if the **Job Configuration** node has content.

It is possible to have several solver sequences under a Solver Configuration node. The particular sequence that is enabled and runs when selecting **Compute** has a green border around its icon. An enabled sequence can be disabled by right-clicking it and selecting **Disable** (⊘). If no sequence is enabled when the study attempts to generate a sequence, a new sequence with default settings is generated. Only one sequence per study can be enabled.

See Also

Job Configurations and Solver Configurations in the *COMSOL Multiphysics Reference Guide*.

Editing Solver Configurations

The study automatically generates the nodes in the **Solver Configurations** branch that represents the solvers, dependent variables, and other study-related functionality that the study steps require. You can edit the nodes in the **Solver Configurations** branch to, for example, adjust solver settings. If you edit any settings in a subnode to a **Solver** node, an asterisk in the **Solver** node's icon indicates that the settings differ from the default settings for the study types in the study. You can show the nodes in the **Solver**

Configurations branch before solving by right-clicking the **Study** node and selecting **Show Default Solver** ( ).

Clearing, Deleting, Disabling, and Enabling a Solver or Solutions

Use a **Clear** function to keep the features and be able to recreate the solver configurations by computing again. Use a **Delete** function to completely remove a solver configuration or solver sequence.

Note

The **Undo** operation is not possible for nodes that are built directly, such as geometry objects, meshes, solutions, and plots.

DISABLING AND ENABLING A SOLVER NODE

- To disable a solver, right-click it and select **Disable** ( ).
- To enable a disabled solver, right-click it and select **Enable** ( ). The icon of the enabled solver has a green border.

CLEARING ALL SOLUTIONS FOR ALL MODELS

To clear all solutions for all models at the same time, from the main menu, select **Edit>Clear All Solutions** ( ).

CLEARING A SOLUTION

To clear a set of solutions under a specific study, right-click the **Solver Configurations** node and select **Clear Solutions** ( ).

CLEARING A SOLVER SOLUTION

Right-click the **Solver** node under **Solver Configurations** and select **Solutions>Clear** ( ) from the context menu.

DELETING A SOLVER NODE

To delete one **Solver** node under **Solver Configurations**, right-click the corresponding node in the **Model Builder** and select **Delete** ( ) or press the Delete key. Click **Yes** or **No** to confirm the deletion.

DELETING ALL SOLVER NODES UNDER SOLVER CONFIGURATIONS

To delete all **Solver** nodes under **Solver Configurations**, right-click **Solver Configurations** and select **Delete Solvers** ( ). Click **Yes** or **No** to confirm the deletion. You can also choose whether or not to remove the Results nodes (data sets and plots, for example) associated with the solver configuration.

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

The study type adds appropriate solvers for the study, so that you do not need to select one yourself. If you nevertheless prefer to make a selection, the first question to ask is whether the problem is stationary or time-dependent.

Most real-world phenomena develop in time, but you might know that the system under study approaches a *steady state* described by a stationary solution.

For a stationary problem, select the stationary solver. When solving the time-dependent coefficient form problem

$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au = f$$

the stationary solver searches for a solution where $\partial u/\partial t = 0$.

The time-dependent solver and the stationary solver both handle linear as well as nonlinear problems.

In some cases you might want to study the natural harmonic oscillations of a time-dependent problem. This involves finding the eigensolutions u with their associated eigenvalues λ in a PDE problem of the following form:

$$\nabla\cdot(-c\nabla u-\alpha u)+\beta\cdot\nabla u+au = \lambda d_a u-\lambda^2 e_a u$$

Such an analysis is particularly interesting in electromagnetics, structural mechanics, acoustics, and wave propagation. To study the eigensolutions and compute the eigenvalues (or eigenfrequencies), select the eigenvalue solver.

In addition to these fundamental solvers, COMSOL Multiphysics includes additional solvers for special applications such as an optimization solver, a time-discrete solver, an AWE solver, and a plugflow solver. Some of these solvers are connected to special functionality in the add-on modules.

See Also

Solution Operation Features in the *COMSOL Multiphysics Reference Guide*

Linear Solvers vs. Nonlinear Solvers

AUTOMATIC NONLINEARITY DETECTION

COMSOL Multiphysics automatically detects nonlinearity, so you normally do not need to decide whether to use a linear or a nonlinear solver.

The automatic detection works through analysis of the variables contributing to the residual Jacobian matrix and the constraint Jacobian matrix. If the algorithm finds that both these matrices are complete and do not depend on the solution, the stationary solver (including parametric sweeps) uses a linear solver algorithm. Otherwise, the solver uses a nonlinear solver algorithm. “Complete” here means that the algorithm only found contributing variables for which the correct Jacobian will be computed.

OVERRIDING THE AUTOMATIC NONLINEARITY DETECTION

In some cases you might want to specify explicitly that the stationary solver uses the linear or nonlinear solver algorithm. Such cases include:

- Linear models where the automatic detection of linearity makes COMSOL Multiphysics use the nonlinear solver. This can happen, for example, for models that involve some less common types of coupling variables (directly or indirectly as part of some boundary conditions). The nonlinear solver usually converges directly for linear problems, but if that is not the case, you can switch to the linear solver.
- Using the linear solver to single-step Newton’s method for a nonlinear problem.
- Using the linear solver to solve a linearized (nonlinear) problem.

To explicitly control which solver algorithm the stationary solver uses, click the **Stationary Solver** node to open its settings window and locate the **Linearity** list in the **General** section. Then, from the **Linearity** list, select **Linear** to use the linear solver, **Nonlinear** to use the nonlinear solver, or **Linear perturbation** for small-signal analysis, for example. With the default setting, **Automatic**, COMSOL Multiphysics detects nonlinearity automatically.

See Also

Stationary in the *COMSOL Multiphysics Reference Guide*

WHICH MODELS ARE NONLINEAR?

How do you determine if a problem is linear or nonlinear? Finding out is not always easy, but for most of the physics interfaces you can apply the following criterion: if any coefficient or material property contains a dependent variable the model is nonlinear. The same holds true for models based on a PDE in the coefficient form, again with the same criterion.

Note a few special cases that arise with some modeling interfaces. First, in the Heat Transfer interfaces, if you include radiation terms for black-body radiation, which depend on temperature according to the Stefan-Boltzmann law, the problem is nonlinear. Second, the Single-Phase Flow interface is always nonlinear unless $\rho = 0$, resulting in the linear Stokes equations.

Whether your problem is linear or nonlinear, it is important to realize that the solvers break it down into one or several linear systems of equations. The choice of solver for linear systems thus affects the solution time and memory requirements also for nonlinear models.

See Also

Linear System Solver Selection Guidelines in the *COMSOL Multiphysics Reference Guide*

Remarks on Solver-Related Model Characteristics

THE IMPORTANCE OF A CORRECT JACOBIAN MATRIX

As mentioned above, the solvers break down each problem—whether linear or nonlinear—into one or several linear systems of equations by approximating the given problem with a linearized problem. The coefficient matrix of the discretized linearized problem is called the *Jacobian matrix* (or *stiffness matrix*). In most cases COMSOL Multiphysics computes a correct Jacobian matrix, especially when using the weak solution form. For other cases, though, the software treats certain terms in the PDEs and boundary conditions as if they did not depend on the sought solution, and then an incorrect Jacobian results.

The consequences of an incorrect Jacobian matrix depend on the solver in use:

- The linear stationary solver and the eigenvalue solver simply give an incorrect solution.
- The nonlinear stationary solver and the time-dependent solver take longer time to convergence to the correct solution, and in some cases the solver even fails to find a solution. However, if the ignored terms have a very weak dependence on the sought solution, the impact on convergence speed is small.

An incorrect Jacobian matrix can occur in the following cases:

- If you supply an incorrect derivative of some of the functions in the **Functions** dialog box, and the software uses this function in some PDE or boundary condition with arguments that depend on the solution.
- If you use a MATLAB M-file function for which you have specified no derivative, and then call this function in some PDE or boundary condition with arguments that depend on the solution. To avoid this problem, define the function and its derivatives.
- If you use nonanalytic functions in a complex-valued problem, such as `real`, `imag`, `conj`, `abs`, or `arg`.
- If you use the `nojac` operator.
- If you use a solution form that is not suited for the equations you are solving. Selecting **Automatic** in the **Solution form** list on the **Advanced** page in the **Solver Parameters** dialog box ensures a correct solution form (this is the default setting).

See Also

- Global and Local Functions
- MATLAB Functions (Global Only)

WORKING WITH COMPLEX-VALUED PROBLEMS

When a problem contains complex numbers, be sure to consider the following two aspects:

- For time-dependent problems, the time-stepping algorithm must know if a problem is complex valued. If your model uses a complex-valued initial solution, COMSOL Multiphysics detects this and classifies the problem accordingly. If the problem is complex-valued because of a complex-valued PDE coefficient or other material property, go to the **Solver Parameters** dialog box, find the **General** page, and select the **Allow complex numbers** check box.

- If you expect to receive complex outputs from real inputs in elementary functions such as `sqrt`, `log`, and `asin`, then go to the **Solver Parameters** dialog box, click the **Advanced** tab, and select the **Use complex functions with real input** check box. By default, COMSOL Multiphysics gives an error message if a real input to a function generates a complex output, for instance `sqrt(-1)`.

Computing a Solution

Computing a solution can be done in a few different ways. The most straightforward way is to right-click the **Study** node and select **Compute** ( ), or to press F8. This always runs the enabled solver sequence with a green border around its icon. However, if a solver sequence is disabled, you can still run it by right-clicking it and selecting **Compute** ( ).

Sometimes, the default settings in the study steps are not sufficient to specify the details of how to obtain a solution. For example, you may want to change a tolerance or use a different time-stepping method. This is done in the attached solver sequences. If you want to run such a solver sequence, which has been manually altered, right-click the solver sequence and select **Compute** ( ). Alternatively, enable the solver sequence, right-click the study, and select **Compute** ( ).

If your solver sequence consists of several solvers you can run a particular solver by right-clicking it and selecting **Compute to Selected** ( ), or by pressing F7. You can also choose **Compute From Selected** ( ) to run the selected solver and all solvers below it in the sequence. Choosing the option **Compute All** ( ), or pressing F8, runs all solvers in the solver sequence.

By default, a study creates a **Solution** data set and plot groups with results plots suitable for the physics interfaces that you compute the solution for. If you do not want to generate plots automatically, clear the **Generate default plots** check box in the **Study Settings** section in the main **Study** nodes' settings windows.

Updating the Model

To update the model for a particular study, right-click the **Study** node and select **Update Solution** ( ) or press F5. Pressing F5 updates the current study if selected or all studies if no study is selected. This is useful in the following situations when you have:

- Added or edited variables and want to use these during analysis without having to solve the model again.
- Edited the equations, material properties, or boundary conditions and want to use these without having to solve the model again.
- Changed the element order and want to interpolate the solution onto the new elements for results analysis or other purposes.
- Remeshed or modified a geometry and want to interpolate the solution onto the new geometry for results analysis or other purposes.

In all these cases the software passes, or interpolates, the solution to the resulting data sets, but does not recompute it to reflect any changes in variables, equations, mesh, or geometry.

Computing the Initial Values

In some situations you may want to evaluate and plot values, expressions, or functions that need not be solved for, such as initial values. To make the initial values available for results evaluation and plotting, generate a solver sequence, right-click the **Dependent Variables** node (u,v,w), and select **Compute to Selected** ( ). By default this plots the initial values of the variables solved for as specified by the **Field** subnode ( ). More precisely, choosing **Initial values** from the **Keep solution** list, makes the variables solved for available for results evaluation and visualization. To evaluate the initial values the settings of the corresponding **Method** list is used. The method **Initial expression** uses the settings of the **Initial Values** node, whereas **Solution** uses a solution object.

Note

The **Compute to Selected** ( ) command runs everything above the **Dependent Variables** node, including any solvers, which might be computationally expensive.

(Note, however, that this runs everything above the **Dependent Variables** node, including any solvers, which might be computationally expensive.)

Also variables not solved for (such as a solution from a previous time dependent or parametric analysis) can be made available for results evaluation and visualization. Select **Variables not solved for** from the **Keep solution** list and edit the corresponding **Method** list accordingly.

Getting Results While Solving

The ability to get numerical results and plots of the intermediate solution while solving can be useful for diagnosing problems and for monitoring the progress of the solution process. COMSOL Multiphysics supports two ways of displaying results while solving:

- Plots in the **Graphics** window: you can select any of the plot groups in the model to display while solving.
- Probe data in the **Results** window and associated line graphics in a separate **Probe Plot** window: you can include any probe defined in the model. See Probes.

In the study step nodes settings windows' **Results While Solving** sections you can control which plots to display while solving: Select the **Plot** check box to activate the plotting in the **Graphics** window. You can then select the plot group to use and, for time dependent simulations, at which time steps to update the plot: the output times or the time steps taken by the solver. You can also control which probes to tabulate and plot the values from. The default is to tabulate and plot the values from all probes in the **Results** window and a **Probe Plot** window.

Progress and Log Information

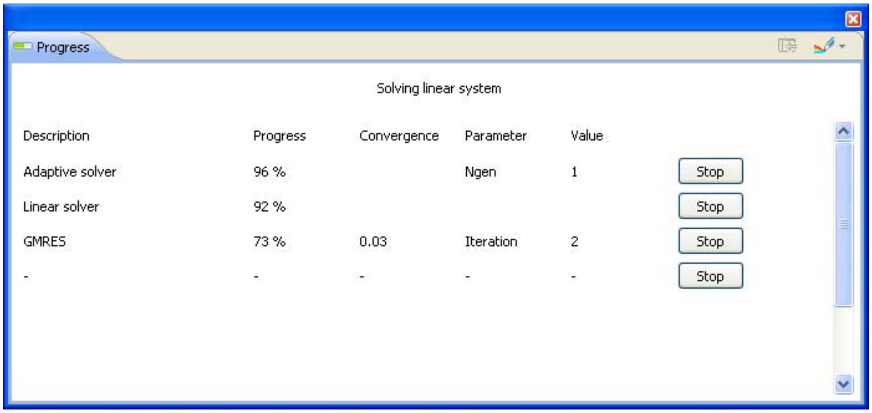
While a problem is being solved it is useful to know its progress. For that purpose COMSOL Multiphysics provides the **Progress** ( ) and **Log** ( ) windows, which are selected from the **View** menu. You can monitor the progress for the solvers during the solution process in the **Progress** window and inspect convergence information and other data from the latest and earlier runs in the **Log** window.

CANCELING SOLVERS

Sometimes you might realize that you are solving the wrong problem, or perhaps it is taking too long; to cancel the process, click the **Cancel** button in the **Progress** window. The solver attempts to cancel the computation as soon as possible and return the program to where it was before starting to solve.

THE PROGRESS WINDOW

The solvers call each other in a hierarchical order: the adaptive solver calls the linear, nonlinear, parametric, eigenvalue, or time-dependent solver; the parametric solver calls the nonlinear or linear solver; the time-dependent, eigenvalue, linear, and nonlinear solvers all call both the assembly and the linear system solver. The solver hierarchy is visible in the **Progress** window ( ) because each solver adds its own line when it is called.



Solving linear system

Description	Progress	Convergence	Parameter	Value
Adaptive solver	96 %		Ngen	1
Linear solver	92 %			
GMRES	73 %	0.03	Iteration	2
-	-	-	-	-

Figure 20-1: The Progress window for the solution of a linear stationary problem using the adaptive mesh refinement solver.

The **Progress** window contains the following columns:

- The **Description** column shows the solver's name.
- The **Progress** column shows an estimate of the solver's progress.
- The **Convergence** column shows an estimate of the solver's convergence if available.
- The **Parameter** and **Value** columns contain solver-dependent information: the adaptive solver shows the adaptive mesh generation number; the time-dependent solver shows the time; the parametric solver shows the parameter value; and the nonlinear solver and iterative linear system solvers show the iteration number.

Using the Stop Buttons

You can use the **Stop** buttons that appear at each solver level to stop the solver's execution. When you click a **Stop** button, COMSOL Multiphysics returns a current approximation to the solution if possible. For example, when you click it at the adaptive solver level, the underlying linear, nonlinear, parametric, eigenvalue, or time-dependent solver continues until it is finished, but the adaptive solver stops at its current generation, immediately returning a solution for analysis. Similarly, you can click the **Stop** button to return the current iteration for the nonlinear solver or an iterative solver. Also use the **Stop** button during time stepping to return all time steps up to the current one. The parametric solver works similarly: to return the solutions for all parameter values up to the current one, click the **Stop** button.

Tip

There is also a **Cancel** button (✖) on the status bar in the lower-right corner of the COMSOL Desktop that you can use to cancel the solver process (that is, stop without returning any solution data).

THE LOG WINDOW

The **Log** window (▤) contains logs from the solvers. When a solver starts, the log window displays logs from all solver calls.

The Adaptive and Parametric Solver Logs

The adaptive solver prints a section in the log for each adaptive generation containing the current number of elements and an error indicator. The parametric solver similarly outputs one section to the log for each parameter value.

The stationary, time dependent, and eigenvalue solvers log their iterations.

The Nonlinear Solver Log

The log from the nonlinear solver contains the following information:

- The iteration number (**Iter**).
- A relative error estimate (**ErrEst**).
- The damping factor used in each Newton step (**Damping**).
- Fraction of Newton and Cauchy steps for Double dogleg (**Newton, Cauchy**).
- The size of the undamped Newton step (**Stepsize**) in the error estimate norm.
- The numbers of residuals (**#Res**), Jacobians (**#Jac**), and linear-system solutions computed (**#Sol**) so far.

The Iterative Linear System Solver Logs

The iterative linear system solvers produce a log that additionally contains the total number of linear iterations (**LinIt**), a relative error estimate (**LinErr**), and the relative residual (**LinRes**). The relative error estimate is a factor times the relative (preconditioned) residual. The relative residual is the Euclidean norm of the residual divided by the norm, $|b|$, of the linear system's right-hand side.

See Also

Convergence Criteria in the *COMSOL Multiphysics Reference Guide*

The Time Dependent Solver Log

The time-stepping algorithm produces a log that contains

- The time step number (**Step**).
- Time (**Time**; output times are indicated with **out**).
- The step size (**Stepsize**).
- The number of residuals (**Res**), Jacobians (**Jac**), and linear system solutions (**Sol**) computed.

You can see also the order of accuracy of the method (**Order**), the number of error test failures in time stepping (**Tfail**), and the number of failures in the nonlinear iterations (**NLfail**). For iterative linear system solvers, the log also contains the total number of linear iterations, a linear error estimate, and a relative residual (see above).

The Eigenvalue Solver Log

The eigenvalue solver produces a log that contains the iteration number (**Iter**), an error estimate (**ErrEst**), the number of converged eigenvalues (**Nconv**), and— if you are using an iterative linear solver— the number of linear iterations (**LinIt**).

The Optimization Solver Log

Note

The **Optimization** solver requires the Optimization Module.

The optimization algorithm used by the solver **SNOPT** is an iterative procedure that involves *major* and *minor* iterations. A major iteration results in a new solution candidate. For each major iteration, the optimization solver solves a quadratic-programming subproblem using an iterative procedure; these iterations are the minor iterations.

The log produced by the optimization solver **SNOPT** contains the following data:

- The cumulative number of minor iterations (**Itns**).
- The current major iteration number (**Major**).
- The number of minor iterations for the current major iteration (**Minor**). This value should be 1 when the solver is close to the solution.
- The step length taken in the current search direction (**Step**). This value should be 1 when the solver is close to the solution.
- The number of times the multiphysics model has been solved (**nPDE**).
- The maximum complementarity gap (**Error**). It is an estimate of the degree of non-optimality of the reduced costs. For convergence, this value should be smaller than the **Optimality tolerance**.
- The current value of the objective function (**Objective**).

The optimization solver **Levenberg-Marquardt** is an iterative procedure used to solve least squares problems. The log produced by the solver **Lenenberg-Marquardt** contains the following data:

- The number of Levenberg-Marquardt iterations (**Itns**).
- The current Levenberg-Marquardt factor (**lmFact**). A small factor typically indicates fast convergence.
- The number of times the multiphysics model has been solved (**nPDE**).
- The maximum absolute value of the gradient (with respect to the control variables) of the objective function (**Gradient**).
- The estimated error based on the gradient, the objective function and the control variables (**Error**). For convergence, this value should be smaller than the **Optimality tolerance**.
- The current value of the objective function (**Objective**).

Convergence Plots

Convergence plots use graphics to show how an error estimate or time step evolves during the solution process for nonlinear, time-dependent, and parametric solvers.

CONVERGE PLOT SETTINGS

In the **Preferences** dialog box, click **Results** and then select the **Generate convergence plots** check box for the convergence plots to appear. This is the default setting. There is also a **Convergence plot buffer size (steps per plot)** field that controls the size of the buffer used for storing the steps in the convergence plot. The default value is 10,000 steps. You can also control whether to generate convergence plots or not for a study from the main **Study** node using the **Generate convergence plot** check box in the **Study** node's settings window.

To control which solvers that generate convergence plots, use the **Convergence Plot Settings** menu button ( ) on the **Progress** window's toolbar. Click the button to select or clear the convergence plots for each solver. For example, for a nonlinear time-dependent model, the menu that appears when you click the menu button contains the following solvers:

- Nonlinear solver
- Time-dependent solver (Generalized-alpha)

By default, all solvers are selected, and the convergence plot for each solver appears in its own **Convergence Plot** window. Clear the check mark for a solver to turn off its convergence plot.

Click the **Copy Convergence Data to Model** button ( ) on the **Progress** window's toolbar to copy the convergence data to a table in a **Results** window.

Note

NaNs (Not-a-Number entries) that appear in the tabulated convergence data represent breaks between multiple convergence cycles in an interactive solution process.

CONVERGENCE INFORMATION IN THE PLOTS

The convergence plots show an error estimate against the iteration number for the nonlinear solver and for the iterative linear system solvers (the Conjugate gradients, BiCGStab, GMRES, FGMRES, and multigrid solvers).

For the nonlinear solver, the convergence plots show the relative error estimate for each nonlinear iteration number. This number also appears in the **Log** window as **ErrEst**.

For the iterative linear system solvers, the error estimate for each linear iteration is a factor times the relative (preconditioned) residual. This number also appears in the **Log** window as **LinErr**. When these solvers are used together with the nonlinear solver, the graphs for the different linear-system solution steps are merged, and the plots use the accumulated number of iterations.

When using the parametric solver, the graphs for the different parameter steps are merged, and the convergence plots use the accumulated number of iterations.

When using a time-dependent solver, the graph in the **Convergence Plot** window shows the reciprocal of the time step size versus the time step. That is, a convergence plot with decreasing values shows that the time-dependent solver takes longer time steps, and vice versa.

The error estimate numbers for the last iteration also appear in the **Convergence** column in the **Progress** window.

Click the **Copy to Figure** button to create a copy of the convergence plot in a separate figure window.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis

Tip

In general, how the perturbation concept (and the study) is named is based on the application. For example, in the AC/DC Module it is referred to as *small-signal analysis*, whereas in the Structural Mechanics Module it is referred to as *prestressed analysis*. For the Batteries & Fuel Cells Module and the Electrodeposition Module, the studies are called *AC impedance*. For the CFD Module, and fluid flow in general, the term *perturbation* is sufficient.

See Also

See these studies in the *COMSOL Multiphysics Reference Guide* for details about availability by module and interface:

- Frequency-Domain, Perturbation
- Small-Signal Analysis, Frequency Domain
- Prestressed Analysis, Eigenfrequency
- Prestressed Analysis, Frequency Domain
- AC Impedance Stationary
- AC Impedance Time Dependent

In this section:

- Frequency-Domain Perturbation Study and Study Step
- Harmonic Perturbation—Exclusive and Contributing Nodes
- Prestressed Frequency Analyses Study

Model

If you have the Acoustics Module and AC/DC Module, see Loudspeaker Driver: Model Library path **Acoustics_Module/Industrial_Models/ loudspeaker_driver**.

If you have the AC/DC Module, see Small-Signal Analysis of an Inductor: Model Library path **ACDC_Module/Inductive_Devices_and_Coils/ small_signal_analysis_of_inductor**.

Frequency-Domain Perturbation Study and Study Step

Use perturbed frequency domain analysis to study small oscillations about a bias solution. With the **Frequency-Domain, Perturbation** study and study step ( ), small-signal analysis (ac/dc), prestressed analysis (structural mechanics), or harmonic perturbation (fluid flow) types of analyses can be performed. When this is added as a Study from the Model Wizard, two study steps are added under the Study node. When it is added as a study step (for example, with the **AC Impedance Stationary** study), it is also part of a two step study that is added under the Study node.

In all cases, the first step is usually a **Stationary** study step that computes the stationary (or bias) solution. The second step is this **Frequency-Domain, Perturbation** step, which computes a perturbed solution of the linearized problem around the linearization point (or bias point) computed in the first step.

Usually, two different right-hand-side contributions (or loads) must be defined for each step. The first step needs a stationary value for the contribution, and the second step needs the perturbed value for the contribution. The definition of these contributions differs between exclusive and contributing features, and this relates to the **Harmonic Perturbation** node, which can be added to a wide variety of physics interface features (for example, the **Electric Potential** and **Electric Ground** nodes for the **Electric Currents** interface).

Model

If you have the Acoustics Module and AC/DC Module, see Axisymmetric Condenser Microphone with Electrical Lumping: Model Library path **Acoustics_Module/Industrial_Models/ condenser_microphone_lumped**.

See Also

For different plot settings made available by using this feature, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

Harmonic Perturbation—Exclusive and Contributing Nodes

A feature that is exclusive has a **Harmonic Perturbation** subfeature. This subfeature adds harmonic perturbations to the right-hand-side contributions of its parent node (for example, a **Boundary Load** on the **Solid Mechanics** interface or a **Terminal** node on the **Electric Currents** interface). In the settings window, the perturbation is entered for these contributions, which is only used when you solve for a **Frequency-Domain, Perturbation** study type. The parent node defines the stationary value for the contribution, which is not present for the **Frequency-Domain, Perturbation** study.

Tip

Harmonic perturbation nodes have a tilde over the top of the node, as in this example of a boundary level node ( ).

Contributing nodes have a yellow dot to the left of the node, as in this example of a boundary level node ( ).

Exclusive nodes are of two types—Override ( ) and Overridden ( )— and in both types a red arrow is used at the top (override) or bottom (overridden) left corners.

Features that are contributing (typically sources) may add their contributions as a harmonic perturbation. To define the stationary value for the contribution, you typically add another feature of the same type with the harmonic perturbation setting cleared.

For exclusive loads there is only one way of doing it due to the exclusivity—as a subfeature. This subfeature cannot, in general, be a full copy of the original feature because only some subsets of data may be changed.

As an example (prescribed displacement in structural mechanics), the prescribed displacement must have the same local system and the harmonic perturbation can only be applied to degrees of freedom already prescribed in the parent node.

Also, all contributing features are free to use the full set of settings. A static point load can be at one point in the global direction, and a local system for the harmonic contribution can be used.

See Also

- Physics Exclusive and Contributing Feature Node Types
- Key to the Nodes and Toolbar Buttons
- For different plot settings made available by using this feature, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

Prestressed Frequency Analyses Study

Vibrating MEMS structures are often prestressed. For example, a cantilever structure could be prestressed by applying a DC voltage bias between the cantilever and a nearby ground plane, then vibrations could be driven at resonance by applying an additional AC bias. Another common example would be a clamped-clamped beam with a residual thermal stress.

COMSOL includes two study types to address this situation—**Prestressed Analysis, Eigenfrequency** ( ) and **Prestressed Analysis, Frequency Domain** ( ).

Each of these study types first performs a stationary analysis of the problem to compute the prestressed state and then performs the appropriate prestressed analysis. In both cases the prestressed state can be computed using any stationary problem that COMSOL can solve.

PRESTRESSED ANALYSIS, EIGENFREQUENCY

The **Prestressed Analysis, Eigenfrequency** study ( ) performs a **Stationary** study (step 1) followed by an **Eigenfrequency** analysis (step 2), which is performed on the prestressed structure. To perform this study, no additional forces need to be added to the physics settings as only the modes and mode frequencies are returned.

PRESTRESSED ANALYSIS, FREQUENCY DOMAIN

The **Prestressed Analysis, Frequency Domain** study ( ) performs a **Stationary** study (step 1) followed by an **Frequency Domain** analysis (step 2), which is performed on the prestressed structure. For this study type, it is necessary to specify the magnitude of the harmonic load, as this determines the magnitude of the system response. In order to do this, a **Harmonic Perturbation** force must be added to the model. For contributing loads (that is to say, loads that can be added without overwriting the same type of node, such as a boundary load in solid mechanics), right-click the node and select **Harmonic Perturbation**. In this case, a load to generate the prestress must be added separately to the model.

For exclusive boundary conditions (that is to say, loads which overwrite previously added loads of the same type, such as the potential load in electrostatics), **Harmonic Perturbation** is added as a subnode to the boundary condition node. Its magnitude is added in the subnode itself.

See Also

For different plot settings made available by using this feature, see Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings.

21

Results Evaluation and Visualization

To help you analyze results COMSOL Multiphysics provides numerous result-evaluation and visualization tools, including advanced graphics, data display, and export functions. This chapter gives details about how to use these tools in the following sections:

Processing and Analyzing Results

During results analysis and visualization there are these main operation features: *data sets*, *plots and plot groups*, *derived values and tables*, *export of data and images*, and *reports*.

See Also	• Results Analysis and Plots in the *COMSOL Multiphysics Reference Guide* • Results in the *COMSOL Java API Reference Guide*

Important	The links to the features described in the *COMSOL Multiphysics Reference Guide* do not work in the PDF, only from the online help in COMSOL Multiphysics.

Tip	To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

DATA SETS

Data sets contain the source of data for plotting, for example, by indicating a solution and geometry or by transforming another data set. Go to Defining Data Sets.

PLOT GROUPS AND PLOTTING

A plot group is a collection of plot features to display simultaneously in the **Graphics** window. You can enable or disable plots in a plot group to determine the most applicable final image for your model or project. The physics interfaces create suitable default plots grouped in descriptive plot groups. Go to About Plot Groups and Plotting.

Cross-Section Plots

Use a combination of data sets and plot groups to create cross-section plots. Go to Creating Cross-Section Plots and Combining Plots.

DERIVED VALUES AND TABLES

Derived values define the evaluation of values of integrals, variables in points, or global variables. The evaluation results are stored in **Tables** and displayed in the **Results** window. Go to Defining Derived Values and Tables.

EXPORTING DATA AND IMAGES

Create a variety of reports using data, images, or animations from plot groups. Go to Exporting Data and Images.

REPORTS

Create reports as HTML documents that contain settings, selections, comments, plots, and other information about the model. Go to Creating Reports.

VIEWS

To display the **Views** node (for 2D or 3D models) under **Results**, click the **Show** button ( ) and select **Advanced Results Options**. Then go to User-Defined Views for information.

Defining Data Sets

See Table 21-1 in the section About Data Sets for links to all the data sets described in this section.

About Data Sets

Data sets contain or refer to the source of data for creating **Plots** and **Reports**. It can be a **Solution**, a **Mesh**, or some transformation or cut plane applied to other data sets—that is, create new data sets from other data sets. Add data sets to the **Data Sets** branch ( ) under **Results**.

All plots refer to data sets; the solutions are always available as the default data set. Characteristics of a data set include:

- Visualization mesh
- Mapping to a previous data set (except for solutions and meshes)
- Ability to refer the evaluation to the previous data set.

The base data set maps to a solution and geometry or some other source of data. An example of a transformation data set is **Revolution 2D**, which sweeps a 2D data set into 3D.

Note

Cut Point, Cut Line, Cut Plane, Edge 2D, Edge 3D, and Surface data sets are used in combination with plot groups and Line, Point, and Surface graphs to create cross-sectional plots and plots for data in points, along lines and edges, and on cut planes and surfaces.

HOW TO ADD A DATA SET TO THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** ( ), right-click **Data Sets** ( ) and select an option from the context menu. Continue defining each data set as described. See Table 21-1 for links to all the data set types.

ADDING A SELECTION TO A DATA SET

For the data sets that contain data defined in the model, such as **Solution** data sets ( ) and **Mesh** data sets ( ), add a selection so that the results and plots use a subset of the geometry. In the **Selection** subnode ( ), select the geometric entities for which you want to include data in the data set using the **Graphics** window and the settings in the **Selection** subnode’s settings window. Select the **Propagate to lower dimensions** check box to make a selection of domains, for example, also include their adjacent boundaries, edges, and points. This can be convenient when using Surface plots on the boundaries of a 3D geometry, for example.

TABLE 21-1: DATA SET TYPES

LINK TO SECTION	ICON	DESCRIPTION AND PLOT USE
Average Evaluation or Integral Evaluation Data Sets		Creates a data set that computes the average of another data set, for example to plot the average.
Contour 2D Data Sets		To analyze on 2D contour lines. Use this data set for 2D arrow plots, 2D line plots, and 1D global plots.
Cut Line 2D and 3D Data Sets	3D 2D	To create lines through 2D and 3D geometry to visualize along the line. Use this data set to create 2D or 3D cross-sections line plots.
Cut Plane Data Sets		Plots on cut planes are made on 3D data sets and can be visualized in either 2D or 3D plot groups.
Cut Point 1D, 2D, and 3D Data Sets	3D 2D 1D	Plot and evaluate a value in a certain point along time or along a parametric solution. Use this data set to create 1D, 2D or 3D cross-sections point plots.
Edge 2D and 3D Data Sets	3D 2D	Plot and evaluate a value along an boundary (edge) in 2D or an edge in 3D. Use this data set to create plots for data on edges.
Function 1D, 2D, and 3D Data Sets	f(x)	Create a data set that can evaluate functions on a domain.
Isosurface Data Sets		To visualize isosurfaces in 3D. Use this data set to create arrow surface plots, surface plots, and contour plots. In addition, the contour data set can be applied to the isosurface data set.

TABLE 21-1: DATA SET TYPES

LINK TO SECTION	ICON	DESCRIPTION AND PLOT USE
Join Data Sets		To join the solution data from two Solution data sets. This makes it possible to evaluate and plot the difference between two solutions, for example.
Maximum Evaluation and Minimum Evaluation Data Sets	Max Min	Creates a data set that computes the maximum or minimum of another data set.
Mesh Data Sets		To make a mesh available for visualization and results analysis. Use a Mesh feature in a plot group to get the actual plot.
Mirror 2D and Mirror 3D Data Sets	3D 2D	To extend a solution defined on one side of an axis to the other side of the axis. Useful for visualization of a solution to an axisymmetric problem or plane of reflection.
Parameterized Curve 2D and 3D Data Sets	3D 2D	To visualize data along a general parameterized curve in 2D or 3D. Use this data set to create a line plot in its original dimension and as a line graph plot in 1D.
Parameterized Surface Data Sets		To visualize data on a general parameterized surface. Use this data set to with a surface plot in its original dimension and as any plot type in 2D.
Parametric Extrusion 1D and 2D Data Sets	2D 1D	To extend another data set by using a parameter, such as time, as a dimension.
Particle Data Sets		To visualize particle traces computed by a Particle Tracing interface.
Revolution 1D and 2D Data Sets	2D 1D	Use a revolution data set to visualize a 2D or 1D axisymmetric solution in 3D and 2D, respectively.

TABLE 21-1: DATA SET TYPES

LINK TO SECTION	ICON	DESCRIPTION AND PLOT USE
Sector 2D and Sector 3D Data Sets	3D 2D	To exploit symmetries in the model to reduce the model size and then being able to use rotations and reflections to plot the solution for the entire geometry.
Solution Data Sets		To make solutions available for visualization and results analysis. Solvers create Solution data sets automatically.
Surface Data Sets		Plot and evaluate a value on surfaces (boundaries) in 3D. Use this data set to create plots for data on surfaces.

See Also

- Results Evaluation and Visualization
- Defining Data Sets
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Expressions and Predefined Quantities
- Creating Cross-Section Plots and Combining Plots
- Data Export

Average Evaluation or Integral Evaluation Data Sets

Select an **Average Evaluation** ( ) or **Integral Evaluation** ( ) data set, found under the **More Data Sets** submenu, to compute the average or integral of another data set for example, to plot the average or integral.

Model

For an integral evaluation data set example see Flow Past a Cylinder: Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**.

For an average evaluation data set example, and if you have the Batteries & Fuel Cells Module, see Edge Effects in a Spirally Wound Li-Ion Battery: Model Library path **Batteries_and_Fuel_Cells_Module/Batteries/li_battery_spiral_2d**.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Average** or **More Data Sets>Integral** from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

Note

To integrate on a **Solution** data set, use a **Selection** to define the geometric entity (point, boundary, edge or domain) to integrate over. Right-click the data set and select **Add Selection**.

3 Under **Settings**:

a Select a **Method**—**Auto**, **Integration**, or **Summation**.

- **Auto**—the default and computes the integral for fields by numerical integration and for reaction forces by summation. Predefined quantities for reaction forces use summation instead of integration. Also, to specify an expression for the integrand, use the `reacf` operator, and the automatic setting chooses the summation method.
- **Integration**—the standard numerical integration method (quadrature).
- **Summation**—a summation method useful for computing reaction forces. The summation method finds all nodes on the boundary, evaluates the expression in the nodes and sum up the values. Reaction force variables are predefined in the structural mechanics interfaces.

b For **Integration order**, the check box is selected by default. COMSOL automatically determines an appropriate integration order for the expression. Clear the check box to make a different entry in the field. COMSOL then performs the integration elementwise using numeric quadrature of the selected order.

c Select a **Geometry level**—**Data set highest**, **Volume**, **Surface**, **Line**, or **Point**.

Contour 2D Data Sets

Use a **Contour** () data set, selected from the **More Data Sets** submenu, for results evaluation on 2D contour lines. Contour lines cannot be parameterized in general, so only 2D arrow plots, 2D line plots, and 1D global plots can be used to visualize the data set.

Tip

Links to all the other data set types are found in Table 21-1.

1. Select **More Data Sets>Contour** ( ) from the **Data Sets** context menu.
2. Under **Data** select a **Data set**. Only 2D data sets can be used.
3. Under **Expression** select predefined expressions based on the physics of the model or enter an **Expression** in the field. Select a **Unit** and the **Description** check box to enter a description (or edit the default).
4. Under **Levels**, select an **Entry method**—**Number of Levels** or **Levels** for the contours.
   - If **Number of Levels** is selected, enter the total number of levels in the **Total levels** field (the default is 5).
   - If **Levels** is selected, enter the values for the contour **Levels** or click the **Range** button ( ) to define specific range of levels.
5. Click the **Plot** button ( ) to show the geometry for the data set.

Cut Line 2D and 3D Data Sets

Use a **Cut Line** data set in 2D ( ) or 3D ( ) to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis features available in 1D are available for cut line data sets as well as 3D plots and results analysis features for edges. These data sets are used to create cross-section line plots.

Model

For a 3D example, see Thin-Film Resistance: Model Library path **COMSOL_Multiphysics/Electromagnetics/thin_film_resistance.**

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Cut Line 2D** ( ) or **Cut Line 3D** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 To specify the cut line, under **Line Data**, select **Two Points** or **Point and direction** from the **Line entry method** list.

4 If **Two Points** is selected:

   a Enter coordinates **x**, **y**, and **z** coordinates as required for the space dimension in the **Point 1** and **Point 2** fields.

   b Select the **Bounded by points** check box to constrain the line between the defined points.

5 If **Point and direction** is selected, enter **Point** and **Direction** values for the **x**, **y**, and **z** coordinates as required for the space dimension.

2D

For 2D models, select the **Additional parallel lines** check box to define multiple lines for plotting or evaluation, for example. Enter **Distances** from the original line in the field, or click the **Range** button ( ) to define a range of distances.

The **Distances** field refers to a direction that is normal to the cut line and which is rotated 90 degrees counterclockwise relative to the cut line's direction. For example, if the cut line is from (0, 0) to (1, 0), then the distances are along the vector (0, 1) from any point on the cut line.

---

6 Select the **Snap to closest boundary** check box to snap the selected points to the closest boundary in the geometry. Use this option when evaluating a variable that is available on boundaries but not in domains.

7 If needed, under **Advanced**, edit the default name for the space variable (`dn1x`, for example) in the **x** field. The space variable name appears in the **Results** window when displaying the data.

8 Click the **Plot** button ( ) to show the geometry for the data set.

Cut Plane Data Sets

For plots on cut planes, use **Cut Plane** data sets ( ), which are made on 3D data sets and can be visualized in either 2D or 3D plot groups. All plots and results analysis features available in 2D are available for cut plane data sets as well as for surfaces in 3D.

The cut plane corresponds to an orthogonal 2D coordinate system embedded in 3D. This data set is used to create 3D cross-section surface plots.

Model

If you have the:

- AC/DC Module, see Magnetic Lens: Model Library path **ACDC_Module/Particle_Tracing/magnetic_lens**
- Batteries & Fuel Cells Module, see Mass Transport Analysis of a High Temperature PEM Fuel Cell: Model Library path **Batteries_and_Fuel_Cells_Module/PEMFC/ht_pem**
- CFD Module, see Airflow Over an Ahmed Body: Model Library path **CFD_Module/Single-Phase_Benchmarks/ahmed_body**
- Chemical Reaction Engineering Module, see Laminar Static Mixer: Model Library path **Chemical_Reaction_Engineering_Module/Mixing/laminar_static_mixer**
- Particle Tracing Module, see Ideal Cloak: Model Library path **Particle_Tracing_Module/Ray_Tracing/ideal_cloak**

Tip

Links to all the other data set types are found in Table 21-1.

---

1 Select **Cut plane** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Under **Plane Data**, select **Quick** or **General** from the **Plane type** list.

The **Plane Type** consists of the sets of planes orthogonal to the coordinate axes applicable for the model geometry—for example, **xy-planes**, **yz-planes**, and **zx-planes** in 3D.

4 If **Quick** is selected, select a set of cut planes orthogonal to a coordinate axis and specify the transverse coordinate by entering the location along the transverse coordinate axis:

a Select **xy-planes**, **yz-planes**, or **zx-planes** from the **Plane** list.

b Enter the **x-**, **y-**, or **z-coordinate** in the field based on the **Plane** selection.

5 If **General** is selected, select a **Plane entry method**—**Three points** or **Point and normal**.

- If **Three points** is selected, enter **Point 1**, **Point 2**, and **Point 3** in the **x-**, **y-**, and **z-**coordinate fields.
- If **Point and normal** is selected, enter **Point** and **Normal** data in the **x-**, **y-**, and **z-**coordinate fields.

6 Select the **Additional parallel lines** check box to define multiple planes for plotting or evaluation, for example. Enter **Distances** from the original line in the field, or click the **Range** button ( ) to define a range of distances for additional cut planes.

7 Click the **Plot** button ( ) to show the geometry for the data set.

8 Under **Advanced**:

a In the **Advanced** section, under **Space variables**, you can change then name of the space variables **x** and **y** for the cut plane's coordinates from their default values (`cpl1x` and `cpl1y`, for example).

b Select the **Define normal variables** check box to create variable names for the cut plane's normal vector.

c In the **Normal variables** section, enter or edit the variable names for the components of the normal vector. Enter **nx**, **ny**, and **nz** values.

Cut Point 1D, 2D, and 3D Data Sets

Use a **Cut Point** data set in 1D ( ), 2D ( ), or 3D ( ) to plot and evaluate a value in a certain point along time or along a parametric solution and to create 1D, 2D, or 3D cross-sectional point plots. The choice of 1D, 2D, or 3D only controls the type of input the data set accepts. For example, a 1D cut point can only be added to 1D data sets, 2D can only be added to 2D data sets, and so forth. Either of these can be used to make a point graph plot along time and so forth.

Model

For a 2D example, see 2D Heat Transfer Benchmark with Convective Cooling: Model Library path **COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d**

For a 3D example, and if you have the:

- MEMS Module, see Prestressed Micromirror: Model Library path **MEMS_Module/Actuators/micromirror**
- Microfluidics Module, see Star-Shaped Microchannel Chip: Model Library path **Microfluidics_Module/Fluid_Flow/star_chip**
- RF Module, see Microwave Oven: Model Library path **RF_Module/RF_and_Microwave_Engineering/microwave_oven**

For a 1D example, and if you have the:

- Plasma Module, see Benchmark Model of a Capacitively Coupled Plasma: Model Library path **Plasma_Module/Capacitively_Coupled_Plasmas/ccp_benchmark**

Tip

Links to all the other data set types are found in Table 21-1.

1. Select **Cut Point 1D** ( ), **Cut Point 2D** ( ), or **Cut Point 3D** ( ) from the **Data Sets** context menu.
   - Select **Cut Point 1D** for one point as cross-section data.
   - Select **Cut Point 2D** or **3D**, for example, to plot the value in a certain point along time or along a parametric solution and use the data set in its original dimension.
2. Under **Data** select a **Data set**.

3 Under **Point Data**, select a method of entering the point coordinates from the **Entry method** list:

- Select **Coordinates** (the default) to enter **x**, **y**, and **z** coordinates as required for the space dimension.
- Select **From file** to enter or browse to a text file with the cut point data.
- Select **Grid** to enter coordinates for gridded data as **x**, **y**, and **z** coordinates as required for the space dimension.
- Select **Regular grid** to specify the number of points in each direction for a regular grid in the **Number of x points**, **Number of y points**, and **Number of z points** as required for the space dimension.

4 Select the **Snap to closest boundary** check box to snap the selected points to the closest boundary in the geometry. Use this option when evaluating a variable that is available on boundaries but not in domains.

5 Click the **Plot** button ( ) to show the geometry for the data set.

Edge 2D and 3D Data Sets

Use an **Edge 2D** (□) data set or an **Edge 3D** (⧉) data set to plot and evaluate a value along an edge (boundary) in 2D or an edge in 3D. Create a line graph or any other plot type in a **1D Plot Group** to plot data along an edge using one of these data sets.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Edge 2D** (□) or **Edge 3D** (⧉) from the **Data Sets** context menu to make it possible to evaluate and plot a value along an edge in visualization and results analysis.

Use an **Edge 2D** data set to plot values on boundaries (edges) in a 2D geometry. Use an **Edge 3D** data set to plot values on edges in a 3D geometry.

2 Under **Data** select a **Data set**, typically a solution.

3 In the **Selection** section, select the edges to include.

Function 1D, 2D, and 3D Data Sets

Adding an **Function 1D**, **Function 2D**, or **Function 3D** node (f(x)) from the **More Data Sets** submenu creates a data set that can evaluate functions on a domain. All functions in the same list as the selected function can be evaluated. The domain is an interval for Function 1D, a rectangle for Function 2D, and a block for Function 3D. The domain does not need to have the same dimension as the number of arguments to the function. To plot the functions, use a line graph in 1D, for example, by pointing to the Function 1D data set in a 1D Plot Group (or similarly for Function 2D and Function 3D).

Model

For an example of a 1D function data set and if you have the AC/DC Module, see Geoelectrics: Model Library path **ACDC_Module/ Other_Industrial_Applications/geoelectrics**.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Function 1D**, **More Data Sets>Function 2D**, or **More Data Sets>Function 3D** from the **Data Sets** context menu.

2 Select a **Function** from the list.

3 Under **Parameter Bounds** enter the following. Available fields are based on the function dimension:

a Enter a **Name**. The **First, Second**, and **Third Parameter** default names are `s`, `t`, and `u`, respectively.

b For each parameter, enter a **Minimum** lower bound for the first, second, and third dimension of the domain.

c For each parameter, enter a **Maximum** upper bound for the first, second, and third dimension of the domain.

4 Enter a **Resolution**. This is the number of points into which each dimension is discretized. The defaults are 1000 (1D), 100 (2D), and 30 (3D).

5 Click the **Plot** button () to show the geometry for the data set.

Isosurface Data Sets

Use an **Isosurface** () data, found under the **More Data Sets** submenu, set to visualize isosurfaces in 3D. Isosurfaces cannot be parameterized in general so use this data set to create arrow surface plots, surface plots, and contour plots. The contour data set can be applied to the isosurface data set.

