



COMSOL Multiphysics®

Reference Manual



COMSOL Multiphysics Reference Manual

© 1998–2013 COMSOL

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

This Documentation and the Programs described herein are furnished under the COMSOL Software License Agreement (www.comsol.com/sla) and may be used or copied only under the terms of the license agreement.

COMSOL, COMSOL Multiphysics, Capture the Concept, COMSOL Desktop, and LiveLink are either registered trademarks or trademarks of COMSOL AB. All other trademarks are the property of their respective owners, and COMSOL AB and its subsidiaries and products are not affiliated with, endorsed by, sponsored by, or supported by those trademark owners. For a list of such trademark owners, see www.comsol.com/tm.

Version:

May 2013

COMSOL 4.3b

Contact Information

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

If you need to contact Support, an online request form is located at the COMSOL Access page at www.comsol.com/support/case.

Other useful links include:

- Support Center: www.comsol.com/support
- Download COMSOL: www.comsol.com/support/download
- Product Updates: www.comsol.com/support/updates
- COMSOL Community: www.comsol.com/community
- Events: www.comsol.com/events
- COMSOL Video Center: www.comsol.com/video
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part No. CM020005

Contents

Chapter 1: Introduction

About COMSOL Multiphysics	2
The COMSOL Modules and Interfacing Options	4
Overview of the Reference Manual	5
The COMSOL Documentation Set	9
About the Documentation Set.	9
Where Do I Access the Documentation and Model Library?	10
Checking for Software Updates	12
Typographical Conventions	12

Chapter 2: The COMSOL Modeling Environment

The COMSOL Desktop Environment	34
Introduction	34
Changing the COMSOL Desktop Layout.	35
Moving Between Windows and Sections on the COMSOL Desktop	36
Changing the COMSOL Desktop Language	37
Printing from the COMSOL Desktop	37
The Main Menu and Toolbar	39
The Model Wizard and Model Builder	41
The Model Wizard	41
Basic Steps to Build a Model	43
The Model Builder Window	45
Opening a Context Menu to Add Nodes.	48
Moving Nodes in the Model Tree.	50
Going to the Source Node	51
Copying, Pasting, and Duplicating Nodes	51
Expanding and Collapsing All Nodes	52

Undoing and Redoing Operations	52
Clearing All Meshes and Clearing all Solutions	53
Sorting Nodes by Space Dimension and Type	53
Other COMSOL Desktop Windows	57
The Node Settings Windows	57
The Graphics Window	58
The Plot Windows	59
The Messages Window	59
The Progress Window.	60
The Log Window.	60
The Table Window	61
The Help Window	62
The Model Library Window	68
The Material Browser Window	70
The External Process Window.	70
Errors and Warnings	71
The About COMSOL Multiphysics Box	72
Keyboard Shortcuts	74
Key to Nodes and Toolbar Buttons	77
The Model Builder Nodes	77
Dynamic Nodes in the Model Builder	78

Chapter 3: Building a COMSOL Model

The Model Wizard and Model Builder	82
The Model Nodes in the Model Builder	83
Adding Nodes to the Model Builder	85
About the Sequence of Operations	85
Deleting, Disabling, and Enabling Nodes	86
Model Administration	88
About the COMSOL Model File Formats.	88
Saving COMSOL Model Files	90

Reverting to the Last Saved Model File.	90
Saving a Model Thumbnail Image	91
Resetting the Model History	91
Saving and Opening Recovery Files	91
Working with Remote Servers.	92
Editing Node Properties, Names, and Identifiers	93
Checking and Controlling Products and Licenses Used	96
Borrowing Licenses	96
Information About Memory Use	97
Viewing Node Names, Identifiers, Types, and Tags	97
 Preferences Settings	100
General Preferences	100
User Interface Preferences	102
Show Preferences	102
Expand Sections Preferences	103
Model Builder Preferences	103
Cluster Computing Preferences	105
Remote Computing Preferences	106
Geometry Preferences.	107
Graphics Preferences	107
Results Preferences	110
Builder Tools Preferences	111
Temporary Files Preferences	113
Memory and Processors Preferences	114
Preferences for Updates	115
LiveLink Products Preferences	116
 The Model Library	118
Opening and Searching the Model Library	118
Changing the Model Library Root Folder.	119
Updating the Model Library Using Model Library Update.	120
Adding a User Model Library	121
 Modeling Guidelines	122
Using Symmetries	122
Effective Memory Management	123
Selecting an Element Type	124

Analyzing Model Convergence and Accuracy	125
Achieving Convergence When Solving Nonlinear Equations.	125
Avoiding Strong Transients	126
Physics-Related Checks and Guidelines	127
Results With Unphysical Values	128
Advanced Physics Sections	132
Resetting to Default.	132
Resetting from Preferences	132
Setting as Preferences	133
Expanding Sections	133
Expanding the Equation Sections	133
Show Equation Sections	133
Show Equation View	134
Show Override and Contribution	134
Show Discretization	134
Show Stabilization	136
Show Advanced Physics Options	138
Show Advanced Physics Options—Context Menu	
Equation-Based Nodes	138
Show Advanced Study Options	139
Show Advanced Results Options	140
The Physics Nodes	141
Specifying Physics Settings	141
Physics Node Context Menu Layout	142
Physics Nodes by Space Dimension	142
Physics Exclusive and Contributing Node Types	144
Physics Node Status.	147
Physics Default Nodes	148
Equation View	149
Specifying Model Equation Settings	153
Specifying Equation Coefficients and Material Properties	153
Modeling Anisotropic Materials	154
Specifying Initial Values.	155

Boundary Conditions	156
Boundary Condition Types	156
Physics Boundary Types	158
Continuity on Interior Boundaries	159
Physics Axial Symmetry Node	160
Constraint Reaction Terms	160
Weak Constraints	162
Constraint Settings	164
Periodic Boundary Conditions	164
Periodic Boundary Condition Example.	165
Computing Accurate Fluxes	168
Flux Computation Methods.	168
Flux Calculation Example—Heat Transfer Model.	170
Using Load Cases	173
Defining Load Groups and Constraint Groups.	173
Load Group.	175
Constraint Group	176
Defining and Evaluating Load Cases	176
Using Units	179
Unit Systems in COMSOL	179
Selecting a Unit System	181
Using Standard Unit Prefixes and Syntax	181
SI Base, Derived, and Other Units	184
Special British Engineering Units	189
Special CGSA Units	190
Special EMU Units	190
Special ESU Units.	191
Special FPS Units	192
Special IPS Units	192
Special MPa Units.	193
Special Gravitational IPS Units	193
Switching Unit System	193
About Temperature Units	194
About Editing Geometry Length and Angular Units	195
Indication of Unexpected, Unknown, or Inconsistent Units	196

Units and Space Dimensions	196
Numerical Stabilization	197
About Numerical Stabilization in COMSOL.	197
An Example of Stabilization	198
Stabilization Techniques	200
References for Stabilization Techniques	205

Chapter 4: Definitions

About Global and Local Definitions	208
Global Definitions	208
Definitions	208
About Parameters, Variables, and Expressions.	209
User-Defined Parameters.	211
User-Defined Variables	212
Common Settings for the Definitions Nodes	213
Operators, Functions, and Constants	217
Unary, Binary, and List Operators and Their Precedence Rules	217
Mathematical and Numerical Constants	218
Mathematical Functions	219
Physical Constants	221
Built-In Operators	222
Predefined and Built-In Variables	240
Predefined Physics Variables.	240
Variable Naming Convention and Scope	240
Variable Classification and Geometric Scope	241
Built-In Global Variables	242
Geometric Variables and Mesh Variables	243
Shape Function Variables	249
Solver Variables	253
Entering Ranges and Vector-Valued Expressions	253
Summary of Built-In Variables With Reserved Names	256

User-Defined Functions	262
About User-Defined Functions.	262
Analytic	263
Elevation	264
External	265
Gaussian Pulse.	266
Image	267
Interpolation	268
MATLAB	275
Piecewise.	275
Ramp	277
Random	278
Rectangle.	278
Step.	279
Thermodynamics Package	279
Triangle	280
Waveform	280
Specifying Discontinuous Functions	281
Model Couplings	283
About Model Couplings and Coupling Operators	283
General Extrusion	291
Linear Extrusion	292
Boundary Similarity	293
One-Point Map	295
Two-Point Map.	295
Edge Map.	295
Identity Mapping	296
General Projection	297
Linear Projection.	298
Integration	299
Average	300
Maximum and Minimum	301
Common Settings for Model Couplings	302
Coordinate Systems	305
About Coordinate Systems	305
Base Vector System	307

Boundary System	309
Cylindrical System	310
Mapped System	312
Rotated System	313
Spherical System	315
Scaling System	316
Identity and Contact Pairs	318
About Identity and Contact Pairs.	318
Identity Pair	320
Contact Pair	322
Probes	324
About Probes	324
Domain Probe, Boundary Probe, and Edge Probe	325
Domain Point Probe	326
Boundary Point Probe	327
Point Probe Expression	328
Global Variable Probe	328
Infinite Element Domains and Perfectly Matched Layers	330
Simulation of Infinite Domains	330
Standard Geometry Configurations	331
Note on Availability	335
The Challenge of Open Boundaries for PMLs in Problems	335
PML Implementation	336
Perfectly Matched Layer	340
Known Issues When Modeling Using PMLs	341
Modeling Unbounded Domains	342
Infinite Element Domain	344
Known Issues When Modeling Using Infinite Elements.	345
References for PMLs and Infinite Element Domains	346

Chapter 5: Visualization and Selection Tools

Selecting and Visualizing in Models	348
About Viewing and Selecting in the Graphics Window	348
About Geometric Entities	349
About Selecting Geometric Entities	350
Pasting Lists of Geometric Entities to Create Selections	354
The Graphics Toolbar Buttons	355
Using the Selection List Window	357
Selecting and Deselecting Geometric Entities	361
Zooming In and Out in the Graphics Window	365
Changing Views in the Graphics Window	365
Moving Around and Rotating 3D Geometry in the Graphics Window	366
Lighting, Transparency, and Wireframe Rendering	367
Hiding and Showing Geometric Entities	368
Named Selections	370
Creating Named Selections	370
Explicit	371
Ball, Box, and Cylinder	373
Union, Intersection, Difference, or Complement	375
Adjacent	376
Opening an Example Model	376
Creating a Selection Node from the Selection List Window	377
Creating a Selection Node from a Settings Window	377
Creating Selections From Geometric Primitives and Operations	378
Creating Named Selections in the Geometry Sequence	379
Explicit Selection	379
Ball Selection, Box Selection, and Cylinder Selection	380
Union, Intersection, Difference, or Complement Selection	383
Adjacent Selection	383
User-Defined Views	385
The View Nodes	385
View 2D	386
Axis	386

View 3D	387
Camera	390
About the 3D View Light Sources and Attributes	391
Directional Light	393
Point Light	395
Spotlight	397
Headlight.	399
Hide Geometry Objects and Hide Geometric Entities.	400
Capturing and Copying Screenshots	402

Chapter 6: Geometry Modeling and CAD Tools

Working with Geometry Sequences	406
Overview of Geometry Modeling Concepts	406
Adding a Model (Geometry)	407
Adding Geometry Nodes.	408
Editing and Building Geometry Nodes.	410
The Current Node in Geometry Sequences	413
Deleting, Disabling, and Enabling Geometry Features	414
Errors and Warnings in a Geometry Sequence.	415
The Geometry and CAD Environment	417
Geometric Primitives	417
Using Work Planes	419
Boolean Operations.	423
Geometry Object Transforms	424
Geometry Object Conversions	425
General Geometry Operations	426
The Geometry Toolbars	427
The Measurements Page	431
Copying and Pasting Geometry Objects	432
The Geometry Node	433
The Finalize Node—Finalizing the Geometry	435
Insert Sequence from File	437
Exporting Geometry to File.	438

Composite Object (Backward Compatibility)	439
Creating a Geometry for Analysis	440
Techniques for Creating Geometries	440
Associative Geometry and Selections of Geometry Objects	441
Selecting the Space Dimension.	442
Geometry Modeling Examples	446
Creating a 1D Geometry Model	446
Creating a 2D Geometry Model	446
Creating a 3D Geometry Model	455
Forming Composite Edges and Faces by Ignoring Vertices and Edges	465
Merging Vertices by Collapsing Edge	471
Geometric Primitives	475
Bézier Polygon.	475
Block	477
Circle	478
Cone	479
Cylinder	481
Eccentric Cone	482
Ellipse	483
Ellipsoid	484
Helix	485
Hexahedron.	488
Interpolation Curve.	488
Interval	490
Parametric Curve	490
Parametric Surface	492
Point	494
Polygon	494
Pyramid	495
Rectangle.	496
Sphere.	497
Square.	498
Tetrahedron.	499
Torus	500

Geometry Operations	501
Array	501
Chamfer	502
Compose.	502
Convert to Curve	503
Convert to Point	504
Convert to Solid	504
Convert to Surface	505
Copy	506
Cross Section	507
Deformed Configuration	507
Delete Entities.	508
Difference	509
Edit Object	510
Extrude	511
Fillet	513
Import.	513
Intersection.	516
Mirror.	517
Move	518
Partition	519
Revolve	520
Rotate.	522
Scale	523
Split.	524
Sweep	525
Tangent	527
Union	528
Work Plane	529
Virtual Geometry Operations	532
Ignore Vertices	533
Ignore Edges	534
Ignore Faces	534
Form Composite Edges	535
Form Composite Faces	535
Form Composite Domains	536
Collapse Edges.	537

Collapse Faces	537
Merge Edges	538
Merge Vertices	538
Mesh Control Operations	539
Mesh Control Vertices	539
Mesh Control Edges	539
Mesh Control Faces	540
Mesh Control Domains	541

Chapter 7: Meshing

Creating Meshes	468
About Mesh Elements for 1D, 2D, and 3D Geometries	468
Adding Meshing Sequences	469
Adding and Building Meshing Operations	469
Editing Meshing Nodes	470
Errors and Warnings in Meshing Sequences	471
Deleting, Clearing, Disabling, and Enabling Meshes	472
Using Several Meshing Sequences of Imported Mesh Type	472
The Meshing Toolbar	473
Meshing Techniques	477
Physics-Controlled Meshing	477
User-Controlled Meshing	478
2D Meshing Options	478
3D Meshing Options	479
Free Meshing	480
About Free Meshing	480
Generating a 2D Free Mesh	481
Generating a 3D Free Tetrahedral Mesh	482
Troubleshooting Free Tetrahedral Mesh Generation	485
Structured Meshes	487
About 2D Mapped Meshing	487

About 2D and 3D Boundary Layer Meshes	487
About 3D Swept Meshes	488
Generating a 3D Swept Mesh	490
Mesh Control Entities	494
About Mesh Control Entities	494
Using Mesh Control Entities to Control Element Size	494
Using Structured and Unstructured Mesh with Boundary Layers	496
Importing and Exporting Meshes	500
About Mesh Export, Import, and Operations on Imported Meshes	500
Exporting a Mesh.	500
Importing Meshes	501
Operations on Imported Meshes.	502
Using Operations on an Imported Mesh	502
Ball	506
Box	507
Create Vertex	508
Cylinder	508
Delete Entities.	509
Finalize	510
Import.	510
Join Entities	512
Logical Expression	513
Backward Compatibility	514
Mesh Object	514
Mesh Element Quality and Size	515
Displaying Mesh Element Quality and the Mesh Element Size	515
Mesh Statistics.	516
Meshing Operations	518
Boundary Layers	518
Convert	520
Copy Domain (2D, 3D), Copy Face (3D), and Copy Edge (2D, 3D)	521
Edge	524
Free Quad	524
Free Tetrahedral	525

Free Triangular	526
Mapped	528
Reference	529
Refine	530
Swept	531
Mesh Attributes	535
Boundary Layer Properties	535
Corner Refinement	536
Distribution	537
Edge Groups	539
Edge Map	540
One-Point Map	540
Scale	541
Size	542
Two-Point Map	545
Avoiding Inverted Mesh Elements	546
Inverted Mesh Elements	546
Using Linear Geometry Shape Order	547
Modifying the Geometry or Mesh	547
Visualizing Inverted Mesh Elements	547

Chapter 8: Materials

Materials Overview	550
About Materials and Material Properties	550
About the Material Databases	551
About Using Materials in COMSOL	553
Opening the Material Browser and Adding Materials	555
User-Defined Materials and Libraries	557
Adding New Materials to a Model	557
Adding an External Material Library	557
Adding a New User-Defined Library	558
Restoring a Deleted User-Defined Library	559

Material Nodes and Material Browser	560
The Materials Branch	560
The Material Browser Window	561
The Material Settings Window.	562
Property Groups	568
Material Properties Reference	571
About the Output Material Properties.	571
Model Inputs	584
Using Functions in Materials	586
Adding a Function to the Material	586
Example of Defining an Analytic Function.	587
Module-Specific Material Databases	590
AC/DC Material Database	590
Liquids and Gases Material Database	591
MEMS Material Database	592
Piezoelectric Materials Database	593
Piezoresistivity Materials Database	594
Batteries and Fuel Cells Materials Database.	595
Semiconductor Materials Database	596
References for the Material Databases.	596

Chapter 9: Overview of the Physics

The Physics User Interfaces	600
Introduction to the Physics User Interfaces	600
Physics Groups in the Model Wizard	601
Physics Guide	602
Selecting Physics	605
Show More Physics Options	607
Equations for the Physics	610
Physics Nodes—Equation Section	610
Physics Nodes Advanced Settings	612

Creating Multiphysics Models	613
Multiphysics Modeling Approaches	613
Using Predefined Multiphysics	613
Adding Physics Sequentially	614
Building a Multiphysics Model Directly	614
Adding Multiphysics Couplings	615
Deleting Physics User Interfaces	616
Chapter 10: The AC/DC User Interfaces	
The Electromagnetics User Interfaces	618
Fundamentals of Electromagnetics	619
Maxwell's Equations	619
Constitutive Relations	620
Potentials	621
Material Properties	622
About the Boundary and Interface Conditions	623
Electromagnetic Forces	624
References for Electromagnetic Theory	624
Theory of Electric Fields	625
Charge Relaxation Theory	625
Theory of Electrostatics	628
Electrostatics Equations	628
The Electrostatics Interface in Time-Dependent or Frequency-Domain Studies	629
Theory of Electric Currents	630
Electric Currents Equations in Steady State	630
Theory of Magnetic and Electric Fields	632
Maxwell's Equations	632
Magnetic and Electric Potentials	632
Gauge Transformations	633

Selecting a Particular Gauge.	633
The Gauge and the Equation of Continuity for Dynamic Fields.	634
Time-Harmonic Magnetic Fields	634
Theory of Magnetic Fields	636
Magnetostatics Equation	636
Frequency Domain Equation	636
Transient Equation	637
The Electrostatics User Interface	638
Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface	639
Charge Conservation	641
Initial Values.	642
Space Charge Density	643
Zero Charge	643
Ground	644
Electric Potential	645
Surface Charge Density	645
External Surface Charge Accumulation	646
Electric Displacement Field	647
Periodic Condition	648
Thin Low Permittivity Gap	648
Line Charge.	649
Line Charge (on Axis)	650
Line Charge (Out-of-Plane)	651
Point Charge	651
Point Charge (on Axis)	652
Change Cross-Section	653
Change Thickness (Out-of-Plane).	654
The Electric Currents User Interface	655
Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface	656
Current Conservation.	658
Initial Values.	660
External Current Density.	661
Current Source	661

Electric Insulation	662
Boundary Current Source	663
Normal Current Density	663
Distributed Impedance.	664
Contact Impedance	665
Sector Symmetry.	667
Line Current Source	668
Line Current Source (on Axis).	668
Point Current Source	669
Point Current Source (on Axis)	669

The Magnetic Fields User Interface 671

Domain, Boundary, Point, and Pair Nodes for the	
Magnetic Fields Interface	673
Ampère's Law	674
Initial Values.	676
External Current Density.	677
Velocity (Lorentz Term)	677
Magnetic Insulation	679
Magnetic Field	680
Surface Current	681
Magnetic Potential	681
Perfect Magnetic Conductor	682
Line Current (Out-of-Plane)	683

Chapter II: Pressure Acoustics

Fundamentals of Acoustics 686

Acoustics Explained.	686
Examples of Standard Acoustics Problems	687
Mathematical Models for Acoustic Analysis	689

Theory for Pressure Acoustics, Frequency Domain 692

Frequency Domain Study.	692
Eigenfrequency Study	694
References for the Pressure Acoustics, Frequency Domain Interface . .	694

The Pressure Acoustics, Frequency Domain User Interface	696
Domain, Boundary, Edge, Point, and Pair Nodes for the	
Pressure Acoustics, Frequency Domain Interface	700
Pressure Acoustics Model	701
Monopole Source	702
Dipole Source	702
Initial Values	702
Sound Hard Boundary (Wall)	703
Normal Acceleration	703
Sound Soft Boundary	704
Pressure	705
Impedance	705
Symmetry	706
Plane Wave Radiation	706
Spherical Wave Radiation	707
Cylindrical Wave Radiation	708
Incident Pressure Field	708
Periodic Condition	709
Interior Sound Hard Boundary (Wall)	709
Axial Symmetry	710
Continuity	710

Chapter 12: Chemical Species Transport

Theory for Transport of Diluted Species	714
Mass Balance Equation	714
Convective Term Formulation	715
Solving a Diffusion Equation Only	716
Crosswind Diffusion	716
Reference	717
The Transport of Diluted Species User Interface	718
Domain, Boundary, and Pair Nodes for the Transport of Diluted	
Species User Interface	721
Dynamic Transport Feature Node	722
Initial Values	723

Reactions	724
No Flux	724
Concentration	725
Flux	725
Inflow	726
Outflow	727
Symmetry	727
Flux Discontinuity	727
Periodic Condition	728
Thin Diffusion Barrier	729
Thin Impermeable Barrier	729

Chapter 13: Fluid Flow

Theory of Laminar Flow	732
General Single-Phase Flow Theory	732
Compressible Flow	735
The Mach Number Limit	735
Incompressible Flow	736
The Reynolds Number	736
Numerical Stability—Stabilization Techniques for Fluid Flow	737
Pseudo Time Stepping for Laminar Flow Models	738
The Projection Method for the Navier-Stokes Equations	740
The Boussinesq Approximation	741
Theory for the Wall Boundary Condition	742
Prescribing Inlet and Outlet Conditions	744
Theory for the Pressure, No Viscous Stress Boundary Condition	745
Theory for the Normal Stress Boundary Condition	746
Discontinuous Galerkin Formulation	747
Particle Tracing in Fluid Flow	747
References for the Single-Phase Flow, Laminar Flow User Interfaces	748
Single-Phase Flow, Laminar Flow User Interface	750
The Laminar Flow User Interface	750
Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow	753
Fluid Properties	754

Volume Force	754
Initial Values.	755
Wall.	755
Inlet.	758
Outlet.	759
Symmetry	761
Open Boundary	762
Boundary Stress	762
Periodic Flow Condition	763
Flow Continuity	765
Pressure Point Constraint	765