> *Tip*
>
> Links to all the other data set types are found in Table 21-1.

1. Select **More Data Sets>Isosurface** () from the **Data Sets** context menu.
2. Under **Data** select a **Data set**.
3. Under **Expression** select predefined expressions based on the physics of the model or enter an **Expression** in the field. Select a **Unit** and the **Description** check box to enter a description (or edit the default).
4. Under **Levels**, select an **Entry method** for the isosurfaces—**Number of Levels** or **Levels**.
   - If **Number of Levels** is selected, enter the total number of levels in the **Total levels** field (the default is 5).
   - If **Levels** is selected, enter the value of the isosurface **Levels** or click the **Range** button () to define a specific range of levels.
5. Click the **Plot** button () to show the geometry for the data set.

Join Data Sets

Use an **Join** () data set join the solutions from two **Solution** data sets. The joined data sets makes it possible to compare solutions from two data sets—for example, to evaluate and visualize the difference between two solutions to the same problem using two different meshes in a mesh convergence study. The Join data set has predefined methods to combine the solutions to get the difference, product, quotient, or sum of the two solution data sets. In addition, two predefined “operators,” `data1` and `data2`, correspond to the solution data in the first and second Solution data set, respectively, and make it possible to compare, for example, solutions from different but compatible models or to combine the two solution data sets using another method than the ones that you can select directly.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Join** () from the **Data Sets** context menu.
2 Select a solution data set as `data1` and another solution data set as `data2` from the **Data** lists in the **Data 1** and **Data 2** sections, respectively.
3 Select which solutions to use from the **Solutions** lists: The default, **All**, uses all solutions. Select **One** to use one of the available solution in a time-dependent, parametric, or eigenvalue solution from the list of solutions that appear.

Note

The following data set combinations support the option to include all solutions from both data sets:

- Both data sets point to the same solution.
- Both data sets point to two different stationary solutions.
- Both data sets point to two different time-dependent solutions.

4 Under **Combination** select a way of combining the solutions from the **Method** list: **Difference** (the default), **Explicit**, **General**, **Product**, **Quotient**, or **Sum**. The **Explicit** method makes it possible to use the "operators" `data1` and `data2` with arguments in a plot expression, for example. The **General** method provides an **Expression** field where you can type a general expression using `data1` and `data2` for the join operation. The default expression, `data1-data2`, is identical to the **Difference** method.

Maximum Evaluation and Minimum Evaluation Data Sets

Select the **Maximum** evaluation ( ) or **Minimum** evaluation ( ) data set, found under the **More Data Sets** submenu, to create a data set that computes the maximum or minimum of another data set.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Maximum** or **More Data Sets>Minimum** from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Under **Settings**, select a **Geometry level**—**Data set highest**, **Volume**, **Surface**, **Line**, or **Point**.

4 Click the **Plot** button ( ) to show the geometry for the data set.

Mesh Data Sets

Add a **Mesh** ( ) data set to make a mesh available for visualization and results analysis, typically for mesh-related quantities such as the mesh element quality, which are possible to plot using a mesh data set without computing a solution. Use a **Mesh** feature in a plot group to get a plot of the actual mesh.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Mesh** ( ) from the **Data Sets** context menu.

2 Select a **Mesh** from the list.

3 Click the **Plot** button ( ) to plot the geometry where the mesh is active.

Mirror 2D and Mirror 3D Data Sets

2D

Use a **Mirror 2D** ( ) data set, selected from the **More Data Sets** submenu, to extend a solution defined on one side of an axis to the other side of the axis. This can be useful for visualization of a solution to an axisymmetric problem.

3D

Use a **Mirror 3D** ( ) data set, selected from the **More Data Sets** submenu, to extend a solution defined on one side of a plane to the other side of a plane.

Tip

Links to all the other data set types are found in Table 21-1.

Model

- For a 3D example, and if you have the Chemical Reaction Engineering Module, see Surface Reactions in a Biosensor: Model Library path **Chemical_Reaction_Engineering_Module/ Surface_Reactions_and_Deposition_Processes/reacting_pillars**
- For a 2D example, and if you have the Plasma Module, see DC Glow Discharge: Model Library path **Plasma_Module/ Direct_Current_Discharges/positive_column_2d**

1 Select **More Data Sets>Mirror 2D** ( ) or **More Data Sets>Mirror 3D** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Define the **Axis Data** (2D models) or **Plane Data** (3D models):

2D

Under **Axis Data**, for 2D models, select an **Axis entry method**—**Two points** or **Point and direction**.

- If **Two points** is selected, enter **Point 1** and **Point 2** for both the **x** and **y** coordinates.
- If **Point and direction** is selected, enter values for the **Point** and **Direction** for both the **x** and **y** coordinates.

3D

Under **Plane Data**, for 3D models, select a **Plane type**—**Quick** (the default) or **General**. If **Quick** is selected, select a **Plane**—**xy-planes**, **yz-planes**, or **zx-planes**.

Enter the **z-coordinate**, **x-coordinate**, or **y-coordinate** based on the selection.

- If **General** is selected, select a **Plane entry method**—**Three points** to enter the mirror axis by specifying three points or **Point and normal** to specify the mirror axis by specifying one point and a normal vector. Then if **Three points** is selected, enter coordinates in the **Point 1**, **Point 2**, and **Point 3** fields for **x**, **y**, and **z** coordinates (SI unit: m).
- If **Point and Normal** is selected, enter **Point** and **Normal** vector coordinates for **x**, **y**, and **z** coordinates (SI unit: m).

4 Under **Advanced**, select the **Define variable** check box to create a **Positive side indicator** variable.

The default variable, `mir1side`, is 1 in the original domain and 0 in the mirror. Use the **Positive side indicator** variable in the **Expression** section of a plot settings window to exclude quantities from the mirror side.

5 Click the **Plot** button ( ).

Parameterized Curve 2D and 3D Data Sets

Use a **Parameterized Curve** data set in 2D ( ) or 3D ( ) to visualize data along a general parameterized curve. Visualize the parameterized curve as a line plot in its original space dimension (2D or 3D) and as a line graph plot in 1D. Select these options from the **More Data Sets** submenu.

Model	For a 3D example, and if you have the Heat Transfer Module, see Radiative Heat Transfer in Finite Cylindrical Media: Model Library path **Heat_Transfer_Module/Tutorial_Models/cylinder_participating_media**

Tip	Links to all the other data set types are found in Table 21-1.

1. Select **More Data Sets>Parameterized Curve 2D** ( ) or **More Data Sets>Parameterized Curve 3D** ( ) from the **Data Sets** context menu.
2. Under **Data** select a **Data set**.
3. Under **Parameter**, enter a **Name** and the **Minimum** and **Maximum** range of the parameter curve.
4. Under **Expressions**, enter functions for the **x**, **y**, and **z** coordinates as required for the space dimension.
5. In the **Resolution** field, enter the number of subdivisions of the parameter range. The default is 1000.
6. Click the **Plot** button ( ).

Parameterized Surface Data Sets

Use a **Parameterized Surface** ( ) data set, selected from the **More Data Sets** submenu, to visualize data on a general parameterized surface. Visualize the parameterized surface as a surface plot in its original dimension and as any plot type in 2D.

Tip	Links to all the other data set types are found in Table 21-1.

1. Select **More Data Sets>Parameterized Surface** ( ) from the **Data Sets** context menu.
2. Under **Data** select a **Data set**.

3 Under **Parameters**, enter the following for both the **First parameter** and **Second parameter** fields.

   a Enter a **Name**.

   b Enter a range of the parameter in the **Minimum** and **Maximum** fields.

4 Under **Expressions**, enter functions for coordinates of the two parameters in the **x**, **y**, and **z** fields.

5 In the **Resolution** field, enter the number of subdivisions of the parameter ranges. the default is 200.

6 Click the **Plot** button ( ).

Parametric Extrusion 1D and 2D Data Sets

The **Parametric Extrusion** data set in 1D ( ) and 2D ( ) extends another data set by using a parameter, such as time, as a dimension. Select these options from the **More Data Sets** submenu.

Model	See The KdV Equation and Solitons for an example of a 1D parametric extrusion data set: Model Library path **COMSOL_Multiphysics/ Equation-Based_Models/kdv_equation**.

Tip	Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Parametric Extrusion 1D** ( ) or **More Data Sets>Parametric Extrusion 2D** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Under **Settings**:

   a Select the **Level scale factor** check box to enter a value or edit the field.

   b The **Separate levels** check box is selected by default. Click to clear the check box.

4 Click the **Plot** button ( ).

Particle Data Sets

Note

Particle data sets require the Particle Tracing Module.

Use a **Particle** (⁘) data set, selected from the **More Data Sets** submenu, to visualize particle traces computed by a Particle Tracing Module interface. The Particle data set is automatically created when solving a model containing one of the Particle Tracing Module interfaces, provided that the **Generate default plots** option is selected in the **Study**. Selections may be added to the particle data set which allows, for example, the number or fraction of particles in a given domain or on a given boundary to be computed during results processing. Visualize the particles using a plot of the particle trajectories in the original dimension, as a Poincaré map, and as a 2D phase portrait.

Model

If you have the AC/DC Module and Particle Tracing Module, see Magnetic Lens: Model Library path **ACDC_Module/Particle_Tracing/magnetic_lens**.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Particle** (⁘) from the **Data Sets** context menu.

2 Under **Particle Solution**, select a **Solution** from the list of solution data.

3 In the **Particle geometry** field, enter the name of the variable for the particle geometry. This corresponds to the hidden geometry on which the particle degrees of freedom are computed. When the Particle data set is generated from the default plots the correct name for the particle geometry is filled in automatically. The particle geometry takes the name `pgeom_<tag>` where `<tag>` is the interface identifier for the specific Particle Tracing interface.

4 In the **Position dependent variables** field, enter the names of the dependent variables for the particles' position using commas to separate the variables. Like the particle geometry, the names of the particle degrees of freedom are filled in automatically when the particle data set is generated from the default plots. The names correspond

to the particle position degrees of freedom with the model identifier prepended. For example, for a 3D model, with tag `<mod1>`, and **Dependent Variables** `qx`, `qy`, and `qz` for the interface, the correct expression to enter is `mod1.qx`, `mod1.qy`, and `mod1.qz`.

5 Click the **Plot** button ( ) to display the geometry for the data in the data set.

Revolution 1D and 2D Data Sets

1D Axi

Use a **Revolution** data set to visualize a 1D axisymmetric ( ) solution in 2D. All plot types in 3D or 2D are available for visualization through the revolution data set.

2D

Use a **Revolution** data set to visualize a 2D ( ) solution in 3D. All plot types in 3D or 2D are available for visualization through the revolution data set.

Model

See Tubular Reactor for a 2D example: Model Library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Revolution 1D** ( ) or **Revolution 2D** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Under **Axis Data**, enter information based on space dimension.

1D

For 1D models, enter a value in the **x** field to specify the revolution point.

2D

For 2D models, from the **Axis entry method** list, select **Two points** to enter the revolution axis by specifying two points or **Point and direction** to specify the revolution axis by specifying one point and a direction vector.

- If **Two points** is selected, enter coordinates in the **Point 1** and **Point 2** fields for **x** and **y** coordinates.
- If **Point and Direction** is selected, enter **Point** and **Direction** vector for **x** and **y** coordinates.

4 Under **Revolution Layers**:

a By default, the number of layers is set to **Normal**. Also choose **Fine**, **Coarse**, or **Custom** from the **Number of layers** list.

a If **Custom** is selected, enter the number of subdivisions about the revolution axis in the **Layers** field (default value: 50).

b Enter the **Start angle** (in degrees) for the revolved model.

c In the **Revolution angle** field, enter the angle (in degrees) to revolve the model to see into the geometry (default value: 360 degrees; that is, a full revolution). Use negative values to revolve the model in the opposite direction.

An axisymmetric geometry in the *rz*-plane is projected to the *xy*-plane and then rotated about the *y*-axis using the start angle and revolution angle.

5 Under **Advanced**:

a Select which plane to map the axisymmetric solution to from the **Map plane to** list:

- Select **xy-plane** (the default) to map the *rz*-plane to the *xy*-plane and then rotate it about the *y*-axis.
- Select **xz-plane** to map the *rz*-plane to the *xz*-plane and then rotate it about the *z*-axis.

b Select the **Define variables** check box to create variable names in the revolved geometry.

c In the **Space variables** section, enter or edit the variable names for the revolved coordinate system. For 1D, enter **x** and **y** values. for 2D, enter **x**, **y**, and **z** values.

d Under **phi**, enter or edit the variable name for **phi** in its field. Phi is the name of the angle variable in the revolved coordinate system. The default names are `rev1x`, `rev1y`, `rev1z`, and `rev1phi`.

6 Click the **Plot** button ().

Sector 2D and Sector 3D Data Sets

Use the **Sector 2D** ( ) and **Sector 3D** ( ) data sets, selected from the **More Data Sets** submenu, to make it possible to plot the solution for the full geometry while reducing computation time and memory requirements for complex geometries by exploiting sector symmetries. The geometry must be of a type that can be transformed through the use of rotation or reflection (mirroring). Rotation and reflection are only available when using an even number of sectors.

For example, suppose that there are *N* sectors in a geometry. A Sector data set first evaluates the input expressions in the source data set, creates *N* copies (one for each sector of the geometry), maps and interpolates the data, and transforms the expression components that correspond to vector fields.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **More Data Sets>Sector 2D** ( ) or **More Data Sets>Sector 3D** ( ) from the **Data Sets** context menu.

2 Under **Data** select a **Data set**.

3 Define the **Axis Data**. For **Sector 2D** data sets, enter values for both the **x** and **y** coordinates (SI unit: m). For **Sector 3D** data sets, select an **Axis entry method**—**Two points** or **Point and direction**. Then:

- if **Two points** is selected, enter **x**, **y**, and **z** coordinates (SI unit: m). for **Point 1** and **Point 2**.
- if **Point and direction** is selected, enter **x**, **y**, and **z** coordinates for the **Point** (SI unit: m) and **Direction**.

4 Under **Symmetry**, enter any integer greater than or equal to 2 into the **Number of sectors** field (the default is 2).

- When the **Number of sectors** entered is an even number, select a **Transformation**—**Rotation** (the default) or **Rotation and reflection**. If **Rotation and reflection** is selected, also enter the **x** and **y** coordinates for the **Direction of reflection axis** for **Sector 2D** data sets, or enter the **x**, **y**, and **z** coordinates for the **Radial direction of**

reflection plane for **Sector 3D** data sets. For an uneven number of sectors, only rotation is available and it requires no additional user input.

5 Under **Advanced**, select the **Define variables** check box to create **Space variables**. These variables evaluate to the coordinate after the transformation (as opposed to x, y, and z, which evaluate to the coordinates in the underlying data set).

The default variables (the number of which are based on space dimension) are `sec2x` for the x coordinate, `sec2y` for the y coordinate, and `sec2z` for the z coordinate.

When the **Define variables** check box is selected, the input expression is enabled once for each sector, something that increases evaluation time by roughly a factor of the number of sectors (N). If the input expression being evaluated contains any of the space variables, then this evaluated mode is enabled automatically.

For cyclic symmetry, that is, when the number of sectors is odd or **Rotation** is selected as the **Transformation**, also enter the **Azimuthal mode number**. The default is 0.

Note

Use the azimuthal mode number to control the source data set and evaluate it with different phases for the different sectors. If the mode number is k, then the phase is shifted with $-2\pi ik/N$ for sector i.

6 Click the **Plot** button () to plot the geometry where the data set is defined.

Solution Data Sets

The **Solution** () data sets make solutions available for visualization and results analysis. Solvers automatically create **Solution** data sets. Right-click to add a **Deformed Configuration** node to the Mesh node.

See Also

- Deformed Configuration in the *COMSOL Multiphysics Reference Guide*
- Remeshing a Deformed Mesh

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Solution** () from the **Data Sets** context menu.

2 Select a **Solution** to make available for visualization and results analysis.

3 If there is more than one model, select a **Model** to define the geometry for the solution data.

4 Select the **Frame** to evaluate the coordinates in—**Mesh**, **Material**, or **Spatial**. The default in most cases is the Material frame, and this rarely needs to be changed. This frame selection is used for all results evaluation that uses the solution data set.

5 Optionally, enter a value for the **Solution at angle (phase)** for complex-valued solutions.

Use the **Solution at angle (phase)** option to evaluate complex-valued expressions by multiplying the solution in the solution data set by a factor of `exp(i*pi*phase/180)` prior to expression evaluation.

Also, optionally apply a **Scale factor** to scale the entire solution by a real factor.

6 Click the **Plot** button ().

Surface Data Sets

Use a **Surface** () data set to visualize data on surfaces (boundaries) of a 3D geometry. Refer to this data set to plot and evaluate a value on a surface using a 2D plot group and a **Surface** or **Contour** plot, for example.

Tip

Links to all the other data set types are found in Table 21-1.

1 Select **Surface** () from the **Data Sets** context menu to make it possible to evaluate and plot a value on a surface in visualization and results analysis.

2 Under **Data** select a **Data set**, typically a solution.

3 Under **Parameterization**, choose from one of these options in the **x- and y-axes** list to specify what the local *x*-axis and *y*-axis represent:

- **Surface parameters**. This is the default setting, which uses the parameters of the 3D surface.
- **xy-plane**, **yz-plane**, **zx-plane**, **yx-plane**, **zy-plane**, or **xz-plane**. The local *x*- and *y*-axes then represent the global *xy*-plane, global *yz*-plane, global *zx*-plane, global *yx*-plane, global *zy*-plane, or global *xz*-plane, respectively
- **Expression**. With this setting, enter any expression, including the global space coordinates, for example, in the **x-axis** and **y-axis** fields. The default values are **x** and **y**, respectively.

4 Under **Selection**, select the boundaries to include in the data set using the available selection tools or by selecting directly in the **Graphics** window.

About Plot Groups and Plotting

A plot group contains one or more plots (for example, combining a surface plot and a streamline plot) using the same data set, such as a solution. You can define plot groups for 1D, 2D, and 3D plots and then create one or a series of plots in a plot group. Several plot groups can also be used in a model. Information in the form of data and images can be used to generate a report, or exported.

- 1D plots include line graphs, point graphs, graphs of global data, and table plots, with options to use FFT to plot a time-dependent solution as a frequency spectrum
- 2D plots include surface plots (color plots), contour plots, arrow plots, line plots, and more specialized plots to visualize, for example, streamlines, particle traces, principal stresses, and coordinate systems
- 3D plots include volume and surface plots (color plots), isosurface plots, and specialized plots like in 2D

To add a plot group, right-click the **Results** node and select, for example, **3D Plot Group** () from the context menu.

Tip	When you are working with **Functions**, you can also click the **Create Plot** button () to create a customized plot of the function under **Results**, including default plot groups and plots.

The physics interfaces create suitable default plots for visualizing the results for the particular physics or application. The default plots appear in plot groups with descriptive names. You can modify and deleted these plots and plot groups and add additional plots to existing or new plot groups. To disable the default plots for a study, clear the **Create default plots** check box in the **Study Settings** section in the main **Study** settings windows.

See Also

- For details about individual plots, and adding color expressions, deformations, filters, and height expressions., see Creating 1D, 2D, 3D, and Polar Plots.
- For details about standard and common settings, such as color and style options, see Common Plot Settings.
- For details about study settings, see The Study Nodes in the *COMSOL Multiphysics Reference Guide*.

Common Plot Settings

Before you begin creating plot groups and plots, this section provides you with some general plotting information which is common to many (or all) of the plots.

- Color Coding for Plot Groups and Plot Types
- Adding Plot Windows
- Specifying the Window to Plot In
- Locking Plot Windows
- Selecting a Data Set for Plots
- Selecting Solutions for Parametric Sweeps
- Expressions and Predefined Quantities
- Plot Titles for Plot Groups and Plot Types
- Using Special Formats and Symbols in Titles
- Legends
- Arrow Positioning
- Principal Components and Positioning
- Defining the Number of Levels
- Selecting Color Tables
- Defining the Color and Data Ranges
- Defining the Coloring and Style
- Defining Element Filters
- Defining Shrinking of Elements
- Entering Quality Settings for Plot Settings Windows
- Inheriting Style Options
- Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings

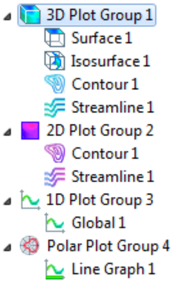
Color Coding for Plot Groups and Plot Types

The plot groups and plot types are color coded based on space dimension. See Figure 21-1 for an example.

1D	All the 1D and polar plots selected from **1D Plot Group** and **Polar Plot Group** nodes are green.

2D	All the 2D plots selected from the **2D Plot Group** node are pink.

3D	All the 3D plots selected from the **3D Plot Group** node are blue.



- 3D Plot Group 1
  - Surface 1
  - Isosurface 1
  - Contour 1
  - Streamline 1
- 2D Plot Group 2
  - Contour 1
  - Streamline 1
- 1D Plot Group 3
  - Global 1
- Polar Plot Group 4
  - Line Graph 1

Figure 21-1: An example of each plot group and its plot color coding.

Adding Plot Windows

The COMSOL Desktop always includes the **Graphics** window, which is the default window for all kinds of plots, but you can also add other plot windows for results plots by right-clicking a plot group node and choosing **Plot In>New Window** or by adding a plot window in the **Window Settings** section of the plot group nodes' settings windows.

Specifying the Window to Plot In

The default for all plot groups is to plot in the **Graphics** window, but you can plot in any other plot window by right-clicking the plot group node and choosing another plot window from the **Plot In** submenu. The **Plot In** submenu also exists on the context menu for each plot type if you want to plot only an individual plot type in a plot group. You can also control where the plots appear and add new plot windows in the **Window Settings** section of the settings windows for the plot group nodes.

To create and update all plots in all plot windows, right-click the **Results** node and select **Plot All**. If more than one plot group use the same plot window, that plot window contains the plots from the last plot group.

Locking Plot Windows

If you want to prevent a plot in a plot window from being overwritten by other plots, you can lock the plot window. Click the **Lock** button (🔒) in the plot window's toolbar to lock that plot window from subsequent plots. An attempt to create a plot in a locked plot window results in a message such as **Window 'Plot 1' is locked** in the **Messages** window. Click the **Lock** button (🔒) again to clear the lock. It is not possible to lock the **Graphics** window, which COMSOL Multiphysics uses for general visualization.

Selecting a Data Set for Plots

Almost every plot type has the Data section where you select a **Data set** from the list. **From parent** means that the plot uses the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

FOR SOME 1D PLOTS

Under **Data** select a **Data set**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. Select:

- **From parent** (the default) to use the same data set as the plot group it belongs to.
- a **Solution** data set to visualize a quantity along a geometric edge.
- a **Cut Line** data set to visualize a quantity along the cut line (a cross section).
- a **Parameterized Curve** data set to visualize a quantity along the parameterized curve.

- **None** to not use any of the available data sets.

See Also

- Selecting Solutions for Parametric Sweeps
- About Data Sets

Selecting Solutions for Parametric Sweeps

This information is useful when defining plots or derived values for **Parametric Solver** and **Parametric Sweep** studies. Under the **Data** section in the **Parameter values** section, you can select the associated parameter values to use.

PLOTS

When setting parameters for parametric sweep studies in the **Results** node, the available settings depend on the problem type. A time-dependent problem, for example, allows you to select both time steps and parameter values. Similarly, an eigenvalue problem contains both eigenvalue and parameter settings. In results features, the time and eigenvalue settings are referred to as the *inner solutions*. Parameters that result in a **Parametric** node under **Job Configurations** are referred to as *outer solutions*. In a graph plot for a parametric eigenvalue solution, for example, the **Axis source data** setting for the x-axis data controls whether you want the inner (that is to say, eigenvalue) or outer (that is to say, parametric) solutions on the x-axis, or any other level in a nested parametric sweep.

DERIVED VALUES

For **Parametric Sweep** studies you also select both time steps or eigenvalues and parameter values. For a parametric sweep with more than one level (which is the case for all time-dependent and eigenvalue parametric sweeps) you also select the data to display in the table columns. When available, from the **Table columns** list, select one of the following options, when applicable:

- **Data only** (the default) to list the expression values in columns with the parameter variations as rows.
- The level of the nested parametric sweep, which can be **t** for time or the names of the parameters at the loop level, to list the selected levels parameter variations across the columns and the parameter variations for the other levels as rows.

- **Inner solutions** to list the inner solution variation in columns with the remaining parameter variations as rows.
- **Outer solutions** to list the outer solution variation in columns with the remaining parameter variations as rows.

Click the **Evaluate** button (≡) to fill a table with the value of the expression for the parameter variations in columns and rows as specified. If you click the **Evaluate** button again, the data is appended to the current table using that table's data settings. If you change the table data format it may not be possible to append that data to an existing table. In that case you need to create a new table by selecting New Table from the **Evaluate** button (≡) menu.

See Also

- Entering Ranges and Vector-Valued Expressions
- Defining Derived Values and Tables

Expressions and Predefined Quantities

When plotting and evaluating results, COMSOL Multiphysics provides a large number of predefined quantities that are specific to the physics interfaces in the model as well as general quantities for the geometry, coordinate systems, and mesh.

COMSOL Multiphysics does not limit the results calculations to predefined quantities; you can plot and evaluate any function by entering the corresponding expression. You can combine user-defined variables, physics interface variables, standard variables, and mathematical and logical functions and numbers. The **Expression** field or list is available for most plot types as well as for integration and data display and evaluation. You can enter any expression directly in the field or insert variables from a list of predefined quantities that you open by pressing Ctrl+Space or by clicking the **Insert Expression** (✚ ▾) button.

In the **Expression** sections in the settings windows for plot nodes you can:

- Click the **Replace Expression** (⇄ ▾) button to select a predefined quantity and replace the contents of the **Expression** field with the corresponding variable.
- Click the **Insert Expression** (✚ ▾) button to insert the corresponding variable at the current position in the **Expression** field.
- Select a **Unit** from the list. You can select from a predefined number of applicable units for the quantity that the variable represents, but you can also type any

compatible unit for that quantity to use a unit that is not in the list (for example, `mi/h` for miles per hour as a unit for a velocity quantity).

- Select the **Description** check box to enter a description (or edit the default).

The predefined quantities that you get access to by clicking one of the above buttons are divided into categories based on where in the model they belong:

- Each physics has its own list of predefined quantities. For multiphysics interfaces, there is typically a common list, with quantities that are not linked to any of the participating physics, and separate lists for each participating physics, with quantities that are specific to each physics.
- A **Definitions** list with variables for coordinate systems and user-defined variables.
- A **Geometry and Mesh** list with variables for spatial coordinates, mesh size, mesh quality, and domain indexes.

EVALUATION OF UNDEFINED QUANTITIES

During the evaluation of expressions, by default COMSOL Multiphysics does not report partially undefined quantities, and the program plots a quantity where it is defined. The plot is empty where the plotted data is undefined or “not-a-number” (NaN). If a results quantity is undefined everywhere, an error occurs for all plot types.

ACCESSING OTHER SOLUTIONS THAN THE SELECTED SOLUTION

When you use the names of the dependent variables in a results expression, COMSOL Multiphysics uses the solution associated with the selected parameter value, eigenvalue, or time for a parametric analysis, eigenvalue analysis, or time-dependent analysis, respectively. To access other solutions in the model, use the `with` operator.

PROCESSING SOLUTIONS WITH A STORED LINEARIZATION POINT

If the solution being processed has a stored linearization point, four different options are available for how to evaluate the expression. The **Evaluation method** list in the **Expression** section contains the following options:

- Choosing **For linearization point** evaluates the expression at the stored linearization point. This is equivalent to evaluating `linpoint(expr)`.
- Choosing **For perturbation** evaluates the expression for the perturbation component of the solution.
- Choosing **For total instantaneous** evaluates the expression at the superposition of the linearization point and the perturbation. The phase and amplitude of the

perturbation can be changed in the corresponding Data set. This is equivalent to evaluating `lintotal(expr)`.

- Choosing **Linearized deviation** linearizes the expression at the stored linearization point and evaluates the linearized expression. This is equivalent to evaluating `lindev(expr)`.

Plot Titles for Plot Groups and Plot Types

Every plot group and plot type have a **Title** section where the **Title type** is selected and set. The options are **Automatic** (the default), **Custom**, **Manual**, or **None**. An **Automatic** title is generated based on the type of plot or plots selected. Select **Manual** to enter free text in the field or **None** for no title. Select **Custom** to add existing information combined with custom prefix and suffix text to the title as described below.

CUSTOM PLOT TITLES

Solution

Under **Solution** select the check boxes as required.

- Select **Data set** to include details about the data set used for the plot.
- Select **Phase** to include information about the phase (when applicable).
- Select **Solution** to include the details about the solution (the time step or parameter values, for example) for the plot (when applicable).

Type and Data

Under **Type and data** select the check boxes as required.

- Select **Type** to include the plot type in the title.
- Select **Description** to include the variable details.
- Select **Expression** to include the variable expression in the title.
- Select **Unit** to include the variable unit.

User

Under **User** enter text as required.

- Enter text in the **Prefix** field to add free text at the front of any Solution title text string. For example, if all the Data set, Phase, and Solution check boxes are selected, this text is first.

Enter text in the **Suffix** field to add free text at the end of any Solution title text string. For example, if all the Data set, Phase, and Solution check boxes are selected, this text is at the end of this information.

See Also

Using Special Formats and Symbols in Titles

Using Special Formats and Symbols in Titles

SUPPORT FOR FORMATTING AND SYMBOLS IN TEXTS

For the titles as well as the *x*-axis, *y*-axis, and *z*-axis labels, you can use formatted strings that include HTML tags, Greek letters, and mathematical symbols. The tables in the following sections provide information about supported format and symbols. In addition to ASCII characters, Greek letters, and the mathematical symbols listed in Table 21-4, COMSOL Multiphysics correctly displays any Unicode-based character that you paste into a title or label field.

HTML TAGS

You can use the following HTML tags in text strings for plot labels and titles:

TABLE 21-2: VALID HTML TAGS

HTML TAG	DESCRIPTION
<B> </B>	Enclosed text is rendered using a bold font.
<I> </I>	Enclosed text is rendered using an italic font.
<SUB> </SUB>	Enclosed text is rendered in subscript with the enclosed text slightly lower than the surrounding text.
<SUP> </SUP>	Enclosed text is rendered in superscript with the enclosed text slightly higher than the surrounding text.
<TT> </TT>	Enclosed text is rendered using a monospaced font.
<U> </U>	Enclosed text is underlined.

GREEK CHARACTERS

The texts in labels and titles in all plots support the following Greek character tags:

TABLE 21-3: VALID GREEK SYMBOL TAGS

TAG	SYMBOL	TAG	SYMBOL
\ALPHA	Α	\alpha	α
\BETA	Β	\beta	β
\GAMMA	Γ	\gamma	γ
\DELTA	Δ	\delta	δ
\EPSILON	Ε	\epsilon	ε
\ZETA	Ζ	\zeta	ζ
\ETA	Η	\eta	η
\THETA	Θ	\theta	θ
\IOTA	Ι	\iota	ι
\KAPPA	Κ	\kappa	κ
\LAMBDA	Λ	\lambda	λ
\MU	Μ	\mu	μ
\NU	Ν	\nu	ν
\XI	Ξ	\xi	ξ
\OMICRON	Ο	\omicron	ο
\PI	Π	\pi	π
\RHO	Ρ	\rho	ρ
\SIGMA	Σ	\sigma	σ
\TAU	Τ	\tau	τ
\UPSILON	Υ	\upsilon	υ
\PHI	Φ	\phi	φ
\CHI	Χ	\chi	χ
\PSI	Ψ	\psi	ψ
\OMEGA	Ω	\omega	ω

MATHEMATICAL SYMBOLS

For texts in titles and axis labels, you can use the following mathematical symbol tags:

TABLE 21-4: VALID MATHEMATICAL SYMBOL TAGS

TAG	SYMBOL	TAG	SYMBOL
\approx	≈	\bullet	•
\sim	~	\partial	∂
\prop	∝	\nabla	∇
\neq	≠	\prod	∏
\equiv	≡	\sum	∑
\lequal	≤	\sqrt	√
\gequal	≥	\integral	∫
\ll	>>	\oplus	⊕
\gg	<<	\otimes	⊗
\plusmin	±	\larrow	←
\infinity	∞	\rarrow	→
\deg	°	\lrarrow	↔
\cdot	·	\ldarrow	⇐
\times	×	\rdarrow	⇒

Legends

The **Show legends** check box is selected by default to display the plotted expressions to the right of the plot. In plots where each line represents a certain time value, eigenvalue, or parameter value, these values also display. For 1D point plots, the legend displays the coordinate (or vertex number).

When **Automatic** is selected from the **Legends** list (the default), select or clear the **Expression** and **Description** and check boxes to control what to include in the automatic legends (by default it includes the description only). If **Manual** is selected from the **Legends** list, enter your own legend text into the table.

Arrow Positioning

Note

Arrow positioning is available when the plot dimension is the same as the highest dimension available. Arrow positioning is available for 2D arrow surface plots in 2D plot groups but not for 2D arrow surface plots in 3D plot groups.

Under **Arrow Positioning**, and based on space dimension, in the **x grid points**, **y grid points**, and **z grid points** fields (**r grid points** and **z grid points** in 2D axial symmetry) select an **Entry method**—**Number of points** or **Coordinates**:

- If **Number of points** is selected, enter the number of **Points** in each direction (default: 15).
- If **Coordinates** is selected, enter **Coordinates** or click the **Range** button ( ) to select and define specific coordinates.

See Also

Entering Ranges and Vector-Valued Expressions

Principal Components and Positioning

These sections are available for the Principal Stress Volume and Principal Stress Surface Plots.

PRINCIPAL COMPONENTS

Under **Principal Components**, select a **Type**—**Principal stress** or **Principal strain**.

- Under **Principal values**, enter information in the **First**, **Second**, and **Third Value** fields. The default are the three principal stresses (`solid.sp1`, `solid.sp2`, and `solid.sp3`, for example, for a Solid Mechanics interface; the prefix is the physics interface identifier), plotted using red, green, and blue arrows, respectively.
- Under **Principal directions**, enter information in the table under **First**, **Second**, and **Third** for the **X**, **Y**, and **Z** coordinate fields. The defaults are the directions (eigenvectors) for the first, second, and third principal stress.

For transient problems, enter a **Time**.

POSITIONING

This section is not available for the 3D Principal Stress Surface plot.

Under **Positioning** select an **Entry method**— **Number of points** or **Coordinates** for the **x grid points**, **y grid points**, and **z grid points**. If **Number of points** is selected, enter the number of **Points** in each direction (the default is 7 for Principal Stress Volume plots and 15 for Principal Stress Surface plots). If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

Defining the Number of Levels

For Contour plots, Contour data sets, and Isosurface data sets, under **Levels** and from the **Entry method** list, select **Number of Levels** or **Levels**. If **Number of Levels** is selected, enter the total number of levels in the **Total levels** field (the default is 20). Otherwise, enter the values of the contour **Levels** or click the **Range** button ( ) to define specific range of levels.

Selecting Color Tables

For many plot types you can select the color table to use for coloring the surfaces, boundaries, contours, streamlines, slices, and so on. These color tables use 1024 colors each. The best way to compare the color tables is to open a model and experiment with the options.

RAINBOW AND RAINBOW LIGHT

Rainbow is the default for plots that support color tables. The color ordering corresponds to the wavelengths of the visible part of the electromagnetic spectrum. It starts at the small-wavelength end with dark blue. The colors range through shades of blue, cyan, green, yellow, and red. The disadvantage of this color table is that people with color vision deficiencies (affecting up to 10% of technical audiences) cannot see distinctions between reds and greens.

RainbowLight is similar but uses lighter colors.

THERMAL, THERMALEQUIDISTANT, AND THERMALLIGHT

Thermal colors range from black through red and yellow to white, corresponding to the colors iron takes as it heats up.

ThermalEquidistant is similar but uses equal distances from black to red, yellow, and white, which means that the black and red regions become larger compared to the Thermal color table.

ThermalLight is similar but uses equal distances from dark red to orange, yellow, and white, which means that the region with the lowest values is red instead of black as it is in the Thermal color table.

CYCLIC

The **Cyclic** color table is useful for displaying periodic functions because it has a sharp color gradient —it varies the hue component of the hue-saturation-value (HSV) color model, keeping the saturation and value constant (equal to 1). The colors begin with red, then pass through yellow, green, cyan, blue, magenta, and finally return to red.

WAVE AND WAVELIGHT

The **Wave** color table is useful for data that naturally have positive and negative attributes in addition to a magnitude. An example of a double-ended scheme, it ranges linearly from blue to light gray, and then linearly from white to red. When the range of the visualized quantity is symmetric around zero, the color red or blue indicates whether the value is positive or negative, and the saturation indicates the magnitude.

People with color vision deficiencies can interpret the **Wave** color table because it does not use red-green-gray distinctions, making it efficient for 99.98% of the population.

WaveLight is similar and ranges linearly from a lighter blue to white (instead of light gray) and then linearly from white to a lighter red.

TRAFFIC AND TRAFFICLIGHT

The **Traffic** color table spans from green through yellow to red. **TrafficLight** is similar but uses lighter colors.

DISCO AND DISCOLIGHT

The **Disco** color table spans from red through magenta and cyan to blue. **DiscoLight** is similar but uses lighter colors.

GRAYSCALE

The **GrayScale** color table uses the linear gray scale from black to white—the easiest palette to understand and order.

These plots are often easier to use for publication in journal articles. People can also better perceive structural detail in a gray scale than with color. Use this plot to increase the probability that a plot is interpreted correctly by people with color vision deficiencies.

GRAYPRINT

The **GrayPrint** color table varies linearly from dark gray (RGB: 0.95, 0.95, 0.95) to light gray (RGB: 0.05, 0.05, 0.05). Choose this to overcome two difficulties that the GrayScale color table has when used for printing on paper—it gives the impression of being dominated by dark colors, and white is indistinguishable from the background.

CUSTOM COLOR TABLES

You can also add your own continuous and discrete color tables as files with RGB data

See Also

Color Tables in the *COMSOL Multiphysics Reference Guide*

Defining the Color and Data Ranges

Under **Range**, select the **Manual color range** and **Manual data range** check boxes to manually override the color range and data range, respectively, with values in the **Minimum** and **Maximum** field, or use the sliders to control values.

Defining the Coloring and Style

Depending on the plot type and space dimension, the following options are available and defined under **Coloring and Style**. The items are listed in alphabetical order.

ARROW BASE

Select **Tail** (the default) to position the arrow’s tail at the arrow position, or **Head** to position the arrow’s head at the arrow position.

ARROW LENGTH

Select an **Arrow length**:

- **Proportional** (the default), so that the length of the arrows is proportional to the magnitude of the quantity they represent.

- **Normalized**, so that all arrows have the same length.
- **Logarithmic**, so that the length of the arrows is proportional to the logarithm of the magnitude of the quantity they represent. This makes arrows representing small values relatively larger. The value in the **Range quotient** field (default: 100) determines the ratio between the smallest and largest values in the range of values for the logarithmic arrow length.

ARROW PLACEMENT

Select a **Placement** of the arrows—**Uniform**, **Elements**, or **Uniform anisotropic**.

- Select **Uniform** (the default) for arrows positioned uniformly on the surface
- Select **Elements** for arrows positioned in the mesh elements (that is, more densely placed arrows where the mesh density is high).
- Select **Uniform anisotropic** to position the arrows using an anisotropic density (that is, more arrows in some directions than in others). If **Uniform anisotropic** is selected, use the **x weight**, **y weight**, and (in 3D) **z weight** fields to give weights for the arrow density in the different directions (using positive scalar weights). The default weights are 1 in all directions. A higher value increases the arrow density in the corresponding direction.

Number of Arrows

When **Uniform** or **Uniform anisotropic** is selected as the **Placement**, also specify the **Number of arrows** (default: 200).

ARROW TYPE

Select an **Arrow type**—**Arrow** or **Cone**.

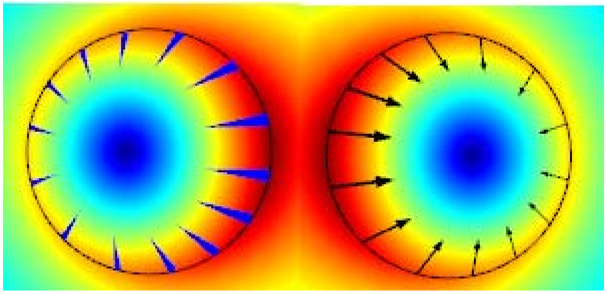


Figure 21-2: Example of 2D model arrow types: cone (left) and arrow (right).

COLOR

For arrows, and unless a **Color Expression** subnode determines the arrow colors, select an arrow **Color** or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

For lines, select a **Color**—**Custom**, **Cycle**, **Black**, **Blue**, **Cyan**, **Gray**, **Green**, **Magenta**, **Red**, **White**, or **Yellow**. If you select **Cycle**, it cycles through all the colors. If you select **Custom**, click the **Color** button to select a custom color from the palette. Enter a line **Width** or use the slider to select.

COLOR LEGEND

Color legend is selected by default. Click to clear the check box if required. The legend displays to the right of the plot.

See Also

Legends

COLOR TABLE

If the default (**Rainbow** in most plots) is not suitable for the plot, try other options. See Color Tables below for details. In some cases, select a **Coloring**—**Color Table** (default) or **Uniform**. If **Uniform** is selected, select a **Color** or **Custom** to choose a different color.

Reverse Color Table

Select the **Reverse color table** check box to reverse the order of the colors in the color table.

Symmetrize Color Range

Select the **Symmetrize color range** check box to obtain a color range centered around zero. This setting is useful for visualizing wave-like solutions with zero bias.

See Also

Selecting Color Tables

CONTOUR TYPE

Select a **Contour type**—**Lines** or **Filled**. If **Lines** is selected, you can also select the **Level labels** check box to display line labels on the graph.

GRID

Select a **Grid**—**None** (the default), **Fine**, **Normal**, or **Coarse**. If **Fine**, **Normal**, or **Coarse** are selected, also choose a **Color** for the grid.

INTERPOLATION AND NUMBER OF INTERPOLATED TIMES

Particle trajectories can appear jagged because the output times for the simulation are too few to result in a smooth plot. You can improve the particle trajectories by using a uniform interpolation of the data for the particle trajectories. From the **Interpolation** list, select **None** for no interpolation (the default), or select **Uniform** to use a uniform interpolation of the data using additional interpolated times defined in the **Number of interpolated times** field. The default is 100 interpolated times.

Note

Interpolation of lines is only available for Particle Trajectories plots, which are available for use with Particle data sets created with the Particle Tracing Module.

LINE STYLE

The line styles available depend on the type of plot and the space dimension and include these options.

- **Cycle**, **Solid**, **Dotted**, **Dashed**, or **Dash-dot**. If **Cycle** is selected, it cycles through all the options.
- **Line**, **Tube**, or **None**. If **Tube** is selected, enter a **Tube radius expression** for the radius of the tube; click the **Replace Expression** button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m.

LINE TYPE

Select a **Line type**—**Line** or **Tube**. For 3D Streamline plots, **Ribbon** is also available. Ribbons are an alternative to tubes for visualization of, for example, the vorticity of a flow field.

- If **Tube** is selected, enter a **Tube radius expression** (the radius of the tube); click the **Replace Expression** button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m.
- If **Ribbon** is selected, enter a width for the ribbons in the **Width expression** field; click the Replace Expression button ( ) to select a predefined expression to replace

the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 m. Select the **Width scale factor** check box to enter a user-defined scaling of the ribbons' width in the associated field. By default, the program scales the width automatically.

LINE MARKERS OR MARKER TYPE

Select a **Marker** type—**None**, **Cycle**, **Asterisk**, **Circle**, **Diamond**, **Plus sign**, **Point**, **Square**, **Star**, or **Triangle**.

If a marker is selected (excluding **None**), then from the **Positioning** list select **Interpolated** or **In data points**. For **Interpolated**, enter the **Number** of markers to display (the default is 8) or use the slider to select. If **In data points** is selected, the markers appear in the data points for the plot (which for a plot of a 1D solution are the mesh nodes).

Note

The line markers are only available for Histogram plots using a continuous function.

LINE WIDTH

Enter a line **Width** or use the slider to select.

PLOT ALONG LINES WHEN ANIMATING

If the plan is to create an **Animation** report, select the **Plot along lines when animating** check box. This is useful for Particle Trajectories, Particle Tracing, and Particle Tracing with Mass plots.

POINT MOTION

Select a **Point motion** to specify what should happen **When particles leaves domain**—**Stick to Boundary** (to plot the points on the boundary at the exit point) or **Disappear** (to not render these points at all).

For static fields, specify the **End time** in the **Advanced** section. It is possible that all particles have left the domain at the selected time. In that case, all points appear at the outflow boundary if **Stick to boundary** is selected, and no points appear if **Disappear** is selected. To make the points appear, specify an earlier end time.

POINT RADIUS

Enter a **Point radius expression**; click the **Replace Expression** button ( ) to select a predefined expression to replace the current expression, or press Ctrl+Space to insert a predefined expression (SI unit: m). The default is 1 mm.

POINT STYLE

Under **Point style**, select a **Type**—**Point**, **None**, or **Comet tail**. If **Point** or **Comet tail** is selected, enter a Point Radius and Radius Scale Factor.

> *Note*
>
> **Comet tail** is available with the Particle Tracing Plots, Particle Tracing with Mass Plots, and the Particle Trajectories Plots and requires a license for the Particle Tracing Module.
>
> Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity; visually, it is the same as the tail of a comet traveling through space.
>
> These additional settings are available when **Comet tail** is selected—Tail and Tail Components and Tail Scale Factor.

RADIUS SCALE FACTOR

Select the **Radius scale factor** check box to enter a scalar number for the scale factor.

RANGE QUOTIENT

If **Logarithmic** is selected as the arrow length, enter a **Range quotient**, which is the ratio between the maximum arrow length and the arrow length below which no arrow is drawn. The default is 100.

SCALE FACTOR

Enter a **Scale factor** for the arrows using a positive scalar number in the field or by using the associated slider (for scale factors between 0 and 1).

TAIL AND TAIL COMPONENTS

Define the length and direction of the comet tail as a vector expression. For the **Tail expression**, click the **Replace Expression** button ( ) to select a predefined expression to insert a predefined expression into the **Tail, x component**, **Tail, y component**, and **Tail, x component** (for 3D plots) fields. The expressions available are based on the physics interfaces used in the model. The default expressions (typically `pt.nvx`, `p.nvy`, and `pt.nvz`) represent the negative of the particle velocity.

TAIL SCALE FACTOR

Select the **Tail scale factor** check box to enter a scalar number between 0 and 1 or use the slider to select.

WIREFRAME

To plot only on the visualization mesh, select the **Wireframe** check box. This displays the surface as a triangular grid.

Defining Element Filters

For Mesh and Volume plots you can specify the elements to display under **Element Filter**. Without filtering, the plots display all elements. Using element filters you can highlight elements based on, for example, their mesh quality, size, or location.

To define an element filter, select the **Enable filter** check box and select a **Criterion**—**Logical expression** (the default), **Random** or **Expression**. For Mesh plots, **Worst quality**, **Best quality**, and **Size** are also available, which filters elements with the worst element, best quality, or size, respectively. When you choose **Size**, the fraction that you specify is the fraction with the smallest elements. For example, a fraction of 0.1 plots the smallest 10% of the elements.

- If **Expression** or **Logical expression** is selected, enter an **Expression** in the field. For example, an expression can be `abs(x-y)` to plot a fraction of elements closest to the line $y = x$ (that is, the fraction that you specify is the fraction where the expression evaluates to the smallest values). An example of a logical expression is `(h>0.1)&&(h<0.4)`, which shows the elements with an element size between 0.1 and 0.4 (`h` is the predefined variable for the mesh element size); another example is `x>0`, which plots elements in the right half-plane only.
- If **Random**, **Expression**, **Worst quality**, **Best quality**, or **Size** is selected, specify the **Fraction** of elements (0–1) to show (the default is 1, which means that all elements are included).

Defining Shrinking of Elements

For Mesh and Volume plots, under **Shrink Elements**, enter an **Element scale factor** between 0 and 1 to scale elements in the mesh plot. The default value is 1, which means no shrinking. Using a smaller value shrinks the size of the elements in the plot accordingly. This can be useful for visualizing individual elements and looking at interior elements in a volume plot.

Entering Quality Settings for Plot Settings Windows

Many plots have a **Quality** section where you can select a plot resolution, enforce continuity, and specify the use of accurate derivative recovery. The steps for this section vary slightly based on the plot but are basically as follows.

1 Under **Quality**, select a plot **Resolution**—**Finer**, **Fine**, **Normal**, **Coarse**, **No refinement** or **Custom**. A higher resolution means that elements are split into smaller patches during rendering.

Note

Custom refinement applies to the base data set. This means that the number of elements in the model can increase radically if the plot uses for example a revolve data set, as the refinement is applied to the solution data set.

2 To enforce continuity on discontinuous data, under **Quality**, from the **Smoothing** list, select:

- **None**: to plot elements independently.
- **Internal** (the default): to smooth the quantity inside the geometry but no smoothing takes place across borders between domains with different settings.
- **Everywhere**: to apply smoothing to the entire geometry.

The default is to smooth the quantity except across borders between domains, where there is often a sharp transition from one material to another or between different types of physics.

3 Under **Quality**, the **Recover** default is **Off** because the accurate derivative recovery takes processing time. This recovery is a polynomial-preserving recovery that recovers fields with derivatives such as stresses or fluxes with a higher theoretical convergence than smoothing (see Accurate Derivative Recovery below for more information).

To edit the default and use accurate derivative recovery, from the **Recover** list select:

- **Within domains**: to perform recovery inside domains.
- **Everywhere**: to apply recovery to all domain boundaries.

ACCURATE DERIVATIVE RECOVERY

Plotting and evaluating stresses or fluxes boils down to evaluating space derivatives of the dependent variables. By default, computing a derivative like `ux` or `uxx` (first and second derivatives of u with respect to x) is done by evaluating the derivative of the shape functions used in the finite element approximation. These values have poorer accuracy than the solution `u` itself. For example, `exitus` if `u` has linear elements. COMSOL Multiphysics then evaluates the derivatives (and `u` itself) using a polynomial-preserving recovery technique by Z. Zhang (see Ref. 1). The recovery is only applied on variables that are discretized using Lagrange shape functions.

The polynomial-preserving recovery is a variant of the superconvergent patch recovery by Zienkiewicz and Zhu that forms a higher-order approximation of the solution on a patch of mesh elements around each mesh vertex. For regular meshes, the convergence rate of the recovered gradient is $O(h^{p+1})$—the same as for the solution itself. Near boundaries the accuracy is not as good, and it might even be worse than without recovery. Results evaluation is about 2–5 times slower when using accurate derivative recovery. For this reason, prefer using recovery of stresses or fluxes for surface plots, contour plots, slice plots, and isosurface plots. Use recovery to a lesser extent for volume plots and line plots that plot values near or on the domain boundaries.

By default, the accurate derivative recovery smoothes the derivatives within each group of domains with equal settings. Thus, there is no smoothing across material discontinuities. You find the setting for accurate derivative recovery in the plot node's settings windows' **Quality** section. Due to performance reason, the default value for **Recover** list is **Off** (that is, no accurate derivative recovery). Select **Within domains** to smooth the derivatives within each group of domains with equal settings. Select **Everywhere** to smooth the derivatives across the entire geometry.

Reference

1. A. Naga and Z. Zhang, "The Polynomial-Preserving Recovery for Higher Order Finite Element Methods in 2D and 3D," *Discrete and Continuous Dynamical Systems—Series B*, vol. 5, pp. 769–798, 2005.

Inheriting Style Options

All 2D and 3D plot types (except Mesh) have the **Inherit Style** section. Use this feature to maintain a consistent style between plots.

After there is more than one plot in a plot group, the **Plot** list makes these plots available to select the attributes you want to maintain between plots. For any plot with this section, and if the check box is applicable to the plot type, the **Color**, **Color and data range**, and **Deform scale factor** check boxes are selected by default. The attributes vary based on the plot type and include:

- Arrow scale factor
- Color
- Color and data range
- Deform scale factor
- Height scale factor
- Ribbon width scale factor
- Point radius scale factor
- Tail scale factor
- Tube radius scale factor

The default **Plot** selected is **None**, which means that styles are not inherited for any plots. If you want to inherit a style, add and select a plot type from the **Plot** list. All attributes automatically inherit the style from the selected plot. To prevent a plot attribute from being inherited, click to clear the check box or select **None** from the **Plot** list.

Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings

See Also

For details about the solvers and studies, including the availability by module, see Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis.

For the plots that incorporate harmonic perturbation, small-signal analysis, or prestressed analysis, there are additional settings that display in the a variety of plot windows.

Expression Evaluated For

Select an option from the **Expression evaluated for** list—**Static solution**, **Harmonic perturbation**, **Total instantaneous solution**, **Average for total solution**, **RMS for total solution**, or **Peak value for total solution**. Each option is described below.

Tip See Special Operators for information about the operators described in this section.

Static Solution

The expression is evaluated by taking the values of any dependent variables from the linearization point of the solution. This is achieved by wrapping the expression in the `linpoint` operator.

Harmonic Perturbation

If **Harmonic perturbation** is selected, the **Compute differential** check box is also available. If the check box is not selected, the expression is evaluated by taking the values of any dependent variables from the harmonic perturbation part of the solution.

If the **Compute differential** check box is selected (default), the differential of the expression, with respect to the perturbation, is computed and evaluated at the linearization point. This is achieved by wrapping the expression in the `lindev` operator. For expressions which are linear in the solution, the two options are the same.

Total Instantaneous Solution

The expression is evaluated by adding the linearization point and the harmonic perturbation and taking the real part of this sum. This is achieved by wrapping the expression in the `lintotal` operator. The phase and amplitude of the harmonic perturbation part can be set in the corresponding data set.

Average for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then averaging over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalavg` operator.

RMS for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then taking the RMS over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalrms` operator.

Peak Value for Total Solution

This is the same as evaluating for **Total instantaneous solution** and then taking the maximum over all phases of the harmonic perturbation. This is achieved by wrapping the expression in the `lintotalpeak` operator.

Creating 1D, 2D, 3D, and Polar Plots

A plot group contains one or more plots (for example, combining a surface plot and a streamline plot) using the same data set, such as a solution. Define plot groups for 1D, 2D, and 3D plots and then create individual or a series of plots in a plot group. Use the information in the form of data and images to generate a report or export to a file.

Attributes can also be added to a plot to modify the plot's behavior—**Deformation** ( ) attributes deform a plot, **Color Expression** ( ) attributes modify the color of a plot, and filtering to only include parts of the plot is selected using the **Filter** attribute ( ). Select an appropriate color table for the plots' color expressions based on your audience and what you plan to do with the final analysis.

These main topics are included in this section. Links to individual plots are found in Table 21-5 and Table 21-6.

- Common Plot Settings Cross References
- 1D Plot Groups
- Polar Plot Groups
- 2D Plot Groups and 3D Plot Groups
- Color Expressions
- Deformations
- Filters
- Height Expressions

See Also

- Results Evaluation and Visualization
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Common Plot Settings
- Creating Cross-Section Plots and Combining Plots
- Working with Geometry

1D Plot Groups

Important

It is not possible to create plots in a higher dimension than the data set being visualized. For example, a 2D plot group cannot be used to visualize a solution for a 1D model. 1D plot groups can be used for all models.

Note

At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot.

The time-related settings only display on the interfaces for time-dependent models.

Use a **1D Plot Group** ( ) to plot the graphs listed in Table 21-5. 1D plot groups are also used to create cross-section plots.

1. Define one or more **Data Sets**.
2. In the **Model Builder**, right-click **Results** and add a **1D Plot Group** ( ).
3. Under **Data,** select a **Data set**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. Also, if applicable, select the times, eigenvalues, or parameters for which you want to plot the solution data.
4. Under **Title**, the **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.
5. Under **Plot Settings**, manually enter axis labels by selecting the **x-axis label** and **y-axis label** check boxes.
6. Under **Axis**, edit these settings as required.
   - Select the **Manual axis limits** check box to edit the **x minimum**, **x maximum**, **y minimum**, and **y maximum** limits already assigned based on the data set.
   - Select one or all of these check boxes: **Preserve aspect ratio**, **x-axis log scale**, **y-axis log scale**. When **Preserve aspect ratio** check box is selected, the distances on the x-axis and y-axis are kept equal.

7 Under **Grid**, edit these settings as required:

- Select the **Manual spacing** check box to edit the predefined **x spacing** and **y spacing** fields.
- Enter values as required in the **Extra x** and **Extra y** fields. Click the **Range** button ( ) to define a range of values.

8 Under **Legend**, select the position for legends in all plots for the plot group from the **Position** list—**Upper right** (the default), **Lower right**, **Upper left**, or **Lower left**. Legends in all plots in the plot group use this position.

9 Under **Window Settings**, edit these settings as required:

- Select a **Plot window**. The **Graphics** window is the default but any other plot window can be selected, or select **New window** to plot in a new window.
- Select the **Window title** check box to enter a custom title (except for the **Graphics** window), which is then available in the **Plot window** list for all models. Click the **Add plot window** button (+) to add a plot window to the list of available windows.

10 Add plots to the group and define it as required. When ready, click the **Plot** button ( ) or right-click the node and select **Plot**.

INTERACTIVE

Use a combination of data sets and plots to create a cross-section point plot and cross-section line plot.

To add plots to a group, right-click the **1D Plot Group** node to select as many as required. Each plot group can have several plots combined to create a meaningful representation of the data.

ADDING PLOTS TO THE 1D PLOT GROUP IN THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** ( ), right-click the **1D Plot Group** ( ) node and select an option from the context menu such as **Line Graph** or **Point Graph**. Continue defining each plot as described (see Table 21-5).

When the plot type is defined, click the **Plot** button ( ), or right-click the node and select **Plot**. The plot displays in the window selected from the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

TABLE 21-5: 1D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION
Far Field Plots		Plot the value of a global variable. This plot is available for the RF Module and the Acoustics Module.
Global Plots		Plot a global scalar quantity as a function of time or a parameter.
Histogram		Plot a histogram showing the distribution of a quantity.
Line Graphs		Plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line.
Nyquist Plot		Display a Nyquist plot, which shows the magnitude and phase of a frequency response.
Point Graphs		Visualize the value in a point along time or a parameter value. The point can be a point in the geometry or a cut point.
Table Graph		Table plots display data from a table with one line per output column.
*1D, 2D, and 3D Cross-Section Point Plots		A point cross-section plot makes it easy to view an expression at an arbitrary set of spatial coordinates and results in a line plot.

* Cross-section plots are not selected from this menu. Instead, these are created it in two steps using data sets and plot groups to generate the plot.

See Also

- Common Plot Settings Cross References
- Polar Plot Groups
- 2D Plot Groups and 3D Plot Groups
- Solver Studies and Study Types
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Common Plot Settings
- Creating Cross-Section Plots and Combining Plots

Polar Plot Groups

The **Polar Plot Group** node ( ) creates a graph for *polar plots*: plots of a function in polar coordinates: the radius *r* and the angle θ. This is useful for visualizing, for example, a radar cross section or other similar polar plots for electromagnetic or acoustic wave models. The available plot types and settings for the **Polar Plot Group** node are similar to those for the 1D Plot Group.

See Also	- 1D Plot Groups - Common Plot Settings Cross References - Common Plot Settings - Combining Plots

2D Plot Groups and 3D Plot Groups

Important	It is not possible to create plots in a higher dimension than the data set being visualized. For example, a 3D plot group cannot visualize a solution for a 2D model. In some cases data sets add dimensions to their parent, for example, Revolve 2D. In this case, the solution is a 2D data set, but the revolved data is a 3D data set, which can be used for 3D plot groups. Similarly, 2D plot groups can be used for 3D solutions when there are, for example, cut planes or other data sets that remove a dimension.

Note	At any time during plot creation, click the **Plot** button ( ) to visualize a data set or plot. The time-related settings only display on the interfaces for time-dependent models.

Use a **2D Plot Group** ( ) to combine several 2D plots into one to visualize all the plots simultaneously. Use a **3D Plot Group** ( ) to combine several 3D plots into one to visualize all the plots simultaneously. For example, combine a surface plot and a streamline plot using the same data set.

1 Define one or more **Data Sets**.

2. In the **Model Builder**, right-click **Results** and select **2D Plot Group** (■) or **3D Plot Group** (■).
3. Under **Data** select a **Data set**. Click the **Go to Source** button (≣) to move to the node that the selection in the list next to the button refers to. Also, if applicable, select the time, eigenvalue, or parameter for which you want to plot the solution data.
4. Under **Title**, the **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.
5. Under **Plot Settings**, edit these settings as required:
   - Select a **View**. The default is **Automatic**. Other **Views** are defined under **Definitions**.
   - (2D only) The **x-axis label** and **y-axis label** check boxes are cleared by default, indicating that empty axis labels are used by default. Select the check boxes to enter labels for the *x*-axis and the *y*-axis.
   - The **Show hidden objects** check box is cleared by default. Click to select it if you want to include geometric entities that are hidden in the plot's view.
   - The **Plot data set edges** check box is selected by default. Click to clear if required. Otherwise, select a **Color** (**Black** is the default) or select **Custom** to click the **Color** button and choose a different color from a color palette. Select a **Frame**—**Material** (the default), **Mesh**, **Geometry frame**, or **Spatial**.
6. Under **Color Legend**, specify how to position the color legends (color scales) for the plots in the plot group. Select **Alternating**, **Bottom**, **Left**, **Left double**, **Right** (the default), or **Right double** from the **Position** list. Selecting **Alternating** puts the first color legend to the right of the plot, the second color legend to the left of the plot, and so on. Selecting **Left double** or **Right double** adds two color legends positioned vertically above each other on the left or right side of the plot.

Note

The default precision for the color legend labels is 5 digits. You can change the precision in the **Preferences** dialog box, using the **Color legend** field under **Precision** on the **Graphics** page.

7. Under **Window Settings**, edit these settings as required:
   - Select a **Plot window**. The **Graphics** window is the default setting, but any other plot window can be selected, or select **New window** to plot in a new window.
   - Select the **Window title** check box to enter a custom title (except for the **Graphics** window), which is then available in the **Plot window** list for all models. Click the

Add plot window button ( ) to add a plot window to the list of available windows.

8 Add a plot to the group and define it as required. When ready, click the **Plot** button (), or right-click the node and select **Plot**.

INTERACTIVE

Use a combination of data sets and plots to create a cross-section point plot, cross-section line plot, or cross-section surface plot.

To add plots to a group, right-click the **3D Plot Group** or the **2D Plot Group** node to select as many as required. Each plot group can have several plots combined to create a meaningful representation of the data.

ADDING PLOTS TO 2D AND 3D PLOT GROUPS IN THE MODEL BUILDER

Under **Results** ( ) in the **Model Builder** (), right-click a **2D Plot Group** ( ) or **3D Plot Group** () node and select one or more options from the context menu. Each plot group can have several plots combined to create a meaningful representation of the data. Continue defining each plot as described (see Table 21-6).

Also add attributes to a plot to modify the plot's behavior—**Deformation** () attributes deform a plot, **Color Expression** ( ) attributes modify the color of a plot, and element selection is selected using the **Filter** attribute (). A **Height Expression** () can also be added to some 2D plots.

When the plot type is defined, click the **Plot** button (), or right-click the node and select **Plot**. The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Arrow Line Plots	3D 2D	Plot a vector quantity as arrows on lines or edges (3D).	Color Expression, Deformation, and Filter
Arrow Surface Plots	3D 2D	Visualize a vector quantity in arrows.	Color Expression, Deformation, and Filter

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Arrow Volume Plots		Visualize a vector quantity as arrows in a volume.	Color Expression, Deformation, and Filter
Contour Plots	3D 2D	Visualize a scalar quantity as a contour plot.	Color Expression, Deformation, and Filter
Coordinate System Volume, Surface, and Line Plots	3D 2D	Plot coordinate systems for 2D and 3D models. Found on the More Plots submenu.	Deformation (2D surface and 3D volume) Deformation and Filter (2D Line, 3D surface, and 3D line)
Far Field Plots	3D 2D	Plot the value of a global variable. This plot is available for the RF Module and the Acoustics Module.	none
Histogram		Plot a histogram showing the distribution of a quantity.	Height Expression
Isosurface Plots		Plot a scalar quantity as an isosurface plot.	Color Expression, Deformation, and Filter
Line Plots	3D 2D	Plot a quantity on lines, boundaries (2D), or edges (3D).	Deformation, Filter, and Height Expression (2D only)

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Max/Min Volume, Max/Min Surface, and Max/Min Line Plots	Max/Min: Volume Surface 2D Surface 3D Line 2D Line 3D	Plot the maximum and minimum of an expression and the points there they are attained within the geometry. Found on the More Plots submenu.	Deformation
Mesh Plots		To display a mesh.	Deformation and Filter
Multislice Plots		Display a scalar quantity on slices in multiple directions inside a 3D domain. Found on the More Plots submenu.	Deformation and Filter
Particle Tracing Plots	3D 2D	Visualize the trajectory of a massless particle subject to a flow field. Found on the More Plots submenu.	Color Expression and Deformation
Particle Tracing with Mass Plots	3D 2D	Make a particle tracing plot considering the particle's mass. Found on the More Plots submenu.	Color Expression and Deformation
Particle Trajectories Plots	3D 2D	Visualize the trajectory of a massless particle subject to a flow field in 2D or 3D. This plot requires the Particle Tracing Module.	Color Expression and Deformation

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Phase Portrait Plots		Visualize particle traces using a phase portrait in 2D. Found under the More Plots submenu. This plot requires the Particle Tracing Module.	Color Expression
Poincaré Map Plots	3D 2D	Visualize particle traces as a Poincaré map in 2D, 2D axial symmetry, and 3D. Found on the More Plots submenu. This plot requires the Particle Tracing Module.	Color Expression
Principal Stress Volume Plots		Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter
Principal Stress Surface Plots	3D 2D	Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter
Scatter Surface Scatter Surface and Scatter Volume Plots	2D	Scatter surface plots, found on the More Plots submenu, visualize a scalar quantity as scattered spheres or disks. The radius of the spheres can be proportional to the value of a quantity.	None
Scatter Volume Scatter Surface and Scatter Volume Plots	3D	Scatter volume plots can be used as alternatives to arrow plots for scalar quantities or when to get a feeling for how entities correlate.	None
Slice Plots		Display a scalar quantity on slices inside a 3D domain.	Deformation and Filter
Streamline Plots	3D 2D	Plot a vector field as a streamline plot.	Color Expression and Deformation

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
Surface Plots	3D 2D	Display a quantity on a 2D domain or a 3D boundary.	Deformation, Filter, and Height Expression (2D plots)
Table Surface Plots		Used with 2D plot groups to visualize the data in a table that represents a matrix of values that are functions of two independent parameters.	Height Expression
Volume Plots		To display a quantity inside a 3D domain.	Deformation and Filter
PLOT ATTRIBUTES			**ADD TO THESE PLOTS**
Color Expressions		To add coloring to the shapes defined by a plot.	Arrow (line, surface and volume), contour, isosurface, particle tracing, particle trajectories, principal stress (volume and surface), and streamline
Deformations		Deform the plot according to a vector quantity, for example, the displacement field in structural mechanics.	All plots except scatter plots.
Filters		Filter the element selection for a plot.	Arrow (line and surface), contour, coordinate system (line and surface), isosurface, line plot, mesh, principal stress, slice, surface, and volume.
Height Expressions		Height attributes introduce a 3D height on a 2D surface plot.	2D line plot, 2D surface plot, 2D table surface plot

TABLE 21-6: 2D AND 3D PLOT TYPES

LINK TO SECTION	ICON	DESCRIPTION	PLOT ATTRIBUTES AVAILABLE
CROSS SECTION PLOTS			
*1D, 2D, and 3D Cross-Section Point Plots		View an expression at an arbitrary set of spatial coordinates.	
*2D Cross-Section Line Plots and 3D Cross-Section Line Plots		Create lines through 2D and 3D geometries to visualize along the line. Also explains how to use the interactive cross-section toolbar buttons.	
*3D Cross-Section Surface Plot		Create planes through a 3D model in a 2D geometry to visualize on the plane. The cut plane corresponds to an orthogonal 2D coordinate system embedded in 3D. Also explains how to use the interactive cross-section toolbar buttons.	
* Cross-section plots are not selected from this menu. Instead, these are created in two steps using data sets and plot groups to generate the plot or use interactive cross-section toolbar buttons.			

See Also

- Common Plot Settings Cross References
- 1D Plot Groups
- Polar Plot Groups
- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Common Plot Settings
- Creating Cross-Section Plots and Combining Plots
- User-Defined Views
- Solver Studies and Study Types

Common Plot Settings Cross References

Under **Results** there are many common sections on each settings window. Table 21-7 provides cross references to the information. Availability depends on the plot type and in some cases, the space dimension. For the **Coloring and Style** section, see Table 21-8.

TABLE 21-7: DETAILS FOR THE COMMON PLOT SETTINGS SECTIONS

SECTION ON THE SETTINGS WINDOW	CROSS REFERENCE
Arrow Positioning	Arrow Positioning
Color (3D plot group, Far Field plots)	Expressions and Predefined Quantities
Coloring and Style	See Table 21-8 below and Defining the Coloring and Style
Data	Selecting a Data Set for Plots
Data for Parametric Solver and Parametric Sweep studies	Selecting Solutions for Parametric Sweeps
Element Filter	Defining Element Filters
Expression	Expressions and Predefined Quantities
Inherit Style	Inheriting Style Options
Legends	Legends
Levels	Defining the Number of Levels
Positioning	Principal Components and Positioning
Principal Components	Principal Components and Positioning
Quality	Entering Quality Settings for Plot Settings Windows
Radius	Expressions and Predefined Quantities and Radius Scale Factor
Range	Defining the Color and Data Ranges
r-Axis Data (polar plots)	Expressions and Predefined Quantities
Shrink Elements	Defining Shrinking of Elements
Title	Plot Titles for Plot Groups and Plot Types and Using Special Formats and Symbols in Titles
y-Axis Data (1D plots)	Expressions and Predefined Quantities

COLORING AND STYLE

TABLE 21-8: CROSS REFERENCES FOR THE COMMON COLORING AND STYLE SETTINGS SECTIONS

SECTION	CROSS REFERENCE
Arrow base	Arrow Base
Arrow color	Color
Arrow length	Arrow Length
Arrow type	Arrow Type
Color legend	Color Legend
Coloring	Color Table
Color table	Color Table and Selecting Color Tables
Grid	Grid
Line color	Color
Line markers	Line Markers or Marker Type
Line style (Line, Color, and Width)	Line Style
Line type	Line Type
Line width	Line Style
Number of arrows	Arrow Placement
Placement	Arrow Placement
Plot along lines when animating	Plot Along Lines When Animating
Point color	Color
Point motion	Point Motion
Point radius	Point Radius
Point style, Point type	Point Style
Radius scale factor	Radius Scale Factor
Range quotient	Range Quotient
Reverse color table	Color Table
Scale factor	Scale Factor
Symmetrize color range	Color Table
Tail and Tail components	Tail and Tail Components
Tail scale factor	Tail Scale Factor
Wireframe	Wireframe

See Also

- About Plot Groups and Plotting
- Creating 1D, 2D, 3D, and Polar Plots
- Defining Data Sets
- Defining Derived Values and Tables
- Creating Cross-Section Plots and Combining Plots
- User-Defined Views
- Solver Studies and Study Types

Arrow Line Plots

Use an **Arrow Line** plot to visualize a vector quantity as arrows on lines using a 2D **Arrow Line** ( ) plot, or lines and edges using a 3D **Arrow Line** ( ). Add **Deformation**, **Color Expression**, or **Filter** subnodes as required. For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

Model

See Figure 21-3 and Figure 21-4 for examples from the AC/DC Module and CFD Module Model Libraries.

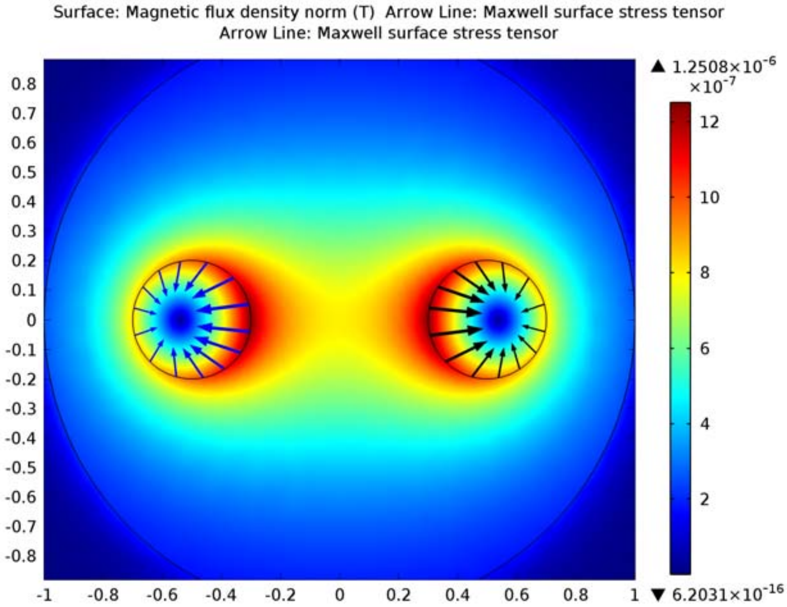


Figure 21-3: A 2D arrow line plot combined with a surface plot. From the AC/DC Module Model Library: Electromagnetic Forces on Parallel Current-Carrying Wires.

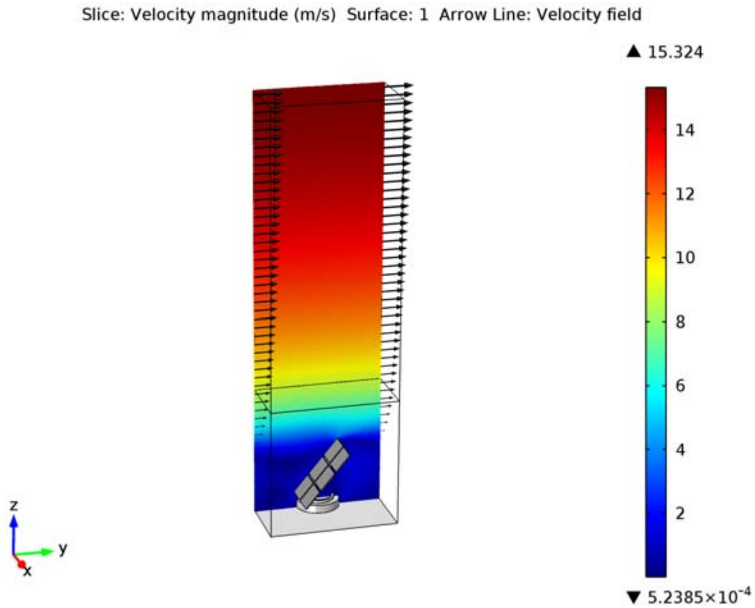


Figure 21-4: A 3D arrow line plot combined with surface and slice plots. From the CFD Module Model Library: Solar Panel.

Arrow Surface Plots

Use an **Arrow Surface** plot to visualize a vector quantity as arrows on a surface using a 2D **Arrow Surface** ( ) or 3D **Arrow Surface** ( ) plot. Add **Deformation**, **Color Expression**, or **Filter** subnodes as required. For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types.

Tip	Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D). Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Model

See Figure 21-5, and Figure 21-6 for examples from the Chemical Reaction Engineering Module Model Library and COMSOL Multiphysics Model Library.

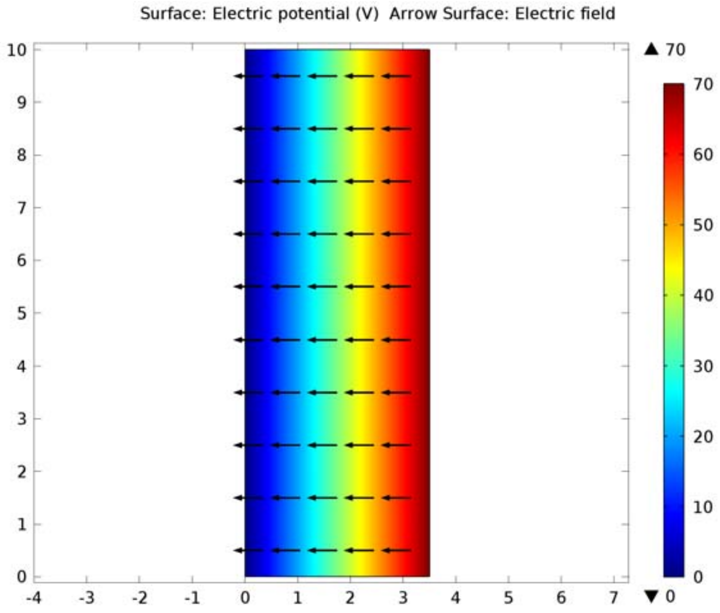


Figure 21-5: A 2D arrow surface plot combined with a surface plot. From the Chemical Reaction Engineering Module Model Library: Isoelectric Separation.

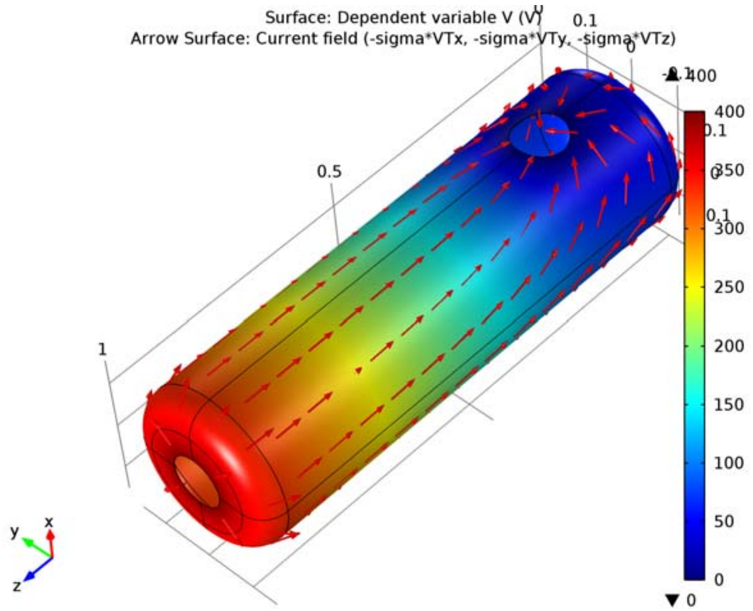


Figure 21-6: A 3D arrow surface plot combined with a surface plot. From the COMSOL Multiphysics Model Library: Shell Diffusion.

Arrow Volume Plots

Use an **Arrow Volume** ( ) plot visualize a vector quantity as arrows in a 3D volume. Add **Deformation**, **Color Expression**, or **Filter** subnodes as required. For example, add a **Color Expression** node to color the arrows with the magnitude of the vector quantity that the arrows represent.

Right-click a **3D Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Model

See Figure 21-7 for an example from the Subsurface Flow Module Model Library.

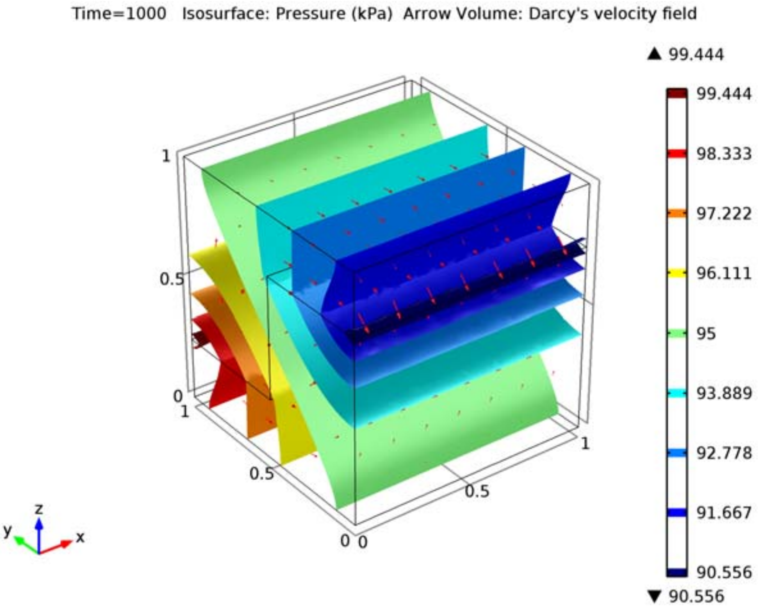


Figure 21-7: A 3D combined arrow volume and isosurface plot. From the Subsurface Flow Module Model Library: Discrete Fracture.

Contour Plots

Use a **Contour** plot to visualize a scalar quantity as a contour in 2D ( ) or 3D ( ) and display the quantity as a set of colored lines. The selected quantity has a constant value on these contour lines, optionally with a 3D height. Add **Deformation**, **Color Expression**, or **Filter** subnodes as required.

Right-click a **2D Plot Group** (🟪) or **3D Plot Group** (🟦) to add these plot types.

Tip	Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D). Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Levels**, **Quality**, **Inherit Style**, and **Coloring and Style**. See below for sections specific to this plot: **Coloring and Style.**

Model	See Figure 21-8 and Figure 21-9 for examples from the COMSOL Multiphysics Model Library and Acoustics Module Model Library.

COLORING AND STYLE

Select a **Contour type**—**Lines** or **Filled**. If **Lines** is selected, also select the **Level labels** check box to display line labels on the graph.

If you select to display level labels, specify the precision (number of significant digits) as a positive integer in the **Precision** field (default: 4). Also select a **Label color**: select a predefined color or select **Custom** to define a custom color by clicking the **Color** button and selecting a color from the color palette that opens.

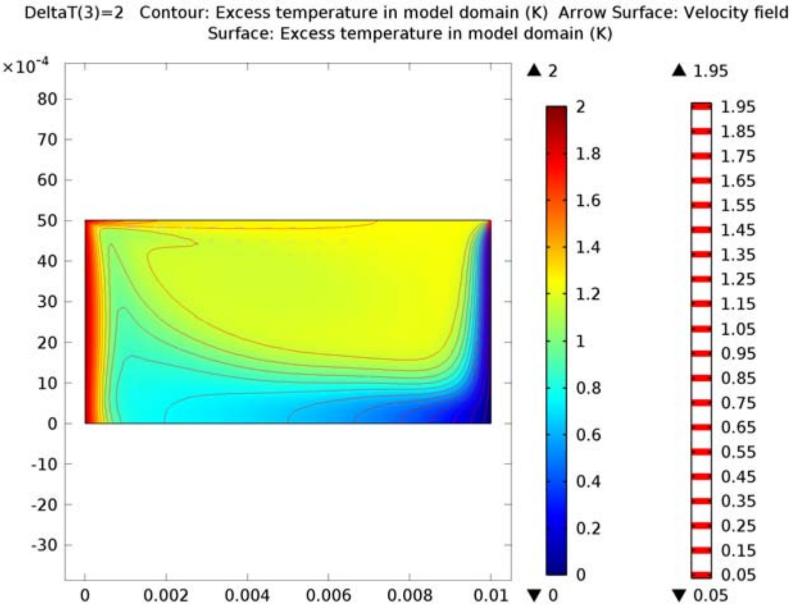


Figure 21-8: A 2D contour plot combined with arrow surface and surface plots. From the COMSOL Multiphysics Model Library: Marangoni Convection.

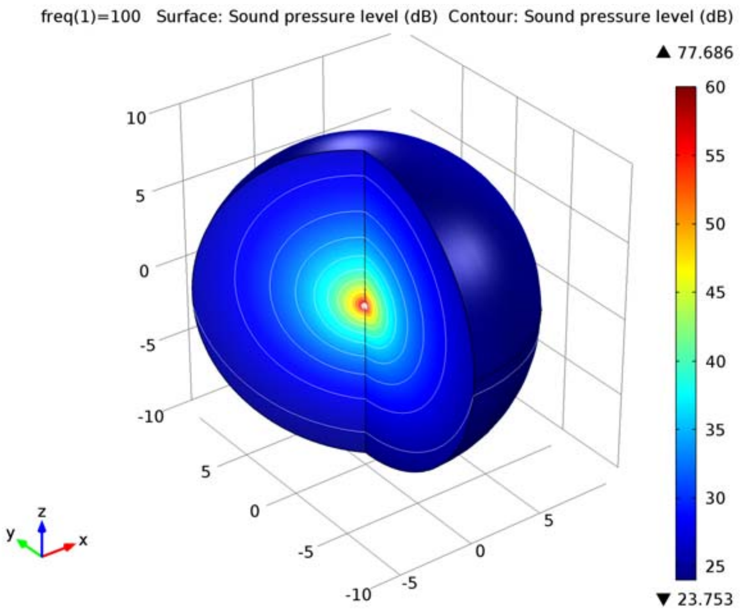


Figure 21-9: A 3D contour plot combined with a surface plot. From the Acoustics Model Library: Doppler Shift.

Coordinate System Volume, Surface, and Line Plots

Add Coordinate System plots in 3D ( ) or 2D ( ). Use the **Coordinate System Volume** (3D), **Coordinate System Surface** (2D and 3D), and **Coordinate System Line** (3D) plots to plot coordinate systems. You can use these plot to visualize the coordinate systems used in, for example, models of piezoelectric devices, where there can be multiple domains, each using its own set of coordinate systems. Right-click to add a **Deformation** or **Filter** (for 2D and 3D Coordinate System Line and 3D Coordinate System Surface plots), if required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

See below for sections specific to these plots: **Coordinate System** and **Positioning**.

Model

See Figure 21-10 for an example from the MEMS Module Model Library.

COORDINATE SYSTEM

For **Coordinate System Surface** in 2D and **Coordinate System Volume** in 3D you can choose, from the **Source** list, if you want to plot a **Coordinate system** (the default) or a **Matrix variable**:

- For **Coordinate system**, choose one of the available coordinate systems from the **Coordinate system** list. The default is **None** (no coordinate system).
- For **Matrix variable**, click the **Replace Expression** button () to select an available matrix variables, which include coordinate transforms to and from the added coordinate systems and physical quantities thar are tensors, such as the thermal conductivity. The selected variable (for example, `ht.k`), then appears next to **Matrix variable**.

For other coordinate system plots, select a coordinate system to plot from the **Coordinate system** list. The default is **None** (no coordinate system), and the list contains any additional coordinate systems that the model includes.

POSITIONING

This section is available for **Coordinate System Volume** ( ) or **2D Coordinate System Surface** ( ) plots.

In the **x grid points**, **y grid points**, and **z grid points** (3D only) fields select an **Entry method**—**Number of points** or **Coordinates**.

- If **Number of points** is selected, enter the number of **Points** in each direction (the default is 15 for 2D Coordinate System Surface plots and 7 for 3D Coordinate System Volume plots).
- If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button () to select and define specific coordinates.

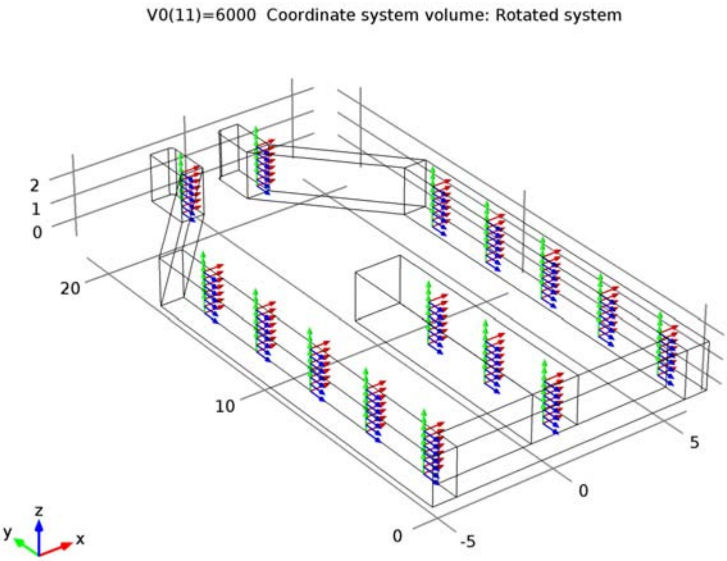


Figure 21-10: A 3D coordinate system volume plot. From the MEMS Module Model Library: Microgripper.

Far Field Plots

Note

The **Far Field** plots are available with the Acoustics Module or RF Module. For a **3D Plot Group** and **2D Plot Group**, select these plots from the **More Plots** submenu.

Default **Far Field** plots are automatically added to any model that uses far field calculations.

The **Far Field** plots are used to plot the value of a global variable.

- For the RF Module, the variables are the far field norm, `normEfar` and `normdBEfar`, or components of the far field variable `Efar`.
- For the Acoustics Module, the variables are the far field pressure `ffc1.Lp_pfar` and sound pressure level `ffc1.Lp_pfar`.

The variables are plotted for a selected number of angles on a unit circle (in 2D) or a unit sphere (in 3D). The angle interval and the number of angles can be manually specified. Also the circle origin and radius of the circle (2D) or sphere (3D) can be specified. For 3D **Far Field** plots you also specify an expression for the surface color.

The main advantage with the **Far Field** plot, as compared to making Line Graphs, is that the unit circle/sphere used for defining the plot directions is not part of the geometry for the solution. Thus, the number of plotting directions is decoupled from the discretization of the solution domain.

Add a **Far Field** plot to any plot group (Polar, 1D, 2D, and 3D).

- 1D or Polar plots ( ) for 2D, 2D axisymmetric, or 3D geometry.
- 2D plot ( ) for 2D axisymmetric or 3D geometry.
- 3D plot ( )for 2D axisymmetric or 3D geometry

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Inherit Style**, and **Coloring and Style**. For 1D and Polar plot groups, see the section also for **Legends**. For 3D plot groups, see the list for **Color**.

See below for sections specific to this plot: **Evaluation**.

1. Define one or more **Data Sets** and add a **Polar Plot Group** ( ), **1D Plot Group** ( ), **2D Plot Group** ( ) or **3D Plot Group** ( ).
2. Right-click the plot node and select **Far Field**. For **3D Plot Groups** and **2D Plot Groups**, the plot is available from the **More Plots** submenu.

EVALUATION

1D Plot Groups and Polar Plot Groups:

1 Under **Angles**, enter the φ **resolution**. The default is 50.

2 Select a **Restriction**—**None** (the default) or **Manual**. If **Manual** is selected, enter values (SI unit: deg) for φ **start** (the default is 0 degrees) and φ **range** (the default is 360 degrees).

3 Under **Center**, enter a value for **z-coordinate** (SI unit: mm). The default is 0.

4 Under **Normal**, enter values for **x**, **y**, and **z**. The defaults for *x* and *y* are 0, and the default for *z* is 1. Enter a **Radius** (SI unit: mm). The default is 1 mm.

2D Plot Groups and 3D Plot Groups:

1 Under **Angles**, enter the θ **resolution**. The default is 10. Enter the φ **resolution**. The default is 20.

2 Select a **Restriction**—**None** (the default) or **Manual**. If **None** is selected, you can also select the **Compute directivity** check box. If the **Compute directivity** check box is selected, the direction for the strongest radiation and the directivity value display in the **Results** window table.

Tip

Select **View>Results** to open the **Results** window.

3 If **Manual** is selected, enter values (SI unit: deg) for:

- θ **start** (the default is 0 degrees)
- φ **start** (the default is 0 degrees)
- θ **range** (the default is 180 degrees)
- φ **range** (the default is 360 degrees)

4 Under **Sphere** from the list, select **Unit sphere** (the default) or **Manual**. If **Manual** is selected, enter a value for **z-coordinate** (SI unit: mm). The default is 0. Enter a **Radius** (SI unit: mm). The default is 1.

Model

When you have the Acoustics Module:

- For a 3D example, see Bessel Panel: Model Library path **Acoustics_Module/Tutorial_Models/bessel_panel.**
- For a 2D axisymmetric example, see Cylindrical Subwoofer: Model Library path **Acoustics_Module/Tutorial_Models/cylindrical_subwoofer**.

When you have the RF Module:

- 2D model example with a Polar Plot Group—Radar Cross Section: Model Library path **RF_Module/Tutorial_Models/radar_cross_section**
- 2D axisymmetric model example with a Polar Plot Group and a 3D Plot Group—Conical Antenna: Model Library path **RF_Module/RF_and_Microwave_Engineering/conical_antenna**
- 3D model example with a Polar Plot Group and 3D Plot Group—Radome with Double-layered Dielectric Lens: Model Library path **RF_Module/RF_and_Microwave_Engineering/radome_antenna**

Global Plots

Use a **Global** ( ) plot to graph a scalar quantity as a function of time or a parameter. Right-click a **1D Plot Group** ( ) or **Polar Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**) **Data**, **Title**, **Coloring and Style**, and **Legends**.

See below for sections specific to this plot: **x-axis Data** or θ **Angle Data.**

Model

See Figure 21-11 for an example from the COMSOL Multiphysics Model Library. A contour data set can be used.

See Also

Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings

X-AXIS DATA (1D PLOTS) OR θ ANGLE DATA (POLAR PLOTS)

From the **Parameter** list, select an option for what the *x*-axis represents—**Solution number**, **Frequency spectrum**, **Phase**, or **Expression**. For **Parametric Sweep** studies, and when there are multiple inner solutions, **Parameter value** and **Time** are also available. Select:

- **Solution number** to use the solution numbers as the *x*-axis data (or θ angle data for polar plots) for an eigenvalue solution or for a parametric solution with more than one parameter.
- **Frequency spectrum** to have COMSOL Multiphysics compute the number of frequencies and the frequency range based on the FFT (fast Fourier transform) of the time-dependent solution.
  - To specify these values manually, select the **Number of frequencies** check box and enter a value in the associated field (the default is based on the number of time samples), or
  - Select the **Frequency range** check box and then enter the bounds of the frequency range in the **Minimum** and **Maximum** fields (in Hz). The FFT algorithm uses resampling based on linear interpolation. The *x*-axis shows the frequency (in Hz). By default, the *y*-axis shows the unscaled Fourier coefficients.
  - Select the **Scale** check box to scale the values on the *y*-axis so that their magnitude reflects the magnitude of the original signal. The values then have the same unit as the input data for the FFT. The *y*-axis title includes the unit if all expressions represented on the *y*-axis have the same unit. The scaling makes the magnitude at 0 Hz equal to the bias or DC component of the original signal. For a pure sinusoid the scaled value is the peak magnitude divided by the square root of 2 ($u_{\max}/\sqrt{2}$).

- **Phase** to specify a range of phase angles for the x-axis data. The default for the **Phase** is `range(0,0.5,2*pi)` (0–360 degrees in steps of 0.5 rad). Click the **Range** button () to define another set of phase angles. Select a **Unit** for the phase angle.
- **Expression** to specify an expression for the x-axis data (or θ angle data for polar plots). If **Expression** is selected, click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select a **Unit** and select the check box to enter a **Description** (or edit the default).
- **Parameter value** to use the x-axis data (or θ angle data for polar plots) stored in the solution for a parametric solution with a single parameter.
- **Time** to use time as the x-axis data (or θ angle data for polar plots) for a time-dependent solution.

Parametric Sweep Studies

Under **x-Axis Data** (θ **Angle Data** for polar plots), for **Parametric Sweep** studies, and when there are multiple levels in the parametric sweep, select an option from the **Axis source data** list to determine which parameter variations to use for the x-axis in the plot:

- The level of the nested parametric sweep (the default setting, using the innermost level), which can be **t** for time or the names of the parameters at the loop level.
- **All solutions** to use all solutions for all parameters, varying along the innermost level and up.
- **Inner solutions** to use the inner solution variation (the time, for example).
- **Outer solutions** to use the outer solution variation.