Chapter 14: Heat Transfer Modeling

Heat Transfer Theory	768
What is Heat Transfer?	768
The Heat Equation	768
A Note on Heat Flux	771
Heat Flux and Heat Source Variables	773
About the Boundary Conditions for the	
Heat Transfer User Interfaces	776
Radiative Heat Transfer in Transparent Media	779
Consistent and Inconsistent Stabilization Methods for the Heat	
Transfer User Interfaces	781
References for the Heat Transfer User Interfaces	782
The Heat Transfer User Interface	783
Domain, Boundary, Edge, Point, and Pair Nodes for the Heat	
Transfer User Interfaces	786
Heat Transfer in Solids.	786
Translational Motion	789
Heat Transfer in Fluids.	790
Initial Values.	793
Heat Source.	793
Thermal Insulation	796
Temperature	796

Outflow	797
Symmetry	798
Heat Flux.	798
Surface-to-Ambient Radiation	800
Periodic Heat Condition	801
Boundary Heat Source.	801
Continuity	803
Thin Thermally Resistive Layer.	803
Line Heat Source.	805
Point Heat Source	806
The Joule Heating User Interface	807
Domain, Boundary, Edge, Point, and Pair Nodes for Joule Heating	810
Joule Heating Model.	812
Electromagnetic Heat Source	814
Initial Values.	814
Boundary Electromagnetic Heat Source	815
About Frames in Heat Transfer	816
Frame Physics Feature Nodes and Definitions	816
Conversion Between Material and Spatial Frames	819

Chapter 15: Structural Mechanics

Solid Mechanics Geometry and Structural Mechanics Physics	
Symbols	824
3D Solid Geometry	824
2D Geometry	825
Axisymmetric Geometry	826
Physics Symbols for Boundary Conditions	826
About Coordinate Systems and Physics Symbols	828
Displaying Physics Symbols in the Graphics Window—An Example	828
The Solid Mechanics User Interface	831
Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics	833
Linear Elastic Material	834

Change Thickness	836
Damping	837
Initial Values.	837
About the Body, Boundary, Edge, and Point Loads	838
Body Load	838
Boundary Load	839
Edge Load	840
Point Load	840
Fixed Constraint	841
Prescribed Displacement	841
Free.	843
Roller	843
Periodic Condition	844
 Theory of Solid Mechanics	 845
Material and Spatial Coordinates	845
Coordinate Systems.	846
Lagrangian Formulation	847
About Linear Elastic Materials	847
Strain-Displacement Relationship.	850
Stress-Strain Relationship.	851
Plane Strain and Plane Stress Cases	851
Axial Symmetry	852
Loads	853
Pressure Loads	855
Equation Implementation.	855
Setting up Equations for Different Studies	856
Damping Models	858

Chapter 16: Equation-Based Modeling

 The Mathematics User Interfaces	 862
 Modeling with PDEs	 864
About Equation Forms.	864
Notational Conventions	865

PDE User Interface Variables	867
The General Form PDE	867
The Coefficient Form PDE	869
Multiple Dependent Variables—Equation Systems	873
Solving Time-Dependent Problems	875
Solving Eigenvalue Problems.	877
About Weak Form Modeling	878
Introduction to the Weak Form	879
The Weak Form PDE	881
Specifying and Interpreting Boundary Conditions.	883
Symmetric and Nonsymmetric Constraints	887
The PDE User Interfaces	890
Creating a Model Using a PDE User Interface	890
Discretization Section Shape Function Types and Element Orders	892
The Coefficient Form PDE User Interfaces	893
The General Form PDE User Interfaces	895
The Weak Form PDE User Interfaces	896
The Classical PDE User Interfaces	897
Domain, Boundary, Pair, Edge, and Point Conditions for PDEs.	898
Initial Values.	899
Coefficient Form PDE	899
General Form PDE	901
Weak Form PDE	901
Source, Edge Source, and Point Source	902
Classical PDE Domain Nodes	903
Dirichlet Boundary Condition	904
Constraint	905
Flux/Source	906
Zero Flux	906
Periodic Condition	907
Destination Selection	907
Theory for the Wave Form PDE	909
Derivation of the Weak Form of the Wave Form PDE.	909
Time Explicit Integrator	911
Local Time Stepping.	912
Reference for the Wave Form PDE User Interface	913

The Wave Form PDE	914
Domain and Boundary Physics for the Wave Form PDE User	
Interface	916
Wave Form PDE	916
Initial Values.	919
Zero Flux	920
Flux/Source	920
Interior Source	921
Interior Flux	921
About Auxiliary Equation-Based Nodes	923
Weak Contribution (PDEs)	923
Weak Contributions on Mesh Boundaries	925
Auxiliary Dependent Variable	926
About Explicit Constraint Reaction Terms	927
Pointwise Constraint	929
Weak Constraint.	931
Discretization (Node)	934
Modeling with ODEs and DAEs	935
Adding ODEs, DAEs, and Other Global Equations	935
Solving ODEs: An Example	936
Solving Algebraic and Transcendental Equations: An Example	936
Distributed ODEs and DAEs	937
The ODE and DAE User Interfaces	938
The Global ODEs and DAEs User Interface.	938
About ODEs, Initial-Value Problems, and Boundary-Value Problems	938
Global Equations	939
Global Constraint	941
Weak Contribution (ODEs and DAEs).	941
The Distributed ODEs and DAEs User Interfaces	942
Distributed ODE.	943
Algebraic Equation	943
The Events User Interface	945
Discrete States	945
Indicator States	946

Explicit Event	946
Implicit Event	947
Reinitialization on Domains, Boundaries, Edges, or Points	947
The Wall Distance User Interface	948
Domain and Boundary Nodes for the Wall Distance User Interface	949
Distance Equation	949
Initial Values.	950
Wall.	950
Theory for Wall Distance	951
The Eikonal Equation	951
Modified Eikonal Equation	951
Reference for the Wall Distance User Interface	952
Curvilinear Coordinates	953
Introduction	953
The Curvilinear Coordinates User Interface	953
Diffusion Method.	955
Elasticity Method.	956
Flow Method	956
User Defined	957
Inlet.	957
Jump	958
Outlet.	958
Wall.	958
Coordinate System Settings.	959

Chapter 17: Sensitivity Analysis

Theory for the Sensitivity User Interface	962
About Sensitivity Analysis	962
Sensitivity Problem Formulation	962
Theory for Stationary Sensitivity Analysis	963
Specification of the Objective Function	965
Choosing a Sensitivity Method.	965

Issues to Consider Regarding the Control Variables	966
Issues to Consider Regarding the Objective Function	968
The Sensitivity User Interface	969
Integral Objective	970
Probe Objective	971
Control Variable Field	971
Global Objective	972
Global Control Variable	972
Adding a Sensitivity User Interface to a Model.	972

Chapter 18: Deformed Geometry and Moving Mesh

Deformed Mesh Fundamentals	976
About Deformed Meshes.	976
Deformed Geometry vs. Moving Mesh	976
Arbitrary Lagrangian-Eulerian Formulation (ALE).	977
About Frames	978
Mathematical Description of the Mesh Movement	981
Derivatives of Dependent Variables	982
Smoothing Methods.	984
Limitations of the ALE Method	985
Tips for Modeling Using Deformed Meshes	985
Remeshing a Deformed Mesh	986
Moving Mesh User Interface	990
Domain and Boundary Nodes in the Moving Mesh User Interface	992
Fixed Mesh	992
Prescribed Mesh Displacement	992
Free Deformation	993
Prescribed Deformation	993
Prescribed Mesh Velocity.	994
Prescribed Normal Mesh Velocity	994
Zero Normal Mesh Velocity.	995
Zero Normal Mesh Displacement	996

Deformed Geometry User Interface	997
Domain and Boundary Nodes for Deformed Geometry	998
Fixed Mesh	998
Prescribed Mesh Displacement	999
Free Deformation	1000
Prescribed Deformation	1000
Prescribed Mesh Velocity	1000
Prescribed Normal Mesh Velocity	1001
Zero Normal Mesh Velocity.	1002
Zero Normal Mesh Displacement	1002

Chapter 19: Studies and Solvers

Studies and the Study Nodes	1004
Introduction to Solvers and Studies	1004
Adding a Study.	1004
Study	1005
Overview of the Study Types	1006
Adding Study Step Nodes	1010
Study Steps and Solver Configurations.	1011
Common Study Step Settings	1013
Working with Studies and Solvers	1014
Selecting Physics and Variables in the Study Steps	1014
Generating Solver Configurations and Job Configurations	1018
Editing Solver Configurations	1019
Clearing, Deleting, Disabling, and Enabling a Solver or Solutions	1019
Computing a Solution	1020
Updating the Model.	1021
Computing the Initial Values	1021
Getting Results While Solving	1022
Progress and Log Information	1022
Convergence Plots	1027
Specifying Values of Dependent Variables.	1028
Remarks on Solver-Related Model Characteristics	1030

Study Step Types	1032
Stationary	1032
Time Dependent	1034
Time Discrete	1037
Time-Dependent Modal	1037
Eigenfrequency	1037
Eigenvalue	1039
Frequency Domain	1040
Frequency Domain Modal	1041
Multigrid Level	1042
Parametric Sweep	1043
Batch	1046
Batch Sweep	1047
Cluster Computing (Study)	1049
Cluster Sweep	1054
Time-Dependent with Initialization, Fixed Geometry	1057
Time-Dependent with Initialization	1058
Time-Dependent, Fixed Geometry	1058
Current Distribution Initialization	1059
Stationary Plug Flow	1059
Frequency-Domain, Perturbation	1059
Small-Signal Analysis, Frequency Domain	1060
Prestressed Frequency Analyses Study	1061
Prestressed Analysis, Eigenfrequency	1061
Prestressed Analysis, Frequency Domain	1062
AC Impedance Stationary	1063
AC Impedance Time Dependent	1063
Linear Buckling	1064
Mode Analysis	1065
Modal Reduced Order Model	1066
Stationary, One-Way Coupled	1067
Stationary, Fluid (Study Step)	1067
Stationary, Solid (Study Step)	1067
Stationary, One-Way Coupled with Initialization	1067
Time Dependent, One-Way Coupled	1068
Time Dependent, Fluid (Study Step)	1068
Time Dependent, Solid (Study Step)	1068
Transient, One-Way Coupled with Initialization	1069

Stationary with Initialization	1069
Transient with Initialization	1070
Wall Distance Initialization	1071
Boundary Mode Analysis	1071
Frequency-Stationary	1072
Frequency-Transient	1073
Mean Energies	1073
Reduced Electric Fields	1074
Particle Trajectories (Study)	1074
Coil Current Calculation	1074
Cyclic Voltammetry	1075
Sensitivity (Study Step)	1075
Optimization (Study Step)	1078
Frozen Rotor	1078
Solver Configurations	1079
The Solver Configurations Node	1079
Show Default Solver	1079
Adding Solver Nodes	1079
Running a Solver Configuration	1079
Errors and Warnings	1080
Solver Overview	1081
Solver Configurations	1081
Solver Operation, Attribute, and Utility Nodes	1081
Solution Operation Nodes and Solvers	1086
Selecting a Stationary, Time-Dependent, or Eigenvalue Solver	1086
AWE Solver	1087
Eigenvalue Solver	1090
Modal Solver	1092
Optimization Solver	1099
Plug Flow Solver	1099
Stationary Solver	1100
Time-Dependent Solver	1106
Time Discrete Solver	1115
Time Explicit Solver	1117
Dependent Variables	1119

References for the Solution Operation Nodes and Solvers	1124
---	------

Solution Attribute Nodes	1126
Adaptive Mesh Refinement	1126
Advanced.	1132
AMS	1137
Automatic Remeshing	1138
Coarse Solver	1139
Control Field	1139
Control State	1140
Direct	1140
Domain Decomposition	1146
Domain Solver.	1149
Field	1149
Fully Coupled	1150
Incomplete LU.	1154
Iterative	1157
Jacobi	1164
Krylov Preconditioner.	1165
Lower Limit.	1167
Lumped Step	1167
Multigrid	1168
Parametric	1174
Postsmoother	1177
Presmoother	1177
Previous Solution.	1177
SCGS	1178
Segregated	1181
Segregated Step	1184
Sensitivity	1187
SOR.	1189
SOR Gauge	1191
SOR Line.	1193
SOR Vector.	1195
State	1198
Stationary Acceleration	1199
Stop Condition	1200
Time Parametric	1202

Vanka	I203
References for the Linear System Solvers and the Preconditions	I207
Solution Utility Nodes I209	
Adaptive Mesh Refinement	I209
Assemble.	I209
Compile Equations	I210
The Statistics Page	I211
Input Matrix.	I212
Store Solution	I212
State Space	I213
Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis I215	
Frequency-Domain, Perturbation Study and Study Step	I215
Harmonic Perturbation—Exclusive and Contributing Nodes	I216
Job Configurations I218	
Parametric	I219
Job Configuration.	I220
Solver	I220
External Class	I221
Geometry Sequence	I221
Meshing Sequence	I221
Save Model to File	I221
Plot Group	I222
Derived Value	I222
Export to File	I222
Batch	I223
Batch Data	I225
External Process	I225
Cluster Computing (Job Configuration)	I226
Using a Job Configuration to Store Parametric Results on File	I229

Chapter 20: Results Analysis and Plots

Postprocessing and Analyzing Results	I234
Results	I235
About the Results Branch	I235
Common Results Node Settings	I236
Selecting a Data Set for Plots	I239
Entering Axis Data for a Data Set	I240
Vector Inputs for Parametric Solver and Parametric Sweep Studies	I240
Expressions and Predefined Quantities.	I242
Defining Plane Data for a Data Set	I243
Plot Titles for Plot Groups and Plot Types	I244
Legends	I245
Using Special Formats and Symbols in Titles.	I246
Arrow Positioning	I248
Principal Components and Positioning.	I248
Defining the Number of Levels	I249
Selecting Color Tables.	I249
Defining the Color and Data Ranges	I251
Defining the Coloring and Style	I251
Defining Element Filters	I257
Defining Shrinking of Elements.	I257
Entering Quality Settings for Plot Settings Windows	I258
Inheriting Style Options	I259
Integration Settings for a Derived Value	I260
Data Series Operation Settings for a Derived Value.	I261
Small-Signal Analysis, Prestressed Analysis, and	
Harmonic Perturbation Plot Settings	I262
Node Properties for Reports	I263
Data Sets	I264
About Data Sets	I264
Adding a Data Set to the Model Builder	I264
Overview of Data Set Types	I265
Average or Integral	I267
Contour	I268

Cut Line 2D and Cut Line 3D	I269
Cut Plane.	I270
Cut Point 1D, Cut Point 2D, and Cut Point 3D	I271
Edge 2D and Edge 3D	I273
Function 1D, Function 2D, and Function 3D.	I273
Isosurface	I274
Join	I274
Maximum and Minimum	I276
Mesh	I276
Mirror 2D and Mirror 3D	I276
Parameterized Curve 2D and Parameterized Curve 3D	I277
Parameterized Surface.	I278
Parametric Extrusion 1D and Parametric Extrusion 2D	I279
Particle	I280
Revolution 1D and Revolution 2D	I281
Sector 2D and Sector 3D	I282
Solution	I284
Surface	I285
 Plot Groups and Plots	 I286
About the Plot Groups	I286
Adding Plots To Plot Groups	I287
Color Coding for Plot Groups and Plot Types.	I293
Working with Plot Windows	I293
Creating Cross-Section Plots and Combining Plots	I294
Plotting and Cross-Section Interactive Toolbar.	I295
1D, 2D, and 3D Cross-Section Point Plots	I298
2D Cross-Section Line Plots	I300
3D Cross-Section Line Plots	I302
3D Cross-Section Surface Plot.	I305
1D Plot Group and Polar Plot Group	I308
2D Plot Group and 3D Plot Group	I310
Arrow Line	I312
Arrow Surface.	I313
Arrow Volume.	I313
Contour	I313
Coordinate System Volume, Coordinate System Surface, and Coordinate System Line	I314

Far Field	1315
Global	1317
Histogram	1319
Isosurface	1321
Line Graph	1321
Line	1322
Matrix Histogram	1322
Max/Min Volume, Max/Min Surface, and Max/Min Line	1323
Mesh	1324
Multislice	1325
Nyquist	1326
Particle Tracing	1326
Particle Tracing with Mass	1330
Particle Trajectories	1333
Filter Node for Particle Trajectories	1333
Phase Portrait	1334
Poincaré Map	1335
Point Graph	1336
Principal Stress Volume	1336
Principal Stress Surface	1337
Scatter Surface and Scatter Volume	1337
Slice	1338
Streamline	1339
Surface	1345
Table Graph	1345
Table Surface	1346
Volume	1347
Color Expression	1347
Deformation	1347
Filter	1348
Height Expression	1349
Derived Values and Tables	1350
About Derived Values	1350
Derived Value Types	1351
Editing and Organizing Results Tables	1352
Volume Average, Surface Average, and Line Average	1354
Volume Integration, Surface Integration, and Line Integration	1355

Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum	1356
Point Evaluation	1357
Global Evaluation	1357
Global Matrix Evaluation	1357
System Matrix	1359
Particle Evaluation	1360
Table	1361
Exporting Data and Images	1364
Export Types	1364
About the Sectionwise Data Format for Data Export	1364
Animation	1365
Data	1370
Mesh	1372
Table	1373
ID Image, 2D Image, or 3D Image	1373
Player	1375
Plot	1376
Reports	1378
About the Report Generator	1378
Generating a Model Report	1378
Report Types	1379
The Report Node	1379
The Documentation Node	1381
The Title Page	1382
The Table of Contents	1383
Sections in the Report	1383
Basic Report Components	1384
Mathematical Symbols and Special Characters	1386
Model Report Components	1397
Definitions Report Nodes	1397
Results Report Nodes	1399
Geometry Report Node	1400
Material Report Node	1401
Mesh Report Node	1401
Model Report Node	1402

Parameters Report Node	1402
Physics Interface Report Node	1402
Root Report Node	1403
Solver Report Node	1404
Study Report Node.	1404

Chapter 21: Running COMSOL

Running COMSOL	1406
COMSOL Client/Server	1406
Parallel COMSOL	1407
COMSOL API	1408
COMSOL Batch	1408
LiveLink for MATLAB	1409
COMSOL Client/Server Architecture	1410
Standalone COMSOL	1410
Running COMSOL as a Client/Server	1410
Running COMSOL with MATLAB	1411
Running COMSOL Client/Server	1413
Advantages of Using COMSOL Client/Server	1413
Running COMSOL Client/Server.	1413
Running COMSOL in Parallel	1416
Shared-Memory Parallel COMSOL	1416
COMSOL and BLAS	1417
Distributed-Memory Parallel COMSOL	1417
Benefits of Running COMSOL in a Distributed Mode	1419
Running COMSOL in Parallel on Clusters	1419
The COMSOL Commands	1424
COMSOL Commands on Windows.	1424
COMSOL Commands on Linux	1435
COMSOL Commands on Macintosh	1452
The COMSOL Convertpre35a Command	1459

Running COMSOL on the Amazon™ Cloud (Amazon EC2™)	
1461	
Nomenclature	1462
Requirements	1462
Introduction	1462
Signing up for an EC2 Account and Logging In	1463
Launching and Using the Remote Instance	1464
Preparing the COMSOL License Management	1469
Installing COMSOL on the AMI	1471
Terminating the Instance	1472
Connecting a COMSOL Desktop GUI to a Remote Instance	1473
Running a Parametric Sweep on Simple Cluster on the Cloud	1478
Connecting to an EC2 Instance from Linux and Macintosh without Java	1479
Creating a Cloud Cluster Using the StarCluster Utility on Linux	1483
Using the Remote Access Functionality in COMSOL using Linux	1487
Troubleshooting	1490
Copying a Custom COMSOL Version to the EC2 or other cloud services	1491

Chapter 22: Glossary

Glossary of Terms	
	1494

Introduction

Welcome to COMSOL Multiphysics®! This book details features and techniques that help you throughout all of your COMSOL modeling in Version 4.3b using the COMSOL Desktop® environment. For example, detailed information about how to build model geometries in COMSOL, 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, documentation set, and other resources.

About COMSOL Multiphysics

COMSOL Multiphysics® is a powerful interactive environment used to model and solve all kinds of scientific and engineering problems. The software provides a powerful integrated desktop environment with a *Model Builder* that gives you a 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 user 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 then internally compiles a set of equations representing the entire model.

You access the power of COMSOL as a standalone product through a flexible graphical user interface (GUI) or by script programming in Java® or the MATLAB® language (requires the 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 assembles and solves the problem using a set of advanced numerical analysis tools. The software runs the 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.

COMSOL 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 in many application areas, including:

- Acoustics
- Bioscience
- Chemical reactions
- Corrosion and corrosion protection
- Diffusion
- Electrochemistry
- Electromagnetics
- Fatigue analysis
- Fluid dynamics
- Fuel cells and electrochemistry
- Geophysics and geomechanics
- Heat transfer
- Microelectromechanical systems (MEMS)
- Microfluidics
- Microwave engineering
- Multibody dynamics
- 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 [Creating Multiphysics Models](#) section discusses multiphysics modeling techniques. Many predefined user interfaces provide easy-to-use entry points for common multiphysics applications.

In its base configuration, COMSOL offers modeling and analysis power for many application areas. For several of the key application areas there are also optional 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 and Interfacing Options

The optional modules, including interfacing options such as the CAD Import Module and bidirectional interfaces such as the LiveLink™ products, are optimized for specific application areas and offer discipline-standard terminology and physics user interfaces. For some modules, additional material libraries, specialized solvers, element types, and visualization tools are also available.



For up-to-date module availability, product descriptions, and a specification chart, go to www.comsol.com/products/multiphysics

Overview of the Reference Manual

The *COMSOL Multiphysics Reference Manual* provides comprehensive information about all modeling steps using COMSOL Multiphysics®. See the individual module manuals for information specific to a specialized Module (see [The COMSOL Modules and Interfacing Options](#) for a list).



As detailed in the section [Where Do I Access the Documentation and Model Library?](#) this information can also be searched from the **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 COMSOL Desktop Environment](#) and an introduction to [The Model Wizard and Model Builder](#).

DEFINITIONS

The [Definitions](#) chapter describes the global and local (model) definitions features. Depending on the geometric scope, you add the 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 Constants](#), [Predefined and Built-In Variables](#), [User-Defined Functions](#), [Coordinate Systems](#), [Model Couplings](#), and [Probes](#).

VISUALIZATION AND SELECTION

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 [Selecting and Visualizing in Models](#), [Named 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. 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 [Working with Geometry Sequences](#), [The Geometry and CAD Environment](#), and [Creating a Geometry for Analysis](#).

MATERIAL

The [Materials](#) chapter introduces you to the material databases included with COMSOL. Topics include a [Materials Overview](#), [User-Defined Materials and Libraries](#), [Material Nodes and Material Browser](#), [Using Functions in Materials](#), and [Module-Specific Material Databases](#).

MODELING

[Building a COMSOL Model](#) explains a range of methods and topics including information about: [The Model Wizard and Model Builder](#), details about [Model Administration](#), [The Physics Nodes](#), [Specifying Model Equation Settings](#), [Advanced Physics Sections](#), [Using Load Cases](#), [Using Units](#), and [Numerical Stabilization](#), for example.