Tip

One example is a time-dependent problem with a geometric parametric sweep. The time steps are the inner solutions, the parameter sweep the outer solutions.

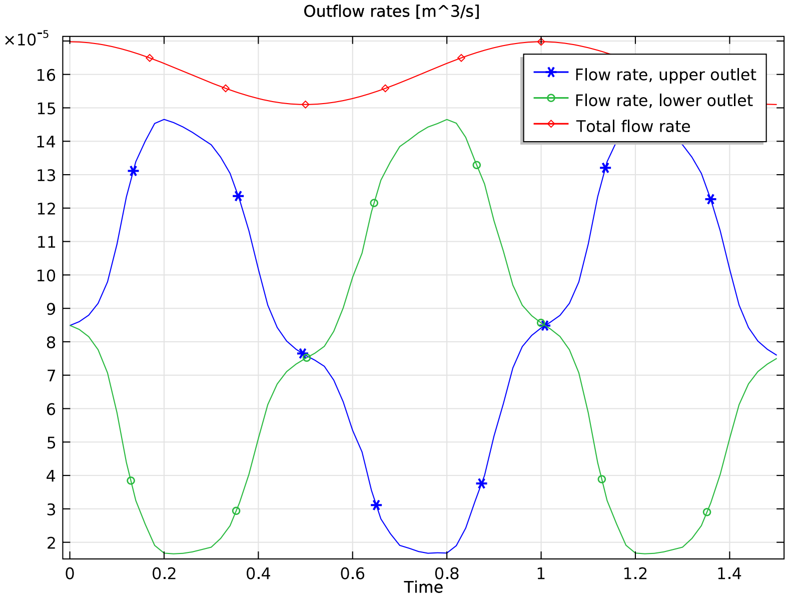


Figure 21-11: A 1D global plot. From the COMSOL Multiphysics Model Library: Fluid Valve.

Histogram

Use a **Histogram** in 1D ( ) or 2D ( ) to plot a histogram that shows how a quantity is distributed over the geometry (mesh volume). In 1D histograms, the *x*-axis in the histogram represents the values of the quantity (as a number of bins or a range of values), and the *y*-axis represents the count of the total element volume in each interval. You can also view the histogram as a plot showing the area in-between contours or isosurfaces. In 2D histograms the *x*-axis and *y*-axis represent the values of two quantities (as a number of bins or a range of values), and the color surface represents the count of the total element volume in each “bin.” The histogram can be normalized and also displayed as a cumulative plot, and it can appear as a discrete or a continuous function. Right-click a **1D Plot Group** ( ) or **2D Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Legends**, and **Quality**. For 2D Histogram nodes, there are two Expression sections: **x-Expression** and **y-Expression**.

See below for sections specific to this plot: **Bins**, **Output**, and **Evaluation**.

Model

See Figure 21-12 for an example from the COMSOL Multiphysics Model Library.

BINS

Select an **Entry method**—**Number of bins** or **Limits**—to define the bins for the histogram's x-axis. Select **Number of bins** (the default) to specify the number of bins (default is 10), or select **Limits** to specify a range of limits (`1 2 3 4`, for example) for the histogram bins. Click the **Range** button ( ) to define a range.

For 2D Histogram nodes, these settings are available for the x-direction and y-direction under **x bins** and **y bins**, respectively.

OUTPUT

Under **Output** specify some properties for the appearance of the histogram. Specify whether to use a continuous or discrete function for the histogram, the normalization, and whether to use a standard or a cumulative histogram.

From the **Function** list, select **Continuous** (the default) to plot the histogram as a continuous function or **Discrete** to plot it as a discrete function (that is, using a constant level in each bin).

From the **Normalization** list, select:

- **Integral** to normalize the histogram so that the integral is equal to 1.

- **None** (the default) to show the actual element volume without any normalization.
- **Peak** to normalize the histogram so that the peak values is equal to 1.

Select the **Cumulative** check box to make the histogram cumulative (that is, the value in each bin is the sum of the values for all bins up to the current one).

EVALUATION

Specify the **Space dimension** and the **Geometry level** for the evaluation. By default, the settings are taken from the data set. For a specific model, some space dimensions and geometry levels might not be applicable.

From the **Space dimension** list, select **Take from data set** (the default) or one of the space dimensions **0**, **1**, **2**, or **3**. The default is sufficient except when the data set is, for example, a cut plane, which can be evaluated for space dimensions 2 or 3.

From the **Geometry level** list, select **Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. Using another geometry level than the data set can be useful, for example, for evaluating over the surfaces of a 3D geometry. For solution data sets, **Take from data set** defaults to the highest dimension where there are any mesh elements.

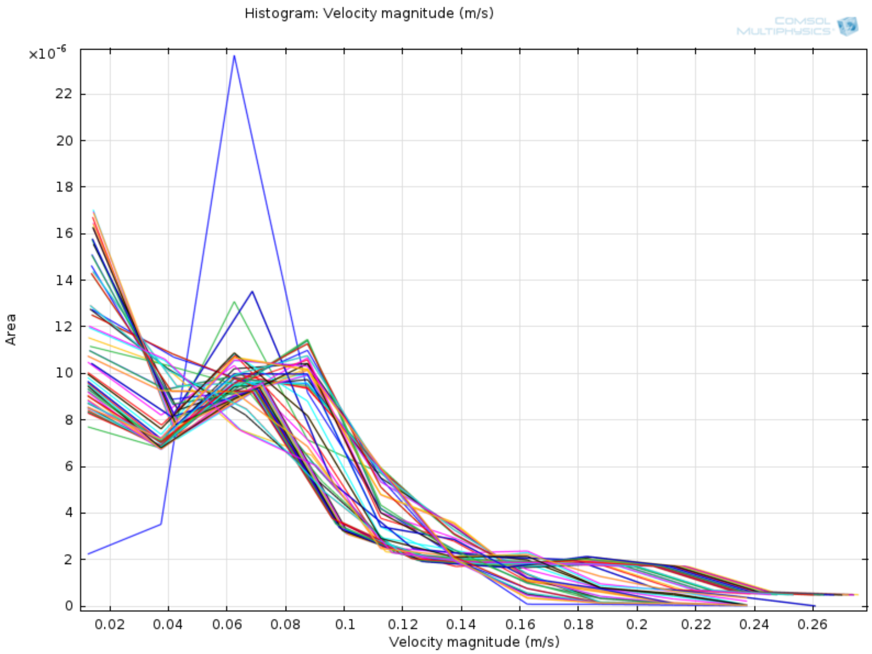


Figure 21-12: A 1D histogram plot. From the COMSOL Multiphysics Model Library: Fluid Valve.

Isosurface Plots

Plot a scalar quantity as an **Isosurface** ( ) plot in 3D. An isosurface plot displays a quantity as a colored set of isosurfaces on which the result has a constant value. The plot can also color isosurfaces based on an independent quantity. You can move the isosurfaces interactively. Add **Color Expression**, **Deformation**, or **Filter** subnodes as required.

Right-click a **3D Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality**, and **Inherit Style**.

See below for sections specific to this plot: **Levels**.

Model

See Figure 21-13 for an example from the Acoustics Module Model Library.

LEVELS

Select an **Entry method** for the isosurfaces—**Number of Levels** or **Levels**.

- If **Number of Levels** is selected, enter the total number of levels in the **Total levels** field (the default is 5).
- If **Levels** is selected, enter the values of the isosurface **Levels** or click the **Range** button ( ) to define specific range of levels.

Note

Before plotting, select the **Interactive** check box to move the isosurfaces defined in this **Isosurface** node interactively using the slider or by entering a shift in the **Shift** field. A zero shift represents the original position of the isosurfaces.

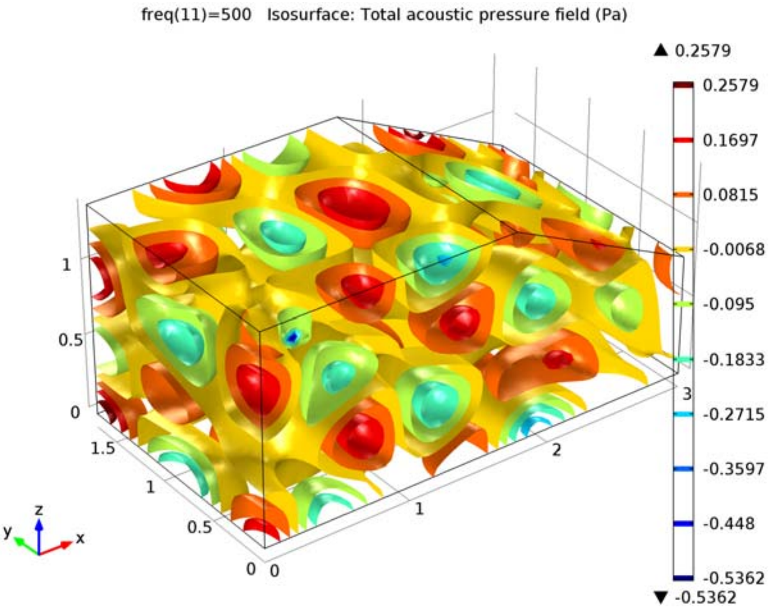


Figure 21-13: A 3D isosurface plot. From the Acoustics Module Model Library: Car Interior.

Line Graphs

Use a **Line Graph** ( ) to plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line. Make a graph plot of a quantity versus another quantity (for example, time).

Right-click a **1D Plot Group** ( ) or **Polar Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**), **Title**, **Coloring and Style**, **Legends**, and **Quality**.

See below for sections specific to this plot: **Selection** and **x-Axis Data** or θ **Angle Data.**

Model

See Figure 21-14 and Figure 21-15 for examples from the COMSOL Multiphysics Model Library and the RF Module Model Library. A parameterized curve data set can be used.

SELECTION (SOLUTION DATA SETS ONLY)

When **Solution** is selected as a **Data Set**, this section displays. Select **Manual** from the **Selection** list to choose geometry directly from the **Graphics** window. Select **All** to add the applicable geometry or any other predefined grouping.

X-AXIS DATA OR θ ANGLE DATA

Note

Line graphs plot one line for each solution. For example, in a parametric sweep, the plot contains one line for each parameter combination.

Select **Arc length** from the **Parameter** list to visualize along an arc length, or select **Expression** to visualize along, for example, a coordinate expression.

If **Expression** is selected, enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

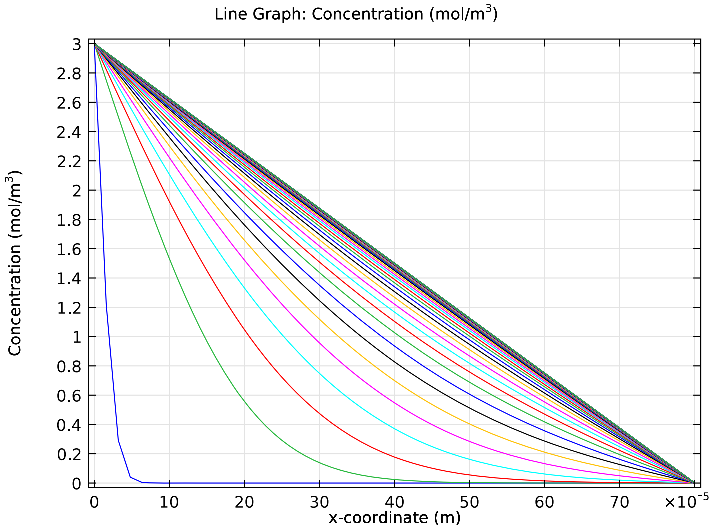


Figure 21-14: A 1D line graph. From the COMSOL Multiphysics Model Library: Effective Diffusivity in Porous Materials.

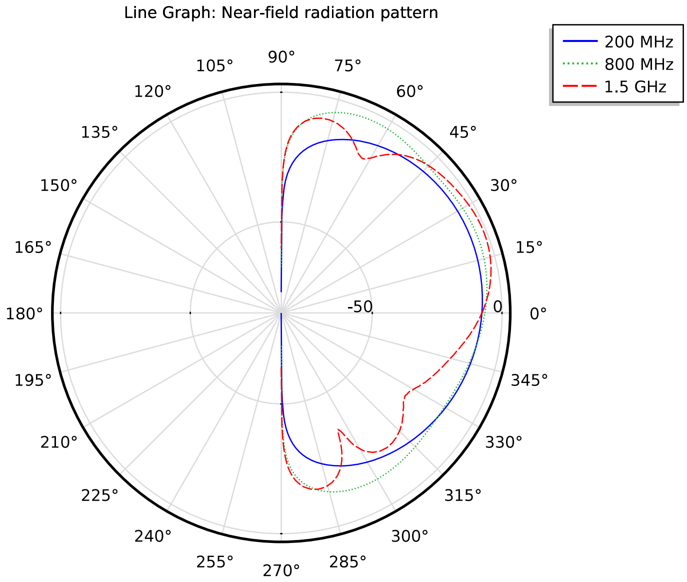


Figure 21-15: A Polar Plot Group line graph. From the RF Module Model Library: Conical Antenna.

Line Plots

Use a **Line** plot to display a quantity on lines—that is, boundaries in 2D ( ) or edges in 3D ( ). Add **Deformation**, **Filter**, or **Height Expression** (2D only) subnodes as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Model

See Figure 21-16, and Figure 21-17 for examples from the Pipe Flow Module and Structural Mechanics Module Model Libraries.

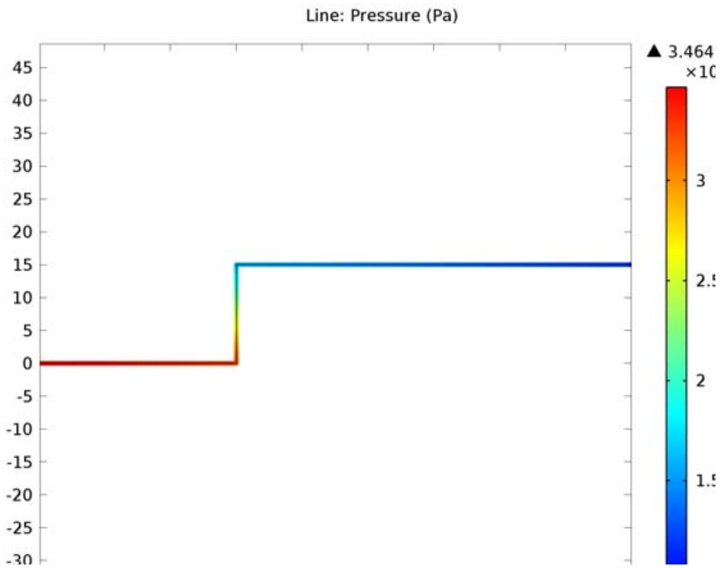


Figure 21-16: A 2D line plot. From the Pipe Flow Module Model Library: Discharging Tank.

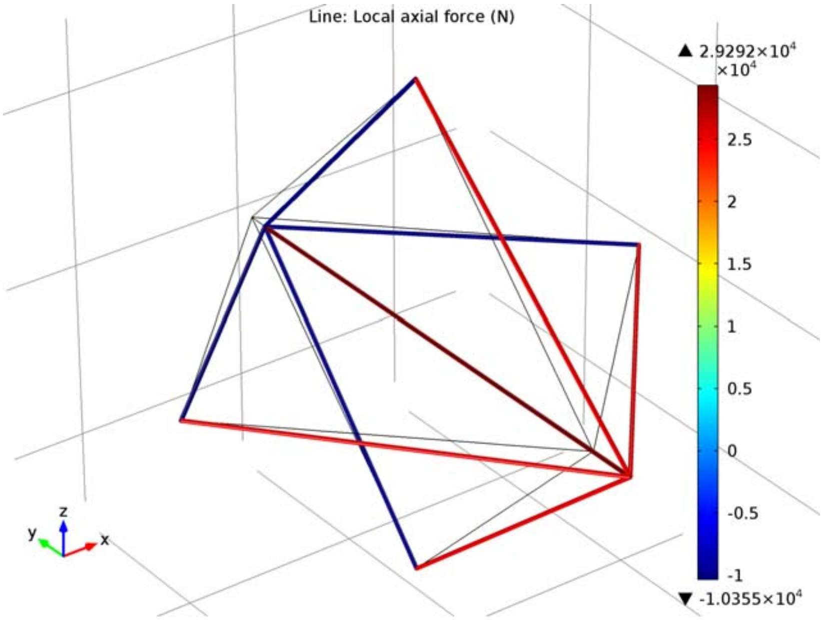


Figure 21-17: A 3D line plot. From the Structural Mechanics Module Model Library: In-Plane and Space Truss.

Max/Min Volume, Max/Min Surface, and Max/Min Line Plots

Use the **Max/Min Volume** ( ), **Max/Min Surface 2D** ( ), **Max/Min Surface 3D** ( ), **Max/Min Line 2D** ( ), and **Max/Min Line 3D** ( ) plots to plot the maximum and minimum values of an expression and the points there they are attained within the geometry.

When plotting the maximum and minimum value, an associated table appears in a **Results** window (underneath the plot with the default COMSOL Desktop layout). The table contains the maximum and minimum values along with the coordinates for the corresponding locations. The coordinate columns' titles contain the space variable names from the data set, if you use a **Cut Plane** data set, for example. Add a **Deformation** subnode as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

See below for sections specific to this plot: **Advanced**.

Model

See Figure 21-18 and Figure 21-19 for examples from the Structural Mechanics Module and MEMS Module Model Libraries.

ADVANCED

Under **Advanced,** define the number of refinements of each mesh element when computing the maximum and minimum by entering a value in the **Element refinement** field (the default is 2). Edit these other settings if required:

- Enter a **Display Precision** for the number of decimals displayed in the labels. The default is 6.
- The **Recover** default is **Off** because recovery takes processing time. To edit the default and use polynomial-preserving recovery and recover fields with derivatives such as stresses or fluxes with a higher theoretical convergence than smoothing, from the **Recover** list, select **Within domains**: to perform recovery inside domains or **Everywhere** to apply recovery to all domain boundaries.
- Select an option from the **Display** list—**Min and max** (the default), **Min**, or **Max**.

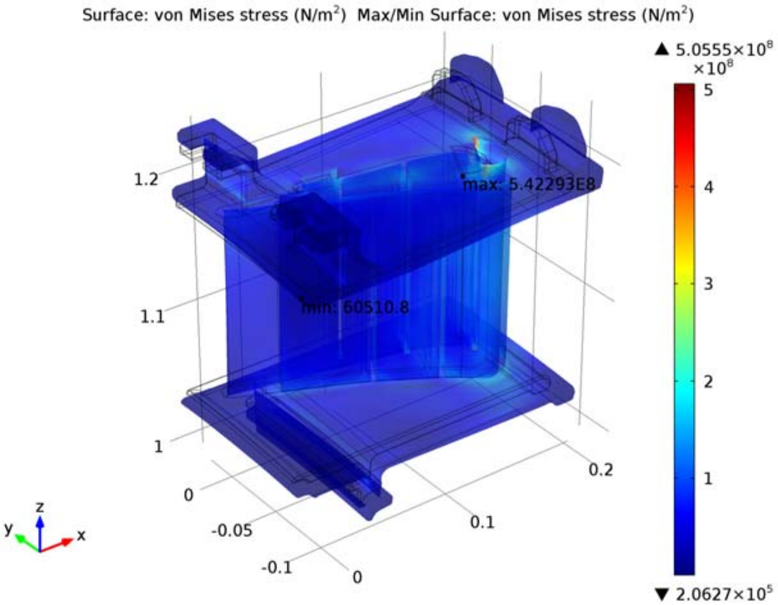


Figure 21-18: A 3D max/min surface plot combined with a surface plot. From the Structural Mechanics Module Model Library: Turbine Stator.

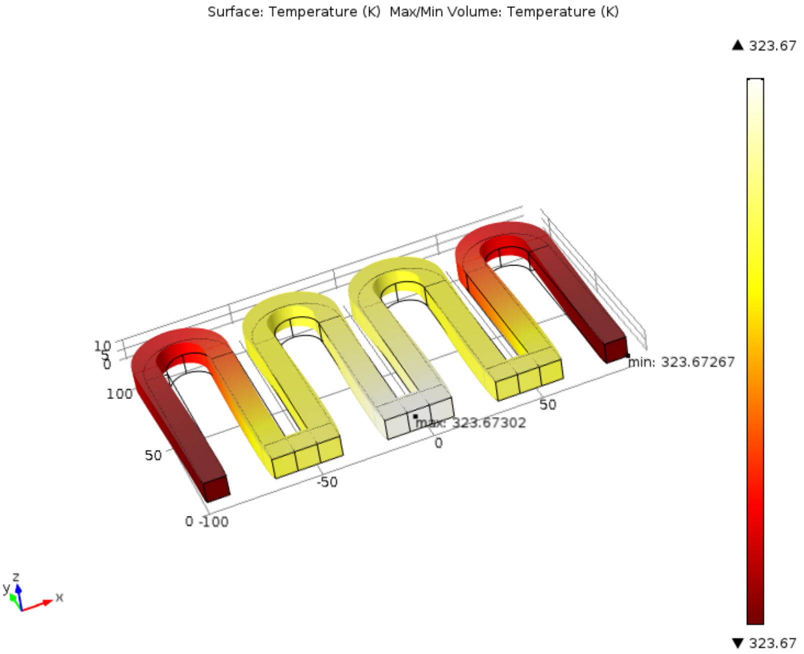


Figure 21-19: A 3D max/min volume plot combined with a surface plot. From the MEMS Module or Structural Mechanics Module Model Library: Thermal Expansion.

Mesh Plots

Use a **Mesh** plot ( ) to display a mesh in 2D or 3D. The plot can display the mesh quality or the mesh size. It is possible to plot the mesh without solving a model using a Mesh data set. Add **Deformation** or **Filter** subnodes as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add this plot type.

Tip

You can also create a Mesh plot by right-clicking the **Mesh** node and selecting **Plot** ( ).

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Element Filter**, and **Shrink Elements**.

See below for sections specific to this plot: **Level** and **Color**.

LEVEL

Select a **Level to** display the mesh—**All**, **Volume** (3D only), **Surface**, **Line**, or **Point**.

If **Surface** is selected, select the base **Element type** to visualize—**All**, **Triangle**, or **Quad**.

3D

For 3D models and if **Volume** is selected, select the base **Element type** to visualize—**All**, **Tetrahedron**, **Prism**, or **Hex**.

COLOR

Under **Color**, select an **Element color**—any basic color, **Quality** (the default) to get an element quality plot; **Size** to get a plot of the local mesh size; **Custom** to select a different color; or **None** to plot with no color.

- Select a **Color table** for the element quality. If the default (**Rainbow**) is not suitable for the plot, try other options.
- Select a **Wireframe color**—any basic color, **Custom** to select a different color, or **None** to plot with no color.

Multislice Plots

Use a **Multislice Plot** (▦) to display a scalar quantity on slices in multiple directions inside a 3D domain. Add **Deformation** and **Filter** subnodes as required.

Right-click a **3D Plot Group** ( ) to add this plot type from the **More Plots** submenu.

Tip	Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D). Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**. See below for sections specific to this plot: **Multiplane Data**.

Model	See Figure 21-20 for an example from the AC/DC Module Model Library.

MULTIPLANE DATA

Under **Multiplane Data**, select an **Entry method**—**Number of planes** or **Coordinates**— for the **x-planes**, **y-planes**, and **z-planes**.

For **Number of planes**, enter the number of planes in the **Planes** field. For **Coordinates**, enter a range of coordinates in the **Coordinates** field or click the **Range** button ( ) to define a range of values.

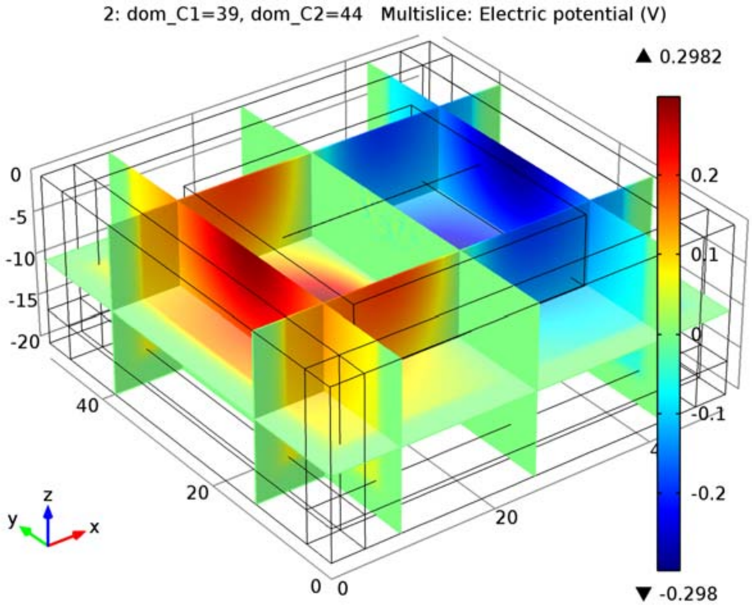


Figure 21-20: A 3D multislice plot. From the AC/DC Module Model Library: Geoelectrics.

Nyquist Plot

Use a **Nyquist Plot** ( ) to plot a Nyquist plot that shows the magnitude and phase of a frequency response. The plot shows the magnitude as the distance from the origin and the phase as the angle using a curve with the frequency as the parameter. Typical data to use for a Nyquist plot include complex-valued impedance data from a frequency domain study.

Right-click a **1D Plot Group** ( ) or **Polar Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Legends**.

Except where noted below, see Global Plots for all of the settings.

- For Global Plots, the **Expressions** section is called **y-Axis Data** (or **r-Axis Data** for polar plots), but the instructions are the same.
- Under **Coloring and Style**, select the **Show unit circle** check box to include a unit circle in the Nyquist plot.

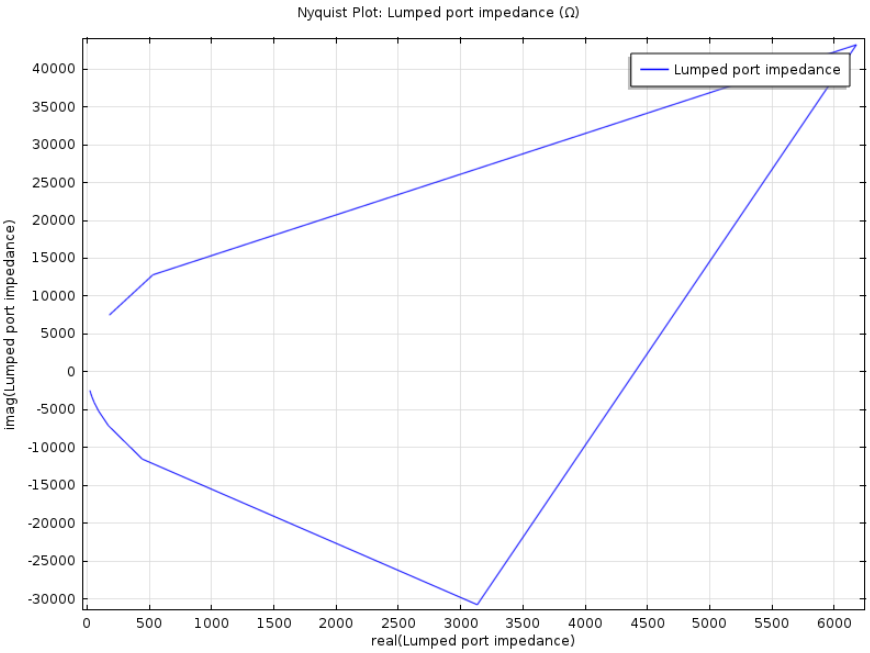


Figure 21-21: A 1D Nyquist plot. Created using the RF Module Model Library model, RF Coil.

Particle Tracing Plots

Note

This plot type is intended for visualizing a small number of particles on simple geometries. The Particle Tracing Module has vastly superior particle tracing capabilities and should be used for all but the simplest of models.

Use a **Particle Tracing** plot to visualize the trajectory of a massless particle subject to a flow field in 2D ( ) or 3D ( ). Visualize *pathlines* (that is, trajectories of particles released in a flow field), which can be time dependent or static. For time-dependent flows, also use a snapshot in time of the flow field as a static field. The motion of the particles does not affect the flow field. Add a **Color Expression** or **Deformation** subnode as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality** (Resolution and Recover only), and **Inherit Style**.

See below for sections specific to this plot: **Particle Positioning**, **Release**, **Quality** (ODE solver settings), and **Advanced**.

Note

There is an additional setting under **Coloring and Style** for this plot.

The **Type** of **Point Style** available includes **Comet tail**. Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity—so visually, it is the same as the tail of a comet traveling through space. Go to Common Plot Settings Cross References for the **Comet tail** settings links.

PARTICLE POSITIONING

3D

For 3D models, enter the initial position of particles in the **x**, **y**, and **z** fields or click the **Range** button ( ) to define a range.

2D

For 2D models, enter the **Positioning** details as described below.

Select a **Positioning**—**Start point controlled** or **Boundary coordinates**. Boundary coordinates are useful, for example, for flow models with one or more inflow boundaries.

- If **Start point controlled** is selected, enter the initial position of particles in the **x** and **y** fields or click the **Range** button ( ) to define a range.
- If **Boundary coordinates** is selected, select an item from the **Named selection** list and select an **Entry method**—**Number of points** or **Boundary parameters**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.
    - If **Number of points** is selected, enter the number of grid **Points** (the default is 10).
    - If **Boundary parameters** is selected, enter the **Relative coordinates** or click the **Range** button ( ) to define a range using the **Range** window.

RELEASE

Under **Release**, specify when to **Release particles**—**Once** (the default), **At intervals**, or **At times**. Select:

- **Once** to release particles once at the first available time, typically at time 0 (zero). To delay the release, select the **Start time** check box and enter a time.
- **At intervals** to release particles at regular intervals starting at the first available time, typically at time 0 (zero). To delay the release, select the **Start time** check box and enter a time. Enter a **Time between releases**. The default is 1.
- **At times** to release particles at an arbitrary time point; enter multiple **Times** to release particles or click the **Range** button ( ) to define a range using the **Range** window.

QUALITY (ODE SOLVER SETTINGS)

Under **Quality**, also define the **ODE solver settings** as required. Go to ODE Solver Settings—Relative Tolerance, ODE Solver Settings—Absolute Tolerance, and ODE Solver Settings—Step Size for details.

ADVANCED

Note

The **Advanced** section contains settings that do not normally need to be adjusted.

Under **Advanced**, also define these settings as required. Go to Advanced—Termination and Advanced—Instantaneous Flow Field for details.

In the **Termination** section, edit the **Maximum number of steps** and **Edge tolerance.** In the **Instantaneous flow field** section, edit the **Plot static flow field even when time dependent** check box, the **Time variable** default, and the **End time active**.

ODE Solver Settings—Relative Tolerance

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section.

Enter a **Relative tolerance** for the ODE solver. The default is 0.001.

- When solving the 2nd-order ODE $m\ddot{x} = F(t, x, \dot{x})$ for x, the solver first rewrites it as two coupled 1st-order ODEs: one for position x and one for velocity $\dot{x}$, each with two components in 2D and three components in 3D.
- The **Relative tolerance** value is the relative error tolerance that the ODE solver uses. It applies to all components of the particle's position and velocity. The solver controls the step size so that the estimated error e in each integration step satisfies

$$\begin{cases} e < \max(\text{atolpos}, \text{rtol} \cdot |x_i|) & (\text{for all components } x_i \text{ of } x) \\ e < \max(\text{atolvel}, \text{rtol} \cdot |\dot{x}_i|) & (\text{for all components } \dot{x}_i \text{ of } \dot{x}) \end{cases}$$

  where rtol is the relative tolerance specified, atolpos is the absolute tolerance for the particle's position components, and atolvel equals the absolute tolerance for the particle's velocity components.

ODE Solver Settings—Absolute Tolerance

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section. Specify the solver's absolute tolerance. The default is **Automatic**. To enter different values, select **Manual** from the **Absolute tolerance** list and enter a **Position**. The **Position** field can contain a single value—it applies to all components of the position and is the absolute tolerance.

ODE Solver Settings—Step Size

Follow these supplementary instructions for the **ODE solver settings** section found under the **Quality** section. Specify the solver **Step size.** The default is **Automatic**—COMSOL uses the initial value of the acceleration (force divided by mass) and the relative and absolute tolerances to determine the initial time step.

- The automatic maximum step size is 10% of the total simulation time for time-dependent flows as well as for static flow fields where the end time is manually specified in the **Advanced** section (in the **Plot static flow field even when time dependent>End Time** field). For static flow fields where the end time is not set manually, there is no upper limit of the step size. However, in this case, the initial time step is less than or equal to 0.1.

To edit the settings, select **Manual** from the **Step size** list and enter values in the **Initial time step** and **Maximum time step** fields.

- The **Maximum time step** is the longest time step the solver takes. It has higher priority than the **Initial time step**; that is, if an initial step size is set larger than the maximum step size, the solver lowers the initial step size to the maximum step size.

Note

The initial step size, whether entered manually or computed automatically, is not necessarily the first step the solver takes but is a first try. If this step leads to an error such that the tolerances are not met, COMSOL Multiphysics lowers it.

Advanced—Termination

Follow these supplementary instructions for the section found under the **Advanced** section. The **Termination** section contains settings that determine when to end the particle tracing simulation.

- To specify an upper limit of the number of time steps, click to select the **Maximum number of steps** check box and edit the default (1000). The particle simulation ends after this number of steps.
- To specify how close to the geometry boundary the path lines are cut when they exit the geometry, edit the **Edge tolerance** default (0.001). This is an absolute tolerance controlling how close to the geometry boundary the pathlines are cut when they exit the geometry. A lower value cuts the line closer to the geometry boundary.

Advanced—Instantaneous Flow Field

Follow these supplementary instructions for the section found under the **Advanced** section. To specify if you want to plot an instantaneous even if the solution is time dependent, select the **Plot static flow field even when time dependent** check box. This freezes the time selected previously— for example, from a **Plot Group** page in the **Data>Time** list—to the value specified and considers this a static flow field.

- Edit the **Time variable** default (`partt`) if required. Normally it is not necessary to change the default name but the name can be used in expressions as well as for the color when coloring the pathlines according to an expression.
- If required, select the **End time active** check box and enter a value.

Particle Tracing with Mass Plots

Note

This plot type is intended for visualizing a small number of particles on simple geometries. The Particle Tracing Module has superior particle tracing capabilities and should be used for all but the simplest of models.

Use a **Particle Tracing with Mass** plot in 2D ( ) or 3D ( ) to visualize the trajectory of a particle with mass and subject to a flow field. Add a **Color Expression** or **Deformation** as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Coloring and Style**, **Quality** (Resolution and Recover only), and **Inherit Style**.

See below for sections specific to this plot: **Equation of Motion** and **Mass and Velocity**.

See Particle Tracing Plots for **Particle Positioning**, **Release**, **Quality** (ODE solver settings), and **Advanced** settings.

Note

There is an additional setting under **Coloring and Style** for this plot.

The **Type** of **Point Style** available includes **Comet tail**. Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity—so visually, it is the same as the tail of a comet traveling though outer space. Go to Common Plot Settings Cross References for the **Comet tail** settings links.

Model

See Figure 21-22 for an example from the COMSOL Multiphysics Model Library.

BACKGROUND INFORMATION

Also, see Particle Tracing in Fluid Flow in the The Fluid Flow Branch section for more information about the predefined expressions for drag-driven particle movement that are available for particle tracing in fluid-flow models.

For particles with mass, COMSOL generates the pathlines by solving the fundamental equation of motion:

$$m\ddot{x} = F(t, x, \dot{x})$$

for the pathline $x(t)$. Here, m is the particle's mass, F equals the force acting upon the particle, and t is time. This is a system of ODEs for x, which COMSOL Multiphysics solves using a pair of Runge-Kutta methods of orders four and five. The solver advances the algorithm with the solution of order five and uses the difference between the order-five and order-four solutions to obtain the local error estimate.

For massless particles, the equation of motion is:

$$\dot{x} = v(t, x)$$

Note

The true formulation of Newton's second law of motion is $\frac{d}{dt}(m\dot{x}) = F(t, x, \dot{x})$. That is, the time derivative of the mass must be considered.

The particle-tracing algorithm does not solve this equation. Thus, an expression is specified for the particle mass that depends on time, the result are incorrect.

Axisymmetric Models

2D Axi

For 2D axisymmetric models, three components for the force are available for particles with mass.

When specifying all three, the algorithm solves for a line in 3D in cylindrical coordinates, but the plot only shows the projection on the axisymmetry plane. In this case, the centripetal force is considered; that is, the algorithm solves the equation

$$\ddot{r} = \frac{F_r}{m} + r\dot{\varphi}^2 \qquad \ddot{\varphi} = \frac{F_\varphi}{rm} - \frac{2\dot{r}\dot{\varphi}}{r} \qquad \ddot{z} = \frac{F_z}{m}$$

where m is the particle mass and (r, φ , z) are the cylindrical coordinates. The variable corresponding to the velocity component in the φ direction (the default name is `partv`) has the dimension length/time, and equals $r\dot{\varphi}$ as $\dot{\varphi}$ has the dimension radians/time.

SETTINGS ON THE PARTICLE TRACING WITH MASS PLOT

EQUATION OF MOTION

Specify the force acting on the particles. Click the **Replace Expression** ( ) or **Insert Expression** (✚) buttons to select predefined expressions based on the physics of the model. Or enter an **Expression**—for 2D enter or select **Fx** and **Fy** components of the force, for 3D enter or select **Fx**, **Fy**, and **Fz** components of the force. Enter a **Description** (or edit the default). When some predefined forces are added, there are additional **Parameters** with a **Value** to enter into a table.

MASS AND VELOCITY

Enter the particle **Mass**. Enter the **Initial velocity**—for 2D enter values for the **x component** and **y component**; for 3D enter values for **x component**, **y component**, and **z component**.

QUALITY (ODE SOLVER SETTINGS)

Under **Quality**, also define the **ODE solver settings** as required and described for Particle Tracing Plots. Go to ODE Solver Settings—Relative Tolerance, ODE Solver Settings—Absolute Tolerance, and ODE Solver Settings—Step Size for details.

ADVANCED

Note

The **Advanced** section contains settings that do not normally need to be adjusted.

Under **Advanced**, define the **Particle velocity variables**. Edit the default variable component names for each particle's velocity. The default names are `partu` (**x component**), `partv` (**y component**), and `partw` (**z component**).

Under **Advanced**, also define these settings as required and described for Particle Tracing Plots. Go to Advanced—Termination and Advanced—Instantaneous Flow Field for details.

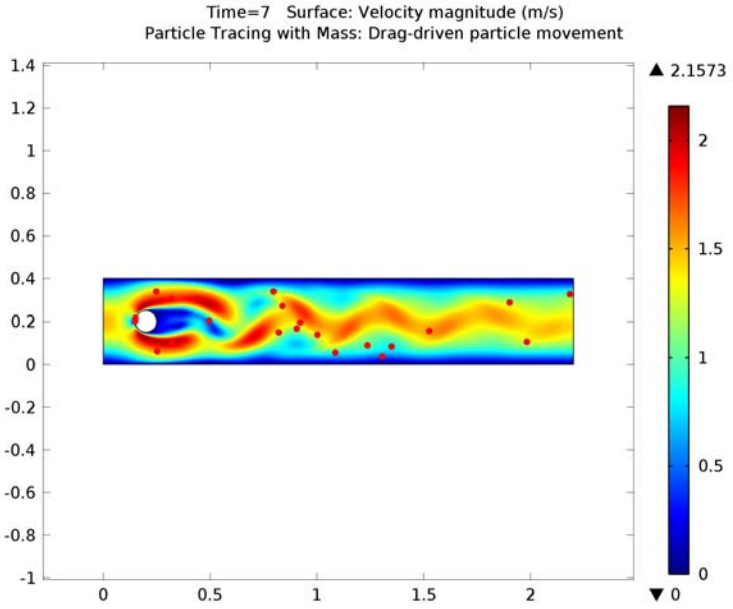


Figure 21-22: A 2D particle tracing with mass plot combined with a surface plot. From the COMSOL Multiphysics Model Library: Flow Past a Cylinder.

Particle Trajectories Plots

Note

This **Particle Trajectories** plot is available with the Particle Tracing Module. However, the plot does not compute the particle trajectories during results processing—the trajectories are computed by one of the physics interfaces in the Particle Tracing Module. The plot can thereby render tens of thousands of particles quickly because the trajectories have already been computed.

Use a **Particle Trajectories** ( ) plot to visualize the trajectory of a massless particle subject to a flow field in 2D or 3D. Add a **Color Expression**, **Deformation**, or **Filter** subnodes as required. For the settings in the **Filter** subnode, see Filter Node for Particle Trajectories.

Right-click a **2D Plot Group** (■) or **3D Plot Group** (■) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

Model

See Figure 21-23 for an example from the Particle Tracing Module Model Library.

Note

There are additional settings under **Coloring and Style** for this plot.

The **Fraction of particles to render** is a scalar value between 0 (no particle) and 1 (all particles; this is the default setting). You can use it to limit the number of particles used in plots of the particle trajectories for models with many particles.

The **Type** of **Point Style** available includes **Comet tail**. Comet tail plots provide a convenient way to indicate the direction of travel of particles at a given point in time. The tail of the comet typically points in the opposite direction to the particle velocity—so visually, it is the same as the tail of a comet traveling though outer space. Go to Common Plot Settings Cross References for the **Comet tail** settings links.

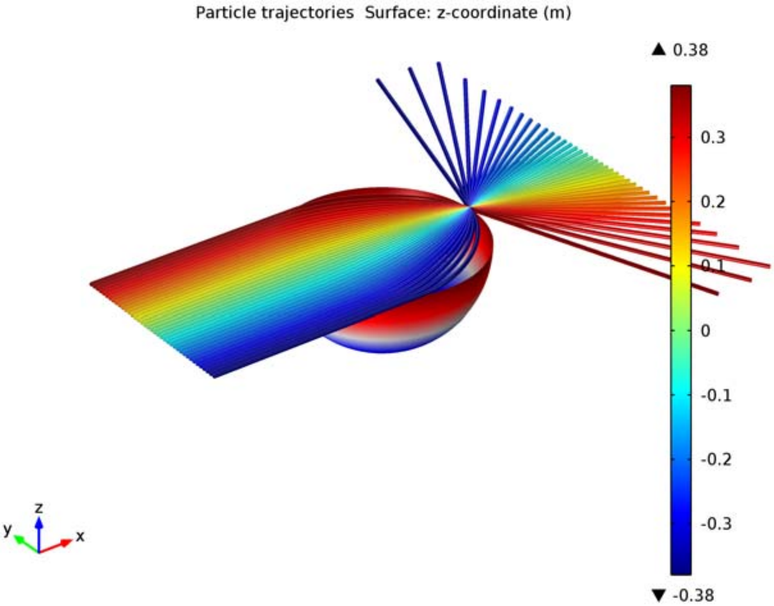


Figure 21-23: A 3D particle trajectories plot. From the Particle Tracing Module Model Library: Rossler Attractor.

Filter Node for Particle Trajectories

Right-click a **Particle Trajectories** plot node ( ) to add a **Filter** subnode ( ), which controls the particle types to include and whether to render all particles or a subset of the particles. The **Filter** node includes the following section:

PARTICLE SELECTION

From the **Particle type** list, select the particle types to include in the particle trajectories plot:

- **All** (the default) to include all particle types.
- **Primary** to include primary particles only.
- **Secondary** to include secondary particles only.

From the **Particles to render** list, select an option for controlling how many particles to render:

- **All** (the default) to render all particles in the particle tracing simulation.

- **Fraction** to only render a fraction of the particles. You specify the fraction as a number between 0 and 1 in the **Fraction of particles** field. The default fraction is 1; that is, to render all particles.
- **Number** to only render a ceratin number of particles, which you specify in the **Number of particles** field. The default is to render 100 particles.

Phase Portrait Plots

Note

This plot is available with the Particle Tracing Module.

Use a **Phase Portrait** plot ( ) for 2D and 2D axisymmetric models to visualize large data sets of particle trajectories. The traditional use of a phase portrait is to plot the particle position on the x-axis and the particle velocity on the y-axis. Each dot in the xy-plane represents a particle. By default, the position is taken as the distance from the origin (0, 0, 0) for 3D models. Add a **Color Expression** as required.

Right-click a **2D Plot Group** ( ) to add this plot type from the **More Plots** submenu.

2D

2D Axi

For 2D and 2D axisymmetric plots, the Phase Portrait view is shared with the model geometry. Therefore, it is necessary to clear the **Plot data set edges** check box found on the **2D Plot Group** or **3D Plot Group** page under **Plot Settings**. Then in the **Graphics** window click the **Zoom extents** button ( ) to see the phase portrait.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

See below for sections specific to this plot: **Expression**.

Model

See Figure 21-24 for an example from the Particle Tracing Module Model Library.

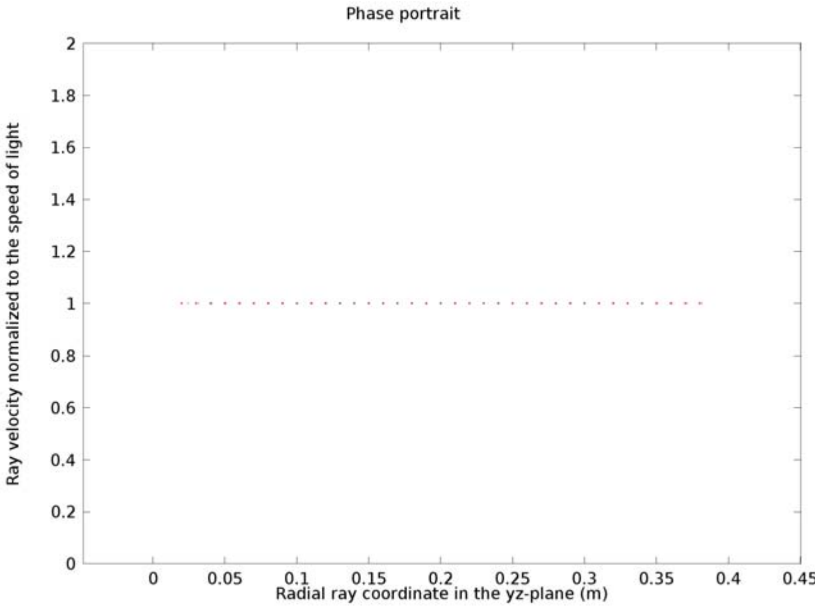


Figure 21-24: A 2D phase portrait plot. From the Particle Tracing Module Model Library: Ideal Cloak.

Expression

Select an option from the **x-axis** list—**Position** or **Manual**. If **Manual** is selected, enter an **Expression** (SI unit: m).

Select an option from the **y-axis** list—**Speed** or **Manual**. If **Manual** is selected, enter an **Expression** (SI unit: m/s).

Tip

The plot is best represented if the magnitude of the *x*-axis data and *y*-axis data are equal. Therefore, it can be useful to normalize the data by selecting **Manual** from the **x-axis** and **y-axis** lists under **Expression** and applying a suitable scaling factor.

Poincaré Map Plots

3D

This plot is available with the Particle Tracing Module and with 3D models only.

Use a **Poincaré Map** plot to visualize particle trajectories using a *Poincaré map* (sometimes called a *first recurrence map*). Add a **Color Expression** as required.

The Poincaré map is constructed by first defining a **Cut Plane** ( ) on the **Particle** data set ( ). Then add a **3D Plot Group** ( ) or a **2D Plot Group** ( ), depending on the dimension of the particle trajectories, and right-click the plot group node to add these plots from the **More Plots** submenu.

This plot type is useful to visualize the particle trajectories in a plot that represents the position of the particles in a section that is usually transversal to the particle trajectories. The Poincaré map represents the particle trajectories in a space dimension that is one dimension lower than the original particle space.

The Poincaré map parent plot group should point to this cut plane (select a **Cut plane** data set under **Data**). The resulting plot places a dot on the cut plane at the location where a particle crossed the plane. The same particle can cross the cut plane multiple times.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Inherit Style**, and **Coloring and Style**.

Model

See Figure 21-25 for an example from the Particle Tracing Module Model Library.

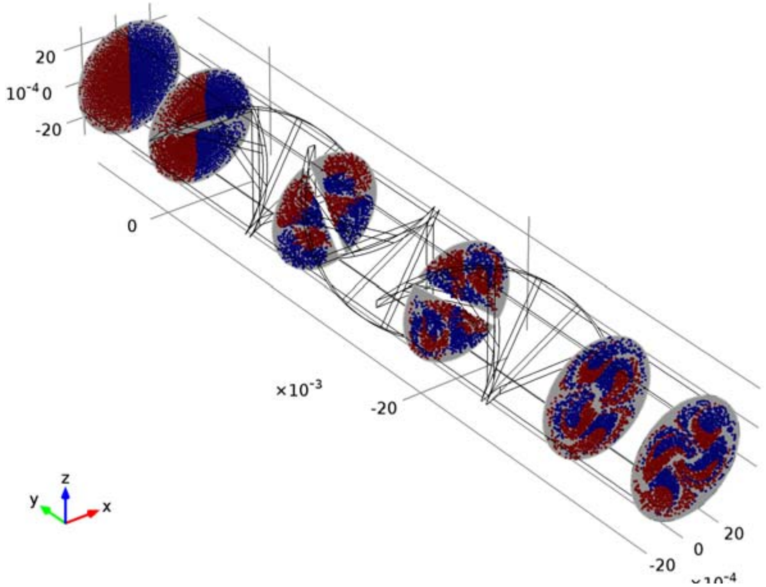


Figure 21-25: Poincaré maps of the particle trajectories at different Poincaré sections. The color is a logical expression indicating which particles had an initial position at x < 0. from the Particle Tracing Module Model Library: Particle Trajectories in a Laminar Static Mixer.

Point Graphs

Use a **Point Graph** ( ) to visualize the value in a point along time or a parameter value. It can be a point in the geometry or a cut point.

Right-click a **1D Plot Group** ( ) or **Polar Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Legends**, **y-Axis** (or **r-Axis**) **Data**, and **Coloring and Style**.

See below for sections specific to this plot: **Selection**.

Model

See Figure 21-26 and Figure 21-27 for examples from the COMSOL Multiphysics Model Library and the RF Module Model Library.

Note

See Global Plots for these settings: **x-Axis Data** or θ **Angle Data**.

SELECTION (SOLUTION DATA SETS ONLY)

Select **Manual** from the **Selection** list to choose geometry directly from the **Graphics** window. Select **All** to add the applicable geometry or any other predefined grouping.

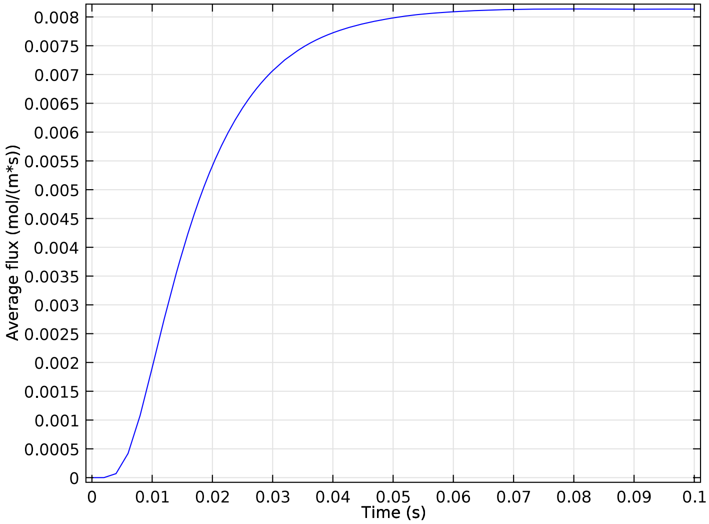


Figure 21-26: A 1D point graph. From the COMSOL Multiphysics Model Library: Effective Diffusivity in Porous Materials.

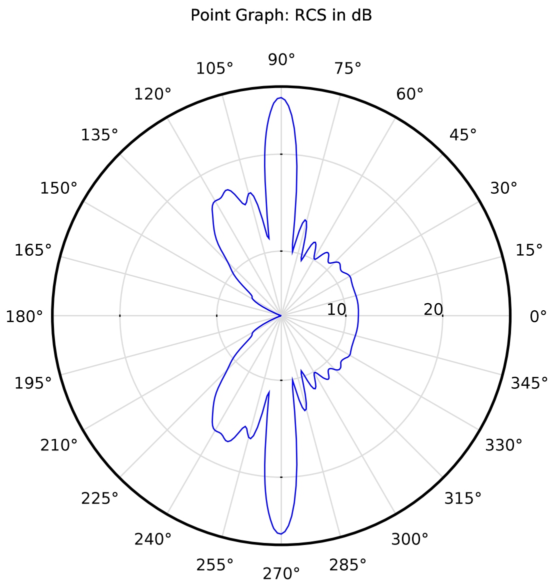


Figure 21-27: A Polar Plot Group point graph. From the RF Module Model Library: Radar Cross Section.

Principal Stress Volume Plots

Use the **Principal Stress Volume** ( ) to plot the principal stress and principal strain in structural mechanics models. The values of the principal stresses σ_1, σ_2, and σ_3 are the eigenvalues of the stress tensor, ordered such that $\sigma_1 > \sigma_2 > \sigma_3$. The same applies for the principal strains ε_1, ε_2, and ε_3. The plots also show the corresponding eigenvectors using arrows. Add a **Deformation**, **Filter**, or **Color Expression** as required.

Right-click a **3D Plot Group** ( ) to add this plot type from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Principal Components**, **Title**, **Positioning**, **Coloring and Style**, and **Inherit Style**.

Model

See Figure 21-28 for an example from the Structural Mechanics Module Model Library.

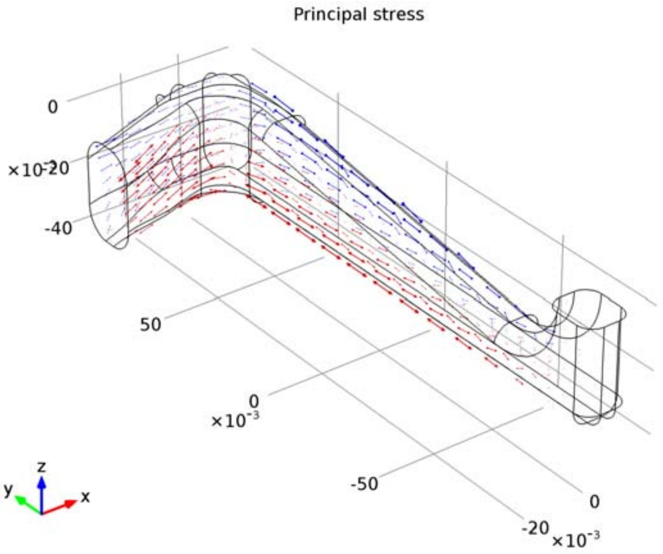


Figure 21-28: A 3D principal stress volume plot. From the Structural Mechanics Module Model Library: Static and Eigenfrequency Analyses of an Elbow Bracket.

Principal Stress Surface Plots

Use the **Principal Stress Surface** plots in 2D (✛) and 3D (✛) to plot the principal stress and principal strain in structural mechanics models. The values of the principal stresses σ_1, σ_2, and σ_3 are the eigenvalues of the stress tensor, ordered such that $\sigma_1 > \sigma_2 > \sigma_3$. The same applies for the principal strains ε_1, ε_2, and ε_3. The plots also shows the corresponding eigenvectors using arrows. Add a **Deformation**, **Filter**, or **Color Expression** as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add this plot type from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Title**, **Inherit Style**, **Principal Components**, **Positioning**, and **Coloring and Style**.

Scatter Surface and Scatter Volume Plots

Use scatter plots to visualize a scalar quantity as scattered spheres on a **2D Scatter Surface** ( ) or in a **3D Scatter Volume** ( ) (as functions of space coordinates or any quantities).

Scatter plots can be used as alternatives to arrow plots for scalar quantities or to represent the correlation between two or more different variables to get a feeling for how quantities correlate. To display a quantity using the color and radius of scattered spheres in the model geometry, use the space coordinates (x, y, and z in 3D) as the expressions for the scatter plot axes.

If you use some other quantities as the variables that determines the scattered spheres' positions on the axes, it is good practice to remove the plotting of the data set's edges (typically the geometry boundaries) by clearing the **Plot data set edges** check box in the main plot group node's settings window. In those cases, the axes in the **Graphics** window no longer represent the space coordinate for the geometry.

The radius and color can both be functions of independent quantities, so a 3D scatter plot can provide information about up to five different quantities as the three axis directions, color, and radius.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types from the **More Plots** submenu.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Radius**, **Color**, **Coloring and Style**, and **Inherit Style**.

See below for sections specific to this plot: **Evaluation Points**.

Model

See Figure 21-29 for an example from the COMSOL Multiphysics Model Library.

EVALUATION POINTS

Under **Evaluation Points**, select an **Entry method** for the **grid points** coordinates based on space dimension (**x grid points** and **y grid points** for 2D; **r grid points** and **z grid points** for 2D axial symmetry; or **x grid points**, **y grid points**, and **z grid points** for 3D).

The evaluation points are located in a block-shaped (3D) or rectangular (2D) grid where the axes represent the expressions defined in the **Expression** section.

- If **Number of points** is selected, enter the number of **Points** in each direction (the default is 15 for 2D Scatter Surface and 7 for 3D Scatter Volume).
- If **Coordinates** is selected, enter **Coordinates** (SI unit: m) or click the **Range** button ( ) to define a range of values.

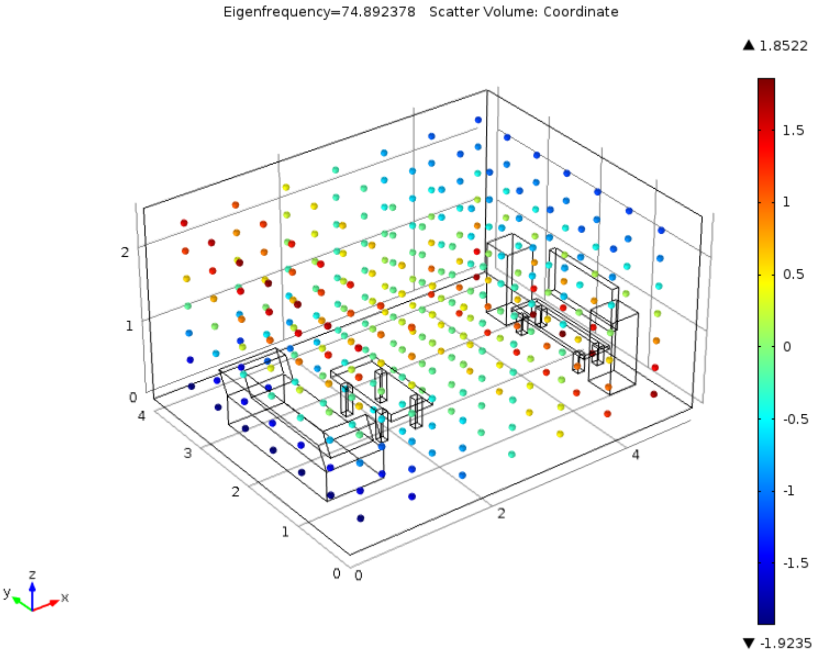


Figure 21-29: A 3D scatter volume plot. From the COMSOL Multiphysics Model Library: Eigenmodes of a Room.

Slice Plots

Use a **Slice Plot** ( ) to display a scalar quantity on slices inside a 3D domain. Add **Deformation** and **Filter** subnodes as required.

Right-click a **3D Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

See below for sections specific to this plot: **Plane Data**.

Model

See Figure 21-30 for an example from the COMSOL Multiphysics Model Library.

PLANE DATA

Under **Plane Data**, select a **Plane Type**—**Quick** (the default) to allow you to specify planes orthogonal to the coordinate axes; or **General** to specify general planes.

If **Quick** is selected:

- From the **Plane** list, select **xy-planes**, **yz-planes**, or **zx-planes** as the set of planes orthogonal to the coordinate axes applicable for the model geometry.
- Select an **Entry method**—**Number of planes** or **Coordinates**.
  - If **Number of planes** is selected, enter **Planes**.
  - If **Coordinates** is selected, enter the applicable (*x*, *y*, or *z*) grid **Coordinates** or click the **Range** button (|⋯|) to define a range of values. Choose a set of cut plane slices to a coordinate axis, specify the transverse coordinate by entering the location along the transverse coordinate axis in the **Coordinates** field.

If **General** is selected:

- Select an option from the **Plane entry method** list—**Three points** or **Point and normal**.
  - If **Three points** is selected, enter **x**, **y**, or **z** coordinates in the **Point 1**, **Point 2**, and **Point 3** fields.
  - If **Point and normal** is selected, enter **x**, **y**, or **z** coordinates in both the **Point** and **Normal** sections.
- If required, select the **Additional parallel planes** check box and select an **Entry method**— **Number of planes** or **Distances**.
  - If **Number of planes** is selected, enter the number of grid **Planes** (the default is 4).
  - If **Distances** is selected, enter the **Distances** (SI unit: m) or click the **Range** button (|⋯|) to define a range of values.

To move the slices interactively, select the **Interactive** check box before plotting. You can then move the slices using the slider or by typing a shift in the **Shift** field. A zero shift represents the original position of the slices.

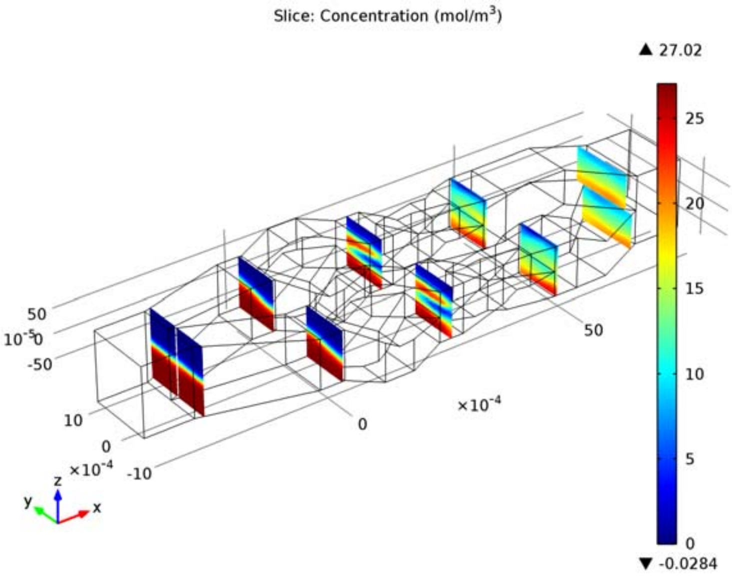


Figure 21-30: A 3D slice plot. From the COMSOL Multiphysics Model Library: Micromixer.

Streamline Plots

Use a **Streamline** plot in 2D ( ) or 3D ( ) to visualize a vector quantity. A streamline is a curve everywhere tangent to an instantaneous vector field. 3D streamline plot is analogous to the 2D streamline plot except that there is no height data setting and the start point selection is different. Add **Deformation** or **Color Expression** subnodes as required.

Right-click a **2D Plot Group** ( ) or **3D Plot Group** ( ) to add these plot types.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality**, and **Inherit Style**.

See below for sections specific to this plot: **Streamline Positioning**, **Selection**, and **Advanced**.

Model

See Figure 21-31 and Figure 21-32 for examples from the AC/DC Module Model Library and the COMSOL Multiphysics Model Library.

Under **Streamline Positioning**, select one of these options from the **Positioning** list—**On selected boundaries** (the default), **Start point controlled**, **Uniform density**, or **Magnitude controlled**. Then follow one of the methods described in Streamline Positioning Section (Continued):

- Method 1: Specifying the Number of Streamlines and Start Boundaries
- Method 2: Specifying Points by Entering Coordinates
- Method 3: Selecting the Specified Number of Start Points in the Geometry
- Method 4: Creating Streamlines with Uniform Density
- Method 5: Creating Streamlines with Variable Density and Magnitude Controlled

SELECTION

Note

The **Selection** section is made available for some data sets when **On selected boundaries** is selected from the **Positioning** list under **Streamline Positioning**.

Select the boundaries from which the streamlines start. By selecting in the **Graphics** window and using the tools in the **Selection** section, select the boundaries for the starting positions for the streamlines.

Define the Advanced Settings for the Streamline Plot as required.

Advanced Settings for the Streamline Plot

Under **Advanced** set these general settings. See also Advanced Section Setting Effects.

- The **Integration tolerance** field default is 0.01 for 3D and 0.001 for 2D. Edit to specify how accurately streamlines are computed.
- The **Maximum number of integration steps** field makes sure that the integration does not continue indefinitely. Edit the default (5000) to control when the computation stops.
- The **Maximum integration time** field sets an upper time limit for the integration. The default is infinity (inf).

- The **Stationary point stop tolerance** can be adjusted to make sure the integration stops near a stationary point in the field. The default is 0.01.
- The **Loop tolerance** field default is 0.01. This is a fraction of the mean of the lengths of the bounding box of the geometry. If a streamline gets closer to its start point than this distance, the streamline snaps to its start point and is plotted as a connected loop. See also Method 5: Creating Streamlines with Variable Density and Magnitude Controlled.
- Select the **Allow backward time integration check box** to integrate points from the starting points both in the direction of the vector field and in the opposite direction.
- Select the **Normalize vector field check box** if required.

STREAMLINE POSITIONING SECTION (CONTINUED)

Method 1: Specifying the Number of Streamlines and Start Boundaries

1 Under **Streamline Positioning**, from the **Positioning** list, select **On selected boundaries**.

> Note
>
> The **Selection** section is made available for some data sets when **On selected boundaries** is selected from the **Positioning** list under **Streamline Positioning**.

2 Under **Selection**, select the boundaries from which the streamlines start. By selecting in the **Graphics** window and using the tools in the **Selection** section, select the boundaries for the starting positions for the streamlines.

3 Enter the **Number** of streamlines (the default is 20).

Method 2: Specifying Points by Entering Coordinates

1 Under **Streamline Positioning**, from the **Positioning** list, select **Start point controlled**.

2 Select **Coordinates** from the **Entry method** list.

3 Enter **x** and **y** (2D), **x**, **y**, and **z** (3D) coordinates (SI unit: m) or click the **Range** button ( ) to define a range of values. Also use a scalar value to represent a fixed value for some of the coordinates.

Method 3: Selecting the Specified Number of Start Points in the Geometry

1 Under **Streamline Positioning**, from the **Positioning** list, select **Start point controlled**.

2 Select **Number of points** from the **Entry method** list.

3 Enter the number of **Points** (the default is 20).

4 From the **Along line or plane list**, select **None**.

Method 4: Creating Streamlines with Uniform Density

The algorithm saturates the entire domain with evenly spaced streamlines.

1 Under **Streamline Positioning**, from the **Positioning** list, select **Uniform density**.

2 Enter the **Separating distance** between the streamlines (the default is 0.05).

The value for the separating distance is a fraction of the mean of the lengths of the bounding box of the geometry. In this case, a streamline stops whenever it gets too close to another streamline or itself (or if any of the general termination criteria specified in the **Advanced** section is fulfilled).

3 The **Advanced parameters** list defaults to **Automatic**. If required, select **Manual** to edit these parameters—**Boundary element refinement**, **Fraction of streamline length to ignore**, **Starting distance factor**, **Terminating distance factor**, or **First start point**.

- Edit the **Boundary element refinement** if streamlines do not behave as expected near boundaries on a coarse mesh—try increasing this number. It is a measurement of the density of points on the boundaries used to set up the structure and is used to measure distances between streamlines. Refining the mesh in the problematic area can also resolve the problem.
- Edit the **Fraction of streamline length to ignore** when a streamline is close to itself, typically for spiraling streamlines. This number controls how big part of the streamline, starting from its start point, that the streamline itself is allowed to get close to.
- The **Starting distance factor** is a factor multiplied with the distance specified in the **Separating distance** field (as a fraction of the mean of the lengths of the bounding box of the geometry—the default value is 0.05). It sets the minimum distance between streamlines and the start point for the next streamline.

  When the domain is close to be saturated with streamlines, new start points tend to be positioned where the streamline has nowhere to go before it gets too close other streamlines, resulting in short streamlines. The higher the value of this factor, the more it disqualifies the start point and thus reduces the number of short streamlines.
- The **Terminating distance factor** is a factor multiplied with the distance specified in the **Separating distance** field. It sets the minimum distance between any pair of streamlines. Thus, this distance is the minimal distance under which the integration of a streamline stops.
- By default the **First start point** list defaults to **Automatic**, and it sets the start point for the first streamline. It is chosen in the element where the highest value of the

velocity of the specified vector field occurs. If required, select **Manual** instead to override the default and enter **x** and **y** coordinates.

Method 5: Creating Streamlines with Variable Density and Magnitude Controlled

To create streamlines with a variable density according to the magnitude of the specified vector field.

1 Under **Streamline Positioning**, from the **Positioning** list, select **Magnitude controlled**.

   The **Magnitude controlled** setting gives proper streamline plots only for incompressible flow fields. In this case, the algorithm places the streamlines so that the flow between each pair of adjacent streamlines is the same throughout the domain, giving streamlines that are more dense where the magnitude of the field is high.

2 This step depends if it is a 2D or 3D model.

2D	For 2D models, enter a **Density** (the default is 20). This value is roughly the number of streamlines. Prior to streamline generation, the software computes a rough estimate of the total flow of the flow field in the model, divides this value with the specified **Density** setting, and uses the resulting value as the flow between each pair of adjacent streamlines.

3D	For 3D models, enter the **Min** (Minimum) **distance** and **Max** (Maximum) **distance** between streamlines (the default **Min distance** is 0.05 and the default **Max distance** is 0.15). These distances are specified as fractions of the mean of the lengths of the bounding box of the geometry. The minimum velocity in the model is mapped to the minimum distance and the maximum velocity to the maximum distance. Thus every point on a streamline and on the boundary has a separating distance associated with it. Given a set of streamlines, the start point for the next streamline is selected using these separating distances.

A streamline stops only if it exits the domain or gets too close to its own start point, using the **Loop tolerance** option in the **Advanced** section (or if any of the general termination criteria specified in the **Advanced** section is fulfilled).

3 If required, from the **Advanced parameters** list, select **Manual** to set advanced parameters as described in Method 4: Creating Streamlines with Uniform Density.

ADVANCED SECTION SETTING EFFECTS

The **Advanced** settings have the following effects:

- When calculating streamlines, the software selects a set of starting points (controlled by the streamline start points and the number of start points).
- The algorithm then finds the vectors of the given vector field at these points by interpolation. It normalizes the vector field if that option is selected.
- The algorithm integrates the points along the direction of the vector using the integration tolerance using a second-order Runge-Kutta algorithm.
- At the new positions, the algorithm finds vector values by interpolation and performs another integration.

This process stops if:

- It reaches a predetermined number of integration steps (controlled by the maximum number of integration steps entry).
- The points end up outside the geometry.
- The points reach a “stationary point” where the vector field is zero. Control the meaning of “zero” with the stationary point stop tolerance.
- It has used a predetermined amount of “time” for integrating (control this parameter with the **Maximum integration time** field).

Finally, the software connects the calculated points for each streamline consecutively with straight lines.

Tip	When integrating, the software uses a pseudo-time that has nothing to do with the time in time-dependent problems. Use the massless particle tracing tool to integrate in time-varying fields and to control the real time in stationary fields.

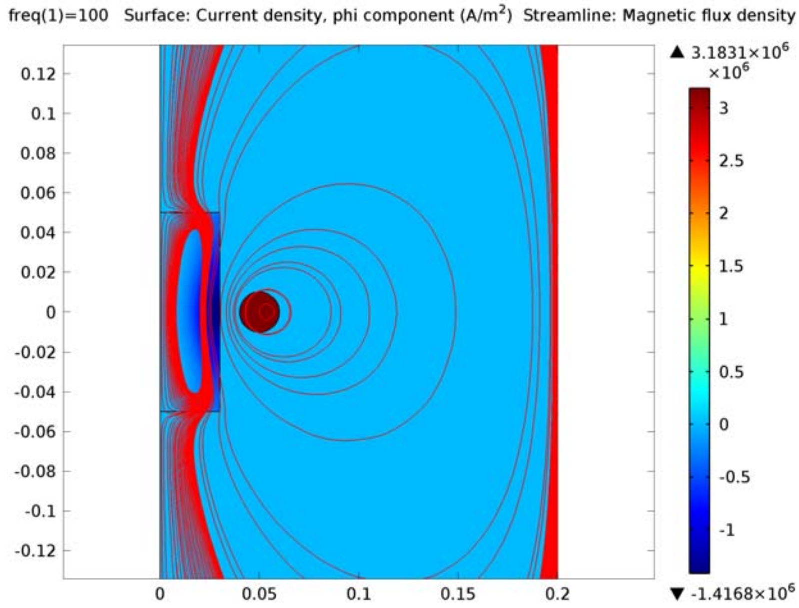


Figure 21-31: A 2D streamline plot combined with a surface plot. From the AC/DC Module Model Library: Eddy Currents in a Cylinder.

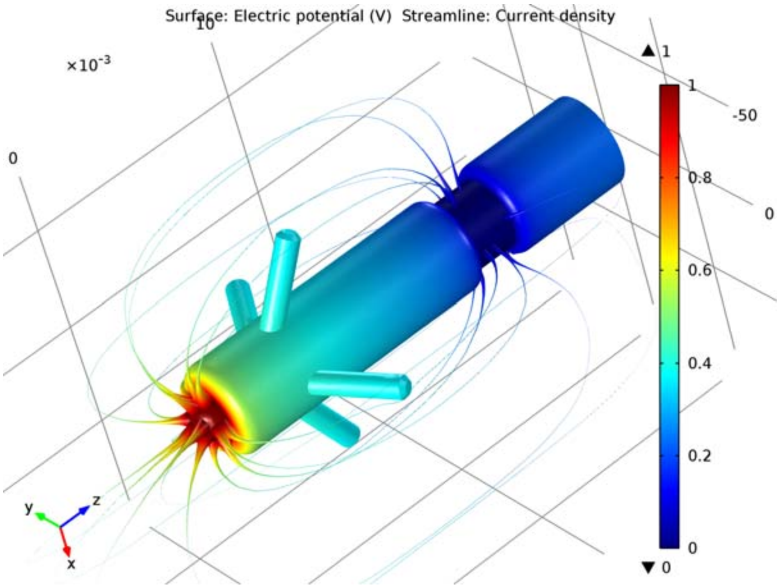


Figure 21-32: A 3D streamline plot combined with a surface plot. From the COMSOL Multiphysics Model Library: Pacemaker Electrode.

Surface Plots

Use a **Surface** plot to display a quantity on a domain in 2D (🟪) or on a boundary in 3D (▣). Add **Deformation**, **Filter**, or **Height Expression** (2D only) subnodes as required.

Right-click a **2D Plot Group** (🟪) or **3D Plot Group** (▣) to add these plot types.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Model

See Figure 21-33 and Figure 21-34 for examples from the COMSOL Multiphysics Model Library.

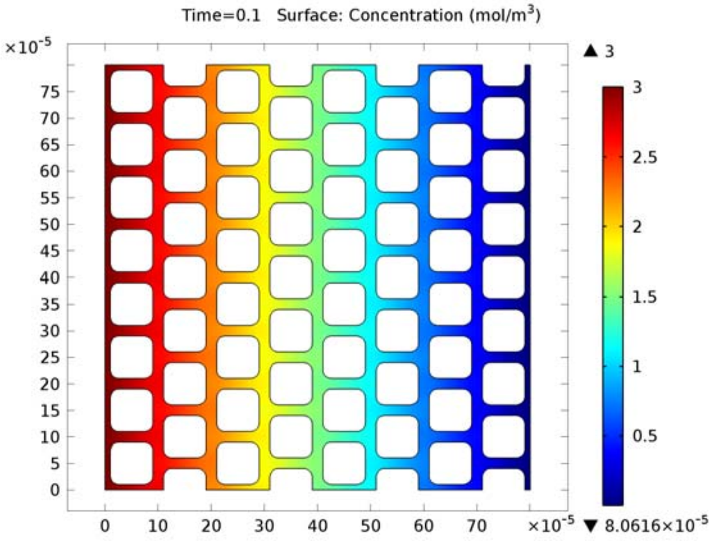


Figure 21-33: A 2D surface plot. From the COMSOL Multiphysics Model Library: Effective Diffusivity in Porous Materials.

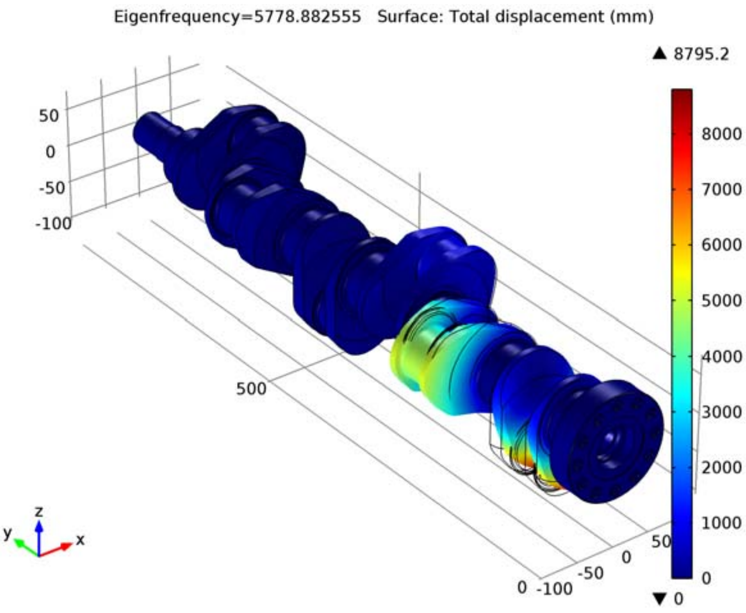


Figure 21-34: A 3D surface plot. From the COMSOL Multiphysics Model Library: Eigenvalue Analysis of a Crankshaft.

Table Graph

Add a **Table Graph** (▦) plot to display data from a table with one line per output column. Right-click a **1D Plot Group** (📈) or **Polar Plot Group** (◎) to add this plot type.

Tip	Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D). Also go to Common Plot Settings Cross References for links to information about these sections: **Legends** and **Coloring and Style**. See below for sections specific to this plot: **Data.**

Define a table to plot. See Adding and Evaluating Tables for a Derived Value.

Model

See Figure 21-33 for an example from the Structural Mechanics Module Model Library.

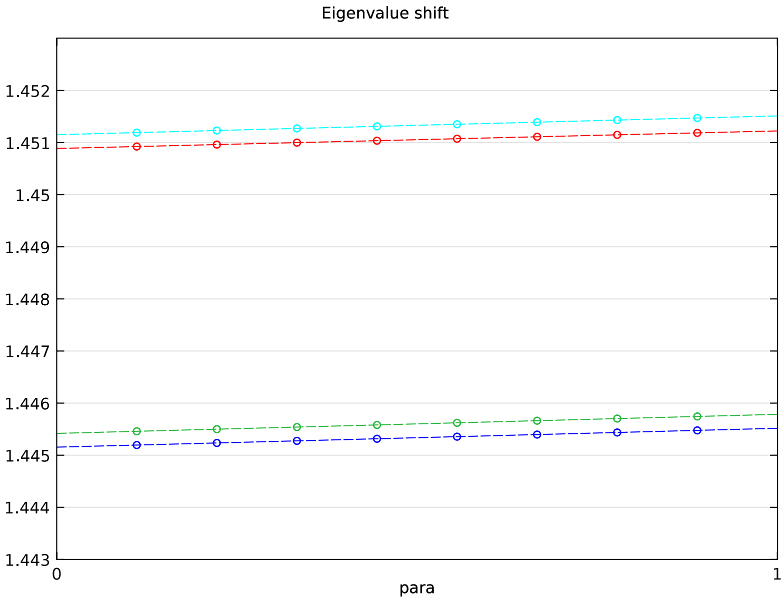


Figure 21-35: A 1D table graph. From the Structural Mechanics Module Model Library: Stress-Optical Effects in Waveguide.

DATA

Select a **Table** or click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

In the **x-axis data** list (or θ **angle data** for polar plots), select the column to use as *x*-axis, select **Row index** to use the table's row indexes (row numbers) as *x*-axis, or leave it at **Automatic** to let the software determine the input from the data in the table.

The **Plot columns** list controls which columns to plot. **All excluding x-axis** (or **All excluding** θ **angle list** for polar plots) indicates all columns not used in **x-axis data** (or θ **angle data**). Select **Manual** instead to specify which columns to plot in the **Columns** list.

Select a **Transformation** of the data from the table—**None** (the default) to use the data directly without any transformation, or select **Frequency domain** to use FFT to transform the data from a time-dependent solution to the frequency domain.

- If **Frequency domain** is selected, the default number of frequencies and frequency range depend on the data, and usually those values do not need changing. To change the values, select the check boxes and enter values for the **Number of frequencies** (the default is 1) and the **Frequency range** in the **Maximum** and **Minimum** fields.

  By default, table plots only display the real data in a table, just as other plot types only display real data unless you use the `imag` function in the expression. To display the imaginary part of complex data in a table, when available, select the **Plot imaginary part** check box. This option is only available when a transformation to the frequency domain is not used.

Table Surface Plots

Use a **Table Surface** ( ) plot to visualize the data in a table that represents a matrix of values that are functions of two independent parameters (for example, as a response surface).

Right-click a **2D Plot Group** ( ) to add this plot type from the **More Plots** submenu. This plot is also available by selecting **Surface Plot** ( ) from the **Results** window's toolbar.

Note

When created from the **Results** window's toolbar, the 2D plot group with the **Table Surface** plot uses **None** in the **Data set** selection. When you add a Table Surface plot to an existing or new 2D plot group, the data set is typically a solution data set, and the plot group includes a plot of the data set edges. This can make the table surface plot hard to see because it uses parameter values on the x- and y-axis and not the 2D geometry's dimension.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Title**, **Range**, **Coloring and Style**, and **Inherit Style**.

See below for sections specific to this plot: **Data.**

DATA

Select a **Table**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to. Select an option from the **Plot data** list—**From table** (the default) or **Manual**.

If **Manual** is selected, select options from the **x-axis data** and **y-axis data** lists, which contain the parameters that define the rows and columns for the table's matrix data, and from the **Data** list, which corresponds to the **Data** list in the **Table** node for the matrix data.

Select the **Plot imaginary part** check box if you want to plot the imaginary part of complex-valued data. For real-valued data, that plot shows a zero imaginary part.

Volume Plots

Use a **Volume** ( ) plot to display a quantity inside a domain in 3D. Add **Deformation** or **Filter** subnodes as required.

Right-click a **3D Plot Group** ( ) to add this plot type.

Tip

Links to all the other plots (including the plot attributes) are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, **Element Filter**, **Shrink Elements**, and **Inherit Style**.

Model

See Figure 21-36 for an example from the COMSOL Multiphysics Model Library.

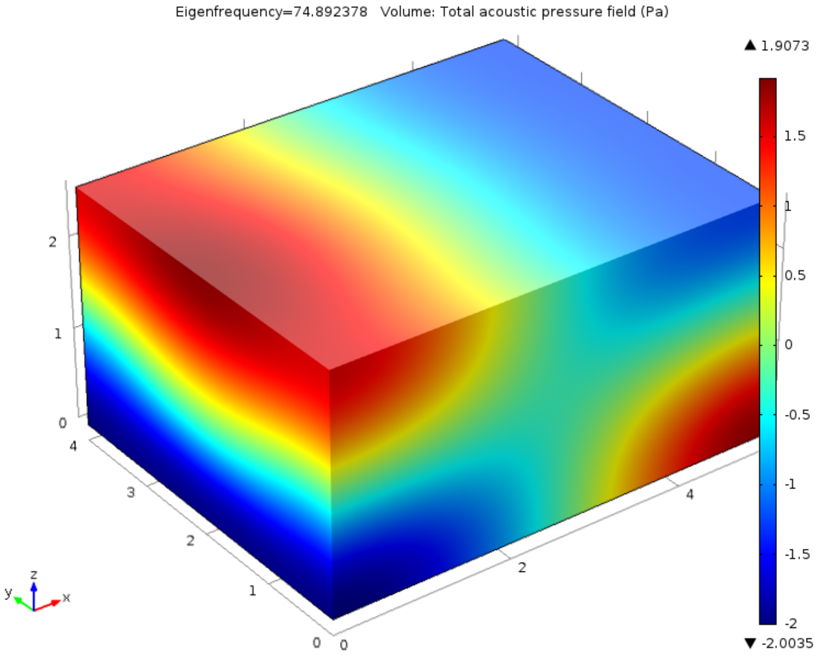


Figure 21-36: A 3D volume plot. From the COMSOL Multiphysics Model Library: Eigenmodes of a Room.

Color Expressions

Use a **Color Expression** feature ( ) to add coloring to the shapes defined by a plot. Add this to these 2D and 3D plots—arrow, contour, isosurface, particle tracing, and streamline.

In the **Model Builder**, add and define a **2D Plot Group** ( ) or **3D Plot Group** ( ). Right-click the plot node (for example, **Streamline**) and select **Color Expression** ( ).

Tip

Links to all the other plots are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Expression**, **Title**, **Range**, and **Coloring and Style**.

Model

See Figure 21-37 and Figure 21-38 for an example from the Subsurface Flow Module Model Library.

.

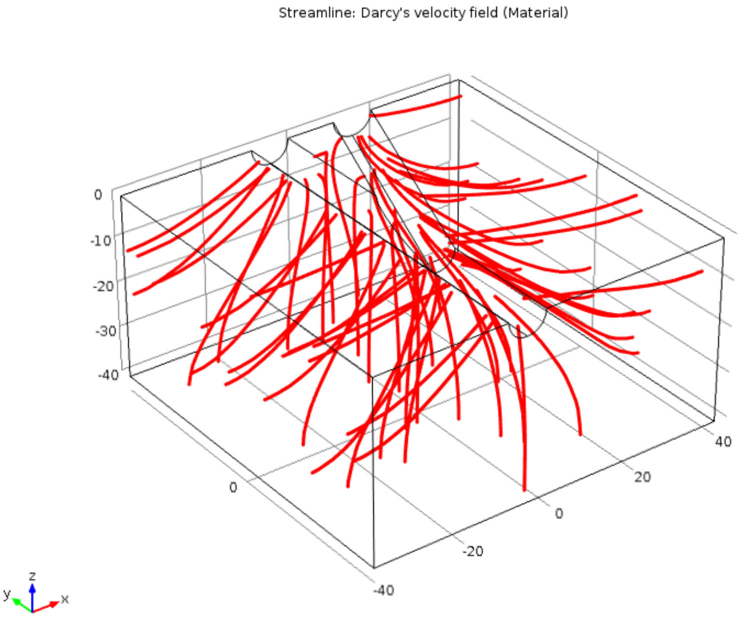


Figure 21-37: A 3D streamline plot with no color expression. From the Subsurface Flow Module Model Library: Open-Hole Multilateral Well.

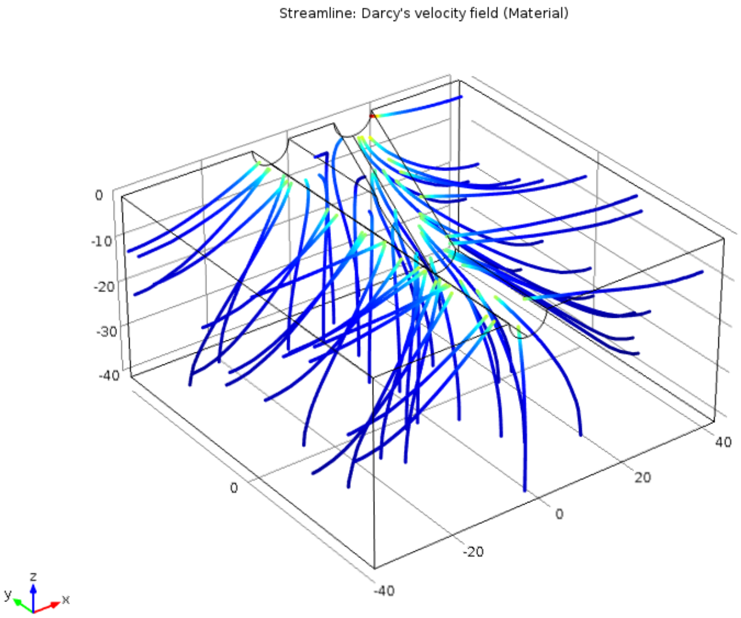


Figure 21-38: A 3D streamline plot with color expression. From the Subsurface Flow Module Model Library: Open-Hole Multilateral Well.

Deformations

Add a **Deformation** feature ( ) to deform the plot according to a vector quantity, for example, the displacement field in structural mechanics. Add this to most 2D and 3D plots—arrow, contour, isosurface, line, slice, streamline, surface, and volume plots. By default, COMSOL scales the deformation to 10% of the geometry.

In the **Model Builder**, add and define a **2D Plot Group** ( ) or **3D Plot Group** ( ). Right-click the plot node (for example, **Arrow Surface**) and select **Deformation** ( ).

Tip

Links to all the other plots are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Also go to Common Plot Settings Cross References for links to information about these sections: **Expression** and **Title**.

See below for sections specific to this plot: **Scale.**

Model

See Figure 21-39 and Figure 21-40 for an example from the Subsurface Flow Module Model Library.

SCALE

Select the **Scale factor** check box to edit the default value for the scale factor.

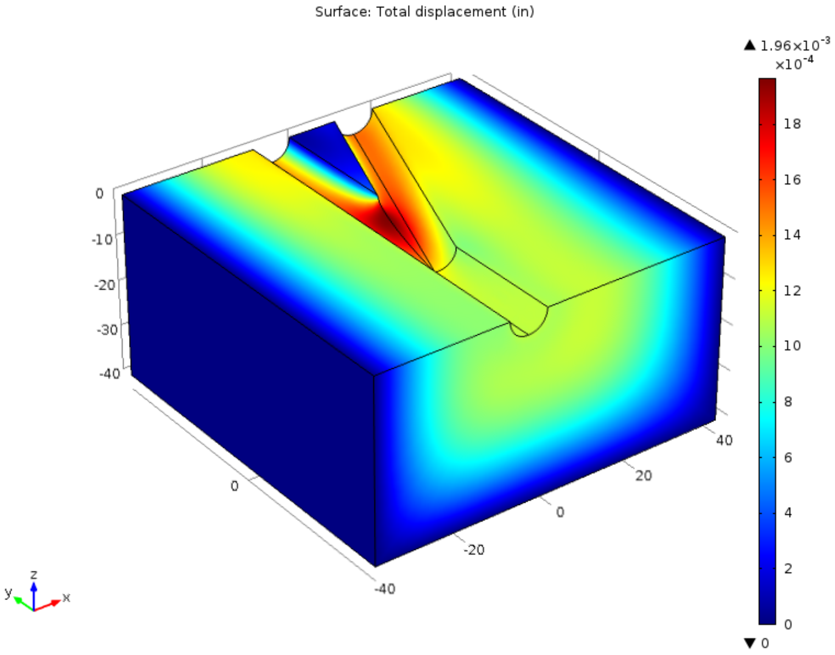


Figure 21-39: A 3D surface plot without deformation. From the Subsurface Flow Module Model Library: Open-Hole Multilateral Well.

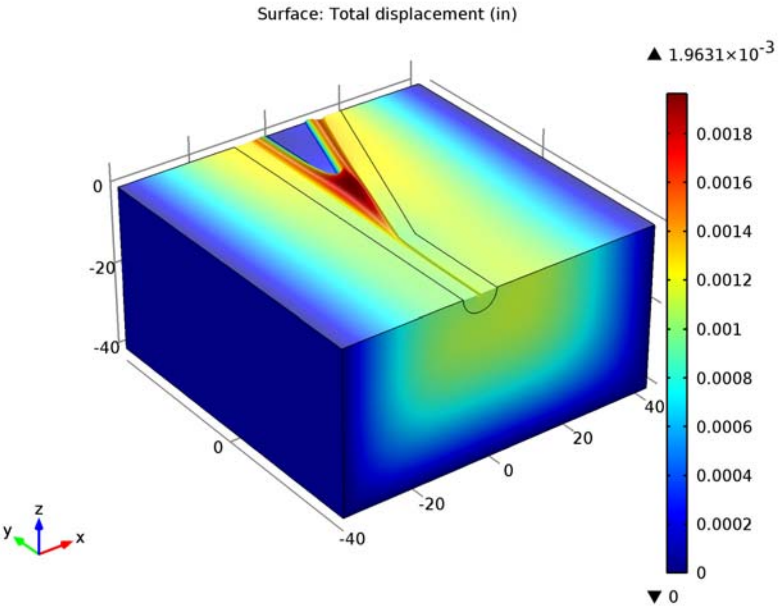


Figure 21-40: A 3D surface plot with deformation. From the Subsurface Flow Module Model Library: Open-Hole Multilateral Well.

Filters

Add a **Filter** ( ) to the following 2D and 3D plots—arrow, contour, isosurface, line plot, slice, and volume. Filters make it possible to filter (limit) the plot using a logical expression that provides a criterion for which parts of the plot to include.

In the **Model Builder**, add and define a **2D Plot Group** ( ) or **3D Plot Group** ( ) with one or more plot nodes. Right-click the plot node (for example, **2D Surface**) and select **Filter** ( ).

Tip

Links to all the other plots are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Model

See Figure 21-41 and Figure 21-42 for an example from the Acoustics Module Model Library.

ELEMENT SELECTION

Under **Element Selection**, enter a **Logical expression for inclusion**. Enter any logical expression using predefined variables. For example, `x>0` filters the plot to only include the part of the geometry where $x > 0$.

Select the **Element nodes to fulfill expression**—**All** (the default), **At least one**, or **At least one but not all**.

- Select **All** to include all elements for which all the element nodes (that is, the entire element) fulfill the criterion in the logical expression.
- Select **At least one** to include all elements for which at least one element node fulfills the criterion in the logical expression (that is, elements that fully or partially fulfill the expression).
- Select **At least one but not all** to include all elements for which at least one of the element nodes but not all of them fulfill the criterion in the logical expression. The

last option is useful for making a filter that shows the plot for a zone around a boundary where the logical expression becomes true.

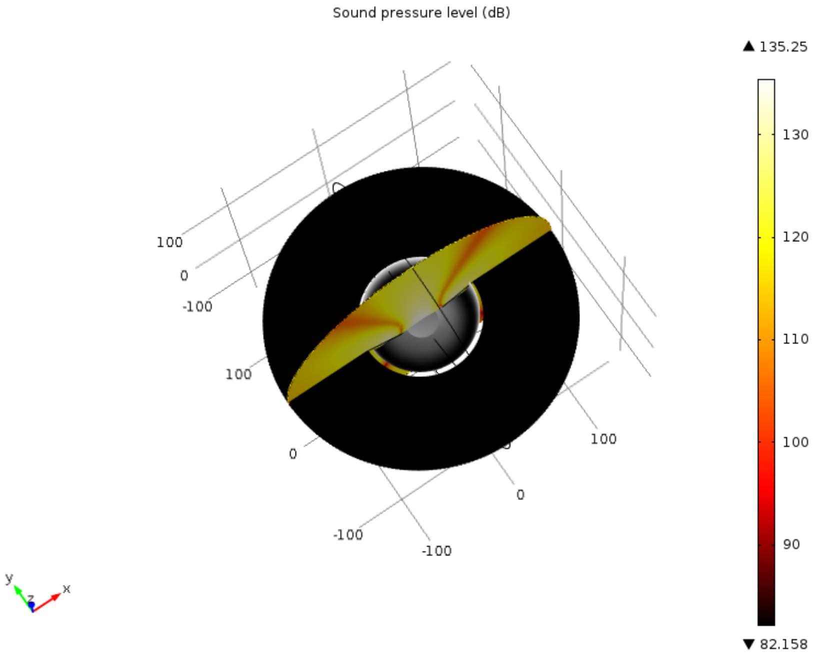


Figure 21-41: A 3D combined plot before a filter is added. From the Acoustics Module Model Library: Baffled Membrane.