PHYSICS OVERVIEW

The [Overview of the Physics](#) chapter briefly describes [The Physics User Interfaces](#) as well as some general [Equations for the Physics](#). Also learn about [Creating Multiphysics Models](#).

AC/DC

The [AC/DC User Interfaces](#) chapter explains the physics user 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 [Pressure Acoustics](#) 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](#) 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](#) 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 Modeling](#) 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](#) 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 MODELING

The [Equation-Based Modeling](#) chapter describes the use of the mathematics interfaces, found under the Mathematics branch () 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 ANALYSIS

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.

DEFORMED MESHES

The [Deformed Geometry and Moving Mesh](#) 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. It also explains the possibilities for importing and exporting meshes in different formats. Topics include [Creating Meshes](#), [Free Meshing](#), [Structured Meshes](#), [Meshing Operations](#), [Importing and Exporting Meshes](#), and [Mesh Element Quality and Size](#).

STUDIES AND SOLVERS

The [Studies and Solvers](#) chapter lists the various types of solvers and studies in COMSOL 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 *Optimization Module Manual* for other supplementary information.

RESULTS AND VISUALIZATION

The [Results Analysis and Plots](#) chapter helps you analyze results in COMSOL and describes numerous result-evaluation and visualization tools, including advanced graphics, data display, and export functions. Topics include [Postprocessing and Analyzing Results](#), [Data Sets](#), [Plot Groups and Plots](#), [Exporting Data and Images](#), and [Reports](#).

The COMSOL Documentation Set

About the Documentation Set

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

- *Introduction to COMSOL Multiphysics*—information about version 4.3b and how to build models using the desktop environment.
- *COMSOL License Agreement*—the license agreement.
- *COMSOL Installation 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 Reference Manual*—the book you are reading covers the functionality of COMSOL Multiphysics across its entire range from geometry modeling to results evaluation and visualization, including the user interfaces for physics and equation-based modeling. It serves as a tutorial and a reference guide to using COMSOL Multiphysics. This book reviews geometry, mesh, solver, and results functionality and provides detailed information about the settings and options. Additionally, it describes some advanced functionality and settings in COMSOL Multiphysics and provides background material and references.
- *COMSOL API for use with Java® Reference Manual*—this book provides details about features and techniques that help you control COMSOL Multiphysics using its application programming interface (API). The COMSOL 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 includes a manual as described in [The COMSOL Modules and Interfacing Options](#). The documentation for the optional CAD Import Module and LiveLinks to CAD packages is available in separate manuals, 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 the capabilities of COMSOL from the MATLAB programming environment.

Where Do I Access the Documentation and Model Library?

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



If you are reading the documentation as a PDF file on your computer, the [blue links](#) do not work to open 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 Reference Manual* describes all user interfaces and functionality included with the basic COMSOL Multiphysics license. This book also has instructions about how to use COMSOL and how to access the documentation electronically through the COMSOL Help Desk.

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

- Press F1 or select **Help>Help** () from the main menu for context help.
- Press Ctrl+F1 or select **Help>Documentation** () from the main menu for opening the main documentation window with access to all COMSOL documentation.
- Click the corresponding buttons (or) on the main toolbar.

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



If you have added a node to a model you are working on, click the **Help** button () in the node's settings window or press F1 to learn more about it. Under **More results** in the **Help** window there is a link with a search string for the node's name. Click the link to find all occurrences of the node's name in the documentation, including model documentation and the external COMSOL website. This can help you find more information about the use of the node's functionality as well as model examples where the node is used.

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.

In most models, 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 **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 WEBSITES

COMSOL website	www.comsol.com
Contact COMSOL	www.comsol.com/contact
Support Center	www.comsol.com/support
Download COMSOL	www.comsol.com/support/download
Support Knowledge Base	www.comsol.com/support/knowledgebase
Product Updates	www.comsol.com/support/updates
COMSOL Community	www.comsol.com/community

Checking for Software Updates

COMSOL provides software updates that improve the software and correct issues found. To check if 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, select the **Check for updates at launch** check box under **Product updates** on the **Updates** page in the **Preferences** dialog box.

Typographical Conventions

All documentation uses 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 help desk in COMSOL, links to other modules, model examples, and documentation sets also work.
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.
<i>italic</i> font	An <i>italic</i> font indicates the introduction of important terminology. Expect to find an explanation in the same paragraph or in the Glossary. The names of other user guides in the COMSOL documentation set also have an <i>italic</i> font.
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.
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, for example, programming examples (after a monospace code or an italic code font) mean that the content in the string can be freely chosen or entered by the user, such as feature tags. For example, <code>model.geom(<tag>)</code> where <code><tag></code> 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, icons are used to help organize the information. These icons vary in importance, but it is 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 for improving model design, and other information that might be useful.
	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 <i>COMSOL Multiphysics Reference Manual</i> .” Note that if you are in the on-line help, the link works.
	Model	The Model icon is used in the documentation as well as in COMSOL from the View>Model Library menu. If you are working on line, 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. 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/Scattering_and_RCS/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 a user interface, node, or theory section, which are based on space dimension.

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 Multiphysics® modeling environment as controlled by COMSOL Desktop and the tools and windows it provides.

The COMSOL Desktop Environment

Introduction

This section describes the major components in the COMSOL Multiphysics® environment. Those components are integrated into 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 again.

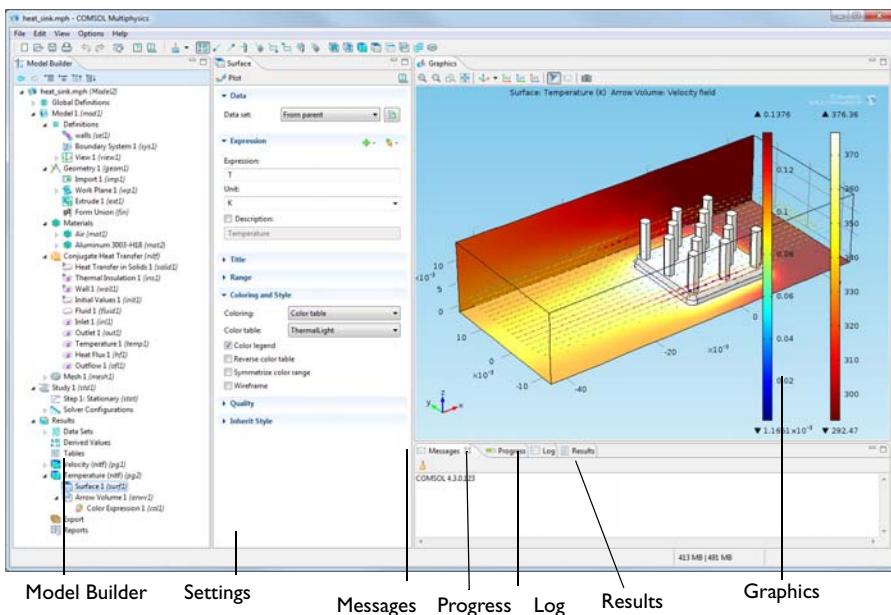


Figure 2-1: The COMSOL Desktop with its major windows in a widescreen layout.

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.

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 window 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 list. Move the mouse to where you want to the group of windows 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; 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 floating window. Select **Detached** again to dock it to 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>** then one of the following:

- **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 the monitor.

CHANGING THE FONT FOR PLOT LABELS AND TITLES

COMSOL uses a default font for texts in plots such as axis labels and titles. 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. 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.



- [Preferences Settings](#)
 - [Key to 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, 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.



- [Keyboard Shortcuts](#)
 - [Key to 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, and Spanish.
- 4 Click **OK**. A message displays indicating that COMSOL 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.



[Preferences Settings](#)

Printing from the COMSOL Desktop

To print the contents in the **Graphics** window or other plot windows, 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 screen shot 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 plot 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 a plot window to a file or the clipboard, click the **Image Snapshot** button () on the Graphics toolbar.



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](#).

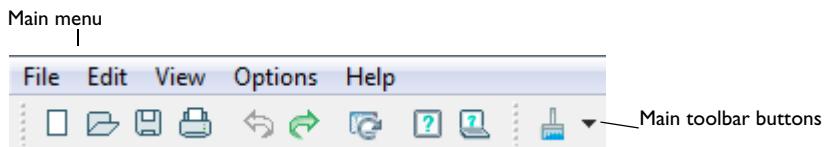


Figure 2-2: The main menu and toolbar buttons.

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. <ul style="list-style-type: none">• 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. <ul style="list-style-type: none">• Keyboard Shortcuts• Copying, Pasting, and Duplicating Nodes• Clearing All Meshes and Clearing all Solutions
View	—	Other COMSOL Desktop Windows
Options	—	<ul style="list-style-type: none">• Preferences Settings• Checking and Controlling Products and Licenses Used
Help	—	Open Help and other online resources, check for updates, and view product information. <ul style="list-style-type: none">• The Help Window• The About COMSOL Multiphysics Box
New or File>New		Create a new model file and open the Model Wizard.

TABLE 2-I: THE MAIN MENU AND TOOLBAR

NAME	BUTTON	DESCRIPTION OR LINK TO MORE INFORMATION
File>New from Physics Builder		Start a new Physics Builder. The option must be enabled under Options>Preferences>BuilderTools. See Builder Tools Preferences .
Open		Open an existing model file located on the computer.
Save or File>Save		Saving COMSOL 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
Help or Help>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 Model Wizard and Model Builder

This section describes the main functionality of the Model Wizard and the Model Builder window on the COMSOL Desktop®.

-
- 
- Other COMSOL Desktop Windows
 - Key to 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*.

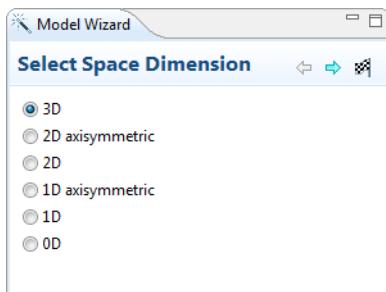


The Model Wizard and Model Builder

THE SELECT SPACE DIMENSION PAGE

Specify the final geometry dimension on the **Select Space Dimension** page—**3D**, **2D axisymmetric**, **2D**, **ID axisymmetric**, **ID**, or **0D**. 0D is used for *physics user 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**

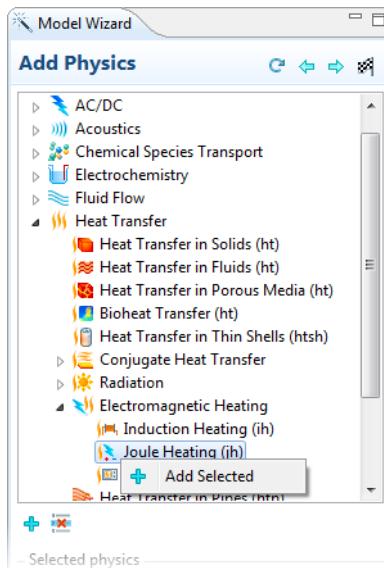
Builder (), but make sure you choose spatial dimensions that this geometry exists in.



THE ADD PHYSICS PAGE

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

The tree organizes the available physics user 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 user interfaces to these application areas as well as attributes to existing physics user interfaces, which are enhanced with additional functionality. The **Recently Used** branch () lists the last five physics user interfaces used in recent modeling sessions.



Selected physics

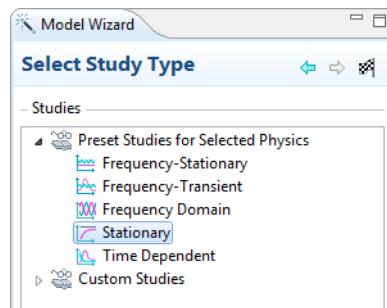
When selecting and adding a physics to the **Selected physics** section, you can review and optionally modify any dependent variable names in the **Dependent variables** section and, for some physics user interfaces, specify the number of dependent variables. For other physics you can edit both the name 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 user 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:

- **Preset Studies**—Study types suggested by a single physics user interface if only one has been chosen.
- **Preset Studies for Selected Physics**—Study types applicable to all physics user interfaces that 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 physics 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 physics being solved for. There is also an empty study type.



The **Selected Physics** table shows all physics 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. Selecting to solve only for some physics 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 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 user interfaces to add to the model. There are also user interfaces for PDEs, ODEs, and DAEs in the **Mathematics** branch. Double-click the physics user interface, click the **Add Selected** () or **Remove Selected** () buttons, or right-click and select **Add Selected**. The **Selected physics** section lists your choices.



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.

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)
	ID
	ID axisymmetric
	2D
	2D axisymmetric
	3D

-
- 
 - 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 node has its own properties and *settings* that are characteristic to it. Branches and subbranches can also contain properties and settings. See [Figure 2-1](#).



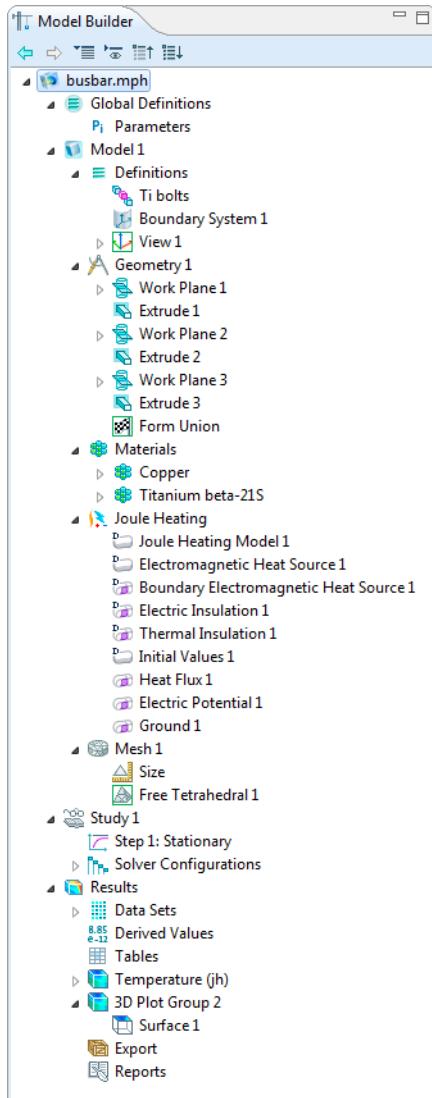
Figure 2-3: The buttons located on the Model Builder window.

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. See also Keyboard Shortcuts .
Next Node		Navigate to the next node in the sequence. See also Keyboard Shortcuts .
Show		Advanced Physics Sections
Expand Sections		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 minimize (left) and maximize (right) 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.



The main branches include:

- The *Global Definitions* branch (): for global definitions, for example, defining parameters and functions that you can use everywhere in the model.
- 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.
- 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.

-
- 
 - [The Model Wizard and Model Builder](#)
 - [Key to Nodes and Toolbar Buttons](#)
 - [Building a COMSOL Model](#)
 - [Studies and Solvers](#)
 - [Results Analysis and Plots](#)
 - [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 user-defined 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 a *Selection* node (), which saves selections of geometric entities (boundaries, for example) that relate to a region or part of the overall geometry for reuse in 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 Material nodes with a defined geometric scope. Material properties required by any of the physics show up automatically in the defined material's settings window.

- Each *physics user interface* forms its own branch below the Materials branch.
- The *Meshes* 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.

-
- 
 - [Geometry Modeling and CAD Tools](#)
 - [Materials](#)
 - [Overview of the Physics](#)
 - [Key to Nodes and Toolbar Buttons](#)
 - [Meshing](#)
-

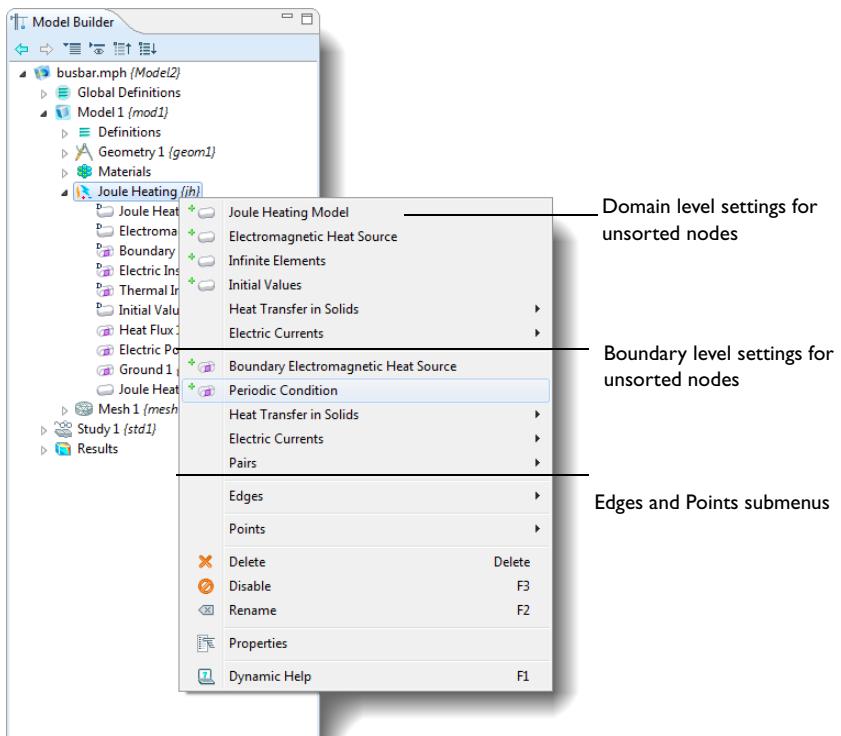
NAVIGATING IN THE MODEL TREE

The following methods are available to select nodes, expand and collapse 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 left-pointing white triangle 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 downward-right pointing black triangle next to the branch icon in the model tree, or press the left arrow key.
- See [Expanding and Collapsing All Nodes](#) for information about how to collapse all branches or to expand all branches.

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. In the context menu, a plus sign next to the icon means a node of that type is added to the Model Builder. After selecting such an option from the list, an associated settings window opens to the right (by default) of the **Model Builder** window (). 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 Macintosh by pressing Ctrl+F10.



 The layout of the context menu depends on whether the nodes are organized by space dimension. The default is unsorted nodes. See [Sorting Nodes by Space Dimension and Type](#) for an example comparing the different context menus.

-
- 
- [Keyboard Shortcuts](#)
 - [Physics Node Context Menu Layout](#)
 - [Key to 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 nodes in the Definitions branch () can be moved relative to other nodes of the same type (functions, selections, and so on).
- The Geometry nodes in the Geometry branch ().
- The Material nodes in the Materials branch ().
- The nodes for physics user interfaces (such as material models, boundary conditions, and sources) in the physics branches, 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 ().
- 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.



For physics 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 whether or not the Ctrl key is pressed.

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, which provide data to the node where they are referenced.

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.

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



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

USING COPY-PASTE FOR NODES

After selecting **Copy** (), click the parent node and from the main menu select, for example, **Edit>Paste Analytic**, or right-click to select **Paste Analytic** () 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 nodes, which can be copied within the same physics user interface or to another identical physics user 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.

-
- 
 - User-Defined Functions
 - Studies and Solvers
 - Plot Groups and Plots
 - Copy
-

Duplicating Nodes

After right-clicking to select **Duplicate** (⊕) (or from the main menu selecting **Edit>Duplicate**), COMSOL 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. Most but not all nodes can be duplicated.

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



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.



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**.



- [Deleting, Clearing, Disabling, and Enabling Meshes](#)
- [Clearing, Deleting, Disabling, and Enabling a Solver or Solutions](#)

Sorting Nodes by Space Dimension and Type



The default setting, found under **Model Builder** in the [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 user 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 user interface, nodes can be organized by space dimension—that is, by geometric entity level: **Domains**, **Boundaries**, **Edges** (3D only), or **Points**.



For physics user interfaces, when either of these options is selected, the way the nodes are organized changes in the Model Builder and 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 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 nodes and user-added nodes are included in the **Model Builder** sequence.

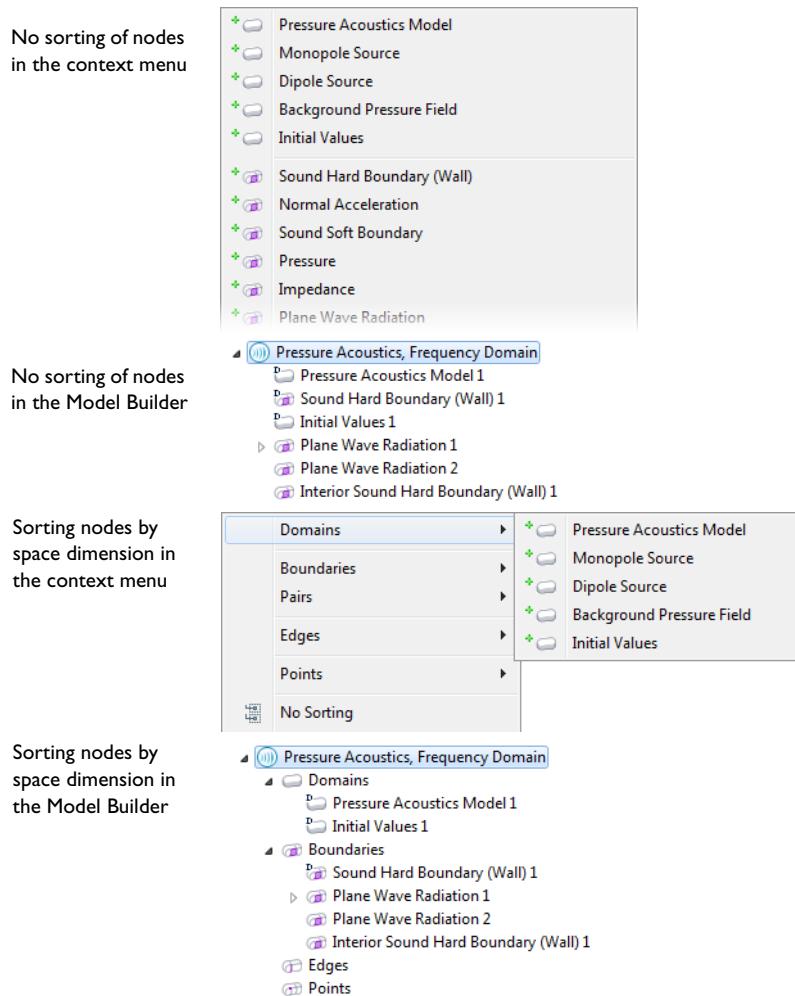


Figure 2-4: The context menu and Model Builder sequence for physics nodes when Sort Nodes by Space Dimension or No Sorting is selected. For the Global Definitions and Definitions nodes, the same principles apply but the nodes are organized by type instead.

ORGANIZING NODES BY TYPE

The same principles apply for the **Global Definitions** and **Definitions** nodes as for the physics 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.



- [Overview of the Physics](#)
 - [Building a COMSOL Model](#)
-

Other COMSOL Desktop Windows

In addition to [The Model Wizard](#) and [The Model Builder Window](#), there are several other COMSOL Desktop® windows that are integral to building your model.