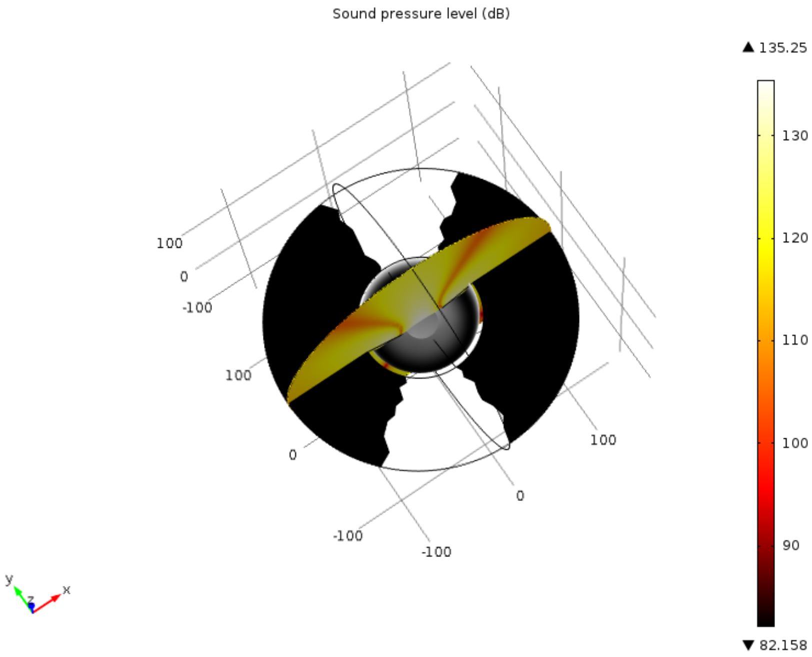


Figure 21-42: A 3D combined plot after a filter is added. From the Acoustics Module Model Library: Baffled Membrane.

Height Expressions

The **Height Expression** subnode ( ) introduces 3D height on a 2D surface plot or 2D table surface plot. Add it to make the height of the plot represent a scalar quantity. 2D Surface, 2D Line, and 2D Table Surface plots support the height expression attribute.

Tip	Links to all the other plots are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

Model

See Figure 21-43 and Figure 21-44 for an example from the COMSOL Multiphysics Model Library.

In the **Model Builder**, add and define a 2D **Surface** plot ( ) or 2D **Table Surface** ( ). Right-click the plot node and select **Height Expression** ( ).

EXPRESSION

Under **Expression**, the **Height data** defaults to **From parent** to use the same data set as the parent plot it belongs to. If **Expression** is selected instead, enter an **Expression**, select a **Unit**, and enter or edit a **Description**. Click the **Replace Expression** ( ) or **Insert Expression** ( ) buttons to select predefined expressions based on the physics of the model. Select the **Description** check box to edit the default description for a predefined expression.

Note

As a subnode to 2D **Histogram** ( ) plots, the **Height** ( ) node's settings window does not contain an **Expression** section.

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**. As a subnode to a 2D **Histogram** ( ) plot, only **Automatic**, **Manual**, and **None** are available.

SCALE

For a manual scaling of the height data, select the check box and enter a **Scale factor** to control the height of the added 3D plot. Enter an **Offset** (default value: 0) or use the associated slider to control the base location (relative to the 2D surface's level).

VIEW

The **Height Expression** subnode makes the plot a 3D plot, which needs a 3D view for the grid, camera, lighting, and other 3D view settings. Select the 3D view to use from the **View** list. The default is **Automatic**, which creates a 3D view if needed. Alternatively, select one of the existing **View 3D** nodes in the model.

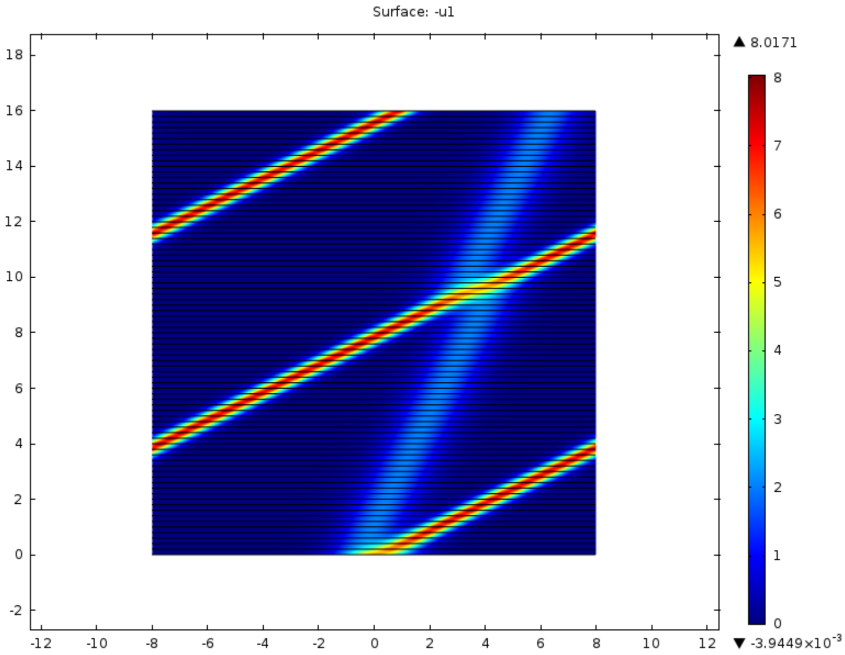


Figure 21-43: A 2D surface plot before a height expression is added. From the COMSOL Multiphysics Model Library: The KdV Equation and Solitons.

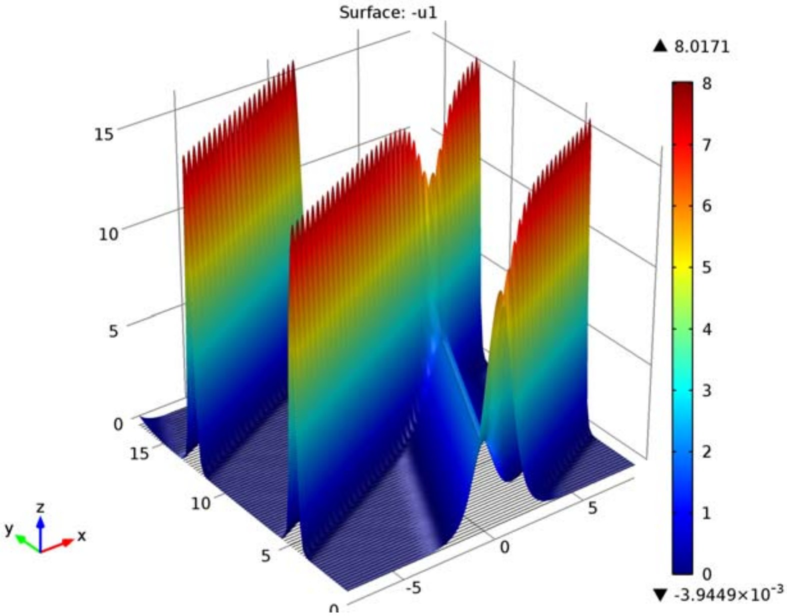


Figure 21-44: A 2D surface plot with height expression. From the COMSOL Multiphysics Model Library: The KdV Equation and Solitons.

Creating Cross-Section Plots and Combining Plots

Cross-section plots are created using a combination of data sets and plot groups. Cross-section plots show the values over time, along a parametric solution, or for several eigenvalues. Cross-section plots visualize a quantity as a family of plots on:

- An arbitrary set of points (in 1D, 2D, or 3D)

  A point cross-section plot makes it easy to view an expression at an arbitrary set of spatial coordinates and results in a line plot. See 1D, 2D, and 3D Cross-Section Point Plots.

Note	Expressions and variables that include derivatives of the dependent variables (for example, stresses in a structural analysis) are not available at isolated geometry vertices (points)

- An arbitrary line (in 2D or 3D). See 2D Cross-Section Line Plots and 3D Cross-Section Line Plots.

  Use Cut Line data sets to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis features available in 1D are available for Cut Line data sets as well as 3D plots and results analysis features for edges.
- Arbitrary planes (in 3D) using a surface plot and cut plane data set. See 3D Cross-Section Surface Plot.

  Use Cut Plane data sets to create planes through a 3D in a 2D geometry to visualize on the plane. All plots and results analysis features available in 2D are available for Cut Plane data sets as well as for surfaces in 3D. The cut plane corresponds to an orthogonal (Cartesian) 2D coordinate system embedded in 3D.

A typical cross-section plot uses a Cut Line 2D data set (which you add in the **Data Sets** branch), which defines a straight line (or set of parallel lines) in a 2D geometry, and a Line Graph in a 1D Plot Group, which uses the Cut Line 2D data set as its data input. You can use the same Cut Line data set for multiple cross-section plots of various quantities along the line that the data set defines, and you can create several Cut Line data sets to plot quantities along different lines of interest.

INTERACTIVE CROSS-SECTION LINE AND SURFACE PLOTS

You can also interactively create cross-section line and surface plots using a combination of cross-section toolbar buttons and clicking the geometry. When you use the cross-section toolbar, plot groups and data sets are automatically added and updated in the Model Builder whenever any line or plane is changed. As soon as a line or plane is completed, a rendering mesh snaps the entry and exit points to the faces of the domain containing the data and to computes the line. See Creating Interactive 2D Cross-Section Line Plots, Creating Interactive 3D Cross-Section Line Plots, and Creating Interactive 3D Cross-Section Surface Plots.

In this section:

- Plotting and Cross-Section Interactive Toolbar
- 1D, 2D, and 3D Cross-Section Point Plots
- 2D Cross-Section Line Plots
- 3D Cross-Section Line Plots
- 3D Cross-Section Surface Plot
- Combining Plots

Plotting and Cross-Section Interactive Toolbar

On the main toolbar for 2D and 3D plots, interactive buttons are available based on the plot type. Use these buttons during the creation of cross-section plots or just in general while creating plots.

TABLE 21-9: PLOTTING AND CROSS-SECTION TOOLBAR

BUTTON	NAME	USE AND RESULT
	Select First Point for Cut Line	Available for 2D and 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click these buttons to plot a cross-section of data between two points.
	Select Second Point for Cut Line	
	Select Cut Line Direction	Available with 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click this button to plot a line perpendicular to a point selected in the Graphics window.

TABLE 21-9: PLOTTING AND CROSS-SECTION TOOLBAR

BUTTON	NAME	USE AND RESULT
	Select Cut Line Surface Normal	Available with 3D plot groups to create a cross-section line plot. Adds a **Cut Line** data set and a **1D Plot Group** with a **Line Graph** that uses this data set. Click this button to plot a line in the same way as a domain point probe, with point and direction.
	Select First Point for Cut Plane Normal	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click these buttons to plot a cross-section of data between the two points along the plane.
	Select Second Point for Cut Plane Normal	
	Select Cut Plane Normal	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click this button to plot a plane perpendicular to a point selected in the Graphics window.
	Select Cut Plane Normal from Surface	Available with 3D plot groups to create a cross-section surface plot. Adds a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot that uses this data set. Click this button to plot a plane.
	Surface	Click this button to add a Surface plot to a 2D or 3D Plot Group.
	Surface with Height	Click this button to add a surface plot with a height attribute to a 2D Plot Group.
	Arrow Surface	Click this button to add an Arrow Surface plot to a 2D Plot Group.
	Streamline	Click this button to add a Streamline plot to a 2D or 3D Plot Group.
	Line	Click this button to add a Line plot to a 2D or 3D Plot Group.
	Slice	Click this button to add a Slice plot to a 3D Plot Group.
	Isosurface	Click this button to add an Isosurface plot to a 3D Plot Group.

TABLE 21-9: PLOTTING AND CROSS-SECTION TOOLBAR

BUTTON	NAME	USE AND RESULT
	Volume	Click this button to add a Volume plot to a 3D Plot Group.
	Arrow Volume	Click this button to add an Arrow Volume plot to a 3D Plot Group.

Notes About Using the Cross-Section Interactive Toolbar

The first time any of the buttons are clicked on the cross-section toolbar, a data set and a plot group containing either a line graph or surface plot are added to the **Model Builder**. No new data set or plot group is created unless the generated data set or plot groups are deleted or disabled. See below for exceptions. COMSOL chooses the default coordinates for the cross section as a vertical line intersecting the data in the middle.

! *Important*	It is important to ensure the areas of the geometry selected contain data when defining the line or plane. When lines or planes are changed, the coordinates and calculations are automatically updated in the data set and in the final plot.

Deleting and Disabling Data Sets and Plot Groups

The following exceptions apply to the data sets and plot groups that are automatically added using the cross-section toolbar.

- If a plot group is *disabled*, no new data set or plot group is created even if you click one of the interactive buttons. You need to enable the plot group to regenerate the cross-section plot.
- If a plot group is *deleted*, click one of the interactive buttons to regenerate the plot group using the cut plane or cut line data set.
- If a data set is *deleted*, and it is used with a plot group, the plot group is also deleted at the same time. However, if the plot group is using another data set it is not deleted.
- If a data set is *disabled*, the associated plot group is not disabled. However, if you want to plot another cross section, click one of the interactive buttons to create a new data set to use with the plot group.

1D, 2D, and 3D Cross-Section Point Plots

CREATING A 1D CROSS-SECTION PLOT USING A CUT POINT DATA SET

A 1D cross-section point plot visualizes a quantity in one or several points in time, along a parameter range, or for several different eigenvalues.

1 Create a **Cut Point 1D** data set.

  Under **Point Data** enter the **x**-coordinates for the plot. Enter one or several space-separated values or a vector of coordinates, for example, `range(0,10,100)`.

2 Click the **Plot** button ().

3 Add a **1D Plot Group** (). Select **Cut Point 1D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-1D Point**.

4 Add a **Point Graph** to this 1D plot group and select **Cut Point 1D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to.

5 Continue to define the **Point Graph** as required.

6 Click the **Plot** button (), right-click the node and select **Plot**, or press F8.

  The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

CREATING A 2D CROSS-SECTION PLOT USING A CUT POINT DATA SET

The 2D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. This example uses the Stresses in a Pulley model from the COMSOL Multiphysics Model Library.

1 Create a **Cut Point 2D** data set.

  - Under **Point Data** enter the *x*- and *y*-coordinate values for the plot. Enter the same number of space-separated values in the **x** and **y** fields. Alternatively, enter a vector of coordinates, for example, `range(0,10,100)`.

2 Click the **Plot** button ().

3 Add a **1D Plot Group** (). Select **Cut Point 2D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-2D Point**.

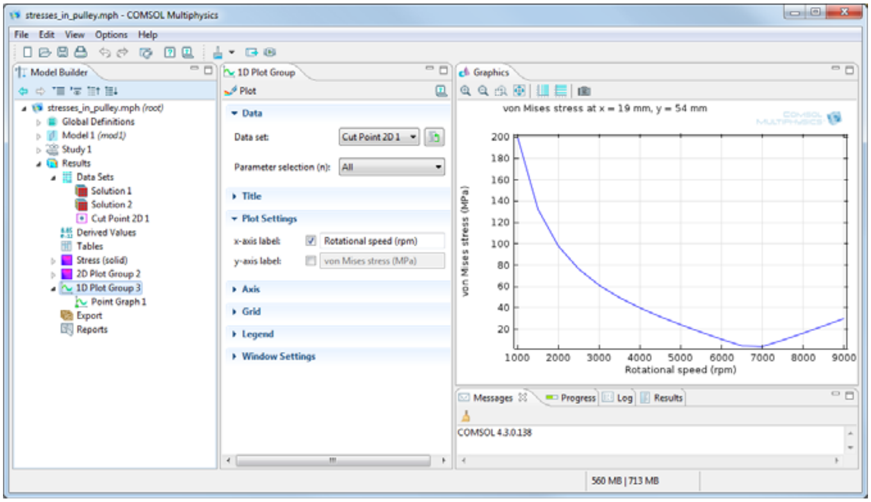
4 Add a **Point Graph** and select **Cut Point 2D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to.

- The *x*-axis corresponds to time, parameter values, or the eigenvalue number.
- The settings in the **y-axis data** area determine the quantity on the *y*-axis. Select from predefined quantities or enter an expression that contains variables.

5 Continue to define the **Point Graph** as required.

6 Click the **Plot** button ( ) right-click the node and select **Plot**, or press F8.

The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.



Data set: Cut Point 2D 1

Parameter selection (n): All

x-axis label: ☑ Rotational speed (rpm)

y-axis label: ☐ von Mises stress (MPa)

COMSOL 4.3.0.138

560 MB | 713 MB

CREATING A 3D CROSS-SECTION PLOT USING A CUT POINT DATA SET

A 3D point cross-section plot visualizes a quantity in one or several points in time, along a parameter range, or for several eigenvalues. This example uses the Star-Shaped Microchannel Chip model from the Microfluidics Module Model Library.

1 Create a **Cut Point 3D** data set.

- Under **Point Data** enter the *x*-, *y*-, and *z*-coordinate values for the plot. Enter the same number of space-separated values in the **x**, **y**, and **z** fields. Alternatively, enter a vector of coordinates, for example, `range(0,10,100)`.

2 Click the **Plot** button ( ).

3 Add a **1D Plot Group** ( ). Select **Cut Point 3D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-3D Point**. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

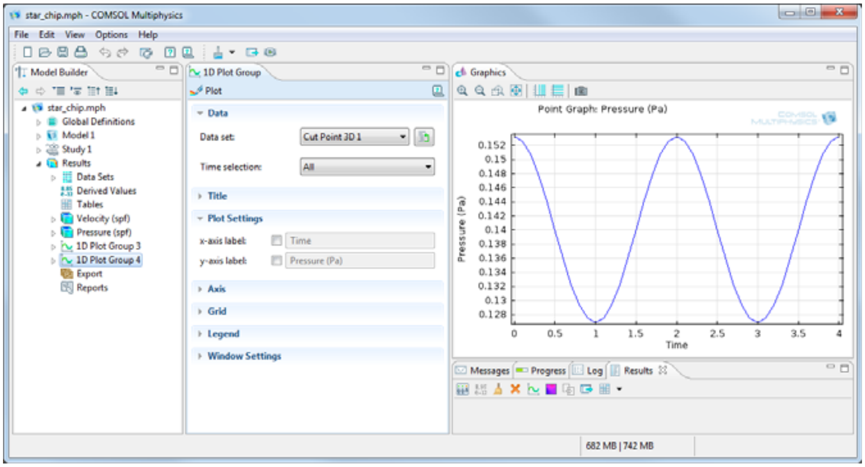
4 Add a **Point Graph** and select **Cut Point 3D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

- The *x*-axis corresponds to time, parameter values, or the eigenvalue number.
- The settings in the **y-axis data** area determine the quantity on the *y*-axis. Select from predefined quantities or enter an expression that contains variables.

5 Continue to define the **Point Graph** as required.

6 Click the **Plot** button ( ), right-click the node and select **Plot**, or press F8.

The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.



Data set: Cut Point 3D 1

Time selection: All

x-axis label: □ Time

y-axis label: □ Pressure (Pa)

2D Cross-Section Line Plots

CREATING A 2D CROSS-SECTION PLOT USING A CUT LINE DATA SET

The 2D line cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues.

1 Create a **Cut Line 2D** data set.
   - Enter the 2D coordinates for the plot. Set the start and end point of the line under **x** and **y**, for Point 1 and Point 2.
   - Select the **Additional parallel lines** check box to visualize on a set of parallel lines. Enter the **Distances** from the line as space-separated values.

2 Click the **Plot** button ( ).

3 Add a **1D Plot Group** ( ). Right-click the plot node and select **Cut Line 2D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-2D Line**.

4 Add a **Line Graph** and select **Cut Line 2D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.
   - Settings under **y-Axis Data** determine the quantity to display on that axis. Either select from predefined **Expressions** or enter an **Expression** containing variables.
   - Settings under **x-Axis Data** determine the quantity on that axis. Select **Arc length** or **Expression** as required. Enter or select an **Expression**, **Unit**, and **Description**.

5 Continue to define the **Line Graph** as required.

6 Click the **Plot** button ( ), right-click the node and select **Plot**, or press F8.

   The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

CREATING INTERACTIVE 2D CROSS-SECTION LINE PLOTS

1 In the **Model Builder**, click a **2D Plot Group** node to display the buttons available on the main toolbar.

2 On the main toolbar, click the **Select First Point for Cut Line** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

3 Click the **Select Second Point for Cut Line** button ( ). Click an end point on the geometry.

   A line connecting the two points is displayed in the **Graphics** window. The first time the cross-section toolbar buttons are clicked, a **Cut Line 2D** data set and a **1D Plot Group** with a **Line Graph** are added to the **Model Builder**.

4 Adjust the cut line as required by clicking the buttons, then clicking the geometry to change where the first and second point start and end (respectively). The

coordinates are updated automatically in the data set and plot group. Click the **1D Plot Group** node to view the updates to the line graph.

5 Continue adjusting the cut line until the line graph representing the points plots the data as required.

3D Cross-Section Line Plots

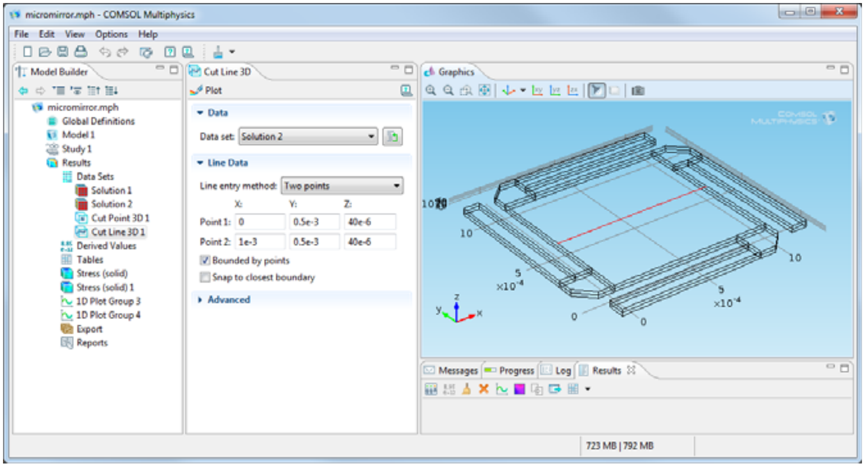
CREATING A 3D CROSS-SECTION PLOT USING A CUT LINE DATA SET

A 3D line/extrusion cross-section plot visualizes a quantity in one or several lines in time, along a parameter range, or for several eigenvalues. This example uses the Prestressed Micromirror model from the MEMS Module Model Library.

1 Create a **Cut Line 3D** data set.

- Enter the 3D coordinates for the plot. Set the start and end point of the line under **x**, **y**, and **z** for **Point 1** and **Point 2**.

2 Click the **Plot** button ( ).



Data

Data set: Solution 2

Line Data

Line entry method: Two points

	X:	Y:	Z:
Point 1:	0	0.5e-3	40e-6
Point 2:	1e-3	0.5e-3	40e-6

☑ Bounded by points

□ Snap to closest boundary

Advanced

3 Add a **1D Plot Group** ( ). Right-click the plot node and select **Cut Line 3D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-3D Line**.

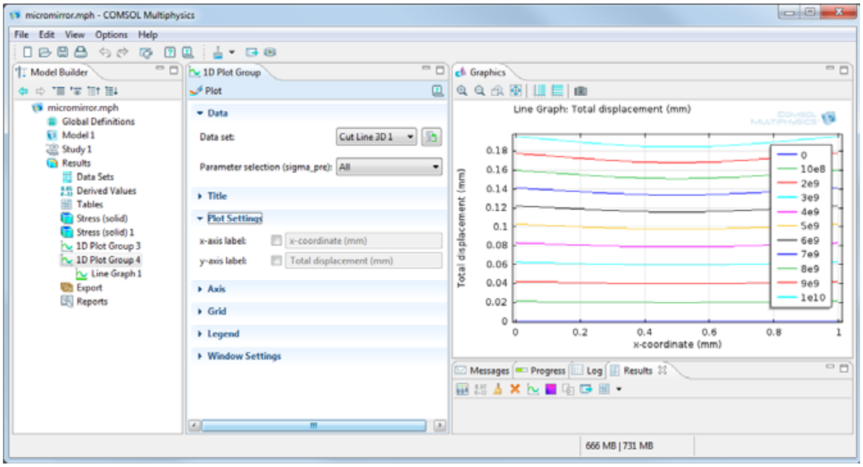
4 Add a **Line Graph** and select **Cut Line 3D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

- Settings under **y-Axis Data** determine the quantity to display on the vertical axis. Either select from predefined **Expressions** or enter an **Expression** containing variables.
- Settings under **x-Axis Data** determine the quantity on the horizontal axis of the cross-section plot. Select **Arc length** or **Expression** as required. Enter or select an **Expression**, **Unit**, and **Description**.

5 Continue to define the **Line Graph** as required. See Line Graphs.

6 Click the **Plot** button ( ) right-click the node and select **Plot**, or press F8.

The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.



micromirror.mph - COMSOL Multiphysics

Data set: Cut Line 3D 1

Parameter selection (sigma_pre): All

x-axis label: □ x-coordinate (mm)

y-axis label: □ Total displacement (mm)

Line Graph: Total displacement (mm)

666 MB | 731 MB

CREATING INTERACTIVE 3D CROSS-SECTION LINE PLOTS

1 In the **Model Builder**, click a **3D Plot Group** node to display the buttons available on the main toolbar. On the main toolbar, click the cross-section buttons as required.

   The first time the cross-section toolbar buttons are clicked, a **Cut Line 3D** data set and a **1D Plot Group** with a **Line Graph** are added to the **Model Builder**.

   Defining a Cut Line:

   a On the main toolbar, click the **Select First Point for Cut Line** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

   b Click the **Select Second Point for Cut Line** button ( ). Click an end point on the geometry. A line connecting the two points is created in the **Graphics** window.

   c Click either of the buttons and then on the geometry to change the start and end points respectively.

   d Click the **1D Plot Group** to view the **Line Graph** based on the selected points.

   Defining a Cut Line—Direction:

   a On the main toolbar, click the **Select Cut Line Direction** button ( ).

   b Click on the geometry to add a line perpendicular to where you clicked. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

   c Click the **1D Plot Group** node to view the **Line Graph** based on the selected points.

   Defining a Cut Line—Surface Normal:

   a On the main toolbar, click the **Select Cut Line Surface Normal** button ( ).

   b Click on the geometry to add a line with a point and direction. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle.

   c Click the **1D Plot Group** node to view the **Line Graph** based on the selected points.

2 Adjust the cut line as required by clicking the buttons, then clicking the geometry to change its coordinates. The data set and plot group are updated automatically with the cut line data. Click the **1D Plot Group** node to view the updates to the line graph.

3 Continue adjusting the cut line until the line graph representing the points plots the data as required.

3D Cross-Section Surface Plot

3D CROSS-SECTION SURFACE PLOT USING A CUT PLANE DATA SET

A 3D surface cross-section plot visualizes a quantity in one or several planes in time, along a parameter range, or for several eigenvalues. This example uses the *Airflow Over an Ahmed Body* model from the CFD Module Model Library.

1 Create a **Cut Plane 3D** data set.

2 Click the **Plot** button ( ), or right-click the node and select **Plot**.

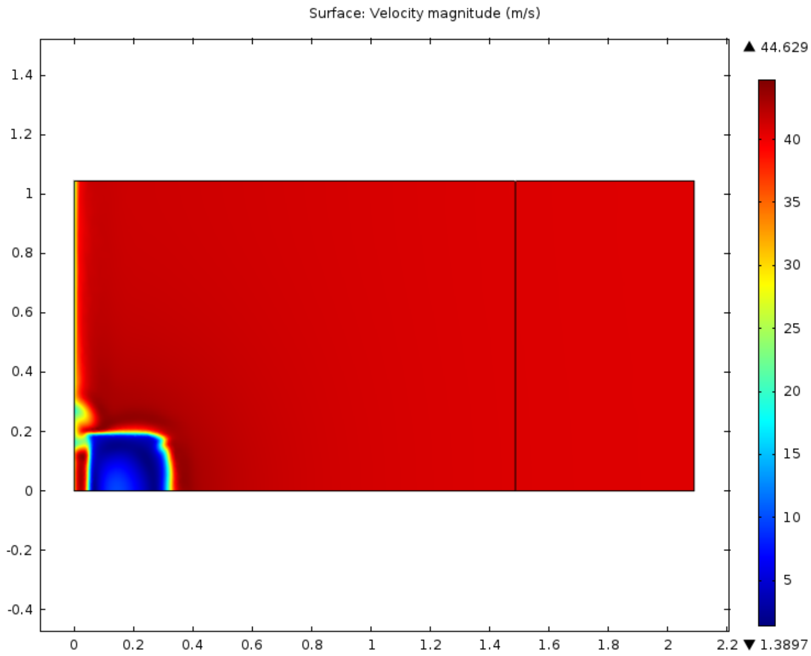
3 Add a **2D Plot Group** ( ). Right-click the plot node and select **Cut Plane 3D** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-3D Surface**.

4 Add a **Surface** plot and select **Cut Plane 3D** as the **Data set** or **From parent** to use the same data set as the plot group it belongs to. Click the **Go to Source** button ( ) to move to the node that the selection in the list next to the button refers to.

5 Continue to define the **Surface** plot as required.

6 Click the **Plot** button ( ), or right-click the node and select **Plot**.

The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.



CREATING INTERACTIVE 3D CROSS-SECTION SURFACE PLOTS

1 In the **Model Builder**, click a **3D Plot Group** node to display the buttons available on the main toolbar. On the main toolbar, click the cross-section buttons as required.

   The first time the cross-section toolbar buttons are clicked, a **Cut Plane 3D** data set and a **2D Plot Group** with a **Surface** plot are added to the **Model Builder**.

   To Define a Cut Plane:

   a Click the **Select First Point for Cut Plane Normal** button ( ). Click a start point on the geometry. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the cut plane looks like if this first click point is chosen for the surface plot.

b Click the **Select Second Point for Cut Plane Normal** button ( ). Click an end point on the geometry. The green highlighted areas show you what the cut plane looks like if this second click point is chosen for the surface plot.

c Click either of the buttons and then on the geometry to change the start and end points respectively.

d Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

To define a normal Cut Plane:

a Click the **Select Cut Plane Normal** button ( ).

b Click on the geometry to add a plane perpendicular to the click location. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the plot looks like if this normal cut plane is chosen for the surface plot.

c Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

To define a normal Cut Plane from Surface:

a Click the **Select Cut Plane Normal from Surface** button ( ).

b Click on the geometry to add a line with a point and direction starting at the click location. COMSOL chooses default coordinates as a vertical line intersecting the data in the middle. The green highlighted areas show you what the plot looks like if this normal from surface cut plane is chosen for the surface plot.

c Click the **2D Plot Group** to view the **Surface** plot based on the selected points.

2 Adjust the cut plane as required by clicking the buttons, then clicking the geometry to change the coordinates. The data set and plot group are updated automatically with the cut plane data. Click the **2D Plot Group** node to view the updates to the surface plot.

3 Continue adjusting the cut plane until the surface plot representing the points plots the data as required.

Combining Plots

Tip

Links to all the other plots are found in Table 21-5 (1D and polar) and Table 21-6 (2D and 3D).

1. Define one or more **Data Sets**.
2. In the **Model Builder,** add one or more **1D** (), **2D** (), or **3D Plot Groups** ().
3. Add individual plots to the plot group. Experiment with how different combinations of plots look by adding a plot, then right-clicking and selecting **Disable**.

   The revised plot displays in the window selected in the **Plot window** list. Right-click the plot node again to **Enable** the plot in the plot group. Or select **Delete** to remove it.
4. Click the **Plot** button (), or right-click individual plot nodes and select **Plot**.

   The plot displays in the window selected in the **Plot window** list. To plot results in another windows, right-click the plot group node or the plot node and select a plot window from the **Plot In** submenu.

Tip

Many of the Model Library plots described are examples of combined plots.

Defining Derived Values and Tables

In this section:

Derived Values

Integrate or compute the average, maximum, or minimum of any quantity to compute derived quantities such as total flux, charges, inductances, reaction forces, and average, maximum, and minimum values.

Use **Derived Values** (8.85 e-12) to define evaluations of numerical results—globally, in a point, or integrated quantities. For 2D and 3D plots, also get numerical results directly in a table by clicking the plot.

For all derived values also apply an operator on a data series (from a parametric or time-dependent study) to compute, for example, the temporal average of a quantity in a point of the domain for which a time-dependent solution is computed. In addition to the average, also compute the integral, maximum, minimum, RMS (root mean square), standard deviation, or variance of the data series. The derived values nodes use data sets (typically solution data sets) that provides the data from which the derived values are computed. If needed, specify the frame and geometry to use in the data set's settings window.

TABLE 21-10: DERIVED VALUE TYPES

DERIVED VALUE	ICON	DESCRIPTION	LINK TO SECTION
Point Evaluation	8.85 e-12	To evaluate expressions or variables defined in a point.	Point Evaluation Derived Values
Global Evaluation	8.85 e-12	To evaluate the numerical value of a global variable.	Global Evaluation Derived Values
Global Matrix Evaluation	8.85 e-12	To define the evaluation of the numerical values for a global matrix variable such as S-parameters in a model with several ports activated as a parametric sweep and a frequency-domain study.	Global Matrix Evaluation
Particle Evaluation	8.85 e-12	To evaluate an expression for all, or a subset of, the particles in a particle tracing model.	Particle Evaluation
AVERAGE SUBMENU			
Volume Average	∿	To evaluate an average over a set of domains in 3D models.	Volume Average, Surface Average, and Line Average Derived Values
Surface Average	∿	To evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
LINE AVERAGE	∿	To evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.	
INTEGRATION SUBMENU			
Volume Integration	∭	To evaluate an integral over a set of domains in 3D models.	Volume Integration, Surface Integration, and Line Integration Derived Values
Surface Integration	∬	To evaluate an integral over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
Line Integration	∫	To evaluate an integral over a set of domains in 1D, boundaries in 2D, or edges in 3D.	

TABLE 21-10: DERIVED VALUE TYPES

DERIVED VALUE	ICON	DESCRIPTION	LINK TO SECTION
MAXIMUM SUBMENU			
Volume Maximum	∿	To evaluate the maximum over a set of domains in 3D models.	Volume, Surface, and Line Maximums and Minimums for Derived Values
Surface Maximum	∿	To evaluate the maximum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
Line Maximum	∿	To evaluate the maximum over a set of domains in 1D, boundaries in 2D, or edges in 3D.	
MINIMUM SUBMENU			
Volume Minimum	∿	To evaluate the minimum over a set of domains in 3D models.	Volume, Surface, and Line Maximums and Minimums for Derived Values
Surface Minimum	∿	To evaluate the minimum over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.	
Line Minimum	∿	To evaluate the minimum over a set of domains in 1D, boundaries in 2D, or edges in 3D.	

See Also

- Adding and Evaluating Tables for a Derived Value
- Adding A Derived Value Node
- Results Evaluation and Visualization
- Defining Derived Values and Tables
- Derived Values Common Settings
- Editing and Organizing Results Tables
- Getting Numerical Results Directly
- Entering Ranges and Vector-Valued Expressions
- Solver Studies and Study Types
- Working with Geometry

Adding A Derived Value Node

1 In the **Model Builder**, under **Results**, right-click **Derived Values** (8.85 e-12).

2 Select an option from the list and continue defining each derived value (see Table 21-10).

3 After adding a **Derived Value** node, click the **Evaluate** button (☰) or right-click the **Derived Values** node and select **Evaluate All** (☰) or **Clear and Evaluate All** (☰).

4 **Evaluate All** calculates and displays the results in the derived values table for all nodes in the Model Builder sequence and keeps the previous data. **Clear and Evaluate All** resets and clears all the data and calculates from the beginning.

The **Results** window opens to view a **Table** of the **Derived values**. By default four digits are displayed; click the **Full precision** button (8.85 e-12) to display as many significant digits as possible. The **Table display precision** level can be changed from the **Options>Preferences>General** menu.

See Also

- Adding and Evaluating Tables for a Derived Value
- Derived Values Common Settings
- Editing and Organizing Results Tables

Derived Values Common Settings

Under **Results**, there are common sections on each settings window. Table 21-7 provides cross references to the information.

TABLE 21-11: DETAILS FOR THE COMMON DERIVED VALUES SETTINGS SECTIONS

SECTION ON THE SETTINGS WINDOW	CROSS REFERENCE
Data	Data and Selecting Solutions for Parametric Sweeps
Data Series Operation	Data Series Operation
Expression	Expressions and Predefined Quantities
Integration Settings	Integration Settings
Parametric Solver and Parametric Sweep studies	Selecting Solutions for Parametric Sweeps
Selection (Solution Data sets)	Selection (Solution Data Sets Only)

Tip

Links to all the derived value types are found in Table 21-10.

DATA

Select a **Data set**. The result is computed by averaging or integrating over the data set selected. Only data sets with data for an applicable space dimension are available.

Integrate over any data set of the right dimension. For example, make a volume integration of a 2D revolved data set, or a surface integration of a cut plane.

For time-dependent and eigenvalue solutions you also select the times or eigenvalues for the solution data to evaluate. For parametric sweeps there are additional settings for selecting solutions; see Selecting Solutions for Parametric Sweeps.

SELECTION (SOLUTION DATA SETS ONLY)

Select **Manual** from the **Selection** list to choose geometry directly from the **Graphics** window. Select **All** to add the applicable geometry or any other predefined grouping.

INTEGRATION SETTINGS

Select a **Method**—**Auto** (the default), **Integration**, or **Summation**.

- **Auto**—the default and computes the integral for fields by numerical integration and for reaction forces by summation. Predefined quantities for reaction forces use summation instead of integration. Also, if an expression is specified for the integrand using the `reacf` operator, the automatic setting chooses the summation method.
- **Integration**—the standard numerical integration method (quadrature).
- **Summation**—a summation method useful for calculating reaction forces. The summation method finds all nodes on the boundary, evaluates the expression in the nodes and sum up the values. Reaction force variables are predefined in the structural mechanics interfaces.

COMSOL automatically determines an appropriate **Integration order** for the expression. To change this, click to clear the check box and enter a number. COMSOL performs the integration elementwise using numeric quadrature of the selected order.

2D Axisymmetric Models

2D Axi	For 2D axisymmetric models, **Volume** and **Surface Integration** is available. COMSOL multiplies the expression (integrand) with `2*pi*r` prior to integration to compute the corresponding volume or surface integral. Complete the next step.

- For **Surface Integration**, select the **Compute volume integral (axial symmetry)** check box.
- For **Line Integration**, select the **Compute surface integral (axial symmetry)** check box.

DATA SERIES OPERATION

These settings are available for all **Derived Value** types. In addition to performing the averaging on each solution in a data series (from a parametric or time-dependent study) an operation can be applied such as the integral or maximum of the averaged quantity for the data series so that the result is, for example, the integral or maximum of the averaged quantity for each step in the data series. The following operators are available from the **Operation** list. Select:

- **None** (the default) to evaluate the average of the data series
- **Average** to evaluate the average of the average of the data series

- **Maximum** to evaluate the maximum of the average of the data series. If **Maximum** is selected, select an option from the **Find maximum of** list—**Real part** or **Absolute value**.
- **Minimum** to evaluate the minimum of the average of the data series. If **Minimum** is selected, select an option from the **Find minimum of** list—**Real part** or **Absolute value**.
- **Integral** to evaluate the integral of the average of the data series
- **RMS** (the *root mean square* or *quadratic mean*)
- **Standard deviation**
- **Variance**

See Also

- Derived Values
- Solver Studies and Study Types
- Entering Ranges and Vector-Valued Expressions

Volume Average, Surface Average, and Line Average Derived Values

The derived average values are useful for calculating averaged quantities for each solution in a data set (a time-dependent solution, for example). Also apply an integral, maximum, or other operation to compute the maximum of an averaged quantity, for example. Under **Results** right click **Derived Values** (8.85 e-12) and select:

- **Volume Average** ( ) to evaluate an average over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Results** window.
- **Surface Average** ( ) to evaluate an average over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Average** ( ) to evaluate an average over a set of domains in 1D, boundaries in 2D, or edges in 3D.

Model

For a line average example, see Tubular Reactor: Model Library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operations**.

Volume Integration, Surface Integration, and Line Integration Derived Values

The derived integration values are useful for calculating integrated quantities for each solution in a data set (a time-dependent solution, for example). Also apply an average, maximum, or other operation to compute the average of an integrated quantity, for example. Under **Results** right click **Derived Values** (8.85 e-12) and select:

- **Volume Integration** (∭) to evaluate an integral over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Results** window.
- **Surface Integration** (∬) to evaluate an integral over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Integration** (∫) to evaluate an integral over a set of domains in 1D, boundaries in 2D, or edges in 3D. Integrate over any data set of the right dimension. For example, make a volume integration of a 2D revolved data set or a surface integration of a cut plane.

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operation**.

- For a surface integration example see Effective Diffusivity in Porous Materials: Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**.

Model

For a line integration example:

- If you have the AC/DC Module, see An RFID System: Model Library path **ACDC_Module/Inductive_Devices_and_Coils/rfid**
- If you have the CFD Module, see Filling of a Capillary Channel—Level Set: Model Library path **CFD_Module/Multiphase_Tutorials/capillary_filling_ls**
- If you have the Microfluidics Module, see Filling of a Capillary Channel—Level Set: Model Library path **Microfluidics_Module/Two-Phase_Flow/capillary_filling_ls**
- If you have the Heat Transfer Module, see Cavity Radiation: Model Library path **Heat_Transfer_Module/Verification_Models/cavity_radiation**

For a volume integration example:

- If you have the AC/DC Module, for a volume and surface integration example, see Eddy Currents: Model Library path **ACDC_Module/Inductive_Devices_and_Coils/eddy_currents**
- If you have the RF Module, see Microwave Oven: Model Library path **RF_Module/RF_and_Microwave_Engineering/microwave_oven**

Volume, Surface, and Line Maximums and Minimums for Derived Values

The derived maximum and minimum values are useful for calculating maximum or minimum quantities for each solution in a data set (a time-dependent solution, for example). Also apply an average, maximum (or minimum), or other operation to compute the maximum or minimum over the entire data set for a quantity, for example. Under **Results** right click **Derived Values** (8.85e-12) and:

- Use **Volume Maximum** (∿) or **Volume Minimum** (∿) to evaluate a maximum or minimum value over a set of domains in 3D models. The result of the evaluation is stored in a **Table** and displayed in the **Results** window.

- Use **Surface Maximum** ( ) or **Surface Minimum** ( ) to evaluate a maximum or minimum value over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- Use **Line Maximum** ( ) or **Line Minimum** ( ) to evaluate a maximum or minimum value over a set of domains in 1D, boundaries in 2D, or edges in 3D.

Model

For a volume maximum example, and if you have the Nonlinear Structural Materials Module, see Polynomial Hyperelastic Model: Model Library path **Nonlinear_Structural_Materials_Module/Hyperelasticity/ polynomial_hyperelastic**

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data**, **Selection**, **Expression**, and **Data Series Operation**.

See below for sections specific to this derived value: **Advanced**.

ADVANCED

Choose to find the maximum or minimum of the real part or the absolute value, which are different for complex-valued data. Choose **Real part** (the default) or **Absolute value** from the **Find maximum of** or **Find minimum of** list.

Point Evaluation Derived Values

Use **Point Evaluation** ( ) to define the evaluation of a variable or an expression in a point. The result is stored in a **Table** and displayed in the **Results** window.

Model

Tapered Cantilever with Two Load Cases: Model Library path **COMSOL_Multiphysics/Structural_Mechanics/tapered_cantilever**

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data**, **Selection**, **Expression**, and **Data Series Operation**.

Global Evaluation Derived Values

Use a **Global Evaluation** (8.85 e-12) to define the evaluation of the numerical value of a global variable.

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data**, **Expression**, and **Data Series Operation**.

Global Matrix Evaluation

Use a **Global Matrix Evaluation** (8.85 e-12) to define the evaluation of the numerical values for a global matrix variable such as S-parameters in a model with several ports activated as a parametric sweep and a frequency-domain study. The **Results** window then displays all values for all frequencies in a parametric sweep.

Tip

Links to all the other derived value types are found in Table 21-10.

Also go to Derived Values Common Settings for links to information about these sections: **Data** and **Expression**.

See below for sections specific to this derived value: **Data Series Operation**.

DATA SERIES OPERATION

Select from the following options for operations on the data series for the inner solutions—typcially a frequency sweep or time series from a study—and the outer solutions—the parametric sweep (for ports in electromagnetics, for example)—from the **Inner solutions** and **Outer solutions** lists:

- Select **None** to use the computed value as it is (the default).
- Select **Average** to use the average of the computed values for the inner solutions or the outer solutions.
- Select **Sum** to use the sum of the computed values for the inner solutions or the outer solutions.

The **Ignore NaN** check box is selected by default so that the results tables do not include NaNs (Not-A-Number values). If you want to see values that a NaNs, clear this check box.

TRANSFORMATION

You can apply a transformation operation to compute the inverse of the matrix variable or to convert between the impedance matrix, **Z**, the admittance matrix, **Y**, and the S-parameter matrix S (for more information about these lumped parameters, see the *AC/DC Module User’ Guide*). From the **Transformation** list, choose one of the following transformations:

- **None** (the default).
- **Inverse**.
- **From S to Y**. For this transformation, also specify the **Characteristic admittance** (SI unit: S). The default value is 1 S. Click the **Range** button ( ) to specify a range of admittance values.
- **From S to Z**. For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is 1 Ω. Click the **Range** button ( ) to specify a range of impedance values.
- **From Y to S**. For this transformation, also specify the **Characteristic admittance** (SI unit: S). The default value is 1 S. Click the **Range** button ( ) to specify a range of admittance values.
- **From Y to Z**.
- **From Z to S**. For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is 1 Ω. Click the **Range** button ( ) to specify a range of impedance values.
- **From Z to Y**.

Note

The transformation operations are only applicable for square matrices.

Particle Evaluation

Note

Particle Evaluation derived value feature requires the Particle Tracing Module.

Use a **Particle Evaluation** ( ) derived values feature to define the evaluation of the numerical value of quantities computed by one of the particle tracing interfaces.

Tip

Links to all the other derived value types are found in Table 21-10. Also see Derived Values Common Settings for links to more information about the **Expression** and **Data** sections.

DATA

Select a **Data set**. Only the **Particle** data sets ( ) are available for selection. If no particle data set is available then the only option is **None**.

The **Select via** option is used to select the times at which the particle evaluation will occur. When **Stored output times** is selected, the **Time** list is populated with the output times from the solver, which is linked to the particle data set. When **Interpolation times** is selected, then enter an array of times for which the particle evaluation should be performed at. Click the **Range** button ( ) to specify a range of times.

EXPRESSION

Select predefined expressions based on the physics of the model or enter an **Expression** in the field. Enter a **Unit** and if required, select the **Description** check box to enter a description (or edit the default). The value of the expression is computed for each particle at the selected times based on the particle location at that time.