-
- [The Model Wizard and Model Builder](#)



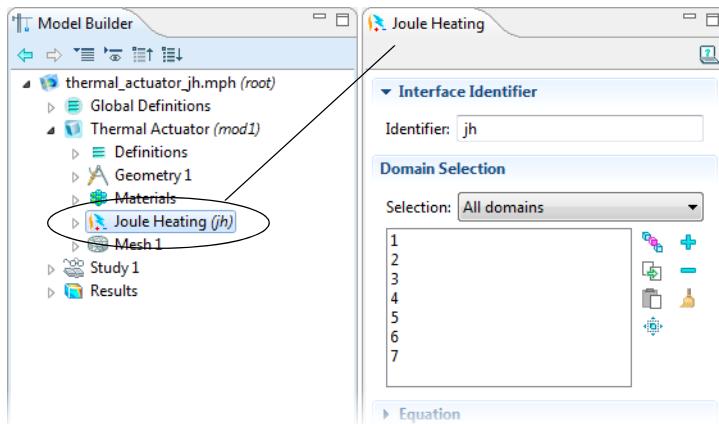
- [Building a COMSOL Model](#)
 - [Information About Memory Use](#)
-

The Node Settings Windows

When any node is clicked in the Model Builder (except a few *container nodes* such as *Definitions* and *Data Sets*), a corresponding settings window opens with the same name as the node. The settings window is an interactive environment for defining operations and model properties.

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** (), **Build All** () , or **Plot** () button, which are 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 directly in the displayed model in the Graphics window, from the Selection List window, or as, for example, **All domains** in the settings window.



-
- [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 when creating the geometry 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](#).



ID, 2D, and 3D toolbars all have the buttons above, plus:



-
- [Visualization and Selection Tools](#)
 - [Selecting and Deselecting Geometric Entities](#)
 - [The Graphics Toolbar Buttons](#)
 - [Plot Groups and Plots](#)
-

The Plot Windows

Plot windows are also graphics windows. COMSOL Multiphysics generates such plot windows for displaying convergence results and to monitor probe values 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:

- Open and saving of model files such as MPH-files.
- Information about geometry objects imported from CAD files.
- Information about the geometry finalization (forming a union or an assembly) and about the number of geometric entities (domains, boundaries, and so on) in the finalized geometry.
- The number of mesh elements and degrees of freedom in the model.

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



- Meshing
 - Studies and Solvers
-

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. 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 ().



- Progress and Log Information
 - Meshing
 - Studies and Solvers
-

The Log Window

The **Log** window () contains information from previous solver runs, including convergence information, solution time, and memory use. 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. 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.

-
- 
- [Progress and Log Information](#)
 - [Preferences Settings](#)
-

The Table Window

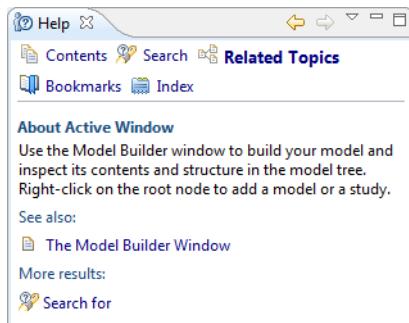
The **Table** window () displays the results from integral and variable evaluations defined in Derived Values nodes or by probes and stored in Table nodes. The window displays by default, but it can also be opened from the main menu by selecting

View>Table. When you select a Table node to display, the window name changes to the name of the Table (**Table 1**, for example).

-
- 
- [Results Analysis and Plots](#)
 - [Derived Values and Tables](#)
-

The Help Window

The **Help** window () provides access to *context help* () in the COMSOL Desktop® and has these pages: [Related Topics](#), [Contents](#), [Search](#), [Bookmarks](#) and [Index](#)



-
- [The Help Documentation Window](#)
 - [Table 2-4](#) for descriptions of the Help buttons
-

CONTENTS

The **Contents** section is a tree-based menu structure of the COMSOL 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 wildcard character.

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 and presentations sections of the website). See [Selecting the Scope From the Documentation Window](#) and [Changing the Search Scope From the Help Window](#) for details.

Selecting the Scope From the Documentation Window

- 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 search scopes from the list.
 - Click **New** to add a new search scope. The **New Scope** window opens where you can define a **List name** for the search scope and add topics from the list to form a custom search scope (to only search among the model libraries, for example).
 - Once a **Scope** is created, click **Edit** to make changes or **Remove** to delete the scope.
- 3 When done with the Scope lists, click **OK** to save or **Cancel** to exit.

Changing the Search Scope From the Help Window

- 1 Select **Help>Help**. Click **Search**.
- 2 In the Help window, expand the **Scope** section.
- 3 The **Default** searches the **Local Help** (the local help documentation) and the **COMSOL Search** (that is, all of www.comsol.com). Click to clear the check boxes as required.
- 4 To create a customized search scope set, click **Default** next to **Scope** to add a **New** scope set, or **Rename**, **Edit**, or **Remove** existing scope sets from the list.
 - Click **New** to add a new search scope set. The **New Scope Set** window opens where you can enter a **Scope Set Name**.
 - Once a **Scope Set** is created, click **Edit** to make changes, click **Rename** to change its name, or click **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** in the **Help** window (or **Edit** from the Scope Set window) to open the **Scope** window. This is where you can customize the searches in Local Help and COMSOL Search.
 - The default **Local Help** setting is **Include all topics**. To refine the search, click **Include only the following topics** and select the check boxes in the list under **Working set contents** as required. The **COMSOL Search** is set to search www.comsol.com. Click to clear or select the **Enable search engine** check box if required.

RELATED TOPICS

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

To view more detailed information, including summaries of the different attributes and settings related to the node, click on the links under **See also**. This takes you to the relevant sections in the COMSOL 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 **Help** button () in the active window's toolbar (not all windows have this toolbar).
- From the main menu, select **Help>Help**.
- Press F1.
- Right-click a node in the **Model Builder** and select **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 show the relevant context-sensitive help information when clicking a window tab or on a model tree

node in the **Model Builder**, regardless of the section of the help system displayed in the **Help** window.

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



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



The first time the **Help** window is opened on your computer, the operating system might issue a firewall security warning. To use Help, allow COMSOL 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. By preloading the help files the context help opens faster when used for the first time. To disable this feature, see [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 documentation (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.

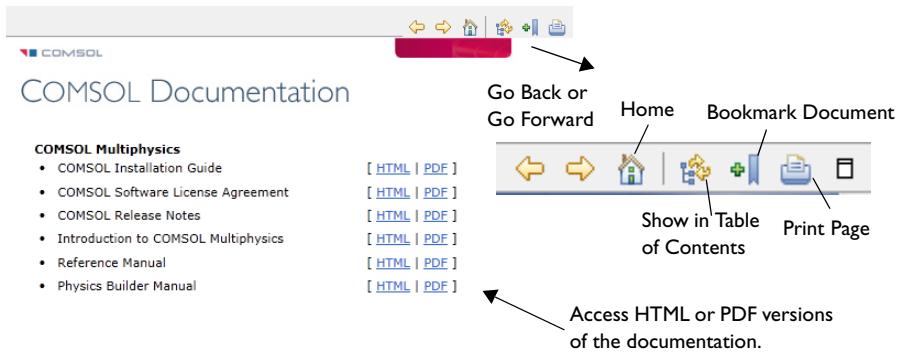


Figure 2-5: The right-hand side of the Help documentation window.

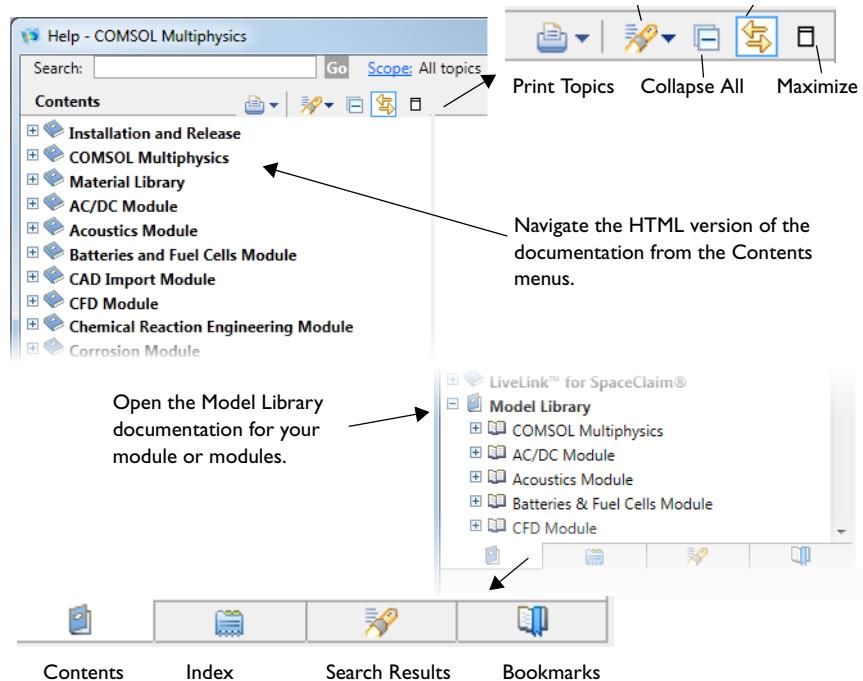


Figure 2-6: The left-hand side of the Help documentation window.

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 backward 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.
Previous	Navigates backward to the previous contents topic.
Show in External Window	Available from the Help window only, this button opens the same page as when you select Help>Documentation from the menu.
CONTENTS	Displays the contents of the documentation 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.

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

WINDOW NAME AND BUTTONS	DESCRIPTION AND TASK
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 its name (or its tag name or identifier prefixed by '@'; see the section [Opening and Searching the Model Library](#) for further details) 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 compact models are indicated by the icon  . You can download the complete MPH-files via **Model Library Update**. As a shortcut to get the complete version of an individual compact model, right-click the model's node in the **Model Library** tree and choose **Download Full Model** (). You can also generate the complete models by building their 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 simple approach does not work.



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.
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 Documentation** button. Alternatively, right-click the model node and select **Open PDF Documentation** 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.

-
- [Model Administration](#)
 - [Opening and Searching the Model Library](#)
 - [Changing the Model Library Root Folder](#)
 - [Updating the Model Library Using Model Library Update](#)
-

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**.



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 to the external processes. You can do operations that are performed in the **External Process** nodes under **Job Configurations** by selecting an external process from the list and then 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 [Table 2-6](#) for descriptions of the toolbar buttons available on this window.

TABLE 2-6: EXTERNAL PROCESS TOOLBAR BUTTONS

NAME	BUTTON	DESCRIPTION
Attach Job		If a job has been detached, click to reattach the job to run external processes and follow the status updates.
Stop All Processes		Sends the stop command to unfinished jobs. Similar to using the Stop button for The Progress Window .
Cancel All Processes		Sends the cancel command to unfinished jobs. Similar to using the Cancel button for The Progress Window .
Stop Process		Stops the selected process. Similar to using the Stop button for The Progress Window .
Cancel Process		Cancels the selected process. Similar to using the Cancel button for The Progress Window .
Rerun Job		Restarts the selected job.

TABLE 2-6: EXTERNAL PROCESS TOOLBAR BUTTONS

NAME	BUTTON	DESCRIPTION
Clear Status		Clears the status of the selected job. Useful when the status indicates that the process is running but the process has failed.
Log		Shows the current log of the selected process.
Open File		Opens the output file from the selected process.
Batch Jobs (generated tables)		Selects the Batch Jobs to view.

	<ul style="list-style-type: none"> • Batch (study type) • Cluster Sweep (study type) • Batch Sweep (study type) • Cluster Computing (Study) • Batch (job configurations) • Cluster Computing (Job Configuration) • External Class
--	--

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 provides 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.

-
- Key to Nodes and Toolbar Buttons
 - Errors and Warnings in a Geometry Sequence
-

The About COMSOL Multiphysics Box

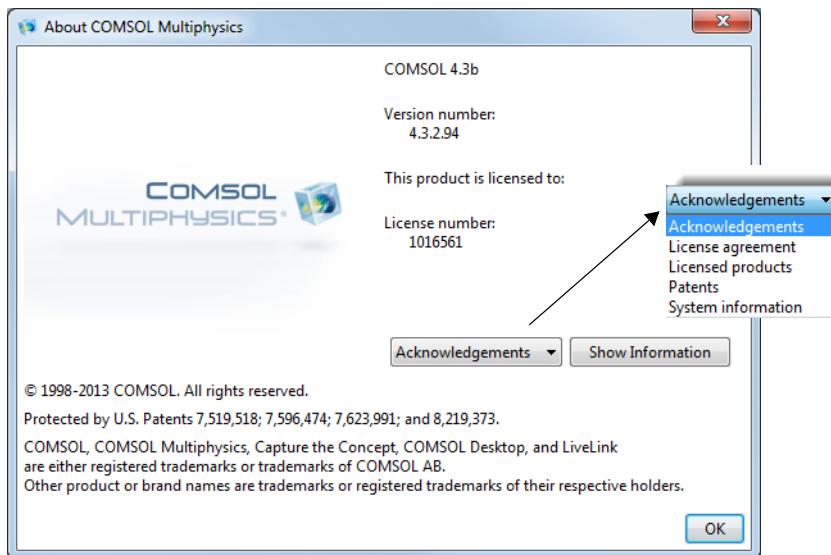


Figure 2-7: The **About COMSOL Multiphysics** box with the **Acknowledgements** list.

From the **Help** menu, choose **About COMSOL Multiphysics** to open the **About COMSOL Multiphysics** box, where you find the following information:

- The version number
- The user or company the product is licensed to
- 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 Multiphysics 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 Macintosh:

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
F1	F1	Display help for the selected node or window
Ctrl+F1	Command+F1	Open the COMSOL Documentation front page in an external Help window
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

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
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+N	Command+N	Create a new model
Shift+Ctrl+N	Shift+Command+N	Start a new Physics Builder (this option must be enabled in Preferences)
Ctrl+O	Command+O	Open a model file
Ctrl+P	Command+P	Print the contents of the plot window
Ctrl+S	Command+S	Save a model file
Ctrl+V	Command+V	Paste copied text
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

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACINTOSH)	ACTION
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.

Key to Nodes and Toolbar Buttons

The Model Builder Nodes

The **Model Builder** has many types of nodes to help you create models and visualize the model structure. These nodes are described throughout this guide. [Table 2-7](#) is a key to the main nodes.

TABLE 2-7: THE MAIN MODEL BUILDER NODES

NODE NAME	NODE	LINKS TO MORE INFORMATION
Data Sets		Data Sets
Definitions		Definitions
Derived Values		Derived Values and Tables
Equation View		Equation View and Physics Nodes—Equation Section
Export		Exporting Data and Images
Geometry		Geometry Modeling and CAD Tools
Global Definitions		About Global and Local Definitions
Materials		Materials
Mesh		Meshing
Model Builder		The Model Builder Window and The Model Wizard and Model Builder
Physics		<ul style="list-style-type: none">Overview of the PhysicsPhysics Nodes by Space DimensionPhysics Node Context Menu LayoutEquation-Based Modeling

TABLE 2-7: THE MAIN MODEL BUILDER NODES

NODE NAME	NODE	LINKS TO MORE INFORMATION
Plots		Plot Groups and Plots and Color Coding for Plot Groups and Plot Types
Reports		Reports
Results		Results Analysis and Plots
Study		Studies and Solvers
Tables		Derived Values and Tables

Dynamic Nodes in the Model Builder

Table 2-8 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 2-8: 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). See The Current Node in Geometry Sequences .
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. See The Current Node in Geometry Sequences .

TABLE 2-8: DYNAMIC NODES—VISUAL AIDS TO IDENTIFICATION

TYPE	ICON	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
Harmonic Perturbation		For example, on a boundary level node for the Electric Currents interface, Electric Ground node . See Harmonic Perturbation—Exclusive and Contributing Nodes .
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 nodes that have been disabled in a Study Step. See Editing and Building Geometry Nodes for Geometry nodes for example.
Pairs		For example, on a 3D Boundary Level node . See Identity and Contact Pairs .
Pairs—Fallback Features		For example, on a 3D Boundary Level pair node . See Identity and Contact Pairs .
Contributing node		For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types and Physics Node Status .
Default node		For example, on a 3D boundary level node. See Physics Default Nodes .
Override		For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types .
Overridden		For example, on a 3D boundary level node . See Physics Exclusive and Contributing Node Types
STUDY STEPS ANALYSIS		
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 .

TABLE 2-8: DYNAMIC NODES—VISUAL AIDS TO IDENTIFICATION

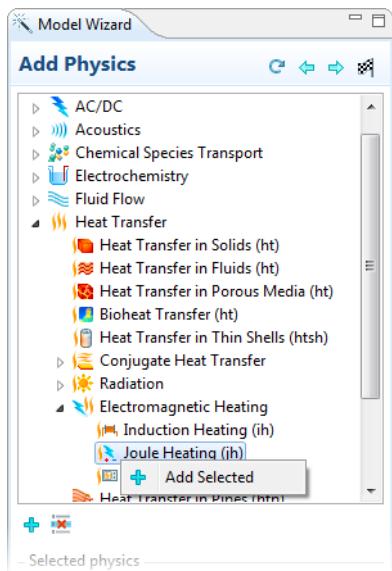
TYPE	ICON	NODE EXAMPLE AND LINK TO MORE INFORMATION (WHERE APPLICABLE)
Provide Degrees of Freedom		For example, a Laminar Flow interface is enabled (not grayed out), shows that the study step provides degrees of freedom (the yellow dot in the lower-right corner), and has a change of state indicated by the asterisk. The yellow dot means that the study step provides degrees of freedom but does not solve for the physics. See Selecting Physics and Variables in the Study Steps .
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. For example, for the Conjugate Heat Transfer interface .
Disabled, provides no degrees of freedom, and shows a change of state		In this example, a Transport in Diluted Species interface is disabled (grayed 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 .
LOAD AND CONSTRAINT GROUPS		
Load Group		This is an example of a Body Load node with a load group . This is for a 2D model at the domain level. See Load Group and Using Load Cases .
Constraint Group		This is an example of a Fixed Constraint node with a constraint group . This is for a 2D model applied to points. See Constraint Group and Using Load Cases .

Building a COMSOL Model

This chapter explains a range of methods and topics used when building models in COMSOL®: From working with the Model Wizard and fundamental concepts for the Model Builder to the use of units. For example of how to build a complete model step by step, see the model libraries for COMSOL Multiphysics® and the add-on modules and the *Introduction to COMSOL Multiphysics* book.

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, and study type.



The physics user interfaces contain predefined physics descriptions and equations for a variety of engineering and scientific disciplines. 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, add studies, and add physics. Right-click the **Root** (top) node and select **Add Model** or **Add Study** 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 in the **Model** branches—**Definitions**, **Geometry**, **Materials**, **Mesh**, and nodes based on the physics interfaces selected (see [Figure 3-1](#)). The **Model** nodes and branches form the sequence of operations that define the model.

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 represents 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. It is also possible to couple variables between **Model** nodes using coupling operators.

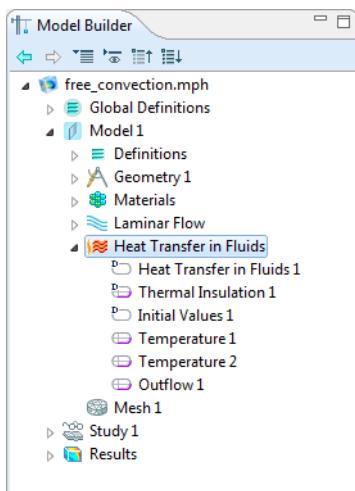


Figure 3-1: An example of the Model Builder default nodes.

These default nodes are normally added under a **Model** node:

- **Definitions**: Contains user-defined variables, selections, views, pairs, functions, probes, model couplings, and coordinate systems, which are defined locally for the model. See [About Global and Local Definitions](#) 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** (A): Contains the sequence of geometric objects and operations (or imported CAD data) that defines the model geometry.
- **Materials** (M): Contains the materials and material properties used as sources for material data in the model. See [Materials](#) for detailed information.

- **Physics** (): Any added physics displays as a node under **Model** (**Heat Transfer** in Figure 3-1 for example).
- **Meshes** (): 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 model identifier is `mod1`, `mod2`, and so on, but you can change it in the **Identifier** field.

MODEL SETTINGS

This section contains general model settings that 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.



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.

-
- 
 - [Editing Node Properties, Names, and Identifiers](#)
 - [Setting the Unit System for Individual Models](#)
 - [Curved Mesh Elements in the COMSOL API for use with Java® Reference Manual](#)
-

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 functionality available as subnodes to that particular node on that branch of the tree.

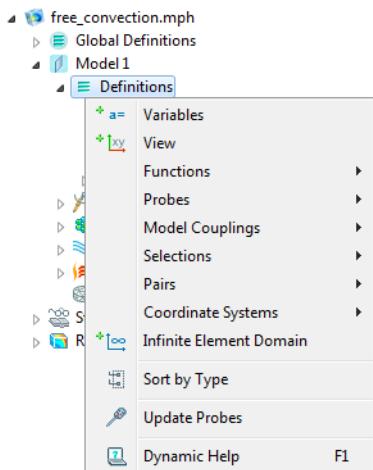


Figure 3-2: A context menu opens when you right-click any node in the Model Builder.

About the Sequence of Operations

COMSOL operates through *sequencing* and evaluates most of the branch nodes in the **Model Builder** from the top-down as a *sequence of operations*. By adding nodes to a model branch in the Model Builder in a certain order, you set up such sequences of operations. The software records a sequence as a macro in the background. This makes it possible to, for example, to parameterize a model and rerun the simulation.

COMSOL then re-evaluates each sequence, automatically updating the geometry, mesh, physics, and solution. A solver sequence, for example, could define your model with one solver and then, using the returned solution, solve it with an alternative solver. 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.

Some nodes under a physics branch can override other nodes higher up in the sequence. How COMSOL treats those nodes depends on whether they are contributing or exclusive nodes.



Physics Exclusive and Contributing Node Types

The sequence of operations means that the order of the nodes in the tree is important. In the following branches of the model tree, the node order makes a difference, and you can move nodes up and down to change the sequence of operations:

- Geometry
- Material
- Physics
- Mesh
- Solver

Also, the order can have some importance in the plot groups in the Results branch as well as for the Perfectly Matched Layers and Infinite Elements nodes in the Definitions branch (those nodes are available with some of the add-on modules).

Deleting, Disabling, and Enabling Nodes

You can change the contents, and actions, of the sequences in the model tree by disabling and deleting nodes. A disabled node does not take part in the evaluation of a sequence. See [Figure 3-2](#). Some nodes, such as container nodes and default nodes in the physics user interfaces (see [Physics Default Nodes](#)), cannot be disabled and deleted. When this is the case, the context menu does not have these options available. You can use Shift-click and Ctrl-click to select multiple nodes that you want to delete, disable, or enable.

- 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.



Instead of disabling and enabling variables and physics nodes to simulate different analysis cases (using different boundary conditions or sources, for example), use the selection of variables and physics in the study steps' **Physics and Variables Selection** sections, or use *load cases* for solving cases with varying loads or constraints.