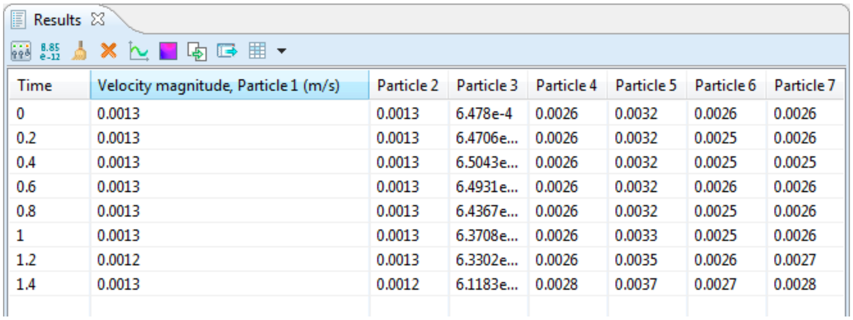
EVALUATION

To specify the particles to evaluate the expression for, the **Particles to evaluate** list contains the following options:

- Select **All** (the default) to evaluate the expression for all particles in the simulation.
- Select **Fraction** to evaluate for a fraction of all particles. Enter a scalar value between 0 and 1 in the **Fraction of particles** field. The default value is 1, which means that the evaluation includes all particles.
- Select **Number** to specify a number of particles to evaluate for in the **Number of particles** field. The default value is 100 particles. If the particle simulation contains fewer particles than the specified number, all particles are included.

The Results Table

Click the **Evaluate** button ( ) or right-click the **Derived Values** node and select **Evaluate All** ( ) or **Clear and Evaluate All** ( ). Select **View>Results** to open the **Results** window and view a table of the particle evaluation values.



Time	Velocity magnitude, Particle 1 (m/s)	Particle 2	Particle 3	Particle 4	Particle 5	Particle 6	Particle 7
0	0.0013	0.0013	6.478e-4	0.0026	0.0032	0.0026	0.0026
0.2	0.0013	0.0013	6.4706e...	0.0026	0.0032	0.0025	0.0026
0.4	0.0013	0.0013	6.5043e...	0.0026	0.0032	0.0025	0.0025
0.6	0.0013	0.0013	6.4931e...	0.0026	0.0032	0.0026	0.0026
0.8	0.0013	0.0013	6.4367e...	0.0026	0.0032	0.0025	0.0026
1	0.0013	0.0013	6.3708e...	0.0026	0.0033	0.0025	0.0026
1.2	0.0012	0.0013	6.3302e...	0.0026	0.0035	0.0026	0.0027
1.4	0.0013	0.0012	6.1183e...	0.0028	0.0037	0.0027	0.0028

The first column in the table is a list of the **Time** values selected from the **Data** section. There are N additional columns, where N is the number of particles chosen in the **Evaluation** section.

The values in each column correspond to the supplied **Expression** for all the selected **Time** values. Each column contains M rows, where M is the number of **Time** values selected in the **Data** section. By default four digits are displayed; click the **Full precision** button (8.85 e-12) to display as many significant digits as possible. If required, the precision level can be changed from the **Options>Preferences>General** menu.

Adding and Evaluating Tables for a Derived Value

Tables ( ) can store the result of **Derived Values** and results from probes, for example. The result are displayed in the **Results** window, which by default is located below the **Graphics** window. There are two ways to add a **Table**:

- In the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, click the **Evaluate (New Table)** button ( ). This evaluates the feature and stores the result in a new table. This button is also used as the **Evaluate** button.
- In the **Model Builder**, right-click the **Tables** node, and select **Table** ( ).

  A **Table** node is added under **Tables**. Click the node to display a table with the selected integration feature's description and values in the **Results** window.

There are three ways to evaluate a **Derived Values** node and put the result in a **Table**:

- In the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, click the **Evaluate** button (≡).
- In the **Model Builder**, right-click the specific **Derived Values** node (for example, **Volume**, **Surface**, or **Line Integration**) and select **Evaluate**.
- In the **Model Builder**, right-click the **Derived Values** node, and select **Evaluate All** (≡) to evaluate all the **Derived Values** nodes. This appends the results to any existing tables. Select **Clear and Evaluate All** ( ) to first clear the affected tables and then evaluate all the **Derived Values** nodes.

In addition, you can specify the output table for data from a probe in the **Table and Windows Settings** section of the probe's settings window. Also, for nested parametric sweeps you can add an accumulated probe table, which provides a matrix of values where the rows and columns represent two independent parameters.

To delete all tables, right-click the main **Tables** ▦ node and select **Delete All** (✖). To clear the contents of all tables, right-click the main **Tables** ▦ node and select **Clear All** ( ).

1 To add a **Table** node, if not already created, right-click the **Tables** (▦) and select **Table** (▦).

2 In the **Data** section, if applicable (for accumulated probe tables, for example) select a format from the **Format** list:

   - Select **All** (the default) to display all data in the table. For many tables this is the only available format.
   - Select **Filled** to create a matrix of data from, for example, a nested parametric sweep with independent parameters. Select the parameters to use for the rows and the columns from the **Rows** and **Columns** lists, and select the data (probed quantity, for example) to use in the table from the **Data** list. Under **Column headers**, select a **Column description** (in addition to the parameter values): **None**, **From data** (the default), or **Manual**. If you select **Manual**, type the description in the **Description** field.

3 Click the **Import** button if you want to use the table for imported data from, for example, a spreadsheet. Select a data file in the **Import** dialog box that opens.

4 In the **Storage** section, select a way to store the table data from the **Store table** list:

- Select **In model** (the default) to store the table data in the model.
- Select **On file** to store the table data in the model to a file that you select by first clicking **Browse** or enter into the **Filename** field.
- Select **In model and on file** to store the table data both in the model and on file.

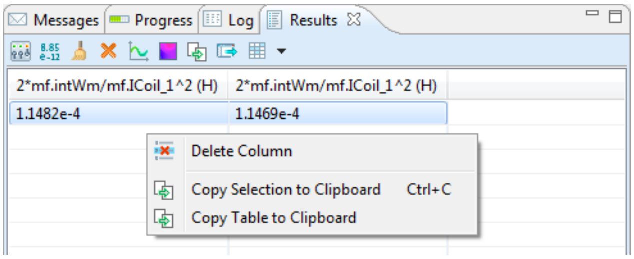
  For the storage in the model, you can specify the buffer size in the **Buffer size (rows)** field, The default is 10,000 rows.

Editing and Organizing Results Tables

When a **Table** (▦) is generated and displays in the **Results** window, a variety of editing options are available as listed in Table 21-12 and in Figure 21-45.

TABLE 21-12: RESULTS TABLE EDITING BUTTONS

BUTTON AND FUNCTION	DESCRIPTION
Full precision (8.85 e-12)	Click to display as many significant digits as possible.
Clear Table (🧹)	Click to clear the data from the table, but keep the table itself. Click the Evaluate button (☰) to regenerate the table data.
Delete Table (✖)	Click to delete the table. There is no undo. If required, click the Evaluate button (☰) to regenerate the table.
Plot (📈)	Click to plot the table in the **Graphics** window.
Copy Table to Clipboard (📋)	Click the button or right-click anywhere in the table and select this option from the context menu. Paste the data in a spreadsheet, for example.
Export (➡)	Click to export the table to a text file in a spreadsheet format.
Delete column	Right-click a column header or anywhere in the table, and select Delete column. There is no undo. If required, click the Evaluate button (☰) to regenerate the table.
Copy selection to clipboard	Select the rows to copy, then right-click anywhere in the table and select this option from the context menu or press Ctrl+C.



2*mf.intWm/mf.ICoil_1^2 (H)	2*mf.intWm/mf.ICoil_1^2 (H)
1.1482e-4	1.1469e-4

Figure 21-45: Context menu and button options for editing results tables.

Getting Numerical Results Directly

2D

3D

For 2D and 3D models, the numerical value of the current plot can be displayed by clicking anywhere in the model geometry.

For a 3D model, the value is for the point where the a ray projected from the point clicked hits the geometry.

COMSOL Multiphysics displays the value at that point along with the point's coordinates in a row in an **Evaluation 2D** or **Evaluation 3D** table that is found in the **Results** window. Each click adds a row to the table. From the **Results** window, plot or copy the table data to a clipboard like any other table.

Exporting Data and Images

In this section:

- Export Feature Types
- Animation Export
- Data Export
- Mesh Export
- Table Export
- Plot 1D, 2D, or 3D Image Export
- Plot Export
- Player Export

Export Feature Types

After a model is completed, you can add various components to the **Export** branch and then generate outputs (animations, data, images, player, or export), or export the information to your computer for use in external documents or for other purposes.

In the **Model Builder**, under the **Export** node, right-click and select an option as listed in Table 21-13.

TABLE 21-13: EXPORT TYPES

EXPORT TYPE	ICON	DESCRIPTION
Animation Export		To define and export a movie or series of images based on a plot group. Play the animation in a web browser or use it in presentations or on a web site.
Data Export		Exports numerical data to file. Data export operates directly on data sets. It is also possible to export mesh data.
Mesh Export		To export a mesh defined by a data set to file.
Table Export		To export the contents of a table to file.
Plot 1D, 2D, or 3D Image Export		To export plot images from a 1D Plot Group, 2D Plot Group, and 3D Plot Group.
Player Export		To create interactive animations.
Plot Export		To export a plot from a 1D Plot Group, 2D Plot Group, or 3D Plot Group.

See Also

- Results Evaluation and Visualization
- Creating Reports

Animation Export

Use **Animation** ( ) to define and export a movie or series of images based on a plot group. Play the animation in a web browser or use it in presentations or on a website.

Important

Use Windows Media Player to play AVI files.

Tip

Use this feature, for example, to export multiple images for different time steps or eigenvalues.

1 In the **Model Builder**, right-click **Export** ( ) and select **Animation** ( ).

2 Under **Scene**, select a **Subject**. The **Subject** list contains the 1D, 2D, or 3D plot groups previously defined.

3 Under **Output**, select an **Output type**—**Image sequence** generates multiple image files; **Movie** generates a single file containing all the images. If **Movie** is selected, go to the next numbered step.

   If **Image sequence** is selected:

   - Enter a **Filename** including a path to save it to your computer.

     - OR -

   - Click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the computer desktop and enter a **Filename** in the **Export Image Sequence** dialog box then select a image file type from the **Save as type** list—**.png**, **.bmp**, or **.jpg**.

     The text entered in the **Filename** field is used for all the images generated. For example, if you enter `image`, select **.png** as the file type, and there are 11 frames in the movie, 11 files are created: `image01.png`, `image02.png`, …, `image11.png`.

4 If **Movie** is selected as the **Output type**, select a **Format** for the movie: **GIF**, **Flash**, or **AVI**.

   a For all movie formats, enter a path and include a **Filename**. Or click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the computer desktop and enter a **File name** in the **Export Animation** dialog box.

   b For all movie formats, enter a number of **Frames per second**.

   Enter information into the following fields based on the **Format**:

   - If **Flash** is selected, the **Interpolate between frames** check box is selected by default. Click to clear the check box if you do not want interpolation between the frames.
   - If **AVI** is selected, enter a **Quality**. This is a scalar value between 0 and 1. The default value is 0.75.

     AVI is a file format that can contain video encoded in different ways.
   - For **Flash** and **GIF**, select the **Open in browser** check box to launch the default web browser to view the output Flash or GIF file.

5 Under **Frame Settings**:

   - From the **Size** list, select **Manual** (the default) or **Current**, which uses the current size of the **Graphics** window. If you select **Current**, the **Record in reverse order** check box is the only additional setting.
   - Select the **Lock aspect ratio** check box to keep the original animation width and height.
   - In the **Width** and **Height** fields, enter the number of pixels (px) for the generated image size. The default value is 640-by-480 pixels.
   - If required, select the **Record in reverse order** check box.

6 Under **Parameter Sweep**, select a **Sweep Type**—**Result parameter**, **Dynamic data extension**, or **Stored solutions** (the default). Each time selected creates a frame in the movie or an individual image file. Go to Stored Solutions, Result Parameter, or Dynamic Data Extension for next steps.

Stored Solutions

Selecting **Stored solutions** (the default) is useful for animating time-dependent solutions or an animation across the eigenmodes for an eigenvalue/eigenfrequency solution or across the parametric solutions for a solution from a parametric sweep.

If **Stored solutions** is selected:

a Select **Stored output times** from the **Select via** list, and enter any vector of times in the **Times** field. For example, enter `range(0,0.01,1)` or `0 10^(range(-3,3))`.

b For time-dependent problems, select **Interpolated times** from the **Select via** list. Enter **Times** or click the **Range** button () to select and define specific times.

c For parametric sweep studies where the are multiple inner solutions (for example, a parameter sweep around a time-dependent solution), select an option from the **Solutions** list—**Inner solutions** or **Outer solutions**.

If **Outer solutions** is selected (that is, the parameters from the parametric sweep), the **Parameter values** list contains all options. Select as required, then select an option from the **Inner type** list—**First**, **Last** (the default), or **All**.

If **Inner solutions** is selected, select the **Parameter value** to animate, then the time steps. See Volume Integration, Surface Integration, and Line Integration Derived Values and Global Plots for more information about the inner and outer solutions.

Result Parameter

Selecting **Result parameter** is useful for animating the changes resulting from a sweep of the values for a defined global parameter (in **Global Definitions>Parameters**). Using a parameter you can, for example, animate a sweep of the position of a slice across the geometry in a slice plot.

If **Result parameter** is selected:

a Select the parameter from the **Parameter** list (or select **None**), which contains all global parameters, and define an interval for the parameter values using the **Start** and **Stop** edit fields.

b Enter the **Number of frames**. The number of frames divided by the number of frames per second gives the total playing time for a movie of a static or eigenvalue solution.

Dynamic Data Extension

Selecting **Dynamic data extension** is useful for animating the dynamics of an eigenmode in an eigenfrequency or eigenvalue solution. In such a dynamic data extension, the full harmonic cycle, which is the default, is the normal choice. You can also use it to animate a stationary solution even if there is no obvious interpretation of the animation.

If **Dynamic data extension** is selected:

a When animating static and eigenfrequency/eigenvalue solutions, select a **Cycle type**:

- **Full harmonic**—a full sine wave (the solution phase grows linearly from 0 to 360°)
- **Half harmonic**—half a sine wave (the solution phase grows linearly from 0 to 180°)
- **Linear**—a linear ramp ($\mathrm{Re}(e^{i\alpha})$, where α is the phase, grows linearly from 0 to 1)

The cycle starts from the angle specified in the **Solution at angle (phase)** edit field when defining a **Solution Data Set**.

b To visualize the time evolution enter the **Number of frames** to include in the animation.

For the **Model parameter** and **Dynamic data extension** sweep types you can specify the **Number of frames** to include in the animation.

Tip

The number of frames divided by the number of frames per second gives the total playing time for a movie of a static or eigenvalue solution.

7 Under **Layout**:

a By default, the **Title**, **Legend**, **Axes**, and **Logotype** (1D and 2D plots) or **Title**, **Legend**, **Grid**, **Axis orientation**, and **Logotype** (3D plots) are included. To edit the default, select the **Include** check box and click to clear one or several of the **Title**, **Legend**, **Axes**, and **Logotype** check boxes (1D and 2D) or the **Title**, **Legend**, **Grid**, **Axis orientation**, and **Logotype** check box (3D).

b Enter a **Font size** (pt).

c Select a **Background**—**Current** or **Color**. If you select **Color** (the default setting), click the **Color** button to select a **Custom** color background to replace the default, which is white. Select **Current** to use the background in the plot group, which is a blue gradient background for 3D plots and white for 2D and 1D plots.

8 Under **Advanced**, enter a **Resolution**. The default is 96 DPI (dots per inch).

9 Select the **Antialiasing** check box to reduce stairstep-like lines and to smooth lines and edges.

10 Click the **Export** button ( ) in the settings window or right-click the node and select **Export**.

The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Data Export

Use **Data** ( ) to export numerical data to file. Data export operates directly on data sets. You can use different types of evaluation points other than the ones in the data set (for example, a grid) and you can export the data on spreadsheet format or a sectionwise format.

See Also	For detailed information about the data formats uses in the exported data files, see Data Formats in the *COMSOL Multiphysics Reference Guide*.

EXPORTING DATA

1 In the **Model Builder**:

Right-click **Export** ( ) and select **Data** ( ).

- OR -

Right-click any data set node, for example, **Solution**, and select **Add Data to Export**. Click the **Data** node under **Export**.

Tip	The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

2 Under **Data** select a **Data set**.

The **Data set** list contains the 1D, 2D, or 3D data sets previously defined, for example, **Solution** or **Cut Line**.

a For **Parametric Sweep** studies select an option from the **Parameter value** list, which contains the parameter value combination for the outer solutions.

b Select an option from the **Select via** list —**Stored output times** or **Interpolated times** (time-dependent models only). If **Stored output times** is selected, the **Times** section is auto-filled with information from the **Data set** selected.

c If **Interpolated times** is selected, enter **Times** or click the **Range** button ( ) to select and define specific times.

3 Under **Expression** select predefined expressions based on the physics of the model or enter an **Expression** in the field. Enter a **Description** or edit the default description as required.

4 Under **Output**, enter a **Filename** including a path to save it to your computer.

- OR -

Click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the computer desktop and enter a **File name** in the **Export Data** window. All are saved as .txt files.

5 Under **Output**, select the **Points to evaluate in**: **From file**, **Grid**, **Regular grid**, or **Take from data set** (the default).

6 Select the **Points to evaluate in**. Depending on the selected type of points to evaluate in, various additional settings are available.

- To use coordinates from a data file, select **From file**. Then specify a **Coordinate filename** for a text file with the coordinates for the data output, or click **Browse** to locate the file.
- To use a grid to define the points to evaluate in, select **Grid** or **Regular grid**. Then specify the **Data format**—**Grid** or **Spreadsheet** (the default). Additionally, for the **Grid**, specify the **x**, **y**, and **z** coordinates for the grid points in the fields or using the **Range** buttons ( ), or, for the **Regular grid**, specify the **Number of x points**, **Number of y points**, and **Number of z points** for the regular grid in the fields (default: 10 points in each direction).
- To use the data points for the data in the data set, select **Take from data set**. Then specify the **Data format**—**Sectionwise** or **Spreadsheet** (the default). For spreadsheet data, select the **Transpose** check box to transpose the data from columns to rows. Additionally, select a **Space dimension**—**Take from data set** (the default), **Global**, **0**,

1, **2**, or **3**. Also select a **Geometry level**—**Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point** (some of these may not be applicable depending on the model's space dimension).

7 Under **Advanced**:

   a The **Include header** check box is selected by default. Click to clear the check box if you do not want to include a header.

   b The **Full precision** check box is selected by default and provides an output that contains all significant figures for data, which is stored as double-precision numbers. To limit the precision in the output to six significant figures, clear this check box.

   c Select the **Sort** check box to sort the data by increasing *x*, *y*, and *z* coordinates. Otherwise, the data is unsorted.

   d Select **Lagrange points** (the default) or **Gauss points** from the **Evaluate in** list to specify where COMSOL evaluates the data: in the Lagrange points (nodes of the Lagrange elements) or in the Gauss points for the Gaussian quadrature. The **Evaluate in** list is only available for Solution data sets.

   e Select a **Resolution**—**Finer**, **Fine**, **Normal** (default), or **Custom**. If **Custom** is selected, enter a **Lagrange-element node-point order** (default: 1). Use a higher order for a finer resolution.

   f Select a **Smoothing** method for smoothing of the data—**None**, **Internal** (the default, for smoothing within domains but not across interior boundaries), or **Everywhere**.

8 Click the **Export** button ( ) in the settings window or right-click the node and select **Export**.

   The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Mesh Export

Use the **Mesh** ( ) feature to export a mesh to file. Mesh export operates directly on data sets and exports the mesh in the frame specified by the data set.

You can also export the mesh data from data sets that contain meshes, for example, a **Mesh** data set or a **Solution** data set. Right-click the **Export** node and select **Mesh**. In the **Mesh** node's settings window, follow these steps to export mesh data.

1. In the **Data** section, select the mesh data set to export from the **Data set** list. The **Data set** list contains the solution and mesh data sets previously defined.
2. In the **Output** section, select a file type among the available formats in the **File type** list and enter a file name including the path in the **Filename** field (or click **Browse...** to specify the filename). You can export the mesh to a COMSOL Native file (binary `.mphbin` or text `.mphtxt`) or to an STL file (only if you select 3D data set).
3. Click the **Export** button () in the settings window or right-click the node and select **Export**.

   The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Table Export

Use the **Table** () feature to export the contents of a table to file. A table export stores the data from any of the tables in the model as a text file.

To export a table, right-click a **Table** node under **Tables** () and select **Add Table to Export**, or right-click the **Export** node () and select **Mesh** (). In the **Table** node's settings window, follow these steps to export table data:

1. In the **Table** section, select the table to export from the **Table** list. The **Table** list contains all tables in the model.
2. In the **Output** section, specify the path and filename for the table data file, or click **Browse** to specify the filename.
3. In the **Advanced** section, clear the **Full precision** check box if you want to export the table data using the table display precision instead.
4. Click the **Export** button () in the settings window or right-click the node and select **Export**.

   The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **Output** section.

Plot 1D, 2D, or 3D Image Export

Use the **1D Image**, **2D Image**, and **3D Image** ( ) features to export plot images as PNG, BMP, EPS, or JPEG image files.

Tip

Use the Animation Export feature, for example, to export multiple images for different time steps or eigenvalues.

1 In the **Model Builder**:

Right-click **Export** ( ) and select **1D image**, **2D Image**, or **3D Image** ( ).

- OR -

Right-click any **Plot Group** node, for example, **3D Plot Group** or **1D Plot Group,** and select **Add Image to Export**. Click the **Image** node under **Export**.

Tip

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

2 Under **Scene**:

a Select a **Subject** from the list.

The **Subject** list contains the 1D, 2D, or 3D plot groups previously defined. **1D Image** uses data from 1D plot groups, **2D Image** uses data from 2D plot groups, and **3D Image** uses data from 3D plot groups.

2D

3D

For 2D and 3D models, select a **View**—**From plot group** uses the plot group settings, or select a predefined **View** setting from the list.

On the toolbar, click the **Refresh** button ( ) to refresh the **Graphics** window with your settings. This is useful when you change the Plot group or View selection.

3 Under **Image**:

a From the **Size** list, select **Manual** to specify the image size manually using the settings below, or select **Current** to use the current size of the **Graphics** windows (in that case, only the **Antialiasing** setting is available).

b Select a **Unit** of dimension—**Pixels (px)**, **Millimeters (mm)**, or **Inches (in)**.

c Select the **Lock aspect ratio** check box to keep the original image width and height.

d In the **Width** and **Height** fields, enter the number of pixels, millimeters, or inches for the final image size.

e Enter a **Resolution**. The default is 96 DPI (dots per inch).

f Select the **Antialiasing** check box (selected by default) to reduce stairstep-like lines and to smooth lines and edges.

4 Under **File**:

a Choose a format for the image file from the **Format** list: PNG, BMP, JPEG, or (*1D only*) EPS. The default format is PNG.

b Enter a **Filename** including a path to save it to your computer.

- OR -

Click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the computer desktop and enter a **File name** in the **Export Image**.

The text entered in the **File name** field is used for all the images generated. For example, if you enter `image`, select **.png** as the file type, and there are 11 frames in the movie, 11 files are created: `image01.png`, `image02.png`, …, `image11.png`.

5 Under **Layout**:

a By default, the **Title**, **Legend**, **Axes**, and **Logotype** (1D and 2D images) or the **Title**, **Legend**, **Grid**, and **Logotype** (3D images) parts of the plot are included. To edit the default, select the **Include** check box and click to clear or select one or several of the **Title**, **Legend**, **Axes**, and **Logotype** check boxes (1D and 2D) or the **Title**, **Legend**, **Grid**, **Axis orientation**, and **Logotype** check boxes (3D).

b Enter a **Font size** (pt).

c Select a **Background**—**Current** or **Color**. If you select **Color** (the default setting), click the **Color** button to select a **Custom** color background to replace the default, which is white. Select **Current** to use the background in the plot group, which is a blue gradient background for 3D plots and white for 2D and 1D plots.

6 Click the **Export** button ( ) in the settings window or right-click the node and select **Export**.

The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **File** section.

Plot Export

Use **Plot** ( ) to export a plot from a **1D Plot Group**, **2D Plot Group**, or **3D Plot Group**.

1 In the **Model Builder**:

Right-click **Export** ( ) and select **Plot** ( ).

- OR -

Right-click any **Plot type** in any or all **Plot Groups**, for example, the **Surface** and **Streamline** plots in a 3D Plot Group, select **Add Plot Data to Export**. Click the **Plot** node under **Export**.

Tip

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

2 Under **Plot**:

a Select a **Plot group**. The **Plot group** list contains the 1D, 2D, or 3D plot groups previously defined.

b Select a **Plot** to export its data. **Plot Groups** can contain one or more individual plots.

3 Under **Output**, enter a **Filename** including a path to save it to your computer.

- OR -

Click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the computer desktop and enter a **File name** in the **Export Data** window. All are saved as .txt files.

4 Under **Output**, from the **Data format** list, select **Sectionwise** or **Spreadsheet**.

See Also

See The COMSOL File Formats in the *COMSOL Multiphysics Reference Guide* for information about these formats.

For streamline plots and particle tracing plots you can also control the amount of data to export. By default, the data contains full information about all points for all particles or streamlines. Select the **Only export start and end points** check box to only include one row with the start and end points for each particle or streamline.

Note

This check box only has an effect on particles plotted as lines because plots of particles as points do not contain full information about the particle trajectories.

5 Under **Advanced**, the following settings area available:
   - The **Include header** check box is selected by default. Click to clear the check box if required.
   - The **Full precision** check box is selected by default. To limit the precision in the output to seven digits.
   - Select the **Sort** check box to sort the data by increasing *x*, *y*, and *z* coordinates. Otherwise, the data is unsorted.

6 Click the **Export** button ( ) in the settings window or right-click the node and select **Export**.

   The file is exported to the location on your computer previously specified. The **Messages** window confirms where the files are exported as specified in the **File** section.

Player Export

A **Player** ( ) creates interactive animations directly in the COMSOL Desktop.

1 In the **Model Builder**, right-click **Export** ( ) and select **Player** ( ).

2 Under **Scene**, select a **Subject**. The **Subject** list contains the 1D, 2D, and 3D plot groups previously defined. Select one of these or select **None**.

3 Under **Parameter Sweep**, select a **Sweep Type**—**Result parameter**, **Dynamic data extension**, or **Stored solutions** (the default). Go to Stored Solutions, Result Parameter, or Dynamic Data Extension for next steps.

Note

Each time selected creates a frame in the movie or an individual image file.

Stored Solutions

Selecting **Stored solutions** (the default) is useful for animating time-dependent solutions or an animation across the eigenmodes for an eigenvalue/eigenfrequency solution or across the parametric solutions for a solution from a parametric sweep.

If **Stored solutions** is selected for a **2D Plot Group**:

a Select **Stored output times** from the **Select via** list, and select any time from the **Times** list.

b For time-dependent problems, select **Interpolated times** from the **Select via** list. **Times** are automatically added from the Stored solutions.

c For parametric sweep studies where the are multiple inner solutions (for example, a parameter sweep around a time-dependent solution), select an option from the **Solutions** list—**Inner solutions** or **Outer solutions**.

If **Outer solutions** is selected (that is, the parameters from the parametric sweep), the **Parameter values** list contains all options. Select as required, then select an option from the **Inner type** list—**First**, **Last** (the default), or **All**.

If **Inner solutions** is selected, select the **Parameter value** to animate, then the time steps. See Volume Integration, Surface Integration, and Line Integration Derived Values and Global Plots for more information about the inner and outer solutions.

Result Parameter

Selecting **Result parameter** is useful for animating the changes resulting from a sweep of the values for a defined global parameter (listed in **Global Definitions>Parameters**). Using a parameter you can, for example, animate a sweep of the position of a slice across the geometry in a slice plot.

If **Result parameter** is selected:

a Select the parameter from the **Parameter** list (or select **None**), which contains all global parameters, and define an interval for the parameter values using the **Start** and **Stop** edit fields.

b Enter the **Number of frames**. The number of frames divided by the number of frames per second gives the total playing time for a movie of a static or eigenvalue solution.

Dynamic Data Extension

Selecting **Dynamic data extension** is useful for animating the dynamics of an eigenmode in an eigenfrequency or eigenvalue solution. In such a dynamic data extension, the full harmonic cycle, which is the default, is the normal choice. You can also use it to animate a stationary solution even if there is no obvious interpretation of the animation.

If **Dynamic data extension** is selected:

a When animating static and eigenvalue solutions, select a **Cycle type**:

- **Full harmonic**—a full sine wave (the solution phase grows linearly from 0 to 360°)
- **Half harmonic**—half a sine wave (the solution phase grows linearly from 0 to 180°)
- **Linear**—a linear ramp ($\mathrm{Re}(e^{i\alpha})$, where α is the phase, grows linearly from 0 to 1)

The cycle starts from the angle specified in the **Solution at angle (phase)** field when defining a **Solution Data Set**.

b To visualize the time evolution of a time-harmonic model, enter the **Number of frames** to include in the animation.

The number of frames divided by the number of frames per second gives the total playing time for a movie of a static or eigenvalue solution.

4 Under **Frames**:

a If the selected sweep type is **Stored solution**, select **All** or **Number of frames** from the **Frame selection** list. Select **All** to play all solutions in the stored solution.

b Enter the **Number of frames** (if applicable). The default is 25 frames.

c Enter the **Frame number** or select it using the slider to get a preview of individual frames. Observe the geometry in the **Graphics** window to see the shown frame number.

5 Under **Playing** you can adjust some settings that affect the playing of the recorded plots:

a Specify how fast the player runs by typing the time (in seconds) to display each frame (default value: 0.1 s) in the **Display each frame for** field.

b Select the **Repeat** check box to repeat the playing instead of playing the plot sequence once.

6 Right-click the **Player** node and select **Generate Frame** (🎞) to create a series of frames (if more than one **Frame number** is selected).

7 Right-click the **Player** node and select **Play** (▷). Watch each **Frame number** cycle from beginning to end in the **Graphics** window. Use these buttons on the **Graphics** window to **Play** (▷), **Stop** (□), **Next** (⏭), and **Previous** (⏮), to cycle through the animation.

Creating Reports

In this section:

- About the Report Generator
- Report Feature Types
- Generating a Model Report
- Basic Report Components
- Model Report Components

About the Report Generator

The Report Generator is a tool for reporting and documenting models created in COMSOL Multiphysics. It creates a record of the entire model including all the settings made during the modeling process.

The report this utility generates is an overview of the model, and includes:

- Model properties
- Geometry
- Physics
- Mesh
- Studies
- Results and visualization

Several reports can be created for each model, and it is possible to configure each report by adding, moving, deleting, and disabling the nodes that define the report. The reports are stored with the model, so you can keep generating reports using the previously configured report contents and update the reports when the model changes.

These reports are easy to publish as electronic documents suitable for the Internet or as Microsoft® Word documents (.docx files). The utility generates the HTML report in a file format that makes it possible to customize the report in any HTML editor. A custom style sheet can be used to format the report.

You can use HTML tags to create hyperlinks and format the report output in text boxes, headings, and captions.

See Also

- Reports in the *COMSOL Multiphysics Reference Guide*
- Report Feature Types
- Generating a Model Report

Report Feature Types

For any model you can add one or more reports to the **Reports** branch and then generate model reports for documentation and information about a model. You can create reports using predefined templates that define different levels of detail in the report:

- **Brief Report**: contains an overview of the model with all results and plots but no details about the physics and variables.
- **Intermediate Report**: contains comprehensive information about the model, including the physics settings and variables but not complete information about the underlying equations, for example.
- **Complete Report**: contains all information about the model, including physics interface details such as weak equation expressions and shape functions. This report is suitable for troubleshooting, for example.
- **Custom Report**: contains an initially empty report, which you can configure using the availably report components.

Note

For all report types, the templates provide a starting point. It is possible to customize all reports by modifying, moving, adding, disabling, and deleting nodes in the reports. You can also switch the level of detail for an existing report, which affects the report nodes added afterward.

See Also

- Reports in the *COMSOL Multiphysics Reference Guide*
- About the Report Generator
- Generating a Model Report

Generating a Model Report

For a model for which you want to create a report, the following steps describe the general procedure:

1. Right-click **Reports** ( ) and choose a template that creates report nodes that describe the model with a suitable level of detail: **Brief Report**, **Intermediate Report**, or **Complete Report**. You can also choose **Custom Report** and **Document**.
2. In the main **Report** settings window, specify the output format (**HTML** or **Microsoft Word**), the location for the output file and the associated folder with images and style sheet information. You can also specify the style sheet to use for HTML reports and how to enumerate the sections in the report.
3. The top node is typically the **Title Page** node ( ). In its settings window you can define the title (defaults to the model's file name), an image to use at the top of the report, author and company information, and add a summary and acknowledgments if applicable.
4. Review the structure and contents of the report. You can add, move, edit, disable, and delete structural elements and report contents.

See Also

- Reports in the *COMSOL Multiphysics Reference Guide*
- Report Feature Types
- About the Report Generator

Basic Report Components

Right-click nodes to select and add these report nodes from the **Basic** submenu—**Bibliography**, **Code**, **Equation**, **Heading**, **Image**, **List**, **Note**, **Table**, and **Text**.

See Also

For all details about these features, see Basic Report Components in the *COMSOL Multiphysics Reference Guide*.

Model Report Components

Right-click nodes to select items from the **Model** submenu. This group of report components provide information about the model are detailed in these sections as described in the *COMSOL Multiphysics Reference Guide*:

- Definitions Report Nodes
- Results Report Nodes
- Geometry Report Node
- Material Report Node
- Mesh Report Node
- Model Report Node
- Parameters Report Node
- Physics Interface Report Node
- Root Report Node
- Solver Report Node
- Study Report Node

❗ *Important*	The links to the features described in the *COMSOL Multiphysics Reference Guide* do not work in the PDF, only from the online help in COMSOL Multiphysics.

💡 *Tip*	To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

22

Running the COMSOL Software

This chapter provides an overview of the different ways that you can run the COMSOL software in addition to the COMSOL Desktop graphical user interface.

In this chapter:

Running COMSOL

The primary way to access COMSOL's functionality is through the COMSOL Desktop. This section describes alternative means of accessing COMSOL's functionality. In this section:

- COMSOL Client/Server
- Parallel COMSOL
- COMSOL API
- COMSOL Batch
- LiveLink for MATLAB

❗ *Important*	The links to features described in other guides does not work in the PDF, only from the online help in COMSOL Multiphysics.

💡 *Tip*	To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

COMSOL Client/Server

COMSOL's client/server architecture lets you access the COMSOL server—COMSOL's computational engine—as a separate process. For example, the COMSOL Desktop can act as a COMSOL client when connected to a COMSOL server.

You must have a floating network license (FNL) to run the COMSOL server and the COMSOL client on separate computers. Any valid COMSOL license is sufficient to run the client and the server run on the same computer.

The client and server need not run on the same platform. For example, you can run the COMSOL Desktop on Windows connecting to a COMSOL server on a Linux or Mac server. In this way you can interactively access a more powerful remote computer.

To start the COMSOL server under windows, just click **COMSOL Multiphysics Server** in the **Client Server** folder under your COMSOL installation on the start menu. On Linux and Mac, type the command `comsol server` to start the COMSOL server. For more options for starting the COMSOL server, see the *COMSOL Installation and Operations Guide* for how to launch a COMSOL server on the different platforms. The section “The COMSOL Command” for your platform describes all options for launching the COMSOL server.

You can connect to a COMSOL server from the COMSOL Desktop by selecting **Connect to Server** on the **File** menu in the COMSOL Desktop. Disconnect from the sever by selecting **Disconnect from Server**.

You can also export and import a model from the COMSOL Desktop to a running COMSOL server and vice versa. This functionality is primarily intended for the LiveLink for MATLAB. Select **Export Model to Server** or **Import Model from Server** on the **File** menu to transfer a model between the COMSOL Desktop and the LiveLink for MATLAB.

Parallel COMSOL

COMSOL supports two mutual modes of parallel operation: shared-memory parallel operations and distributed-memory parallel operations, including cluster support.

SHARED MEMORY PARALLEL MODE

The shared memory parallel mode is suitable for running COMSOL on modern multicore or multiprocessor computers.

This parallel mode of operation is available for all platforms and all license types.

By default, COMSOL uses the shared memory parallel mode and allocates all cores on the computer.

To control the number of core used and other options, see the section *The COMSOL Command* for your platform.

DISTRIBUTED MEMORY PARALLEL MODE

This parallel mode lets you run COMSOL on a Windows HPC cluster or a Linux cluster.

Note

See Running COMSOL in Parallel for details about running COMSOL in parallel architectures, including Windows and Linux clusters.

For more options on how to run COMSOL on a cluster from the command line, see the *COMSOL Installation and Operations Guide*. Click The COMSOL Commands for a Windows platform, The COMSOL Commands for a Linux platform, or The COMSOL Commands for a Mac platform.

Important

The links to features described in other guides does not work in the PDF, only from the online help in COMSOL Multiphysics.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

BASIC CLUSTER CONCEPTS

The following terms occur frequently when describing the hardware for cluster computing and shared memory parallel computing:

- Compute node: The *compute nodes* are where the distributed computing occurs. The COMSOL server resides in a compute node and communicates with other compute nodes using MPI (message-passing interface).
- Host: The *host* is a hardware physical machine with a network adapter and unique network address. The host is part of the cluster. It is sometimes referred to as a *physical node*.
- Core: The *core* is a processor core used in shared-memory parallelism by a computational node with multiple processors.

The number of used hosts and the number of computational nodes are usually the same. For some special problem types, like very small problems with many parameters, it might be beneficial to use more than one computational node on one host.

COMSOL API

The COMSOL API is a Java-based programming interface for COMSOL functionality. The COMSOL Java API can be used for developing standalone applications based on COMSOL functionality.

The most basic use of the COMSOL Java API is to run a Model Java-file from the COMSOL Desktop or with the COMSOL batch command.

To run a Model Java-file from the COMSOL Desktop, compile it using the COMSOL compile command. This command is called `comsolcompile` on Windows and `comsol compile` on other platforms. The compilation gets you a model class file corresponding to the model Java-file. Launch the model class file by selecting **Open** on the **File** menu, and selecting a **Model Class File** under **File name**.

To create a standalone application using the COMSOL Java API you need to develop a text based or GUI-based interface to the functionality and compile it using the COMSOL compile command. The application can be run in *standalone mode* that links your Java application directly to the COMSOL Java code (as a single process). You can also choose to run the application in client/server mode by connecting to a COMSOL server.

The COMSOL API is described in the *COMSOL Java API Reference Guide*.

See the *COMSOL Multiphysics Installation and Operation Guide* for how to launch the COMSOL compile command on the different platforms. The sections the COMSOL Command for the respective platforms describe how the COMSOL server can be launched.

COMSOL Batch

The COMSOL batch mode makes it possible to run COMSOL without a graphical user interface. The COMSOL batch mode of operations allows you to run both Model Class-files and Model MPH-files.

You can control the options for running COMSOL in batch mode from the **Study** node in the **Model Builder**. To enable the Batch feature, click the **Show** button ( ) and select **Advanced Study Options**. Then in the **Model Builder**, right-click a **Study** node and select **Batch**. Also see Batch in the *COMSOL Multiphysics Reference Guide*.

You can also run COMSOL batch entirely from a command prompt. See the *COMSOL Installation and Operations Guide* for how to launch the COMSOL batch command on the different platforms. The sections *The COMSOL Command* for the respective platforms describe how to launch the COMSOL server.

In batch mode you can monitor the memory usage reported in the log as lines of the form

```
Memory: RAM/MAXRAM VIRT/MAXVIRT
```

where `RAM` is the current memory usage in MB, and `VIRT` is the current virtual memory usage in MB. The maximum measured usage is reported in `MAXRAM` and `MAXVIRT`, respectively. The log only reports changes to the memory usage. You can also monitor the current progress, which is reported as lines in the log of the form

```
Current Progress: 53%
```

where the percentage indicates the currently estimated progress.

LiveLink for MATLAB

The LiveLink for MATLAB provides access to a COMSOL Model from MATLAB. On the MATLAB prompt you access the COMSOL model through a client/server connection to a running COMSOL server. You access the model through the COMSOL Java API and the MATLAB Java interface. In addition, there are M-file wrapper functions that help you perform tasks such as displaying graphics using MATLAB figure windows or fetching data from the model object.

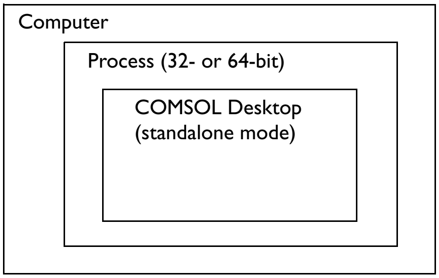
COMSOL Client/Server Architecture

In this section:

- Standalone COMSOL
- Running COMSOL as a Client/Server
- Running COMSOL with MATLAB

Standalone COMSOL

The most straightforward way of running COMSOL is as a standalone application.



Running COMSOL as a Client/Server

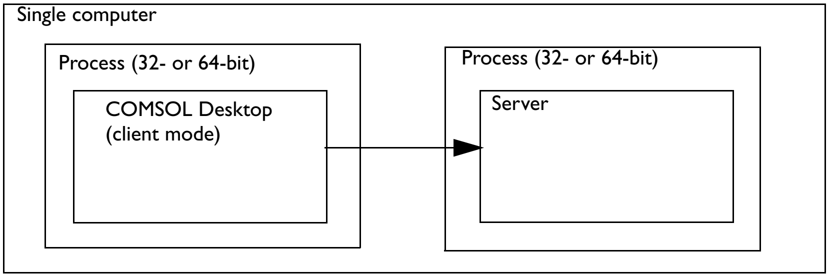
The COMSOL client and server applications are available on all platforms.

Note

The license server is not the same as a COMSOL server. The license manager can run on a computer different from both the ones used by COMSOL Desktop and COMSOL server.

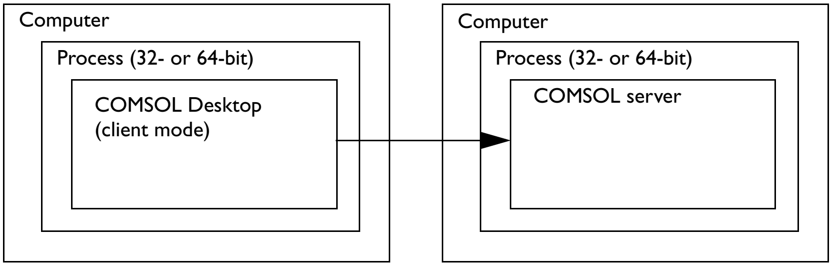
RUNNING COMSOL MULTIPHYSICS CLIENT/SERVER ON THE SAME COMPUTER

Both the COMSOL client and the COMSOL server can run on the same computer and with all available license types: named user license (NSL), CPU locked license (CPU), and floating network license (FNL).



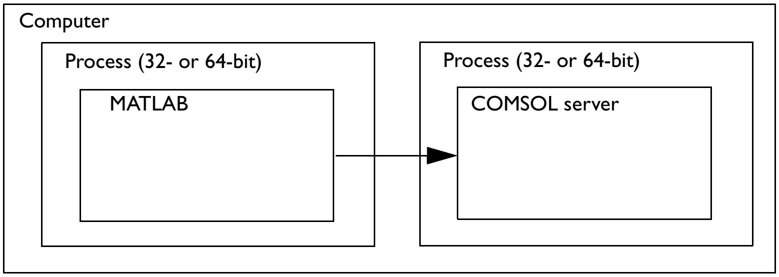
RUNNING COMSOL CLIENT/SERVER ON DIFFERENT COMPUTERS

The COMSOL client and COMSOL Multiphysics server can also run on different computers, but this configuration requires a floating network license (FNL).



Running COMSOL with MATLAB

COMSOL can run together with MATLAB on the same machine using COMSOL’s client/server architecture. The command `comsol server matlab` launches this configuration. The command `comsol matlab` can be use to launch just the client and to connect to a remote server.



RUNNING A THIN CLIENT

When running inside MATLAB the COMSOL client runs in a so-called thin client mode. This means that few COMSOL shared libraries are loaded. Thus all operations that require shared libraries are performed on the COMSOL server.

Running COMSOL Client/Server

The COMSOL Desktop can run in a separate process as a client to a COMSOL server. The COMSOL Desktop client uses a TCP/IP connection to connect to the COMSOL server. The client and server need not run on the same platform. You must have a floating network license (FNL) to run the COMSOL server and the COMSOL client on separate computers.

You can also use the COMSOL client/server when running COMSOL with MATLAB. To do so, start a COMSOL client on a separate computer and connect to a COMSOL server started from within a MATLAB process. In this section:

- Advantages of Using COMSOL Client/Server
- Running COMSOL Client/Server

Advantages of Using COMSOL Client/Server

The COMSOL client/server configuration frees your desktop computer of lengthy computations, dispatching your jobs to a dedicated computer. The computer that runs the COMSOL server could have more memory and a faster CPU than your desktop computer.

Tip

Running the COMSOL server and the COMSOL client separately on the same computer increases the total memory available to solve problems. This is particularly interesting because the 32-bit limit on addressable memory can be the limiting factor for complex models. The COMSOL server components do not use the memory required for the graphical user interface, freeing that memory for the actual computations on the server.

Running COMSOL Client/Server

STARTING A COMSOL SERVER

When you have access to the Windows desktop, start the COMSOL server from the **Start** menu. Go to **All Programs**, select **COMSOL 4.3** and then **Client Server**, and select **COMSOL Multiphysics 4.3 Server**. If starting the COMSOL server from a terminal window in Windows, use the command `<COMSOL installation directory>\bin\win32\comsolserver.exe` for the

32-bit version or
`<COMSOL installation directory>\bin\win64\comsolserver.exe` for the 64-bit version.

On Linux, use the `comsol server` command to start a COMSOL server.

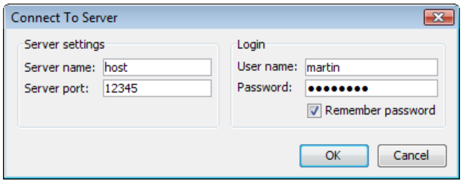
On Mac OS X, use the **COMSOL Multiphysics Server** application, or if you connect to Mac OS X from another computer, use the `comsol server` command in the terminal window.

INITIALIZING THE COMSOL SERVER

The first time you start a COMSOL server on a computer, you are asked for a user name and password. By default, your user name and a hashed password is stored on your computer's hard drive. You can avoid storing user name and password on disk by providing the `-passwd nostore` target option to the COMSOL server command. When the COMSOL server is started, the server displays the port number. The server also displays a message each time you log in from a client.

CONNECTING THE DESKTOP TO A SERVER

From the COMSOL Desktop select **Connect to Server** on the **File** menu. In the **Connect to Server** dialog box, specify the hostname of the server, the TCP/IP port number of the server (the port number of displayed by the server).



Connect To Server

Server settings

Server name: host

Server port: 12345

Login

User name: martin

Password: ●●●●●●●●

☑ Remember password

The user name and password are the ones you used when starting the server, and the port number is printed by the server when it is started.

DISCONNECTING THE DESKTOP FROM A SERVER

You can close the connection to the server or MATLAB using the menu item **Disconnect from Server**. Doing so transfers all data from the server to the client and makes it a standalone version of COMSOL Multiphysics. Unless the server was started with the option `-multi on` it exits.

CONNECTING TO A SERVER FROM MATLAB

From MATLAB you can use the commands `Modelutil.connect` and `Modelutil.disconnect` to connect and disconnect from a COMSOL server. The connection to the server is necessary to access and manipulate a model.

See Also

COMSOL Java API Reference Guide

DISCONNECTING MATLAB FROM A SERVER

To connect to a COMSOL server, use the menu item **Connect to Server**. This opens a dialog box where you can enter the server and login information

EXPORTING TO A COMSOL SERVER

Primarily for use with the LiveLink for MATLAB, use **Export Model to Server** to transfer the model in the COMSOL Desktop to a MATLAB session. This model transfer works while another client is connected to the server. You have this situation when running the LiveLink for MATLAB.