- [Using Load Cases](#)
- [Selecting Physics and Variables in the Study Steps](#)

Model Administration

A variety of tasks can be done to organize and simplify the model building process

-
- 
- Preferences Settings
 - The Model Library
 - Advanced Physics Sections
 - Changing the COMSOL Desktop Layout
 - The COMSOL Desktop Environment
-

About the COMSOL Model File Formats

There are three COMSOL model file formats: MPH-files, Model Java-files, and Model M-files.

COMSOL MPH-FILES

The default standard file format with the extension .mph. The files contain 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 modules are MPH-files.

The MPH-files in the COMSOL model libraries can have two formats:

- *Full MPH-files* include 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* include all settings for the model but have no built meshes and solution data to save space (a few compact 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 **Download Full Model** item ().

File Locking

Only one user can open and edit an MPH-file at the same time. If you try to open an MPH-file that is already open in another user's COMSOL Desktop, that MPH-file is locked, and you get an option to open the MPH-file in a read-only mode (click **Open As Read-Only**). That means that you can edit the model but you cannot save it unless you save the MPH-file under another name. When an MPH-file is locked, COMSOL creates a separate lock file with the same filename as the MPH-file plus the extension **.lock**, stored in the same directory as the locked MPH-file. If a lock file remains after all COMSOL Desktop sessions have ended (which can happen if the COMSOL Desktop session is ended in a nonstandard way), you can reset the lock when trying to open the file the next time by clicking **Reset Lock and Open**.



Linux and Macintosh do not support operating system locking of files.

On those platforms, locking is supported to help users avoid editing the same COMSOL model file, but it is possible to ignore the file locking and delete the lock files.

MODEL JAVA-FILES

Editable script files that contain sequences of COMSOL commands as Java[®] code (see the *COMSOL API Reference Manual* 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 commands as a MATLAB M-file. Run the Model M-files in MATLAB like any other M-file scripts. You can also edit the files in a text editor to include additional COMSOL commands or general MATLAB commands.



Running files in the Model M-file format requires the COMSOL LiveLink™ for MATLAB[®].

-
- 
- [The Model Library](#)
 - [Saving COMSOL Model Files](#)
 - [Reverting to the Last Saved Model File](#)
 - [Saving a Model Thumbnail Image](#)
 - [Saving and Opening Recovery Files](#)
-

Saving COMSOL Model Files

-
- 
- [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 MODEL 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:

Click the **Save** button (), press Ctrl+S, or from the **File** menu, select **Save**.

If the model has been saved before and you want to create a copy:

From the **File** menu, select **Save As**.

SAVING AS A MODEL JAVA-FILE

From the **File** menu, select **Save As Model Java-File**.

SAVING AS A MODEL M-FILE

From the **File** menu, select **Save As Model M-File**.

In all cases, navigate to a the location where you want to save the model and enter a **File name**, and then 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 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 and Opening Recovery Files

COMSOL 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 **Temporary files**. Then, under **Recovery**, clear or select the **Save recovery file** check box. This change takes effect the next time that you start COMSOL. You can also specify where to store the recovery files in the **Folder for recovery files** field or using the **Browse** button.

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 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, from the **File** menu, choose **Open Recovery File**. 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 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 nonparametric problem, the last converged Newton iteration is stored in the recovery file; selecting **Continue** then resumes the solving from this stored state.

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 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.

-
- 
- 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). To edit a model identifier, in the **Model Builder**, click a **Model** node. The **Model** node's settings window opens. Edit the default identifiers (**mod1**, **mod2**, and so on) as required in the **Identifier** field. See [Figure 3-4](#) for an example.

RENAMING A NODE

To rename a node in the **Model Builder** (except container nodes with fixed names such as **Global Definitions**), 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 (except “container nodes” such as **Materials**) and select  **Properties** from the context menu (see [Figure 3-2](#)). 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 and is mainly used when using the COMSOL API and the optional LiveLink™ for MATLAB®.
- 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

The **Used Products** section lists any COMSOL module that is used by that node. This makes it possible to identify which nodes that make use of the license for a 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 (or click another node and then click the node again).

ROOT NODE PROPERTIES

The settings window for the root node includes a **Node Properties** section with additional information about the model file. It also includes 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 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 **License number** is the number of the license used the last time the model was saved. If you open a model made by someone else you see that user's license number. If you save the model you see your own license number.
- 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

The **Used Products** section lists any COMSOL modules used by the models in the file.

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** for the entire model file. 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.

-
- | | |
|---|---|
|  | <ul style="list-style-type: none"> • Preferences Settings • Saving a Model Thumbnail Image • Using Units |
|---|---|
-

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 without restarting COMSOL).

- 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 the ones that are already in use).



The settings are local for each COMSOL 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.



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**.

! 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 currently uses. The two numbers, displayed as, for example, **405 MB | 453 MB**, represent the physical memory 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** number is the subset of the virtual address space used by COMSOL that is physically resident; that is, it is the amount of physical memory (RAM) in “active” use.
- The **Virtual memory** number is the current size of the virtual address space that COMSOL uses.

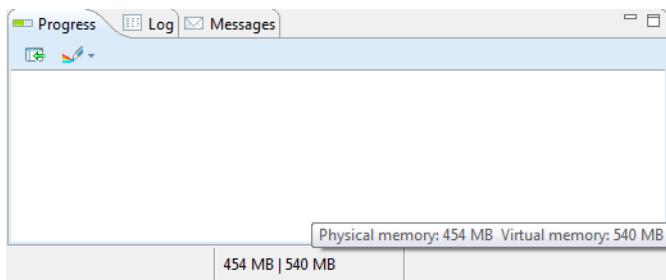


Figure 3-3: An example of memory use displayed in COMSOL.

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.

- *Identifiers* are primarily used as names for functions and operators when called in models and as identifiers for the **Model** nodes that you use for variable scoping.
- *Types* are the unique descriptions for each type of node.

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 user 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 3-4](#) for examples of the different label types.

The options in the list modify the view 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, for model nodes, physics 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:

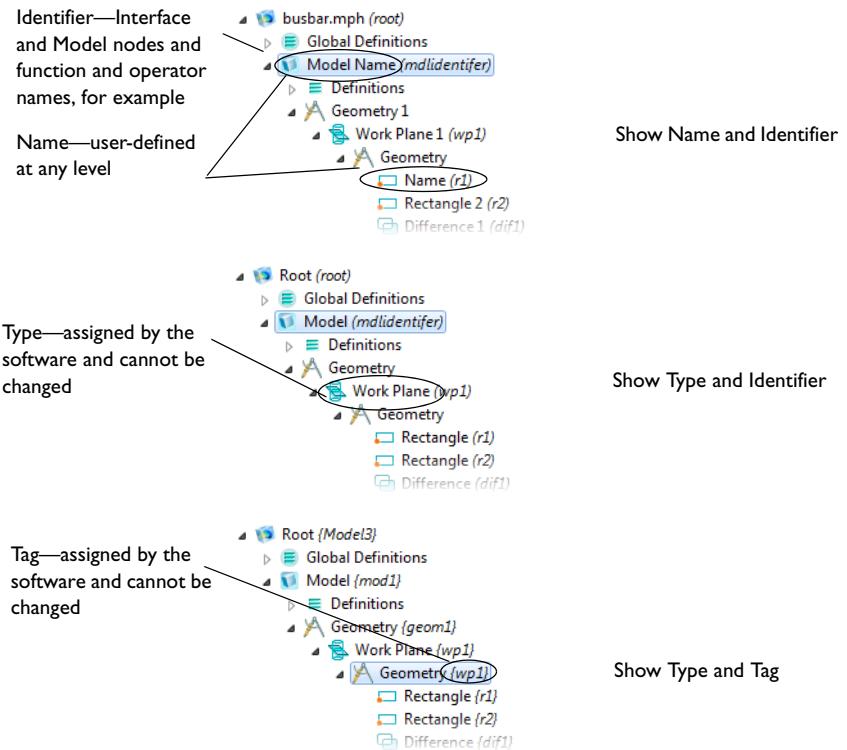


Figure 3-4: Examples of the available label combinations on the View menu.



Editing Node Properties, Names, and Identifiers

Preferences Settings

To make changes to how items are displayed throughout COMSOL, 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 preferences settings.



For some setting changes, a message window displays to tell you that COMSOL needs to be restarted for the changes to take effect.

General Preferences

Select **Options>Preferences** from the Main menu and then click **General** to edit the following.

PRECISION

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 display using the specified precision but appear with full numerical precision when the field is active. The value in the **Table display precision** field controls the display precision for data values in the result tables.

WEB BROWSER

Under **Web browser** (Windows and Linux only), you can choose which browser COMSOL should use to show pages on the COMSOL website and help contents when using the web browser help mode (see below). The following settings are available:

- On Windows: Choose the **Program** setting **System default** to use the default system web browser. Alternatively, choose **Custom** and then give the path to an **Executable** for a different browser installed on your computer.
- On Linux: Type the path to the web browser directly in the **Executable** field, or click the **Browse** button and then point to the executable file on the file system.

LANGUAGE

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 to display the desktop in the selected language.

HELP

Under **Help**, when the **Sticky help** check box is not selected, it means the **Help** window always updates its contents to show the relevant context-sensitive help information when clicking a window tab or a node in the **Model Builder**. When the **Preload help files** check box is selected, it means that the help files are preloaded when launching COMSOL. 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 browser. For the integrated mode, the **Show PDF-files in new browser window** check box controls how PDF documents are launched. Clear this check box to use your default system browser's settings.



The Help Window

LOG

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.

USERNAME

Under **Username**, edit the **Name** field to change the username.

HISTORY EXPORT

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 Preferences

Select **Options>Preferences** from the Main menu and then click **User interface** to edit the preferences under **Look and feel** to 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 an area below the window's name. The name of all settings windows is **Settings**.



If you change the toolbar style, you must restart COMSOL for the change to take effect.

Show Preferences



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 [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** from the Main menu and then click **Show** to edit the following.

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**.

WAVE FORM PDE

Select the **Show Wave Form PDE interface in Model Wizard** check box to enable that physics user interface.



The Wave Form PDE

Expand Sections Preferences



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 [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** from the Main menu and then click **Expand sections** to edit the following.

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 Preferences

Select **Options>Preferences** from the Main menu and then click **Model Builder** to edit the following.

SORTING

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 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 user 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 user 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 user 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.



Sorting Nodes by Space Dimension and Type

 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.

- 
- Details for the following cluster type settings are described in [Cluster Computing \(Job Configuration\)](#).
 - [Running COMSOL on the Amazon™ Cloud \(Amazon EC2™\)](#)
-

Select **Options>Preferences** from the Main menu and then click **Cluster computing** to edit the following.

SETTINGS

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**.
- Enter extra arguments to MPI using the **Additional MPI arguments** field if required.
- 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.
- Enter the details for the **Bootstrap server**.
- 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 commands in the **Prepend command** and **Postpend command** fields as required to add 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 (seconds)**, the interval time that the running process has before signaling that it is alive. The default is 300 seconds.

Remote Computing Preferences

-
- 
 - [Working with Remote Servers](#)
 - [Running COMSOL on the Amazon™ Cloud \(Amazon EC2™\)](#)
 - Details for the Remote Access settings are described in [Cluster Computing \(Study\)](#)
-

Select **Options>Preferences** from the Main menu and then click **Remote computing** to edit the following.

SETTINGS

Under **Settings**:

- Select a **Remote invoke command**—**None** (the default), **SSH**, or **User**. If **SSH** is selected, go to [SSH Settings](#) for additional settings. If **User** is selected, also enter a **Command**.
- Select a **File transfer command**—**None** (the default), **SCP**, **Putty**, or **User**.
- Enter details for the **Remote hosts**.
- Select a **Remote OS**—**Native** (the default), **Windows**, or **Linux**.

SSH Settings

If **SSH** is selected as the **Remote invoke command**, enter these additional details:

- Select an **SSH command**—**SSH** (the default), **Putty**, or **User**.
- If **User** is selected as the **SSH command**, then enter a **Command**.

Then for all **SSH command** options:

- Enter the **SSH directory** path or click **Browse** to navigate to a specific location.
- Enter the **SSH key file** or click **Browse** to navigate to a specific location.
- Enter the **Forward ports**.

- Enter the **Port host**.
- Enter the **SSH user** details.

Geometry Preferences

Select **Options>Preferences** from the main menu and then click **Geometry** to edit the following.

GEOMETRY REPRESENTATION AREA

Under **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.

AUTOMATIC REBUILD

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.

VISUALIZATION

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.



The CAD Import Module kernel requires the CAD Import Module.



The Geometry and CAD Environment

Graphics Preferences

Select **Options>Preferences** from the main menu and then click **Graphics** to edit the following.

VISUALIZATION

Under **Visualization**, edit the following sections.

Rendering Type

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.



If you change the renderer, you must restart COMSOL for the change to take effect.

Optimize For

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.



[The Material Settings Window](#)

Multisampling

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.



Multisampling is only available with hardware rendering using graphics cards that support multiple samples per pixel.

Detail

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.

Show Logo on Canvas

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.

Show Materials

The **Show materials** check box is selected by default and displays the material texture and color as defined in [The Material Settings Window](#).

Show Physics Symbols

Select the **Show physics symbols** check box to display physics symbols that are available with some physics. The check box is not selected by default.



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 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 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 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 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

- Select the **Load system fonts** check box to always load system fonts.
- Select a font **Family** and font **Size** (the default is 9).



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 Preferences

Select **Options>Preferences** from the main menu and then click **Results** to edit the following settings:

AUTOMATIC UPDATE OF PLOTS

Under **Automatic update of plots** you can set the preferences to update 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.

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**, specify preferences for the plots that you can use to monitor 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.
- 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.

The values in the **Maximum filled matrix size** field (default: maximum 1000 rows or columns). This value limits the table size when the table contains a system matrix as a filled matrix so that you do not create tables that require lots of memory and disk space. The system matrices are usually large and sparse, and the default output format in the **System Matrix** node's settings window is the sparse format, which only adds the nonzero matrix elements (and their locations in the matrix) to the table.

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 in 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 empty, the COMSOL 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 in 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 Preferences

Select **Options>Preferences** from the Main menu and then click **Builder Tools** to edit the following, which control some properties for the Physics Builder and its tools.

SETTINGS

In the **Settings** area, select the **Enable Physics Builder** check box to make the Physics Builder available from the **File** menu.



When this option is enabled, select **File>New from Physics Builder** to start a new Physics Builder.

EDITOR

In the **Editor** area you can specify some properties for the code editor:

- Select the **Auto indent** check box to activate automatic indentation (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.

COMMENTS AND DOCUMENTATION

In the **Comments and documentation** area you can control whether to show the nodes for adding comments and documentation to the physics user interfaces and applications that you create.

Show Comments

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.

Show Documentation

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.

Enumerate Sections to Level

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.

Section Levels in Table of Contents

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.

Generate Separate Files at Section Level

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.

Documentation Plug-in Prefix

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.

Vendor

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 Preferences

Select **Options>Preferences** from the Main menu and then click **Temporary files** to edit the following.

RECOVERY

The **Save recovery file** check box is selected by default to save recovery files to disk during the solution process for time-dependent, parametric, and nonlinear solvers. Click to clear as required.

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.



Saving and Opening Recovery Files

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 Preferences

Select **Options>Preferences** from the Main menu and then click **Memory and processors** to edit the following.

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 and then enter a filename in the **Filename** field, or click **Browse** to choose a name and location for the model files. 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 COMSOL commands from the command line.

Preferences for Updates

Select **Options>Preferences** from the Main menu and then click **Updates** to edit the following.

PRODUCT UPDATES

Select the **Check for updates at launch** check box to make the program check for updates each time COMSOL 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 to use when communicating with COMSOL's website for performing a Model Library Update.

The **Configuration** list has the following options:

- **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) and port number in the **Server** and **Port number** fields. The default port number is 443, which is the default for HTTP secure (HTTPS). If the proxy server requires authentication, you are asked to provide username and password the first time you call Model Library Update in each COMSOL session.



The Model Library

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 [General Preferences](#)).
- **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 Preferences

Select **Options>Preferences** from the main menu and then click **LiveLink products** to edit the following. 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.

The Model Library

This section describes how to work with the model libraries and the Model Library.

-
- 
- [Model Administration](#)
 - [Preferences Settings](#)
 - [Preferences for Updates](#)
 - [About the COMSOL Model File Formats](#)
-

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 areas. 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.

The following steps show 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 default, 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. Alternatively, you can limit the search to tag names and identifiers by using the prefix '@', for example, @genext or @ehs. To search for a specific physics user interface, use the scoping syntax @physics:<identifier>. For example, type @physics:jh to find all models that use the Joule Heating physics.

Beside @physics, the supported scopes include @geom, @mesh, @probe, @result, @selection, and @study.



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.

- 3 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 Documentation** 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.



- [The Model Library Window](#)
- [The Help Window](#)

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, use 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.

Updating the Model Library Using Model Library Update



Using the **Model Library Update** service requires Internet access. For a default installation, you also need to run COMSOL as administrator. See the section [Preferences for Updates](#) for instructions on how you can modify your installation to avoid this restriction.

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.



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.
- 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.

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**.
-



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 ().

Modeling Guidelines

To model large-scale problems and for successful modeling in general, COMSOL makes it possible to tune solver settings and to use symmetries and other model simplifications 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.

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 physics user 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 user interfaces have symmetry conditions directly available as nodes that you can add to the model tree.

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 requires 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 user 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. For details about solvers, see the [Studies and Solvers](#) chapter.

ESTIMATING THE MEMORY USE FOR A MODEL

Out-of-memory messages can occur when COMSOL 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 but also on the couplings between variables in a model. 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 COMSOL Desktop, where the program displays the amount of physical memory 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 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 user interfaces). See [Selecting an Element Type](#).
- Making sure that the mesh elements are of a high quality. Mesh quality is important for iterative linear system solvers. 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, COMSOL uses first-order or second-order Lagrange elements (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.



For information about editing shape functions, see [Equation View](#).

Analyzing Model Convergence and Accuracy

It is important that the numerical 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 model libraries are *benchmark models* that include comparisons to established results or analytical solutions.

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.



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 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. Most physics user interfaces for modeling of fluid flow and chemical species transport provide artificial diffusion as part of the default settings.
-



Stabilization Techniques

- 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, 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 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 under-constrained and it is difficult to get convergence.
- Make sure that the mesh is 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.



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 Unphysical Values

WHERE AND WHY DO UNPHYSICAL VALUES APPEAR?

In some models small unphysical values can occur due to numerical artifacts or other model-related reasons. 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.

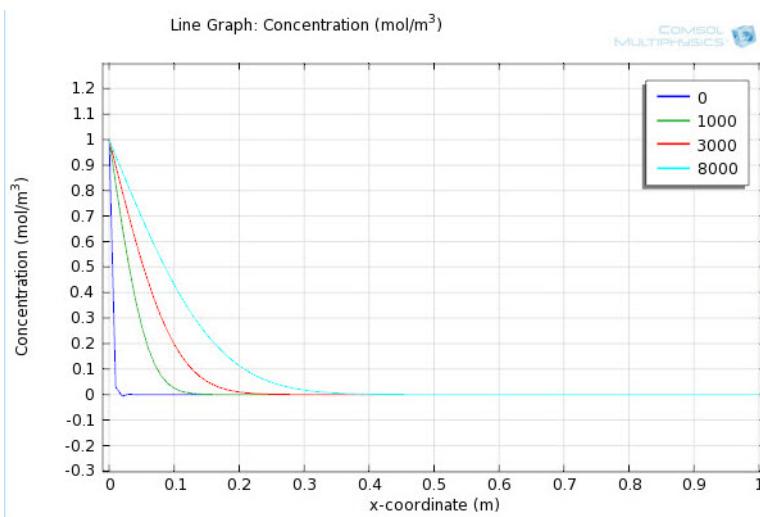
Some reasons for why these unphysical values occur:

- 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. Take care when using interpolated data or a piecewise polynomial function to define a temperature-dependent material property, for example. If you allow extrapolation outside of the defined range of input values, the material property values may not be valid. Also, 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.



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. COMSOL 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 nonnegative. 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.

Advanced Physics Sections

To display the functionality described in this section, click the **Show** button (>Show) 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**.

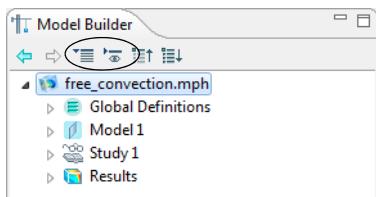


Figure 3-5: The location of the **Expand Sections** and **Show** buttons in the **Model Builder**.



These options are also accessed (and activated) by selecting the check boxes from the **Options>Preferences>Show** and **Options>Preferences>Expand sections** windows. See [Preferences Settings](#).

Resetting to Default

From the **Model Builder**, click the **Show** button (>Show) 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 (>Show) 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.



These options are also accessed (and activated) by selecting the check boxes from the **Options>Preferences>Expand sections** window. See [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 nodes' 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 nodes' settings windows. See also [Physics Nodes—Equation Section](#).

Show Equation View

To display the **Equation View** node under all physics nodes in the **Model Builder**, click the **Show** button () and select **Equation View**.

-
- | | |
|---|--|
|  | <ul style="list-style-type: none">• Equation View• 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 nodes (for material models, 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.

-
- | | |
|---|---|
|  | <ul style="list-style-type: none">• Physics Exclusive and Contributing Node Types• Listing Overrides and Contributions• Overridden Selections |
|---|---|
-

Show Discretization

	<p>There are two categories of discretization—a section on the physics user interface node’s settings window (described here) and adding a Discretization (Node) for global equation-based modeling.</p>
---	--

To enable this setting, click the **Show** button () and select **Discretization**. The options available for each node are described individually in 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 in the model. Some physics use special element types or a reduced element order for some of the field variables.

Some additional information is included in the section [Discretization](#) [Section Shape Function Types and Element Orders](#) of this manual and in the section [Finite Elements](#) in the *COMSOL API for use with Java® Reference Manual*.

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.

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.

Accurate Boundary Fluxes

Some physics can create and compute variables that accurately represent the flux across all boundaries. To enable these variables, select the **Compute boundary fluxes** check box.

Optionally, a smoothing can be applied when computing the boundary flux variables. You add smoothing by selecting the **Apply smoothing to boundary fluxes** check box.

Complex-Valued 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 the split complex variables setting is not active the value type is ignored.



For information about how to specify the splitting of complex variables, see [Compile Equations](#).

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** user interfaces, it displays a **Stabilization** section instead.

Numerical stabilization is available for physics user 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 user 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.



[Numerical Stabilization](#)

CONSISTENT STABILIZATION

There are two *consistent stabilization methods*: **Streamline diffusion** and **Crosswind diffusion**. Usually, both check boxes for these methods are selected by default and should remain selected for optimal performance. 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

In most cases 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 cases, 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

In some cases, 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 by making the system somewhat less dominated by convection. If possible, minimize the use of the inconsistent stabilization method because by using it you no longer solve the original problem.

By default there is no isotropic diffusion selected. If required, select the **Isotropic diffusion** check box and enter a **Tuning parameter** δ_{id} as a scalar positive value. The

default value is 0.25 (a reasonable value to start with is roughly 0.5 divided by the element order). A higher value adds more isotropic diffusion.

Show Advanced Physics Options

To enable this feature, click the **Show** button () and select **Advanced Physics Options**. This displays the **Advanced** section on physics settings windows. This selection also activates the [Show Advanced Physics Options—Context Menu Equation-Based 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 user interface contains some advanced options.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**. This section controls how constraints are enforced, usually in boundary conditions.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak form implementation.

Show Advanced Physics Options—Context Menu Equation-Based Nodes

In the **Model Builder**, when you right-click any physics user interface node and open the context menu, there are additional nodes that you can add to the model. To enable these nodes, click the **Show** button () and select **Advanced Physics Options**. Then, from the **More** menu, you can add these nodes:

- **Weak Contribution.** See [Weak Contribution \(PDEs\)](#) and [Weak Contribution \(ODEs and DAEs\)](#).
- [Weak Constraint](#)
- [Pointwise Constraint](#)
- **Weak Contribution on Mesh Boundaries.** This feature is similar to Weak Contribution but is active on mesh boundaries. See [Weak Contribution \(PDEs\)](#) and [Weak Contribution \(ODEs and DAEs\)](#).

From the **Global** menu add these nodes if desired:

- **Global Equations.** Also see [Adding ODEs, DAEs, and Other Global Equations](#).
- **Global Constraint.** Also see [Symmetric and Nonsymmetric Constraints](#) and [Constraint](#).
- **Weak Contribution.** See [Weak Contribution \(PDEs\)](#) and [Weak Contribution \(ODEs and DAEs\)](#).
- **Discretization.** See [Discretization \(Node\)](#) and [Discretization Section Shape Function Types and Element Orders](#).



Clicking the **Show** button () and selecting **Advanced Physics Options** also displays the **Advanced** section on physics user interface nodes' settings windows, which is described in [Show Advanced Physics Options](#).



These settings can also be activated from the **Preferences** dialog box. See [Preferences Settings](#).

Show Advanced Study Options

Under the **Study** node, click the **Show** button () and select **Advanced Study Options** to enable these nodes that you can then add to the **Model Builder**:

- **Job Configurations.** See [Job Configurations](#).
- **Solver Configurations.** See [Solver Configurations](#).



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, Cluster Sweep,** and **Batch Sweep.** See [Cluster Computing \(Study\)](#).
- **Batch**

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 [Editing Solver Configurations](#).

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](#).

Show Advanced Results Options

To display the **Views** node under **Results**, click the **Show** button () and select **Advanced Results Options**.



- [User-Defined Views](#)
 - [Results Analysis and Plots](#)
-

The Physics 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

Each *physics user interface* includes nodes for specifying all input data for a specific physics in a model:

- Material properties
- Boundary and interface conditions
- Equations (for equation-based modeling)
- Initial conditions

In addition, you can 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 user interfaces.

All settings windows for the specification of the physics and equations accept parameters and variables as input data.



- [Materials](#)
 - [About Global and Local Definitions](#)
-

Physics Node Context Menu Layout

The *context menu* opens when you right-click a physics user interface node. This menu is divided into four sections for most physics: the first section contains domain settings, the second boundary settings, the third edge settings, and the fourth has point settings.



There can be menu items with the same name but applied at different geometric entity levels.

As shown in [Figure 3-6](#), the context menu layout is also based on whether the nodes are not sorted (the default) or if **Sort by Space Dimension** is selected.

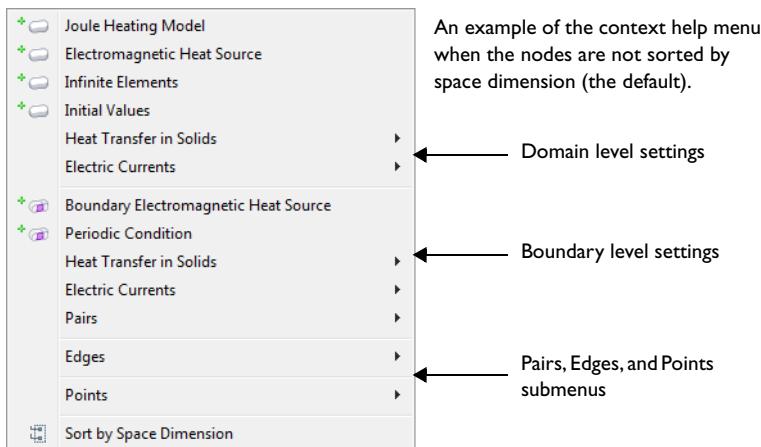


Figure 3-6: The physics context menu layout for a 3D model with unsorted nodes.



Sorting Nodes by Space Dimension and Type

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 3-1](#) and [Figure 3-7](#). The nodes also correspond to [The Graphics Toolbar Buttons](#), some of

which are also based on space dimension.



See [Physics Exclusive and Contributing Node Types](#) and [Physics Node Status](#) for examples of other differences to how the nodes display in the Model Builder.

TABLE 3-I: PHYSICS NODES BY SPACE DIMENSION

NODE NAME	3D	2D AND 2D AXISYMMETRIC	1D AND 1D AXISYMMETRIC
Domain Level			—
Domain Level, default node			
Boundary Level			—■
Boundary Level, default node			
Boundary Level, Pairs			
Point Level			—
Edge Level		—	—

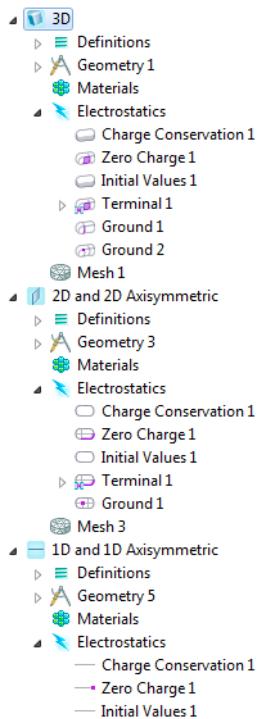


Figure 3-7: An example of the physics nodes listed in Table 3-1 and as displayed in the Model Builder based on model space dimension.

Physics Exclusive and Contributing Node Types

The nodes for the physics 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 as shown in Figure 3-8.



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).



What the node looks like in the Model Builder is based on the space dimension. See [Physics Nodes by Space Dimension](#).

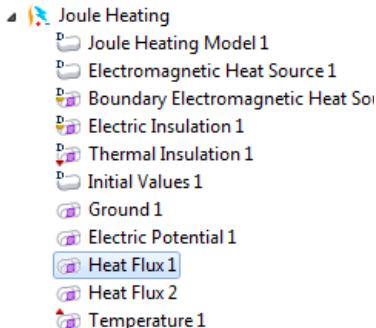


Figure 3-8: An example of exclusive and contributing nodes in a 3D model physics interface operating sequence and when Ungroup Nodes is selected.

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 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 shows the nodes the current nodes contributes with indicated using a yellow dot to the left of the icon, for example, in this boundary level icon  . See also [Figure 3-8](#) for an example.

LISTING OVERRIDES AND CONTRIBUTIONS

If your preferences include showing the **Override and Contribution** section in the settings windows for physics 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 3-9](#) 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 dot to the left of the icon, for example, in this boundary level icon .

 If you disable physics nodes 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.

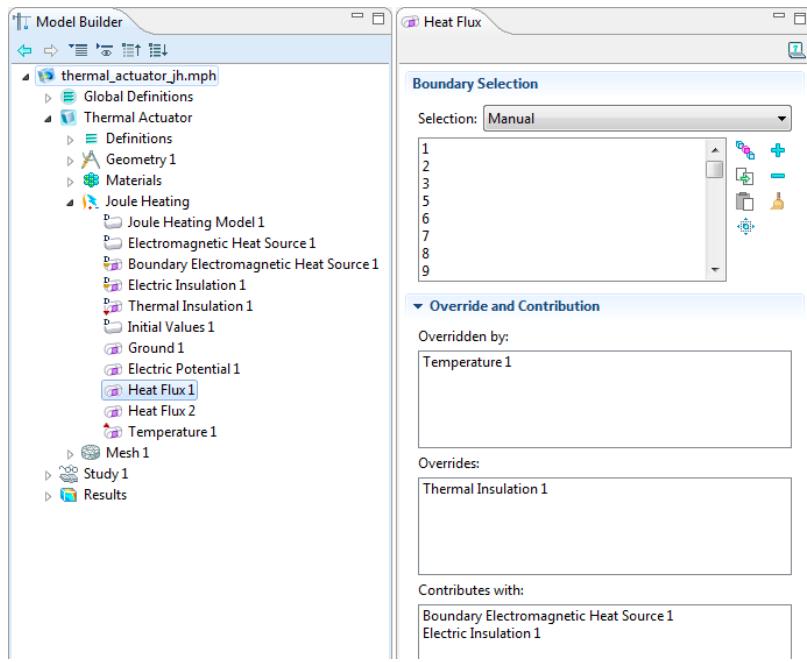


Figure 3-9: The Override and Contribution section lists other physics nodes that the selected node is overridden by, overrides, or contributes with. This is an example for a 3D model and with Ungroup Nodes selected.

-
- Physics Node Status
 - Key to Nodes and Toolbar Buttons
 - Physics Default Nodes
-

Physics Node Status

The status of a physics node depends on if it is a default 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 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 node (such as interior boundaries for an boundary condition that is only applicable for exterior boundaries), the **Selection** list then shows **(not applicable)** next to entries 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 interface branches.

-
- 
- [Physics Default Nodes](#)
 - [Physics Exclusive and Contributing Node Types](#)
 - [Physics Nodes by Space Dimension](#)
 - [Key to Nodes and Toolbar Buttons](#)
 - [Deleting, Disabling, and Enabling Nodes](#)
-

Physics Default Nodes

When you add a physics, COMSOL automatically adds a corresponding physics branch in the model tree, which typically includes a number of default 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 user 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.

In most cases, 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. It is not possible to delete such default nodes, but you can copy and duplicate all default nodes. Some multiphysics user interfaces also add default nodes with no initial selection, which are possible to delete from the model. Default nodes include a *D* (for “default”) in the upper-left corner () to indicate their special status. The copy or duplicate of a default node is a node of the same type but behaves as a normal 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.



Some physics also add standard nodes directly when you add them to a model. They represent functionality that is likely to be useful 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.

Equation View

Equation View () 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.



Editing the predefined expressions for variables, equations, and constraints means that the equations are altered and that COMSOL solves the model using the new expressions.

You can 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 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 node. An orange color for the expression that defines the variable is a warning that the unit of the expression does not match the expected unit for the variable that it defines.



For information about the Equation displays available, see [Physics Nodes—Equation Section](#).

VARIABLES

This section has a table with the variables that the physics node defines. The table includes these columns:

- **Name:** the name of the variable.
- **Expression:** the expression, using COMSOL syntax, that defines the variable.
- **Unit:** the unit for the variable (in the active unit system). If the unit of the expression does not match the unit of the variable, the expression is displayed in orange.
- **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 node defines and their shape functions. This is primarily applicable to equation model nodes; for most physics nodes 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).



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.



The PDE user interfaces and the ODEs and DAEs user 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 node generates. This is typically the case for boundary conditions of constraint types, such as prescribed displacements, temperature, or velocities. Many other physics nodes 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.
- For some variables, **Meta** indicates that the variable definition are fully updated when solving the model. It is therefore not recommended to edit the expressions for such variables.
- 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

Specifying Model Equation Settings

The fundamental mathematical model, representing the physics in a physics user interface, is contained in physics nodes with selection on the same space dimension as the physics itself. The first node under a physics branch is of this type and sets up default equations where the physics is active. These equations are controlled by specifying:

- Material properties, which are entered as coefficients in the equations,
- A coordinate system, in which anisotropic material properties are entered, and
- A material model, which selects an equation suitable for a given type of material.



Not all physics allow anisotropic materials or more than one material model. Therefore, these settings may not be present.

The default node uses the same material model, and thus the equations, everywhere. Material properties can vary between different parts of the feature's selection, if the property is specified as taken **From material**. Add additional nodes to use different material models for different parts of the geometry, or to use different **User defined** material property values.

In equation-based modeling, provided by the Mathematics branch user interfaces, the form of the equation is fixed for each particular node type. Each given equation form contains a number of free coefficients, which then can be specified in the settings.



Equation-Based Modeling

Specifying Equation Coefficients and Material Properties

To specify an equation 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](#)).

- Built-in [Mathematical and Numerical Constants](#).
- Spatial coordinates, time, and the dependent variables in any physics in the model as well as their spatial derivatives and time derivatives.
- [Physical Constants](#)—built-in universal physical constants.
- User-defined parameters, variables, coupling operators, and functions, including external functions and MATLAB functions (requires the COMSOL LiveLink™ for MATLAB®). See [Operators, Functions, and Constants](#).
- Built-in functions and operators such as `d` and `mean`.

You can use these types of variables, constants, functions, and operators in all settings for the physics user interfaces; many types of variables are also available anywhere in the model.

Modeling Anisotropic Materials

Anisotropic materials respond differently to an excitation depending on its direction. Because excitations are generally vectors and the corresponding response a vector density, material properties are usually rank-2 tensor densities. For example, the following material properties are anisotropic tensor densities: diffusion coefficient, permittivity, thermal conductivity, and electrical conductivity.

These properties are, in principle, specified in matrix form and defined by their components in the coordinate system selected in the node settings. At most four components are used in 2D and at most nine components in 3D. When the material contains symmetries, you may specify only a few coefficients which are expanded to a matrix using the following patterns:

- **Isotropic** (the default)—enter only one value c .

$$C = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & c \end{bmatrix}$$

- **Diagonal**—enter the diagonal components for an anisotropic material with the main axes aligned with the model’s coordinate system.

$$C = \begin{bmatrix} c_{11} & 0 & 0 \\ 0 & c_{22} & 0 \\ 0 & 0 & c_{33} \end{bmatrix}$$

- **Symmetric**—enter a symmetric matrix using the diagonal components and the upper off-diagonal components.

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}$$

- **Anisotropic**—enter the full 2-by-2 (2D) or 3-by-3 (3D) matrix for an anisotropic material:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Specifying Initial Values

An **Initial Values** node is added by default to each physics user interface.

In some types of analyses initial values must be provided:

- 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 user interface node. In the settings window, enter the **Initial Values** for all dependent variables (fields) in the physics. The default initial values are usually zero.

For some physics 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 no other values are specified.

To use different initial values in different domains, right-click the main physics node and select **Initial Values** to add additional nodes to the Model Builder.

See [Dependent Variables](#) for more information about handling and plotting initial values.

Boundary Conditions

In the interior of the selection where a physics is active, its behavior is governed by its model equations and material properties. *Boundary conditions* apply to the geometric entities separating this region from the unspecified outside, and also to interior entities of the same dimension. Therefore, boundary conditions on a 3D solid object apply to the exterior surfaces of the solid, and to interior surfaces embedded in the solid. On a shell geometry in 3D space, boundary conditions instead apply to the edges of the shell surface. In general, boundary conditions apply to geometric entities whose dimension is one less than the physics user interface's dimension.

All physics branches that contain a default model equation node also contain a default boundary condition node. This boundary condition is active on all exterior boundaries of the physics' selection, except on the symmetry axis of axisymmetric 2D models. On interior boundaries, an implicit continuity condition applies, which makes the physics field (the temperature, for example) continuous across interior boundaries.

Boundary Condition Types

There are two fundamental ways to specify what is happening at a boundary, and two corresponding fundamental boundary condition types:

- *Flux conditions* specify how the surroundings affect and interact with the model at the boundary, often expressed as an applied force, flux, or current. This type of boundary condition is also called a *Neumann boundary condition*.
- *Constraints* specify the result of the interaction between the model and its surroundings, expressed as expected values of the dependent variables. This type of boundary condition is also called a *Dirichlet boundary condition*.

The two types are closely related because in a well-posed model, every flux condition results in some unique values of the dependent variables, and every constraint requires a unique flux to enforce the expected values. Which type of condition to use depends on what is known about the conditions at the boundary: if the flux is known, the model computes the dependent variables for you; if the values of the dependent variables are known, the model computes the flux.

FLUX CONDITIONS

Flux boundary conditions specify the component of a vector or tensor quantity in the direction normal to the boundary, per unit area of the boundary. Typical examples of flux conditions are the specification of:

- A [Boundary Load](#) in a solid model, which prescribes the stress acting on the boundary.
- [Heat Flux](#) in a heat transfer model, which prescribes the heat per unit area flowing into (or out of) the model across the boundary.
- A [Normal Current Density](#) in an AC/DC model, which prescribes the electrical current per unit area entering (or exiting) the model at the boundary.

There are also more advanced types of flux conditions, where the flux or force is calculated based on local values of dependent variables and other parameters. For example, a [Convective Heat Flux](#) boundary condition on a heated body computes the heat flux based on a heat transfer coefficient and the temperature difference to the surroundings.



Convective Heat Flux requires the CFD Module or Heat Transfer Module.



In COMSOL, by convention, the force acting *on* the model or the flux *into* the model is specified. That is, specify how the surroundings affect the model and not how the model affects its surroundings.

CONSTRAINTS

Constraint boundary conditions specify the value of one or more dependent variables at the boundary, or a relationship between two or more dependent variables. Typical examples include specifying:

- A [Prescribed Displacement](#) of the boundary of a solid object.
- That the velocity is zero on a [Wall](#) boundary in a CFD model.
- The [Temperature](#) at the boundary of a heated solid.
- The [Electric Potential](#) on an electrode in an AC/DC model.

Examples specifying a relation between dependent variables include [Roller](#) conditions on solids and [Wall](#) conditions for slip flow.

Because constraint conditions generally specify the value of a dependent variable, they also provide a reference level for that variable—which a flux condition normally does not. In many types of physics, the model equations together with only flux boundary conditions uniquely describe the local behavior of the dependent variable, but leave the global level undefined. From a physical point of view, the absolute value of the dependent variables are often of less interest, but the existence of a single, unique, solution is essential for some solvers.

Therefore it is often necessary to apply at least one constraint condition in a model, to provide a global reference value for the dependent variables. For example, it is common to designate one of the electrodes in an AC/DC model as **Ground**, which constrains the electric potential there to zero and gives a reference with which to compare other parts of the model.



In most physics, the default boundary condition is of *flux* type and does *not* fix a reference level for the dependent variable. Therefore when solving certain study types, notably **Stationary** studies, you must manually add at least one boundary condition of *constraint* type for the model to be well defined.

- 
- Constraint Reaction Terms
 - Weak Constraints
 - Constraint Settings
-

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.

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 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 geometry.

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.

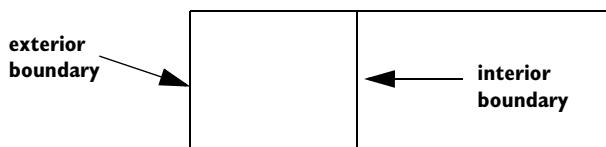


Figure 3-10: Examples of exterior and interior boundaries.

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. 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).



Identity and Contact 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 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.



- [Physics Nodes by Space Dimension](#)
- [Key to Nodes and Toolbar Buttons](#)

Constraint Reaction Terms

Enforcing a constraint condition is more or less a matter of finding a corresponding flux condition that leads to the desired values of the dependent variables. The hidden flux conditions introduced this way appear as *reaction terms* in the system of equations modeling the physics. These reaction terms normally have a physical meaning and correspond to a flux condition, for example:

- The reaction term enforcing a [Prescribed Displacement](#) on a solid model is a *reaction force*, similar to a [Boundary Load](#) boundary condition.
- The reaction term enforcing a [Pressure](#) in an acoustics model is a [Normal Acceleration](#).
- The reaction term enforcing a [Temperature](#) in a heat transfer model is a [Heat Flux](#).

The reaction terms in the model equations can be scaled in different ways, affecting mainly the numerics and solvers. In a model with more than one dependent variable, it is also possible to distribute the reaction fluxes, or forces, over the variables in different ways—while still enforcing the original constraint.

SYMMETRIC REACTION TERMS

Most boundary conditions of constraint type, by default, introduce reaction terms in such a way that an otherwise symmetric systems of equations remains symmetric. This

makes constraints *bidirectional* in the sense that all dependent variables that appear in a constraint expression are also affected by the reaction terms.

To illustrate this, suppose a **Prescribed Displacement** boundary condition is applied on a solid model, specifying that the x -displacement of the boundary, u , is proportional to the y -displacement, v , with a constant of proportionality, k , which is a function of the boundary temperature T :

$$u = k(T)v \quad (3-1)$$

If fully symmetric reaction terms are used to enforce this constraint, reaction forces are applied on both displacement components u and v , as well as a reaction heat flux in the heat transfer equation. Applying symmetric reaction terms this way, on completely different equations, usually makes no sense.

In particular, the solid displacement equation and the heat transfer equation have different units. Because you can choose length and temperature unit independently, the relative scale of the equations is undefined and the symmetry of the coupled system irrelevant. Further you would not, from a physical point of view, expect a constraint on the displacement of a solid boundary to directly affect the temperature field in a model.

RESTRICTED AND NONSYMMETRIC REACTION TERMS

As an alternative to the default (symmetric) application of reaction terms, you can choose to have these affect only the equations and variables in the physics interface where the constraint boundary condition is added. For the example in [Equation 3-1](#), the reaction terms can be restricted to act on the displacement variables and equations in the Solid Mechanics user interface, leaving the temperature unaffected. Many different restrictions of this type are possible, in principle, and COMSOL generally provides two alternatives:

- The most consistent and general way to avoid spurious reaction terms affecting other physics is to start from the globally symmetric formulation and remove the terms entering equations belonging to other physics interfaces. This limits the reaction terms to affecting the current physics in such a way that if there are no other physics in the model, the reaction terms preserve the symmetry. For [Equation 3-1](#) this means that reaction terms are distributed over both u and v equations, in proportions $1:k(T)$.
- The other alternative is to apply the reaction terms only on certain individual variables. Another way to look at [Equation 3-1](#) is to read it as prescribing a value for the x -displacement u , rather than prescribing a given relation between u and v . Accepting that view, it is reasonable to insert reaction terms only acting on u . Such

reaction terms, in general, do not preserve symmetry even for a single physics interface.

Weak Constraints

The standard method to enforce constraints in COMSOL applies the constraints pointwise at node points in the mesh. At each node point, only local values of the dependent variables are affected by the constraint, making the constraints independent of each other. The solvers can therefore eliminate both the constrained degrees of freedom and the constraint force terms, effectively reducing the system of equations and decreasing the number of degrees of freedom being solved for.

Weak constraints enforce the constraint in a local average sense, using shape functions as weights. Reaction terms are explicitly included in the system of equations, which is extended with Lagrange multiplier variables. These Lagrange multipliers in general have a physical meaning and an interpretation as a constraint force or flux. Whereas a standard constraint decreases the number of degrees of freedom by the number of unique constraints, weak constraints increase the degrees of freedom by the same number.