IMPORTING FROM A COMSOL SERVER

Primarily for use with the LiveLink for MATLAB, use **Import Model from Server** to transfer the model from a MATLAB session to the COMSOL Desktop. This model transfer works while another client is connected to the server. You have this situation when running the LiveLink for MATLAB.

Running COMSOL in Parallel

COMSOL supports two mutual modes of parallel operation. One mode is based on the distributed memory model and runs on several nodes on a Linux or Windows cluster; see Distributed-Memory Parallel COMSOL. The other mode is based on the parallel shared memory model. This section includes these topics:

- Shared-Memory Parallel COMSOL
- COMSOL and BLAS
- Distributed-Memory Parallel COMSOL
- Benefits of Running COMSOL in a Distributed Mode
- Running COMSOL in Parallel on Clusters

Shared-Memory Parallel COMSOL

Most multiprocessor machines and dual-core/multicore machines support the shared memory model; however, it is not supported by several nodes on a cluster. The solvers, assembly, and meshing in COMSOL Multiphysics benefit from shared memory parallelism. By default COMSOL uses all cores available on the machine for shared-memory parallelism.

BENEFITS OF RUNNING COMSOL SHARED-MEMORY PARALLEL

All iterative solvers and smoothers except Incomplete LU are parallelized. Some smoothers have blocked versions. The blocked versions are usually more parallel than the nonblocked versions. The finite element assembly also runs in parallel. Usually the speedup depends on the problem size; problems using a lot of memory usually have better speedup.

The PARDISO sparse direct linear solver runs in parallel. The SPOOLES sparse direct linear solver also runs in parallel. The MUMPS direct solver benefits from shared memory parallelism; however, to a slightly lesser extent than PARDISO and SPOOLES.

The free mesher in 3D runs in parallel over the faces and domains of the geometry object being meshed. For this reason, the speedup when running on several processors depends strongly on the domain partitioning of the corresponding geometry. Meshing a geometry with only one domain, such as an imported CAD part, gives almost no speedup at all. On the other hand, meshing a geometry with several domains, such as an imported CAD assembly with many parts, can give significant speedup, especially if the number of elements in the mesh is large.

3D

These plots run in parallel in 3D: slice plots, isosurface plots, volume plots, line plots, deformed-shape plots, and streamline plots.

2D

These plots run in parallel in 2D: surface plots, contour plots, and line plots as well as deformed-shape plots run in parallel.

A significant part of the parallel speedup in computations comes from functions of the type BLAS (basic linear algebra subprogram; see the next section). If you want to run the software in parallel, it is important that the BLAS library you use supports parallelism. The BLAS libraries shipped with COMSOL do that.

Running in parallel usually requires extra memory. If you run out of memory, try to lower the number of used cores as explained in the Installation and Operations Guide. The speedup depends on the processor load. For instance, if your system has m processors and n of them are used by other active programs, do not set the number of processors to a number that is greater than $m - n$. The reason is that the programs compete for the same resources, which slows all of them considerably.

COMSOL and BLAS

BLAS is a set of functions for basic linear algebra operations. Vendors often supply BLAS libraries optimized for their hardware. A large portion of the computational engine in COMSOL relies on BLAS. Included with COMSOL are the BLAS libraries ACML (AMD Core Math Library) optimized for AMD processors with SSE2 support, MKL (Math Kernel Library) optimized for Intel processors. You can also supply your own BLAS library optimized for your hardware. By default COMSOL automatically tries to detect an appropriate BLAS library. The defaults in COMSOL are:

- MKL on Intel processors.
- ACML on AMD processors that support SSE2 instructions; otherwise MKL is used.

You can override the default as explained in the *COMSOL Installation and Operations Guide*. If the library you want to use is unavailable or incorrectly installed, COMSOL switches back to the default library.

Distributed-Memory Parallel COMSOL

The Linux and Windows versions of COMSOL support a distributed memory mode. The distributed mode starts a number of computational nodes set by the user. Each computational node is a separate process running a COMSOL instance. A computational node is not the same as a physical node (computer), but they can coincide. When running in distributed mode, COMSOL uses MPI for communicating between the processes in the distributed environment.

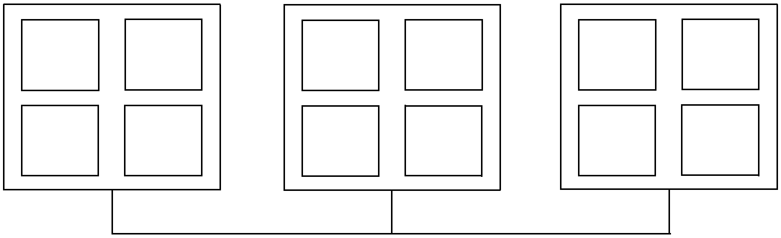


Figure 22-1: Schematic of a cluster with 3 physical nodes (computers) with 4 processors each.

The distributed-memory mode can be combined with COMSOL's ability to benefit from the shared-memory model. All modes that COMSOL can run in are able to use distributed memory mode.

Note

In client/server mode, the computer or cluster acting as server must be accessible from the client through a TCP/IP connection. If you are not able to connect to the server, you can use the COMSOL Batch to solve models on the cluster or do parametric sweeps. You can also use the Cluster Computing study to set up a COMSOL Batch job from within the GUI.

For the schematic in Figure 22-1, you can choose any number of computational nodes between 1 and 12. Each node, in turn, can use between 4 and 1 processors for shared memory. By default, COMSOL uses as many processors as are available on each physical node for shared-memory parallelism on Windows. This is suboptimal if the number of computational nodes is not the same as the number of physical nodes. It is recommended that you explicitly set the number of processors. For the schematic example, if you run 6 computational nodes, the optimal value for number of processors is 2. The number of processors used is $6 \cdot 2 = 12$.

For the same example, assuming you are the sole user of the system for the duration of the computation and that your problem requires a lot of memory, use 3 computational nodes with 4 shared memory cores each. If, on the other hand, your problem is small, use 12 computational nodes with 1 shared memory core each. This way you make the best use of shared-memory and distributed-memory parallelism for each problem.

You do not need a cluster to benefit from COMSOL's ability to utilize the distributed-memory model. On a multiprocessor computer you can use multiple computational nodes. This can be useful for small-sized parameter sweeps, for example. Make sure that the number of computational nodes times the number of processors does not exceed the number of available processors; otherwise performance deteriorates significantly.

Benefits of Running COMSOL in a Distributed Mode

The following direct solvers are supported by COMSOL when running in distributed mode:

- MUMPS
- SPOOLES

PARDISO is not supported in distributed mode. MUMPS is used instead.

The following iterative solvers are supported:

- Iterative solvers: BiCGStab, CG, GMRES, and FGMRES
- Smoothers and preconditioners: SOR, SOR Gauge, SOR Line, SOR Vector, and Vanka
- Multigrid

COMSOL can also run parameter sweeps using the distributed mode. The simplest way to start a distributed parameter sweep is to select the **Distribute parametric sweep** check box in the **Cluster Computing** node's settings window. The simplest way to modify an existing model is to add the **Cluster Computing** study and select **Compute** in the study node's settings window. To enable the cluster computing feature, click the **Show** button ( ) and select **Advanced Study Options**. Then in the **Model Builder**, right-click a **Study** node and select **Cluster Computing** ( ).

Running COMSOL in Parallel on Clusters

You control the options for running COMSOL on a cluster from the **Study** node in the **Model Builder**. To enable the cluster computing feature, click the **Show** button ( ) and select **Advanced Study Options**. Then in the **Model Builder**, right-click a **Study** node and select **Cluster Computing** ( ).

Note

You must have a floating network license (FNL) to run COMSOL in distributed memory parallel mode. See Cluster License Handling for more information.

See Also

Cluster Computing in the *COMSOL Multiphysics Reference Guide*

Model

The Micromixer—Cluster Version and Joule Heating of a Microactuator—Distributed Parameter Version models show how to set up a model for running COMSOL in parallel on a cluster: in the first case for faster solution of a large fluid-flow model using distributed solver jobs, and in the second case for a distributed parametric sweep.

Important

The links to the features described in other guides does not work in the PDF, only from the online help in COMSOL Multiphysics.

Tip

To locate and search all the documentation, in COMSOL, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

The following sections describe how to run cluster jobs on Windows and Linux.

Running a Cluster Job on Windows

This section outlines the main steps for running a cluster job on Windows. Before you start, check that the installation of COMSOL follows these guidelines:

- Make sure that the COMSOL installation directory is shared between all the compute nodes and the head node on a shared network disk.
- Make sure that the license manager is available and up and running.
- If you work on a desktop PC, which is recommended, install COMSOL on that local PC. Also install Windows HPC Pack on the desktop PC before you start. Windows HPC Pack makes it possible to access the cluster from your workstations. It is free and ships with the Microsoft HPC Server 2008 (HPCS 2008). An alternative is to log in to the cluster via Remote Desktop, for example.

To run a cluster job, follow these steps:

1 Start COMSOL Multiphysics.

2 In a complete model, right-click the **Study** node and select **Cluster Computing** ( ).

3 You can make a distributed parametric sweep to take advantage of the cluster with any solver settings. If you want to use a distributed solver job for other models without a parametric sweep, make sure that you enable a direct solver for the active solver node under **Solver Configurations**. To do so, right-click the **Direct** node ( ) and select **Enable**. The MUMPS and SPOOLES direct solvers support distributed solver jobs.

Tip

If the **Solver Configurations** node does not exist, right-click the **Study** node and select **Show Default Solver** ( ).

4 In the **Cluster Computing** node's settings window, select **HPCS 2008** from the **Cluster type** list. This provides access to all parameters that you need for communication with the cluster.

5 To submit the job, right-click the **Study** node ( ) and select **Compute** ( ).

6 You can define more details in the settings window for the **Cluster Computing** node ( ).under **Job Configurations** ( ). When you submit a job, COMSOL Multiphysics adds a **Cluster Computing** node. If you want to change or inspect its settings before submitting the first job, right-click the **Study** node ( ) and select **Show Default Solver** ( ).

7 After submitting the job to the cluster, you can monitor the progress in the **Progress** window ( ) and the **Log** window ( ). The **Progress** window shows the progress of the batch data and external processes, and the **Log** window contains a log with information about the solver operations for each parameter in a parametric sweep, for example. You can also get details about a cluster job in the Windows Job Manager, which is available in the HPC Pack.

RUNNING A CLUSTER JOB ON LINUX

Before you begin, make sure that the license manager is up and running and reachable from all compute nodes and the headnode. Skip the steps 1) and 3) if you are running COMSOL on the machine from where you wish to start the cluster job.

1 Start the COMSOL server on the Linux system with the command `comsol server`. Notice the port number that is displayed (for example, *COMSOL 4.3 started listening on port 2036*).

2 Start COMSOL Multiphysics on your desktop computer.

3 Select **File>Connect to Server**. In the **Connect to Server** dialog box, use the login credentials that you entered at the startup of the COMSOL server.

4 In a complete model, right-click the **Study** node and select **Cluster Computing** ( ).

5 You can make a distributed parametric sweep to take advantage of the cluster with any solver settings. If you want to use a distributed solver job for other models without a parametric sweep, make sure that you enable a direct solver for the active solver node under **Solver Configurations**. To do so, right-click the **Direct** node ( ) and select **Enable**. The MUMPS and SPOOLES direct solvers support distributed solver jobs.

Tip

If the **Solver Configurations** node does not exist, right-click the **Study** node and select **Show Default Solver** ( ).

6 In the **Cluster Computing** node's settings window, select **General** from the **Cluster type** list for Linux clusters. This provides access to all parameters that you need for communication with the cluster.

7 To submit the job, right-click the **Study** node ( ) and select **Compute** ( ).

8 You can define more details in the settings window for the **Cluster Computing** node ( ) under **Job Configurations** ( ). When you submit a job, COMSOL Multiphysics adds a **Cluster Computing** node. If you want to change or inspect its

settings before submitting the first job, right-click the **Study** node ( ) and select **Show Default Solver** ( ).

9 After submitting the job to the cluster, you can monitor the progress in the **Progress** window ( ) and the **Log** window ( ). The **Progress** window shows the progress of the batch data and external processes, and the **Log** window contains a log with information about the solver operations for each parameter in a parametric sweep, for example.

You can do the same cluster simulation from the command line using, for example, a scheduler script. Some example commands:

```
comsol -nn 2 mpd boot -f machinefile -v -d
comsol -nn 2 batch -inputfile comsoltest.mph -outfile output.mph
-batchlog b.log
comsol mpd allexit
```

CLUSTER LICENSE HANDLING

To run COMSOL Multiphysics simulations in distributed memory parallel mode (on a cluster), you must have a floating network license (FNL). Look for the keyword CLUSTERNODE in your license file. When running a cluster job, COMSOL uses the following license components and license check-out procedures:

- On the headnode, one seat of the COMSOL and COMSOLGUI features each are checked out.
- On each of the compute nodes, only one CLUSTERNODE feature is checked out, and it is not counted. This means that you have unlimited number of cluster nodes available for every seat (job) of the floating network license.
- When running a batch job through a scheduler, COMSOL's license manager checks out the noncluster COMSOL license keys (COMSOL, COMSOLGUI, CADIMPORT, CHEM, and so on) from one of the distributed processes. All other processes in the batch job only check out a CLUSTERNODE license key. So, license keys can be checked out from any physical node in the cluster depending on where the scheduler starts the processes.

23

Glossary

This Glossary of Terms contains terms related to finite element modeling, mathematics, geometry, and CAD as they relate to the COMSOL Multiphysics software and documentation. For more application-specific terms, see the glossaries in the documentation for most of the add-on modules. For references to further information about a term, see the index.

Glossary of Terms

adaptive mesh refinement A method of improving solution accuracy by adapting the mesh to the problem's physical behavior.

affine transformations Geometric transformations that are combinations of linear transformations and translations.

algebraic multigrid (AMG) An *algebraic multigrid* solver or preconditioner performs one or more cycles of a multigrid method using a coarsening of the discretization based on the coefficient matrix. Compare to *geometric multigrid (GMG)*.

anisotropy Variation of material properties with direction.

application programming interface (API) An *API* provides a set of documented functions and methods for interacting with a software product.

arbitrary Lagrangian-Eulerian formulation (ALE formulation) A formulation for a moving mesh where dependent variables represent the mesh displacement or mesh velocity. The COMSOL Multiphysics solvers have built-in support for the mesh movement.

arc A segment of the circumference of a circle or ellipse.

Argyris element A 2D, 6-node triangular finite element with a 5th-order *basis function* providing continuous derivatives between elements.

aspect ratio The ratio between the longest and shortest element or geometry dimension.

assemble Taking the local element stiffnesses, masses, loads, and constraints to form the *stiffness matrix*, *mass matrix*, load vector, constraint matrix, and constraint residual vector.

associative geometry An algorithm that maps data associated with a geometry to the new *geometric entities* when the geometry is modified.

backward differentiation formula (BDF) A multistep formula based on numerical differentiation for solutions to *ordinary differential equations*. A BDF method of order n computes the solution using an nth-grade polynomial in terms of backward differences.

basis function A function φ_i in the *finite element space* such that the ith degree freedom is 1, while all other degrees of freedom are 0. For the Lagrange finite element space, φ_i is a linear or higher order polynomial on each mesh element with value 1 in node i and 0 in all other nodes.

Bernstein polynomial See *Bézier basis.*

Bézier basis A set of polynomial functions that occur in the definition of a *Bézier curve*. These polynomial functions are often called *Bernstein polynomials.*

Bézier curve A *rational Bézier curve* is a parameterized *curve* formed as the quotient of two polynomials expressed in the Bézier basis. It is a vector-valued function of one variable. The coefficients of a rational Bézier curve are geometrically interpreted as *control points* and *control weights.* A *nonrational Bézier curve* is a rational Bézier curve with all weights equal, thereby making the denominator polynomial equal to a constant. A nonrational Bézier curve is also called an *integer Bézier curve*.

Bézier patch, Bézier surface A *Bézier patch* or *Bézier surface* is a surface extension of the *Bézier curve*. The *Bézier patch* is a function of two variables with an array of control points.

bidirectional constraint See *constraint.*

Boolean operations Boolean operations are used to construct a *geometry object* from other solid geometry objects and rebuild it in a new form. At least two primary geometry objects are required to create a resultant new geometry object. That new object depends on the type of Boolean operation:

- Union (add): the resultant geometry object occupies all the space of the initial geometry objects
- Difference (subtract): the resultant geometry object occupies all the space of the first geometry object except for the space inside the second geometry object.
- Intersection: the resultant geometry object occupies only the space common to the initial geometry objects

boundary A *geometric entity* with a space dimension one less than the space dimension for the geometry (for example, a *face* in a 3D geometry). In a mathematical context, the symbol ∂Ω represents the boundary of the domain Ω. Sometimes *boundary* is used in a narrower sense meaning an *exterior boundary.* See also *interior boundary*, *exterior boundary.*

border The interface between two parts in an *assembly.*

boundary modeling A geometry modeling method to create a geometry by defining its boundaries. Compare to *solid modeling* and *surface modeling*.

brick element See *hexahedral element*.

chamfer A CAD feature that trims off a corner with a plane or straight line.

Cholesky factorization A memory-saving version of *LU factorization* where U is the transpose of L. It requires that the coefficient matrix A ($A = LU$) be a symmetric positive definite matrix. See also *LU factorization* and *positive definiteness*.

coefficient form PDE A PDE in the coefficient form is a PDE formulation suited for linear PDEs

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot(-c\nabla u - \alpha u + \gamma) + \beta\cdot\nabla u + au = f & \text{in } \Omega \\ \mathbf{n}\cdot(c\nabla u + \alpha u - \gamma) + qu = g - h^T\mu & \text{on } \partial\Omega \\ hu = r & \text{on } \partial\Omega \end{cases}$$

composite geometry object, composite solid object Geometric objects made up by combining *primitive geometry objects* and other composite objects. See also *constructive solid geometry*, *primitive geometry object*, and *Boolean operations*.

COMSOL Desktop The integrated simulation environment for the COMSOL Multiphysics products with a number of windows such as the Model Builder window, the Graphics window, and each feature node's settings window.

COMSOL Multiphysics binary file A binary data file with the extension `.mphbin` that contains geometry objects or mesh objects. Earlier versions of COMSOL Multiphysics used the file extension `.flb`.

COMSOL Multiphysics text file A text data file with the extension `.mphtxt` that contains geometry objects or mesh objects.

condition number A measure of the possible error in a solution due to ill-conditioning of the equations. See also *ill-conditioning*.

constant A named model property that has a constant numeric value.

constraint Restriction imposed upon the dependent variables, typically as a *Dirichlet boundary condition*. *Neumann boundary conditions* are not regarded as constraints. When Dirichlet boundary conditions are introduced, the finite element algorithm makes a corresponding change to the Neumann boundary conditions so that the resulting model becomes solvable. For a *bidirectional constraint*, COMSOL Multiphysics accomplishes this change by adding the transpose of the constraint matrix h times a vector of *Lagrange multipliers* to the right-hand side of the Neumann boundary condition. For a *unidirectional constraint*, the extra term is often some matrix times the vector of Lagrange multipliers. In a mechanical model, the extra term is called a *constraint force*.

constructive solid geometry (CSG) A solid-modeling method that combines simple solid shapes, or *primitives*, to build more complex models using Boolean operations. See also *solid modeling* and *primitive*.

control point *Bézier* and *NURBS* curves and surfaces are defined by a set of points known as *control points*. The locations of the these points control the curve's shape.

control weight Scalar values assigned to *control points* to further control the shape of a curve or surface.

contour plot A plot that shows the variation of a solution component or other quantity. Points with equal values of the plotted quantity are connected with contour lines.

convergence The tendency for a finite element solution to approach the exact solution within well-defined and specified tolerances, for example, by reducing the mesh element size or the time step.

coupling operator An operator used to couple data within a model (geometry) or between different models (geometries). See also *extrusion coupling operator*, *projection coupling operator*, and *integration coupling operator*. Coupling variables provided similar functionality in earlier version of COMSOL Multiphysics, but coupling operators can be reused with different arguments (for example, for integrating different quantities over the same domain).

curl element See *vector element*.

curve The path of a point moving through space. See also *Bézier curve*, *NURBS*, and *manifold*.

curve object A geometry object consisting of only *edges* and *vertices*, for example a geometry object representing a *curve*.

curve segment An individual polynomial or rational polynomial curve. Compounded curves consist of several *curve segments*.

degree of freedom (DOF) One of the unknowns in a discretized finite element model. A degree of freedom is defined by a name and a *node point*. The degree of freedom names often coincide with the names of the dependent variables. The local degrees of freedom are all degrees of freedom whose node points are in one mesh element.

deformed geometry A geometry where the shape changes with a moving-mesh algorithm. It also the name of a *physics interface* for modeling deforming geometries. This is similar to the Parameterized Geometry physics interface in earlier versions of COMSOL Multiphysics.

dependent variable A varying quantity whose changes are arbitrary, but they are regarded as produced by changes in other variables. For example, temperature is a function of the space coordinates and time. In a narrower sense, the dependent variables, or *solution components*, are the unknowns in a mathematical PDE model. Compare to *independent variable*.

differential-algebraic equation (DAE) A set of equations that includes both differential and algebraic equations. A DAE is classified in terms of its *index*, a positive integer, which is related to the minimum number of differentiations needed to transform a DAE to an ODE form.

direct solver A solver for a system of linear equation that uses some variant of Gaussian elimination. Compare to *iterative solver*.

Dirichlet boundary condition A Dirichlet boundary condition specifies the value of the function (dependent variable) on a boundary. Dirichlet boundary conditions are sometimes called *essential boundary conditions* or *constraints*. For a coefficient form PDE the Dirichlet boundary condition is $hu = r$. See also *constraint*.

discretization The process of dividing a continuous system into a finite number of elements with finite size. The difference between the finite-element representation and the real system, the discretization error, drops as the size of the elements decrease. For a time-dependent analysis, a discretization of time into steps provides an idealized behavior of the variations in the solution during these steps.

divergence element A finite element often used for electromagnetic vector fields. The degrees of freedom on the boundary of a mesh element correspond to normal components of the field. Also *Nédélec's divergence element*.

domain A topological part of the modeling space in a geometry model. The geometric representation of a domain is a line segment (interval) in 1D, an area in 2D, and a volume in 3D. In a mathematical context, the symbol Ω represents the domain where the equations are defined.

drop tolerance A nonnegative scalar used in the incomplete LU preconditioner for the iterative solvers. See *incomplete LU factorization*.

dynamic model See *time-dependent model*.

edge, edge segment A *geometric entity* representing a bounded part of a *curve*. An *edge* or *edge segment* is a *boundary* in a 2D geometry. See also *domain*.

edge element See *vector element*.

eigenvalue PDE A PDE that describes an eigenvalue problem with unknown eigenmodes (eigenfunctions) u and eigenvalues λ. The *coefficient form* eigenvalue PDE is:

$$\lambda^2 e_a u - \lambda d_a u + \nabla \cdot (-c\nabla u - \alpha u) + \beta \cdot \nabla u + au = 0$$

elliptic PDE A linear stationary 2nd-order elliptic PDE has the form

$$\nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f$$

where c is positive or negative definite, for example, Poisson's equation.

embed To insert a 2D geometry into a 3D geometry model.

error Deviations from the correct solution, primarily due to: poor modeling; *discretization* (such as insufficiently fine mesh, poor elements, or insufficiently short time steps); and roundoff and truncation (depending on numerical representation, *ill-conditioning*, or the solution algorithms).

error estimate An estimation of the error in the numeric solution to a problem, either locally or globally, primarily for use by an adaptive mesh refinement. See also *adaptive mesh refinement*, *error*.

equivalent boundaries *Boundaries* that are rigid transformations of each other and have compatible meshes. See also *periodic boundary condition*.

essential boundary condition See *Dirichlet boundary condition*.

Eulerian formulation [An Eulerian formulation means that the partial differential equations that describe some physics are formulated in a *spatial frame* (coordinate system), with coordinate axes fixed in space. An Eulerian formulation is common for fluid flow when the focus is on specific locations in space through which fluid flows. Compare to *Lagrangian formulation*.

expression variable A user defined variable that is defined on any geometry domain in terms of *dependent variables*, *independent variables*, *constants*, *application scalar variables*, and other *expression variables*. Global expression variables are valid in all geometries; scalar expression variables are valid in the current geometry.

extended mesh A data structure that includes the full finite element mesh. See also *mesh*, *node point*.

extended multiphysics A model that includes nonlocal couplings and dependencies between variables, where the value at a point is the result of a computation elsewhere in the domain or in another geometry defined in the same model. *Coupling operators* provide the ability to project or extrude values from one geometry or domain to another. Compare to *multiphysics*.

exterior boundary An *exterior boundary* for a dependent variable *u* is a *boundary* such that *u* is defined only on one of the adjacent domains, that is, a boundary to the exterior of the computational domain. See also *boundary*.

extrude To create a 3D geometry object from a 2D geometry object in a *work plane* by translating (extruding) it along a path, often a straight line.

extruded mesh A 3D mesh created by extrusion of a 2D mesh. An extruded mesh can contain *hexahedral elements* and *prism elements.*

extrusion coupling operator An operator in the destination that takes values from the source by interpolation at points that depend on the position of the evaluation points in the destination.

face, face segment A domain describing a bounded part of a *surface* in a 3D geometry. A *face* or *face segment* is a *boundary* in a 3D geometry. See also *domain*.

face object A geometry object with no topological information on domains. Typically a trimmed surface is represented as a face object.

FEM See *finite element method*.

field variables Dependent variables and variables derived from them. Compare to *expression variables.*

fillet A curved transition from one boundary to another, creating a rounded corner.

finite element In the mathematical sense, a *mesh element* together with a set of *shape functions* and corresponding *degrees of freedom*. The linear combinations of the shape functions form a space of functions called the *finite element space*. In the traditional FEA sense, the concept of a finite element also includes the discretized form of the PDEs that govern the physics. COMSOL Multiphysics generally uses *finite element* in the mathematical sense.

finite element analysis (FEA) A computer-based analysis method for field problems using the *finite element method*.

finite element method (FEM) A computational method that subdivides an object into very small but finite-size elements. The physics of one element is approximately described by a finite number of *degrees of freedom (DOFs)*. Each element is assigned a set of characteristic equations (describing physical properties, boundary conditions, and imposed forces), which are then solved as a set of simultaneous equations to predict the object's behavior.

finite element space The linear space of functions where the finite element approximation to the solution of a PDE problem is sought. The functions in the finite element space are linear combinations of *basis functions* (*shape functions*).

flux vector The vector is as below. See also *generalized Neumann boundary condition* and *normal flux*.

$$\Gamma = -c\nabla u - \alpha u + \gamma$$

frame [A *frame* is a coordinate system that is fixed in space, to a material, to the geometry, or to a mesh. The frames make it possible to use an *Eulerian formulation* or a *Lagrangian formulation* for various physics in a model and to combine them using the arbitrary Lagrangian-Eulerian (ALE) method. The following frame types are available: *material frame* (*reference frame*), *geometry frame*, *mesh frame*, and *spatial frame*.

free mesh An *unstructured mesh* that can represent any geometry. Compare to *mapped mesh*.

free mesher The mesh generator creating *free meshes*. The mesh generator creating *triangular elements* is also referred to as the *free triangle mesher*, and the mesh generator creating *quadrilateral elements* is also referred to as the *free quad mesher*.

free quad mesher The mesh generator creating unstructured quadrilateral meshes.

free triangle mesher The mesh generator creating unstructured triangular meshes.

Gauss point Sometimes improperly used as a synonym for *integration point*. See also *integration point*.

general form PDE A PDE in the general form is a PDE formulation suited for nonlinear PDEs

$$\begin{cases} e_a\dfrac{\partial^2 u}{\partial t^2} + d_a\dfrac{\partial u}{\partial t} + \nabla\cdot\Gamma = F & \text{in } \Omega \\ -\mathbf{n}\cdot\Gamma = G + \left(\dfrac{\partial R}{\partial u}\right)^T\mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

generalized Neumann boundary condition A generalized Neumann boundary condition (also called a *mixed boundary condition* or a *Robin boundary condition*) specifies the value of a linear combination of the *normal flux* and the *dependent variables* on a boundary. For a coefficient form PDE, the generalized Neumann boundary condition is

$$\mathbf{n}\cdot(c\nabla u+\alpha u-\gamma)+qu=g-h^T\mu$$

The generalized Neumann condition is often called just *Neumann condition* in the documentation.

geometric entities The basic parts that constitute the finalized geometry: *domains*, *boundaries* (*faces*), *edges*, and *points* (*vertices*).

geometric multigrid (GMG) A *geometric multigrid* solver or preconditioner performs one or more cycles of a multigrid method, using a coarsening of the discretization based on a coarsening of the mesh or a reduction in the order of the shape functions. Compare to *algebraic multigrid (AMG)*.

geometry frame In the geometry frame (coordinate system) the domain is fixed and identical to the original geometry. There is a one-to-one map from the geometry coordinates to the current spatial configuration of the domain, but this map may become quite ill-conditioned as the domain deforms. There is also a one-to-one map from the mesh coordinates to the geometry coordinates, which is trivial until the first remeshing event. After remeshing, both the map from mesh to geometry coordinates and the map from geometry to spatial coordinates may be ill-conditioned, while the composite map from mesh coordinates to spatial coordinates is well-behaved. This composite map is the one used by the ALE method, while the geometry coordinates are in practice only used as reference for the Deformed Geometry interface and for postprocessing.

geometry level The types of topological entities within a geometry model that describes bounded parts of the manifolds in the model, and also the relations between different manifolds in the geometry. The different geometry levels are the *vertex*, *edge*, *face*, and *domain* levels. An entity of dimension one less than the space dimension is referred to as a *boundary*. See also *manifold*.

geometry model A collection of topological and *geometric entities* that form a complete geometric description of the model.

geometry object The objects that represent a geometry model. See also *curve object*, *face object*, *primitive geometry object*, *solid object*.

grid A *grid* usually refers to sets of evenly-spaced parallel lines at particular angles to each other in a plane, or the intersections of such lines. Compare to *mesh*.

Hermite element A finite element similar to the *Lagrange element*. The difference is that there are degrees of freedom for the (1st-order) space derivatives at the mesh vertices. See also *Lagrange element*.

hexahedral element A 3D mesh element with eight corners and six faces, also referred to as *brick element*; sometimes also called *hex element* as a short form.

higher-order element A finite element with *basis functions* that consists of polynomials of degree 2 or higher.

hybrid geometry modeling Creating a geometry model using a combination of *boundary modeling/surface modeling* and *solid modeling*.

hyperbolic PDE A typical example of a linear 2nd-order hyperbolic PDEs is the *wave equation*

$$e_a\frac{\partial^2 u}{\partial t^2} + \nabla \cdot (-c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + au = f$$

where e_a and c are positive.

IGES file An *IGES file* contains 3D CAD data, including the 3D geometry, in an open format according to the *Initial Graphics Exchange Specification*. IGES file can be imported to COMSOL Multiphysics using the CAD Import Module.

ill-conditioning An ill-conditioned system is sensitive to small changes in the inputs and is susceptible to roundoff errors. See also *condition number*.

imprint An imprint of the smaller boundary on the larger boundary that makes the parts in a *pair* match. An imprint inserts points on the boundary in 2D and creates a curve on the boundary in 3D.

incomplete LU factorization An approximate *LU factorization* where small matrix elements are discarded to save memory and computation time. The *drop tolerance* is a relative measure of the smallness of the elements that should be discarded. See also *LU factorization*.

independent variable A variable that can cause variation in a second, *dependent variable*. The independent variables are most often space coordinates and time. Compare to *dependent variable*.

index, for DAE See *differential-algebraic equation*.

index vector See *domain group*.

initial condition The starting values for the dependent variables in a time-dependent analysis and for *nonlinear iterations* or other iterative solvers.

integration coupling operator An operator that evaluates integrals of expressions over the source and returns a single scalar value when used in the destination. Similar functionality is available to evaluate the average, minimum, and maximum values.

integration point See *numerical integration formula*.

interactive meshing Building a mesh in an incremental fashion where each meshing operation acts on a set of geometry domains.

interior boundary An *interior boundary* for a dependent variable u is a *boundary* such that u is defined on both adjacent domains or in no adjacent domain. See also *boundary.*

interval The domain between two vertices in a 1D geometry. Also called a *domain*.

isoparametric element A finite element that uses the same *shape function* for the element shape coordinates as for the *dependent variables.*

isosceles triangle A triangle with at least two equal sides (and two equal angles).

iteration See *iterative solver.*

iterative solver A solver for a system of linear equations that uses an iterative method, calculating a sequence of more and more accurate approximations to the solution. Each step in this sequence is one *linear iteration*. This should not be confused with the Newtons iterations (*nonlinear iterations*) that occur in the solution of a nonlinear system of equations. Compare to *direct solver* and *nonlinear iteration*.

Jacobian matrix A matrix containing the first derivative of a vector-valued function of a vector variable. In particular, it is the derivative of the *residual vector* with respect to the *solution vector*. When used in this narrower sense, the term *stiffness matrix* is sometimes used.

Lagrange element A *finite element* with polynomial shape functions of a certain *order*. The value of the function is used as the *degree of freedom*, and the node points are evenly distributed within the mesh element.

Lagrange multiplier An extra dependent variable introduced in the Neumann boundary condition when a constraint is added. See also *constraint*.

Lagrangian formulation A Lagrangian formulation means that the partial differential equations that describe some physics are formulated in a *material frame* (coordinate system) with coordinate axes fixed to the material in its reference configuration and following the material as it deforms. The Lagrangian formulation is common for solid mechanics because it makes anisotropic material properties independent of the current spatial orientation of the material. Compare to *Eulerian formulation*.

linear iteration A step in a linear iterative solver. See *iterative solver*. Compare to *nonlinear iteration*.

linear PDE An equation where both sides are sums of a known function, the unknown functions, and their partial derivatives, multiplied by known coefficients that only depend on the *independent variables*. Other PDEs are called *nonlinear*.

LU factorization For a linear system of equations, a version of Gaussian elimination that produces a factorization $A = LU$ of the coefficient matrix, where L and U are the lower and upper triangular matrices, respectively. This makes it easy to quickly solve a number of systems with the same coefficient matrix. See also *direct solver*.

manifold A mathematical function describing a *surface*, *curve*, or *point* in a geometry model of any dimension.

mapped mesh A *structured mesh* with *quadrilateral elements* generated by mapping using transfinite interpolation.

mapped mesher The mesh generator creating *mapped meshes*.

mass matrix The matrix E that multiplies the second time derivative of the *solution vector* in the linearized discretized form of a PDE problem. If there are no second time derivatives (that is, if $E = 0$), then the term mass matrix is often used for the matrix D that multiplies the first derivative of the solution vector (the D matrix is otherwise called the *damping matrix*).

material frame The material frame defines a coordinate system that is fixed to the material in its reference configuration and follows the material as it deforms. The material frame is used in connection with a *Lagrangian formulation*. This frame is also referred to as a *reference frame*.

mesh A subdivision of the domains of a geometric model into, for example, triangles (2D) or tetrahedra (3D). These are examples of *mesh elements*. See also *grid*, *structured mesh*, and *unstructured mesh*.

mesh element The individual elements in the mesh that together form a partitioning of the geometry, for example, *triangular elements* and *tetrahedral elements*. See also *finite element*.

mesh frame In the mesh frame (coordinate system) the domain is fixed until an automatic or manual remeshing operation is performed, as well as between remeshing events. There is always a one-to-one map from the mesh coordinates to the current spatial configuration of the domain. Otherwise, the mesh frame can be defined freely and separately from both the *spatial frame*, the *material frame* and the *geometry frame*. The *arbitrary Lagrangian-Eulerian formulation (ALE formulation)* modifies the Eulerian formulation to describe all physical quantities as functions of the mesh coordinates instead of spatial coordinates to allow moving boundaries.

mesh vertex An endpoint or corner of a mesh element. See also *node point* and *vertex*.

method of lines A method for solving a time-dependent PDE through a space discretization, resulting in a set of ODEs.

mixed boundary condition See *generalized Neumann boundary condition*.

mode reduction A model-reduction technique for reducing systems with many degrees of freedom, for example large finite element models, to a form with fewer degrees of freedom for dynamic system simulations and analysis. See also *state-space model*.

model input *Model inputs* are fields such as temperature and velocities that act as inputs for materials and model equations. The model inputs can be fields computed by other physics interfaces or user defined values.

Model Java-file A file that contains Java commands based on the COMSOL Multiphysics *API*. Use the API to extend and modify the Model Java-file. Compiling and running a Model Java-file creates the COMSOL Multiphysics model.

Model MPH-file A binary data file with the extension `.mph` that contains a COMSOL Multiphysics model. Often also just called model file. Earlier versions of COMSOL Multiphysics used the file extension `.fl`.

Model M-file An M-file containing commands that create a COMSOL Multiphysics model. A Model M-file is a text file that is similar to a *Model Java-file* and that can be modified and used with MATLAB. If you have a MATLAB license and a license for the COMSOL LiveLink for MATLAB, the COMSOL Desktop can load a Model M-file. Compare with *Model MPH-file*.

model object An object (data structure) that contains all data for a model. This is the fundamental data structure in a COMSOL Multiphysics model.

Model Wizard Part of the COMSOL Desktop, the Select Space Dimension, Select Physics, and Select Study Type pages are part of the Model Wizard selections that start building a model.

MRI data *Magnet resonance imaging (MRI) data* is an image data format, primarily for medical use. MRI produces high-quality images of the inside of the human body. 3D MRI data is usually represented as a sequence of 2D images.

multidisciplinary models Multidisciplinary models combine PDE-based finite element modeling with other mathematical modeling techniques such as dynamic simulation in areas like automatic control and signal processing.

multigrid A solver or preconditioner for a linear system of equations that computes a sequence of increasingly accurate approximations of the solution by using a hierarchy of coarsened versions of the linear system (having fewer degrees of freedom). See also *algebraic multigrid*, *geometric multigrid*.

multiphysics Multiphysics models include more than one equation and variable from different types of physics. These variables can be defined in different domains. The equations can be coupled together through equation coefficients that depend on variables from other equations. Compare to *extended multiphysics*.

natural boundary condition See *Neumann boundary condition*.

Neumann boundary condition A Neumann boundary condition specifies the value of the *normal flux* across a boundary. Neumann boundary conditions are sometimes called *natural boundary conditions*. Compare to *generalized Neumann conditions*.

Newton's method An iterative solver method, also called the *Newton-Raphson method*, for solving nonlinear equations. See also *nonlinear iterations*.

Newton-Raphson method See *Newton's method*.

node point Any point in the mesh element where the degrees of freedom are defined. The node points often include the mesh vertices and possible interior or midpoint locations. See also *degree of freedom* (DOF) and *mesh vertex*.

nonlinear iteration A Newton step in the solution of a nonlinear PDE problem. Each nonlinear iteration involves the solution of a linear system of equations. Compare to *linear iteration*.

nonlinear PDE See *linear PDE*.

norm A scalar measure of the magnitude of a vector or a matrix. Several types of norms are used to measure the accuracy of numerical solutions.

numerical integration formula A numeric-integration method that approximates an integral by taking the weighted sum of the integrand evaluated at a finite number of points, the *integration points* (sometimes improperly called *Gauss points*). Also called *quadrature formula*.

normal flux The normal component of the *flux vector* at a boundary.

NURBS The *nonuniform rational B-spline (NURBS)* is a popular curve and surface representation scheme. A NURBS representation can be divided into a number of *rational Bézier curves* or surfaces.

order of a finite element The degree of the polynomials that define the *shape functions* (*basis functions*).

ordinary differential equation (ODE) An equation involving functions and their derivatives. The derivatives are with respect to one independent variable only. Compare to *partial differential equation (PDE)*.

parabolic PDE A typical example of a linear 2nd-order parabolic PDE is the *heat equation*

$$d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+au = f$$

where d_a and c are positive.

parameter A constant that takes on different values for each model in a parametric analysis. See also *constant*.

partial differential equation (PDE) An equation involving functions and their partial derivatives; that is, an equation that includes derivatives with respect to more than one independent variable. Compare to *ordinary differential equation (ODE)*.

periodic boundary condition A boundary condition where the values of the solution appear in a periodic pattern, typically so that the value of the solution on one boundary is equal to the value on another boundary. See also *equivalent boundaries*.

phasor A complex number or a vector of complex numbers representing a sinusoidally varying current or voltage.

physics interfaces Sets of features for different types of physics in the COMSOL Desktop environment. The physics interfaces contain predefined equations and boundary conditions and a set of features for setting up models for a certain type of physics. In earlier versions of COMSOL Multiphysics, the application modes provide similar functionality.

pivot Usually a value on the main diagonal of the *stiffness matrix*. *Pivoting* is the interchanging of rows and columns in order to place a particularly large element in the diagonal position. The value of the diagonal term when it is used to eliminate values below it is called the *pivot value*.

point A location in space.

point object A geometry object with only *vertices*.

positive definiteness A symmetric matrix is *positive definite* when all its eigenvalues are positive.

preconditioner The convergence rate of iterative methods depends on the spectral properties of the coefficient matrix. A *preconditioner* is a matrix that transforms the linear system into one that has the same solution but that has more favorable spectral properties. See also *algebraic multigrid*, *geometric multigrid*, *incomplete LU factorization*, *iterative solver*, and *SSOR*.

primitive, primitive geometry object A geometry object with a basic shape such as a cube or a sphere. Add primitives to a model, using arbitrary sizes and positions, and combine them to form complex shapes. See also *constructive solid geometry*, *composite geometry object*, and *Boolean operations*.

prism element A 3D mesh element with six corners and five faces, also referred to as *wedge element*.

projection coupling operator An operator that takes values from the source by evaluating line integrals over lines whose positions are dependent on the position of the evaluation points in the destination.

quadrature formula See *numerical integration formula*.

quadrilateral element A 2D mesh element with four corners and four edges; sometimes also called *quad element* as a short form.

rational Bézier curve See *Bézier curve*.

reference frame See *material frame*.

residual vector The vector L in the discretized form of a PDE problem. In the absence of *constraints*, the discrete form of a stationary equation is $0 = L(U)$ where U is the *solution vector*.

revolve To create a 3D geometry object from a 2D geometry object in a *work plane* by rotating it around an axis.

revolved mesh A 3D mesh created by revolving a 2D mesh. A revolved mesh can contain *hexahedral elements* and *prism elements*.

Robin boundary condition See *generalized Neumann boundary condition*.

shape function A *basis function* described in local element coordinates. See also *basis function*.

shift A value σ around which an eigensolver searches for eigenvalues.

simplex element *Triangle element* in 2D and *tetrahedral element* in 3D.

solid A description of a part of the modeling space. See also *domain*.

solid modeling A 3D geometry modeling method that describes both the boundary and interior of the geometry using solid objects. See also *constructive solid geometry (CSG)* and *solid*.

solid object A geometry object representing one or several *solids*.

solution component See *dependent variable*.

solution matrix A matrix that contains a sequence of solutions as columns. A steady-state problem results in a *solution vector*, but eigenvalue problems, time-dependent problems, and parametric analyses produce a *solution matrix*. See also *solution structure*.

solution structure A data structure that includes the *solution vector* or *solution matrix* and any associated data such as parameter values, output times, or eigenvalues.

solution vector A vector with all the *degrees of freedom* (values of the *dependent variables*) as its components. See also *solution matrix* and *solution structure*.

solver sequence A recorded sequence of named solver settings and commands that can be replayed by a single solver call.

sparse matrix Matrix for which the number of zero elements is large enough to justify special data types and algorithms that avoid operations on zero elements.

spatial frame The spatial frame defines a coordinate system with coordinate axes fixed in space. The spatial frame is used in connection with a *Eulerian formulation*.

split To divide a geometry object into its minimal parts.

stability A solver for a time-dependent model is *unconditionally stable* if the initial conditions are not amplified artificially and the roundoff errors do not grow, regardless of the size of the time step. A solver is *conditionally stable* if there is a maximum value of the time step above which the numerical solution is unstable.

state-space model A linear time-invariant representation of a dynamic system as a set of 1st-order *ODEs* of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{y} &= Cx + Du \end{aligned}$$

where x is the state vector, u is the input, and y is the output. A, B, C, and D are the constant dynamics, input, output, and direct transmission matrices, respectively.

static model See *stationary model.*

stationary model A model where the dependent variables do not change over time. It typically represents a steady-state solution. Also called *static model* or *steady model.*

steady model See *stationary model.*

stiffness matrix See *Jacobian matrix.*

streakline The locus of particles that have earlier passed through a prescribed point in space. See also *streamline.*

streamline A curve that is everywhere tangent to the vector field (in particular a velocity field) at a given instant of time. Sometimes called a *flow line* or *flux line*. See also *streakline.*

streamline-diffusion stabilization A numerical technique for stabilization of the numeric solution to a PDE by artificially adding diffusion in the direction of the *streamlines.*

strong form A partial differential equation in the *strong form* is the standard formulation as an equality of functions. The strong form is divided into the *coefficient form* and the *general form*. Compare to *coefficient form*, *general form*, and *weak form.*

structured mesh A mesh for which all elements and nodes have the same topology. Compare to *unstructured mesh.*

surface A mathematical function (manifold) from 2D to 3D space.

surface normal A vector perpendicular to the surface.

surface modeling A 3D geometry modeling method to describe a geometry by defining its bounding surfaces. Compare *boundary modeling* and *solid modeling*.

swept mesh A 3D mesh generated by sweeping a face mesh along a domain.

symmetric matrix A matrix that equals its own transpose.

symmetric successive overrelaxation (SSOR) A *symmetric successive overrelaxation (SSOR)* preconditioner uses classic SSOR iterations.

symmetry The invariance of an object attribute or of the object itself under a well-defined operation or transformation such as inversion, rotation, or reflection. A *symmetry* allows for a reduction of the model geometry so that appropriate boundary conditions account for the redundant portions of the geometry. Axial symmetry is a common type of *symmetry.*

symmetry boundaries See *equivalent boundaries.*

test function See *weak form.*

tetrahedral element A 3D mesh element with four corners, six edges, and four triangular faces.

time-dependent model See *transient model.*

transient model A model where at least one of the dependent variables changes over time, for example, the heat equation or the wave equation. Also called *dynamic model*, *time-dependent model*, or *unsteady model.*

triangular element A 2D mesh element with three corners and three edges.

trimmed surface If the parameter space of a surface is divided into “valid” and “invalid” regions, the image of the valid regions is called the *trimmed surface*. This corresponds to the part of the surface limited by a closed loop of edges lying on the surface.

unidirectional constraint See *constraint.*

unstructured mesh A mesh without a specific pattern where the elements can have different shapes and the nodes can have different connectivities. Compare to *structured mesh*.

unsteady model See *time-dependent model*.

vector element A finite element often used for electromagnetic vector fields. Each mesh element has degrees of freedom corresponding only to tangential components of the field. Also called *curl element*, *Nédélec's edge element*, or just *edge element*.

vertex A point in a geometry model, often an endpoint of a geometry segment or an intersection of *geometric entities* of higher degree such as *edges* or *faces*. A vertex is referred to as a *point* for the specification of point sources and other PDE modeling. See also *domain*.

weak constraint A reformulation of a *Dirichlet boundary condition* as a *weak form* equation. When using a weak constraint, the corresponding *Lagrange multiplier* becomes a *solution component* (*dependent variable*).

weak form A partial differential equation in the *weak form* is a more general formulation than the strong form. It is produced by multiplying the *strong form* PDE with an arbitrary function called the *test function* and integrating over the computational domain. Most physics interfaces in COMSOL Multiphysics are implemented using a weak form. Compare to *strong form*.

wedge element See *prism element*.

well-posed A well-posed numerical model has at least one solution, and depends continuously on its input, such as initial conditions, boundary conditions, and the geometry.

work plane An embedded 2D work space that can be positioned relative to the coordinate planes or an already existing geometry. Using *work planes* makes it possible to define a geometry in terms of previously created geometry objects such as *points*, *edges*, and *faces*.

Index