A weak constraint is respected only on average over each Lagrange multiplier shape function, rather than pointwise at mesh nodes. When it is possible to satisfy the constraint everywhere on each mesh element, standard and weak constraints in general lead to the same solution. Conversely, when constraints are contradictory or impossible to satisfy everywhere, standard and weak constraints may distribute the error differently, and therefore lead to slightly different solutions.

Weak constraints can be of use in the following situations:

- Standard constraints must never contain time derivatives of the dependent variables. Weak constraints do not have this limitation because they allow the same variables as any other term in the combined system of equations. Note that reaction terms cannot be applied symmetrically to time derivatives but must be selectively applied to individual variables.
- When the reaction force or flux is needed during a solution, because it enters into a coefficient somewhere, the Lagrange multiplier from a weak constraint may provide an accurate value (see [Computing Accurate Fluxes](#)). The corresponding variables

computed from derivatives of the dependent variables are not as accurate and may, if used, introduce considerable errors in the solution.

- When constraints are strongly nonlinear, weak constraints may allow faster and more robust convergence. For nonlinear constraints, the true linearized subproblem solved in each solution step depends on the value of the Lagrange multiplier variables from the previous step. When using standard constraints, this information is discarded between solution steps. Using weak constraints, the Lagrange multiplier values are instead retained between steps because they are part of the solution vector.

Compared to standard, eliminated, constraints, weak constraints may also have the following drawbacks:

- Discontinuous constraints result in (theoretically) infinite Lagrange multipliers. In practice large oscillations result.
- Pointwise and weak constraints on the same set of variables on adjacent boundaries (that is, 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 you 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).
- 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.
- Because extra unknowns are introduced for the Lagrange multipliers, the size of the problem increases compared to the standard constraint elimination method.
- The Lagrange multiplier variables added by the weak constraints have a different unit than the main system variables and can therefore be of a completely different order of magnitude. This may lead to scaling problems. Usually the automatic variable scaling in the solvers is sufficient, but there are cases when manual scaling is needed.
- Weak constraints introduce zeros on the main diagonal of the Jacobian matrix of the discretized system, which therefore cannot be positive definite. This makes certain linear solvers and preconditioners unavailable for solving problems with weak constraints. In particular, the conjugate gradients iterative solver does not work, and neither does the SOR class of preconditioners and smoothers. Instead try another iterative solver and use the Vanka algorithm with the Lagrange multipliers as the

Vanka variables, or use the incomplete LU factorization algorithm as preconditioner.

Constraint Settings

Most constraint nodes have a **Constraint Settings** section which is only available when **Advanced Physics Options** is selected from the **Show** menu (≡). This section provides settings controlling how reaction terms are applied and whether standard or weak constraints are used. Choose to **Apply reaction terms on:**

- **All physics (symmetric)** to apply reaction terms symmetrically on all dependent variables taking part in the constraint.
- **Current physics (internally symmetric)** to apply reaction terms symmetrically only on the dependent variables in the physics where the constraint is added. This leaves other physics unaffected by the constraint.
- **Individual dependent variables** to apply reaction terms only on selected variables. For most physics, this makes the constraint *unidirectional* and often nonsymmetric.

Select the **Use weak constraints** check box to replace the point-wise standard constraints with weak constraints. Note that this introduces additional equations and dependent variables.



Not all constraints provide all the above options. Some reaction term methods may be missing and weak constraints may not be allowed. Some constraint nodes may also implement additional options.

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 so that $u(x_0) = u(x_1)$ (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 so that $u(x_0) = -u(x_1)$. For fluid flow user 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.



For some physics 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 use, see [Periodic Condition](#).



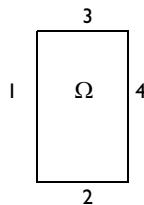
This model makes use of periodic boundary conditions: [The KdV Equation and Solitons](#): Model Library path [COMSOL_Multiphysics/Equation-Based_Models/kdv_equation](#).

Periodic Boundary Condition Example

MODEL DEFINITION

Consider an eigenvalue equation on a rectangle of dimension $1 \times \sqrt{\pi}$.

$$\begin{aligned} -\Delta u &= \lambda u && \text{on } \Omega \\ u(x_1, y) &= u(x_4, y) && \text{on Boundaries 1 and 4} \\ u(x, y_2) &= u(x, y_3) && \text{on Boundaries 2 and 3} \end{aligned}$$



RESULTS AND DISCUSSION

lambda(6)=50.279769 Surface: Dependent variable u

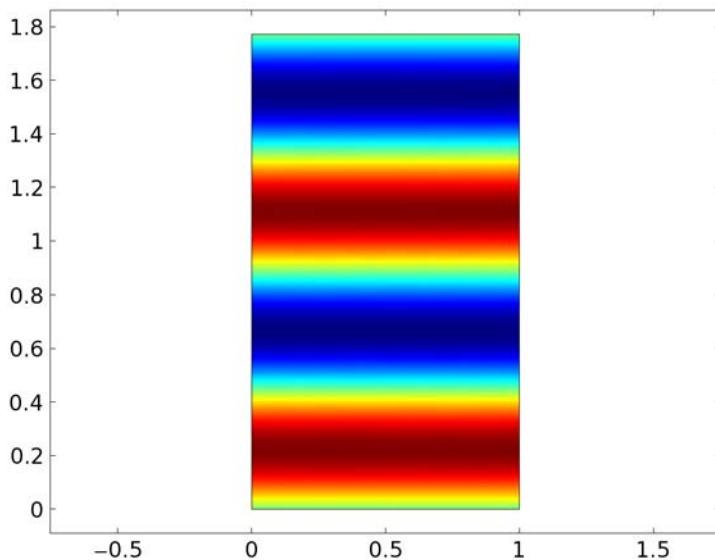


Figure 3-11: The periodic pattern in the solution (6th eigenmode).

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 phase of the solution.

MODELING INSTRUCTIONS

Model Wizard

- 1 Start COMSOL.
- 2 In the **Model Wizard**, select **2D** from the **Select Space Dimension** list. Click **Next** (➡).
- 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** (RECT).
- 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 **Coefficient Form PDE** and select **Periodic Condition**.
- 2** Select boundaries 1 and 4 in the **Graphics** window and add them to the **Boundary Selection** list.
- 3** Right-click **PDE** and select **Periodic Condition**.
- 4** Select boundaries 2 and 3 in the **Graphics** window and add them to the **Boundary Selection** list.

In this case, COMSOL 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>Step 1: Eigenvalue** 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 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

Computing Accurate Fluxes

Flux Computation Methods

COMSOL provides three ways of computing accurate fluxes and reaction forces:

- The first approach involves the reaction force operator (`reacf`) that makes it possible to compute integrals of reaction forces or fluxes during analysis. See [reacf](#) 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 [Weak Constraints](#) for more information about using weak constraints.

-
- Some physics provide a third way of computing accurate fluxes. Under the **Discretization** section, select the **Compute boundary fluxes** check box. The solver then computes variables storing an accurate boundary flux from each boundary into the adjacent domain (in addition to the standard extrapolated value). On interior boundaries, there are two flux variables corresponding to the flux into the domains on either side of the boundary. Unlike the other methods, these variables are available also on unconstrained boundaries. This method is active by default in Coefficient Form PDE, General Form PDE, heat transfer, and mass transport user interfaces. There is also an **Apply smoothing to boundary fluxes** check box that is selected by default. The smoothing can provide a more well-behaved flux value close to singularities.

When using weak constraints in physics user interfaces, the Lagrange multipliers are additional dependent variables in those physics user 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 in the physics user interfaces:

TABLE 3-2: INTERPRETATION OF LAGRANGE MULTIPLIERS

PHYSICS	QUANTITY
Electrostatics	Surface charge density
Magnetic Fields	Surface current
Electric Currents	Current density
Heat Transfer	Heat flux
Transport of Diluted Species	Flux
Solid Mechanics	Force per area
Pressure Acoustics	Normal displacement (acceleration for eigenfrequency studies)
Laminar Flow	Total force per area

The sign of the Lagrange multiplier is the same as the one used when applying the corresponding quantity explicitly in a flux condition. As a general rule, the sign corresponds to an action by the surroundings on the model, rather than the opposite.

COMSOL 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 user interfaces, though.



Lagrange multipliers in axial symmetry are not equal to the reaction flux per area but rather per length and full revolution.



Flux Calculation Example—Heat Transfer Model

The reaction forces are computed from the value of the residual vector L at every node point where a constraint is applied. Therefore the reaction forces should be thought of as discrete values at each node point rather than continuous fields.

The boundary flux variables are computed in a similar way to the reaction forces but with two important differences:

- First, on each boundary, the contributions to the residual vector from the boundary and from the adjacent domains are computed separately. This makes it possible to compute the flux into each adjacent domain even when there is no constraint on the boundary so that the full residual vector is zero.
- Second, the nodal fluxes computed from the residual vector are further processed and represented as a continuous field on the boundary. The integral of this flux field over a boundary is equal to the sum of the nodal fluxes.

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 boundaries 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, 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 (1 W/m^2) 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 The following step is only needed to show how to use a Lagrange multiplier for an accurate flux. Built-in variables for accurate fluxes are available directly also without this step. 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>Integration>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_in Solids>Boundary_fluxes>Total normal_heat flux, accurate** (the variable `ht.ntflux_acc`).
- 4 Click the **Evaluate** button () at the top of the settings window. The total normal heat flux across these boundaries appears in the **Table** window under **Total normal heat flux, accurate (W/m)** and is exactly equal to the influx of 1 (the normal flux is by convention positive in the direction of the normal). If you instead evaluate **Heat Transfer_in Solids>Boundary_fluxes>Total normal_heat flux, extrapolated** the value is about 0.986, which is close to but not exactly 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_{1m}).
- 6** Click the **Evaluate** button () at the top of the settings window. The total heat flux across these boundaries appears in the **Table** window under **Lagrange multiplier for temperature** and is -1 —exactly equal to the influx (but with opposite sign) without the need for a computationally expensive extremely fine mesh. This makes this method useful for physics where built-in accurate flux variables are not available.

Using Load Cases

For linear stationary problems it can be of interest to see the solution for several different loads 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 supports load cases that are combination of loads, with optional weights, and constraints.



- Study Steps and Solver Configurations
- Stationary

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 structural mechanics as well as boundary conditions such as heat flux (a load) and temperature (a constraint) in heat transfer support load groups and constraint groups. You can create load groups and constraint groups in two ways—from the **Global Definitions** node’s context menu or a physics node context menu. Both methods add the node under **Global Definitions**.

Add a Load or Constraint Group from the Global Definitions Context Menu

Add 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 display under the **Load and Constraint Groups** node (). See [Figure 3-12](#).

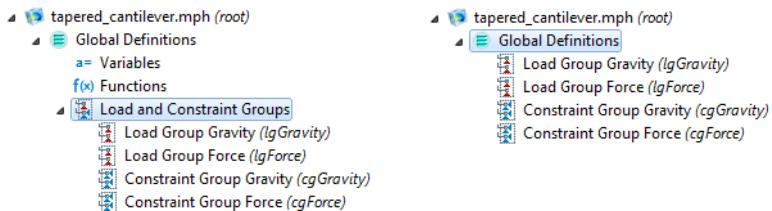


Figure 3-12: An example of the node grouping when Sort by Type is selected (left) and when No Sorting is selected (right).

Add a Load or Constraint Group from a Physics Node Context Menu

Right-click a physics node for any load or constraint (for example, a Fixed Constraint, Heat Source, or Boundary Load node) 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 under Global Definitions and at the same time assigns that physics node (a load or a constraint) to that group.

Assign a Load or Constraint to a Group

To assign a load or a constraint to a load group or constraint group, right-click the physics node for a load or constraint and from the **Load Group** or **Constraint Group** submenu choose one of the following (see Figure 3-13):

- **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.

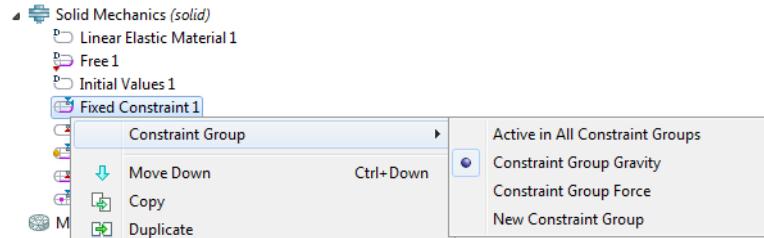


Figure 3-13: An example of the context menu options when a load or constraint physics node is right-clicked, in this case, the Fixed Constraint node.

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 (see [Defining and Evaluating Load Cases](#)).

When the Load Group or Constraint Group is applied to a node under a physics, the node indicates this visually. For example, the **Fixed Constraint** and **Roller** nodes have the blue **Constraint Group** symbol in the upper right corner and the **Body Load** and **Boundary Load** nodes have the red **Load Group** symbol in the upper right corner as in [Figure 3-14](#).

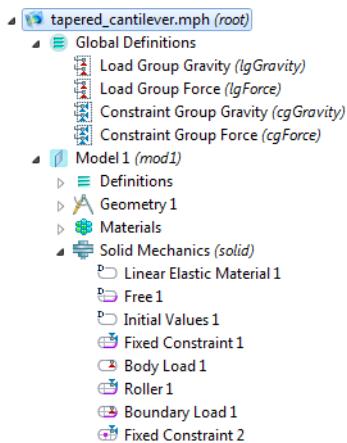


Figure 3-14: An example of the Load and Constraint Groups defined under Global Definitions with loads and constraints applied to nodes under Solid Mechanics.

-
- [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.
-

Load Group

Add a **Load Group** () to the **Global Definitions** branch 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 or heat transfer model to analyze the effects of various loads or sources.

GROUP IDENTIFIER

In the **Identifier** field, the default group identifier (lg1) can be modified if you want to use a more descriptive name, for example, **lgGravity**. You can also create load groups from physics nodes for structural mechanics that represent loads and support load cases: Right-click the physics node and choose **Load Group>New Load Group**.

Constraint Group

Add a **Constraint Group** () to the **Global Definitions** branch 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.

GROUP IDENTIFIER

In the **Identifier** field you can modify the default group identifier (cg1) if you want to use a more descriptive name (for example, **cgForce**). You can also create constraint groups from physics nodes for structural mechanics that represent constraints and support load cases: Right-click the physics node and choose **Constraint Group>New Constraint Group**.



If you select **Sort by Type** from the context menu, either right-click the **Global Definitions** node or the **Loads and Constraints Groups** node () to add a **Load Group** or **Constraint Group**.

Defining and Evaluating Load Cases

You define load cases in the **Stationary** study node's settings window. Follow these steps to create load cases:

- 1 In the **Model Builder** under **Study**, in the **Stationary** study () settings window click to expand the **Study Extensions** section.
- 2 Select the **Define load cases** check box (see [Figure 3-15](#)).
- 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 in order 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 load's direction.

The screenshot shows the 'Study Extensions' section of the COMSOL interface. It includes two main parts: 'Continuation' and 'Define load cases'.

Continuation: A table with columns 'Continuation parameter' and 'Parameter value list'. There are four rows, each with a row editor icon at the bottom.

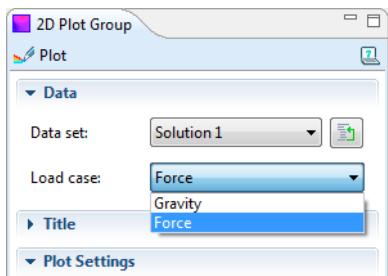
Continuation parameter	Parameter value list

Define load cases: A table with columns 'Load case', 'IgGravity', 'Weight', 'IgForce', 'Weight', 'cgGravity', and 'cgForce'. Two rows are shown: 'Gravity' and 'Force'.

Load case	IgGravity	Weight	IgForce	Weight	cgGravity	cgForce
Gravity		1.0		1.0		
Force		1.0		1.0		

Figure 3-15: An example of the Stationary node's Study Extensions section. Click in the table of load cases to select or remove loads and constraints from the load case.

When you have defined all load cases you can compute the solution. COMSOL 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** 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** ().



- Working with Studies and Solvers
- Stationary study step

Using Units

COMSOL 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 the unit tables, “N/A” means that no unit symbol is available.

Unit Systems in COMSOL

COMSOL 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, 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 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 newton and megapascal (hence the name "MPa system") are naturally derived units of force and pressure, respectively. 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 nondimensionalized (de-dimensionalized or dimensionless) 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.



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 3-3](#) as a guide for the format to enter.

TABLE 3-3: 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}
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, $200[\text{ft}]$ and $3e6[\text{kg/m}^3]$.

Both the name and the symbol can be used for a unit. For example, $2.4[\text{ampere}]$ and $2.4[\text{A}]$ are both valid to indicate an electric current in SI 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 $50[\text{cm}]$). Also, if L is a variable defined as $2[m]$, $L[1/s]$ evaluates to $2[\text{m/s}]$.

The following examples show how to apply the unit syntax:

- Adding two quantities of the same kind that use different units: $0.5[\text{ft}]+33[\text{mm}]$. COMSOL converts the result to the base unit system's length unit.
- Using multiplication with a unit to get consistent units for two quantities that you want to add, for example, $14[\text{kg}]+\text{ht.rho}[\text{m}^3]$, which works if ht.rho represents the density for a heat transfer model. You can also concatenate several units, for example, $3.6[\text{N}][\text{m}]$, which is equivalent to typing $3.6[\text{N*m}]$ and evaluates to $3.6\text{ N}\cdot\text{m}$.



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 then assumes that this value has the unit of the unit for that quantity (as indicated in the settings window).

All data in the material databases and Material Library product use SI units and this unit syntax.

DECLARING UNITS FOR PARAMETERS, VARIABLES, AND FUNCTIONS



It is important to be aware of the following aspects of unit handling.

When using parameters, variables, and functions in expressions:

- If user-defined parameters or variables are used in the physics it is good practice to use the unit syntax to define them. The settings windows for parameters and variables display the resulting unit, in the models base unit system, of user-defined parameters and variables. 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 is otherwise undefined.

- Most user-defined and built-in functions expect dimensionless inputs and outputs, so it is good practice to use inputs such as time dimensionless using unit syntax. If the input is not dimensionless, COMSOL 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.
- If other units than the base unit system’s units are used or if SI prefixes are included, the conversion to base 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** settings window displays the quantity and unit in the base 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** if the base unit system is SI.

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 (for example, `10[m/s]` uses the SI unit for velocity).

- [Table 3-4](#) lists the SI units for the seven base quantities.
- [Table 3-5](#) lists the SI derived units supported in COMSOL.
- [Table 3-6](#) 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 3-6](#) are only available when using such non-SI unit systems.

- [Table 3-7](#) lists other SI derived units without special names or symbols.

TABLE 3-4: BASE SI UNITS

BASE QUANTITY	UNIT 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 3-5: SI DERIVED UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
absorbed dose	gray	Gy
capacitance	farad	F
conductance	siemens	S
dose equivalent	sievert	Sv
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
inductance	henry	H
magnetic flux	weber	Wb
magnetic flux density, magnetic induction	tesla	T
plane angle	radian	rad
power	watt	W
pressure	pascal	Pa

* See the additional notes following the next table.

TABLE 3-6: ADDITIONAL UNITS IN COMSOL

QUANTITY	NAME	SYMBOLS	VALUE
acceleration	galileo	Gal	0.01 m/s ²
dipole moment	debye	D	3.33564095·10 ⁻³⁰ C·m
dynamic viscosity	poise	P	0.1 Pa·s
energy	British thermal unit*	BTU, Btu	1055.05585 J
energy	calorie*	cal	4.184 J
energy	electronvolt	eV	1.6021765314·10 ⁻¹⁹ J
energy	erg	erg	10 ⁻⁷ J
force	dyne	dyn	10 ⁻⁵ N
force	kilopond*	kp, kpf	9.80665 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
length	mile*	mi	1609.344 m
length	microinch	uin	0.0254·10 ⁻⁶ m
length	milliinch	mil, thou	0.0254·10 ⁻³ m
length	nautical mile*, nautimile	nmi	1852 m
length	yard	yd	0.9144 m
magnetic field strength	oersted	Oe	10 ³ /(4·π) A/m
magnetic flux density	gauss	G	10 ⁻⁴ T
mass	atomic mass unit, dalton	u, amu, Da	1.660538782·10 ⁻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	9.869233·10 ⁻¹⁶ m ²
plane angle	degree	deg	π/180

TABLE 3-6: ADDITIONAL UNITS IN COMSOL

QUANTITY	NAME	SYMBOLS	VALUE
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
pressure	inches water*	inAq, inH2O	249.089 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·T+459.67
temperature	Rankine**	R, Ra	5/9·T
time	year*	a, yr	31556952 s
time	day	d	86400 s
time	hour	h	3600 s
time	minute	min	60 s
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
volumetric 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 UNITS IN Table 3-5 AND Table 3-6

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.
kilopond	The kilopond (kp) or kilogram-force (kpf) is a gravitational metric unit of force. Refer to this unit using kilopond, kp, or kpf only.
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.
ohm	To declare the SI unit for electric resistance, ohm, use [ohm]. COMSOL then displays this as Ω .
inches water	The value of 1 inch of water is defined as the pressure exerted by one inch of water for a pure water density of 1000 kg/m ³ at 4 degrees Celsius and standard gravity of 9.80665 m/s ² . Refer to this unit using the symbol only (inH ₂ O or inAq).
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 3-7: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

DERIVED 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)

TABLE 3-7: EXAMPLES OF SI DERIVED UNITS WITHOUT SPECIAL NAMES

DERIVED QUANTITY	NAME	SYMBOL
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 3-8: 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 3-9: DERIVED BRITISH ENGINEERING UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
force	pound-force	lbf

The British thermal unit is also available as Btu or BTU.



If the British engineering unit system is the base unit system, COMSOL constructs derived units from the applicable SI base units and the units listed in [Table 3-8](#) and [Table 3-9](#). 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 3-10: 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 3-11: DERIVED CGSA UNITS IN COMSOL

DERIVED 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 3-12: SPECIAL BASE UNITS IN THE EMU UNIT SYSTEM

BASE QUANTITY	NAME	SYMBOL
length	centimeter	cm
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 3-13: DERIVED EMU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
acceleration	galileo	Gal
capacitance	abfarad	N/A
conductance	absiemens	N/A

TABLE 3-13: DERIVED EMU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
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 3-14: 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 3-15: DERIVED ESU UNITS IN COMSOL

DERIVED 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

TABLE 3-15: DERIVED ESU UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
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 3-16: 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 3-17: DERIVED FPS UNITS IN COMSOL

DERIVED 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 3-18: 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 3-19: 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 3-20: DERIVED MPa UNITS IN COMSOL

DERIVED 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 3-21: 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 3-22: DERIVED GRAVITATIONAL IPS UNITS IN COMSOL

DERIVED QUANTITY	NAME	SYMBOL
force	pound-force	lbf
pressure	psi	psi

Switching Unit System

If the unit system is switched during modeling, COMSOL does not convert the data in the model (except for length units and angular units in the geometry if specified). All physical constants and data in the material libraries are in SI units and defined using

the unit syntax, so you can use them with any unit system because COMSOL converts these values to the corresponding values in the model's unit system.



The units of other input data must be declared for the model using the unit syntax (or manually convert the numerical values to the new unit system).

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. One example is the Stefan-Boltzmann law for blackbody 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 3-23](#) for a list of acceptable unit syntax.

TABLE 3-23: TEMPERATURE UNITS

SCALE	UNIT
Celsius	[degC]
Fahrenheit	[degF]
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 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 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:

- $100[\text{degC}]$ is an expression that has temperature as the dimension. COMSOL interprets it as an absolute temperature and evaluates it as 373.15 K.
- $373.15[1/\text{K}]$ is interpreted as an absolute inverse temperature (but no conversion is necessary from kelvin to kelvin).
- $373.15[1/\text{degC}]$ evaluates to $100[1/\text{K}]$ using the offset of 273.15 degrees between kelvin and degrees Celsius.
- $100[\text{degC/K}]$ 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 interpret $100[\text{degC/K}]$ as an absolute temperature, split the expression using two separate expressions such as $100[\text{degC}]*1[1/\text{K}]$, which equals 373.15. This is also what occurs when you use a variable (TC, for example) defined as $100[\text{degC}]$. 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.



The length unit for the geometry does not affect the units that include length in the physics interfaces or any other part of COMSOL.

-
- 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**.

When importing 3D CAD geometries, you can choose to use the length unit from the CAD file or the length unit from COMSOL.



The Geometry Node

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 $273[\text{K}]+3[\text{ft}]$, 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.

If an unexpected or inconsistent unit appears in a text field for a physical property, COMSOL 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 $2930[\text{K}]$ as 293 kg/m^3 and $2930[\text{mK}]$ 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 user 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 user 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.

Numerical Stabilization

About Numerical Stabilization in COMSOL

This section discusses the numerical stability of the generic scalar convection-diffusion transport equation

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + F \quad (3-2)$$

where β is the convective velocity vector, c is the diffusion coefficient, u is a transported scalar, and F is a source term. The underlying finite element discretization method in COMSOL Multiphysics is the Galerkin method. When discretizing [Equation 3-2](#) using the Galerkin method, it is well known that the resulting numerical problem becomes unstable for an element Péclet number (Pe) larger than one ([Ref. 1](#)):

$$Pe = \frac{\|\beta\| h}{2c} > 1 \quad (3-3)$$

where h is the mesh element size. The Péclet number is a measure of the relative importance of the convective effects compared to the diffusive effects; a large Péclet number indicates that the convective effects dominate over the diffusive effects.

Oscillations can occur where any of the following conditions exist and the Péclet number exceeds one:

- A Dirichlet boundary condition can lead to a solution containing a steep gradient near the boundary, forming a boundary layer. If the mesh cannot resolve the boundary layer, this creates a local disturbance.
- A space-dependent initial condition that the mesh does not resolve can cause a local initial disturbance that propagates through the computational domain.
- A small initial diffusion term close to a nonconstant source term or a nonconstant Dirichlet boundary condition can result in a local disturbance.

As long as diffusion is present, there is—at least in theory—a mesh resolution beyond which the discretization is stable. This means that the spurious oscillations can be removed by refining the mesh. In practice, this method is seldom feasible because it can require a very dense mesh. Instead, it is common practice to use stabilization methods; that is, methods that add artificial diffusion. COMSOL provides several of these methods and described in [An Example of Stabilization](#).

An Example of Stabilization

This example uses the Heat Transfer user interface. To illustrate the concepts, consider the problem

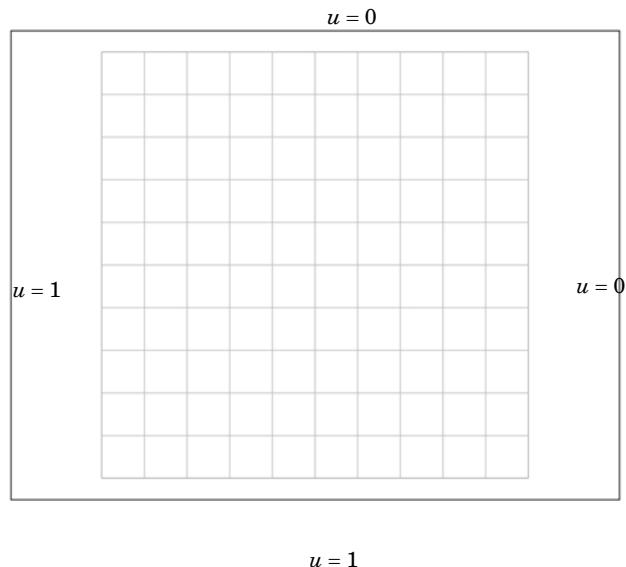
$$\cos\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial x} + \sin\left(\frac{\pi}{3}\right)\frac{\partial u}{\partial y} = 10^{-4}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + 1 \quad (3-4)$$

solved on the unit square. [Equation 3-4](#) is discretized using 10 times 10 biquadratic Lagrangian elements. The boundary conditions are:

- $u = 1$ for $x = 0$
- $u = 1$ for $y = 0$
- $u = 0$ for $x = 1$
- $u = 0$ for $y = 1$

[Figure 3-16](#) shows the mesh and boundary conditions. In general, using uniform meshes for transport problems is not recommended. Nevertheless, this example uses a uniform mesh to demonstrate the different stabilization techniques.

The expected solution rises slowly and smoothly from the left and lower boundaries and has sharp boundary layers along the upper and right boundaries. [Figure 3-17](#) shows a reference solution obtained using 100-by-100 biquadratic Lagrangian elements with streamline diffusion and crosswind diffusion (see the next section). The arrows indicate the direction of β .



$$u = 1$$

$$u = 0$$

Figure 3-16: The computation domain, mesh, and boundary condition for Equation 3-4.

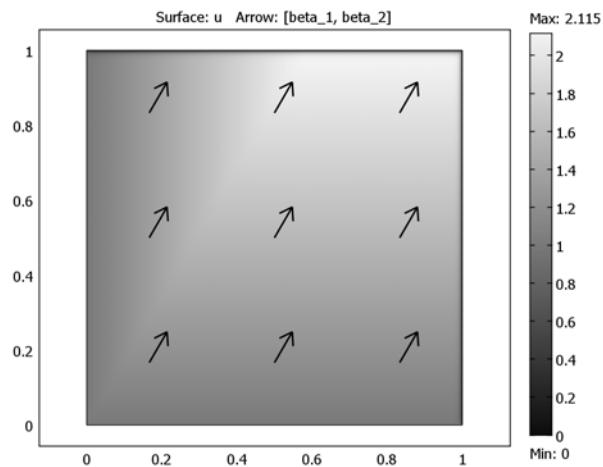


Figure 3-17: Reference solution of Equation 3-4. Solved using 100 times 100 biquadratic elements with streamline diffusion and crosswind diffusion.

The cell Péclet number for this example is

$$\text{Pe} = \frac{1 \cdot 0.1}{2 \cdot 10^{-4}} = 500 >> 1$$

[Figure 3-18](#) displays the solution obtained using the mesh in [Figure 3-16](#) and (unstabilized) Galerkin discretization. As can be expected with such a high Péclet number, the unstabilized solution shows little, if any, resemblance to the reference solution in [Figure 3-17](#). The right plot in [Figure 3-18](#) shows a cross-sectional plot along the dashed line, $y = 0.8$ and the corresponding reference solution. Notice that the unstabilized solution is completely destroyed by oscillations.

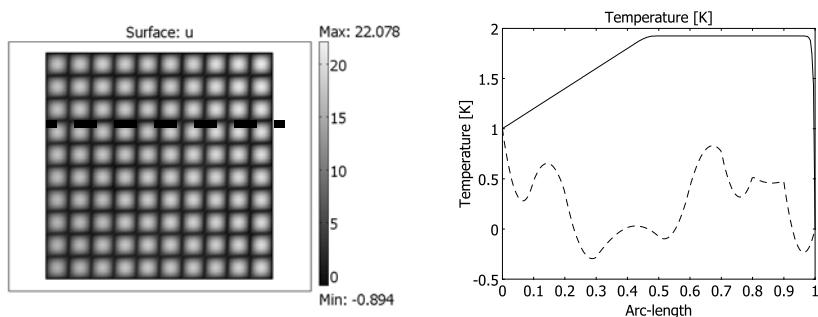


Figure 3-18: Equation 3-4 solved using unstabilized Galerkin formulation. The right plot compares the unstabilized solution (dashed line) along the dashed line in the left plot ($y = 0.8$) with the reference solution (solid line).

The [Stabilization Techniques](#) section explores how different stabilization techniques affect the solution of this example.

Stabilization Techniques

Several techniques for handling numerical instabilities without the need for mesh refinement are available. They all have in common that they add terms to the transport equation. These terms introduce numerical diffusion (artificial diffusion, artificial

viscosity, or numerical viscosity are other common names) that stabilize the solution. To display these sections, click the **Show** button () and select **Stabilization**.

-
- 
 - [Show Stabilization](#)
 - [Consistent and Inconsistent Stabilization Methods for the Heat Transfer User Interfaces](#)
 - [Numerical Stability—Stabilization Techniques for Fluid Flow](#)
-

CONSISTENT STABILIZATION

A consistent stabilization method adds numerical diffusion in such a way that if u is an exact solution to [Equation 3-2](#), then it is also a solution to the problem with numerical diffusion. In other words, a consistent stabilization method gives less numerical diffusion the closer the numerical solution comes to the exact solution.

INCONSISTENT STABILIZATION

An inconsistent stabilization method adds numerical diffusion in such a way that if u is an exact solution to [Equation 3-2](#), then it is not necessarily a solution to the problem with numerical diffusion. In other words, an inconsistent method adds a certain amount of diffusion independently of how close the numerical solution is to the exact solution.

ISOTROPIC DIFFUSION

Adding isotropic diffusion is equivalent to adding a term,

$$c_{\text{art}} = \delta_{\text{id}} h \|\beta\|$$

to the physical diffusion coefficient, c . Here δ_{id} is a tuning parameter. This means that you do not solve the original problem, [Equation 3-2](#), but rather the modified $O(h)$ -perturbed problem

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot ((c + c_{\text{art}}) \nabla u) + F \quad (3-5)$$

Hence, isotropic diffusion is an inconsistent stabilization method. If $\delta_{\text{id}} = 0.5$, the new cell Péclet number can be expressed as

$$\text{Pe} = \frac{h \|\beta\|}{2(c + c_{\text{art}})} = \frac{h \|\beta\|}{2c + h \|\beta\|}$$

Clearly, as $\|\beta\|$ approaches infinity, Pe approaches, but never exceeds, one. While a solution obtained with isotropic diffusion might not be satisfactory in all cases, the added diffusion definitely dampens the effects of oscillations and impedes their propagation to other parts of the system. It is not always necessary to set δ_{id} as high as 0.5 to get a smooth solution, and it should be chosen smaller if possible. A good rule of thumb is to select $\delta = 0.5/p$, where p is the order of the basis functions. The default value is $\delta_{id} = 0.25$.

[Figure 3-19](#) shows the effect of isotropic diffusion on [Equation 3-4](#) with $\delta_{id} = 0.25$. Although the solution is smooth, the comparison with the reference solution in the right plot reveals that the isotropic diffusion introduces far too much diffusion.

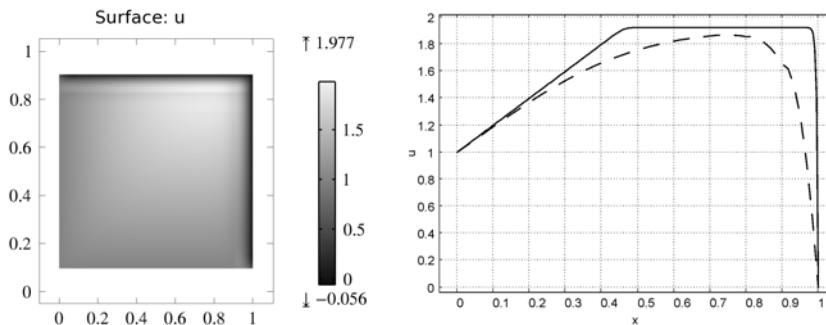


Figure 3-19: [Equation 3-4](#) solved using isotropic diffusion. The right plot compares the stabilized solution (dashed line) along $y = 0.8$ with the reference solution (solid line).

STREAMLINE DIFFUSION

The streamline diffusion method in COMSOL is a consistent stabilization method. When applied to [Equation 3-2](#), it recovers the streamline upwind Petrov-Galerkin (SUPG) method, but it can also recover functionality from the Galerkin least-squares (GLS) method. Both methods are described below. For theoretical details, see [Ref. 1](#) and [Ref. 2](#).

Streamline Upwind Petrov-Galerkin (SUPG)

The theory underlying SUPG is a bit too complicated to describe here, but the resulting expressions can be shown to be closely related to upwinding schemes in finite difference and finite volume methods. SUPG can be shown to add a smaller amount of stability than isotropic diffusion (see [Ref. 3](#)), but while the accuracy of isotropic diffusion is at best $O(h)$, the accuracy of SUPG can be shown to be at least $O(h^{p+1/2})$ where $p \geq 1$ is the order of the basis functions.

Figure 3-20 displays the effect of SUPG on the solution of [Equation 3-4](#). The solution closely follows the reference solution away from the boundary layers, but at the boundary layers, oscillations occur. This is a typical behavior for streamline diffusion: the solution becomes smooth and exact in smooth regions but can contain local oscillations at sharp gradients.

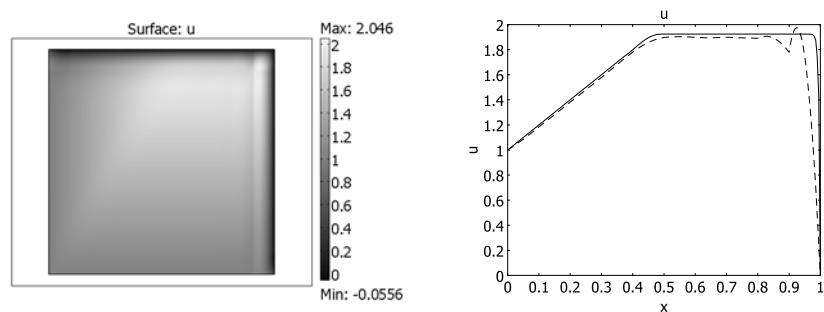


Figure 3-20: Equation 3-4 solved using streamline diffusion. The right plot compares the stabilized solution (dashed line) along $y = 0.8$ with the reference solution (solid line).

Galerkin Least-Squares (GLS)

Galerkin least-squares (GLS) is a more advanced version of SUPG, with which it shares many features. GLS, for example, is also a consistent method and has the same order of accuracy as SUPG. To understand the differences between GLS and SUPG, consider the following extended form of [Equation 3-2](#):

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nabla \cdot (c \nabla u) + su + F \quad (3-6)$$

where s is a production coefficient if $s > 0$ and an absorption coefficient if $s < 0$. If $s \neq 0$, the numerical solution of [Equation 3-6](#) is characterized by the Péclet number (see [Equation 3-3](#)) and the element Damköhler number:

$$Da = \frac{|s| h}{\|\beta\|}$$

A new dimensionless number can be formed by combining the Damköhler number and the Péclet number:

$$2DaPe = \frac{|s| h^2}{c} \quad (3-7)$$

The (unstabilized) Galerkin discretization becomes unstable if $2\text{DaPe} > 1$ (Ref. 4), that is, if the production/absorption effects dominate over the viscous effects. GLS differs from SUPG in that GLS relaxes this requirement while SUPG does not.¹

CROSSWIND DIFFUSION

Streamline diffusion introduces artificial diffusion in the streamline direction. This is often enough to obtain a smooth numerical solution provided that the exact solution of Equation 3-2 (or Equation 3-6) does not contain any discontinuities. At sharp gradients, however, undershoots and overshoots can occur in the numerical solutions as can be seen in Figure 3-20. Crosswind diffusion addresses these spurious oscillations by adding diffusion orthogonal to the streamline direction—that is, in the crosswind direction.

Crosswind diffusion methods are consistent, but they are also nonlinear. This means that the discrete equation system becomes nonlinear even if the original equation (Equation 3-2 or Equation 3-6) is linear, which can increase the computational cost.



Use crosswind diffusion if it is important to avoid undershoots or overshoots. Typical examples are concentrations that must not become negative and mass fractions that must be between zero and one.

The crosswind diffusion option adds a weak contribution as suggested in Ref. 5. For the scalar example here, the term reads

$$-\nu^h \frac{\partial c}{\partial x_i} g^{ij} \frac{\partial c}{\partial x_j}$$

where g^{ij} is the covariant metric tensor. The coefficient ν^h is for Navier-Stokes systems a modified version of the Hughes-Mallet (HM) formulation of Ref. 6. In the scalar case, the modified HM formulation reduces effectively to the form suggested in Ref. 6. Additionally, Ref. 7 suggests to reduce ν^h for higher-order elements. The COMSOL formulation multiplies ν^h with a factor

$$(\sqrt{2})^{1-N}$$

where N is the shape function order.

Figure 3-21 shows the example problem (Equation 3-4) solved using streamline diffusion and crosswind diffusion. Oscillations at the boundary layers are almost

1. The streamline diffusion stabilization in COMSOL is GLS but without any viscous terms in the test operator in the stabilization term.

completely removed (compare with [Figure 3-20](#)), but it has been achieved by the introduction of some extra diffusion. In general, crosswind diffusion tries to smear out the boundary layer so that it becomes just wide enough to be resolved on the mesh ([Figure 3-16](#)). To obtain a sharper solution and remove the last oscillations, the mesh needs to be refined locally at the boundary layers.

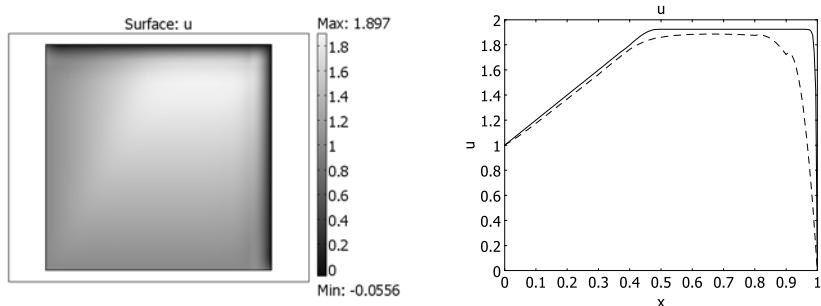


Figure 3-21: Equation 3-4 solved using streamline diffusion and crosswind diffusion. The right plot compares the stabilized solution (dashed line) along $y = 0.8$ with the reference solution (solid line).

References for Stabilization Techniques

1. O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu, *The Finite Element Method for Fluid Dynamics*, 6th ed., Elsevier, 2005.
2. R. Codina, “Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation,” *Comput. Methods Appl. Mech. Engrg.*, vol. 156, pp. 185–210, 1998.
3. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Student literature, 1987.
4. G. Hauke, “A Simple Subgrid Scale Stabilized Method for the Advection-Diffusion-Reaction Equation,” *Comput. Methods Appl. Mech. Engrg.*, vol. 191, pp. 2925–2947, 2002.
5. G. Hauke and T.J.R. Hughes, “A comparative study of different sets of variables for solving compressible and incompressible flows,” *Computer Methods in Applied Mechanics and Engineering*, vol. 153, pp. 1–44, 1998.
6. E.G.D. do Carmo and A.C. Galeão, “Feedback Petrov-Galerkin methods for convection-dominated problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 88, pp. 1–16, 1991.

7. E.G.D. do Carmo and G.B. Alvarez, “A new upwind function in stabilized finite element formulations, using linear and quadratic elements for scalar convection-diffusion problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 2383–2402, 2004.

4

Definitions

This chapter describes the available functionality in the Global Definitions and Definitions branches such as parameters, variables, functions, and coordinate systems.

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**.

Global Definitions

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

- **Variables** (): user-defined variables that are used anywhere to simplify the specifications of some properties. See [User-Defined Variables](#).
- **Parameters** (): user-defined global, scalar values that are used to parameterize any part of the model. See [User-Defined Parameters](#).
- **Functions** are function templates for creating user-defined functions as analytic functions or functions based on common function types such as step functions, ramps, and random functions. See [User-Defined Functions](#).
- **Groups** (): Add **Load Group** () and **Constraint Group** () nodes for use in load cases. See [Using Load Cases](#).

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** (): user-defined variables used to simplify the specifications of some properties.
- **View** (): create a user-defined view to visualize the model. See [User-Defined Views](#) in the [Visualization and Selection Tools](#) chapter.
- **Functions** are function templates for user-defined functions as analytic functions or functions based on common function types such as step functions, ramps, and random functions. See [User-Defined Functions](#).
- **Probes** (). To monitor the development of a scalar-valued quantity (real or complex-valued number) from a dynamic simulation. See [Probes](#).
- **Model Couplings**: the user-defined operators for model couplings are useful for creating integrated quantities and identity mappings, adding nonlocal couplings,

and many other applications. See [Model Couplings](#).

- **Selections** (): create a user-defined set of geometric entities for reuse throughout the model. See [Named Selections](#) in the [Visualization and Selection Tools](#) chapter.
- **Pairs:** There are two types of pairs:
 - **Contact Pair** () , to 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** () , to specify two boundary selections (also available for edges and points) that overlap but belong to different parts of an assembly. Then assign a boundary condition to connect the physics in the two parts in an interface.
- Pairs are only available and necessary when the model geometry is an assembly. See [Identity and Contact Pairs](#).
- **Coordinate Systems** () . Create coordinate systems for use in the physics user interfaces. See [Coordinate Systems](#).
- **Perfectly Matched Layers (PML)** () and **Infinite Elements** () . 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 toward 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 to create a PML. See [Infinite Element Domains](#) and [Perfectly Matched Layers](#).

About Parameters, Variables, and Expressions

PARAMETERS AND VARIABLES

Parameters and *variables* are used to parameterize and organize your model. They are different in their definition and use:

- Parameters are user-defined constant scalars with global scope that are available for use throughout the Model Tree. In particular, they can be used for parameterization in the Geometry, Mesh, and Study branches. Important uses include:
 - Parameterizing geometric dimensions
 - Parameterizing mesh element sizes
 - defining parametric sweeps.

A *parameter expression* can contain: Numbers, other parameters, mathematical constants, physical constants, functions of parameter expressions, unary operators,

and binary operators. Parameters can have units. For example, a parameter can be defined as `(exp(-pi*i)+a)*c_const`, where `a` is another parameter, but it is often a scalar numerical value for use in a parametric sweep, where that value is updated during the sweep.

- A *variable*'s expression can contain numbers, parameters, mathematical constants, physical constants, other variables, functions of variable expressions, unary operators, and binary operators. Variables can also depend on dependent variables (the solution) and their derivatives, and they can have units. For example, a variable can be defined as `pi*(R_tube^2-r_tube^2)`, where `R_tube` and `r_tube` are two other variables or parameters, for example defined as `10[mm]` and `25[mm]`, respectively. Variables cannot be used in the Geometry, Mesh, and Study branches (except for solver stop conditions) to simplify model definitions. They can have global scope or scope local to a model depending on where they are defined. A variable with local scope is limited to a geometric entity level within a model: the entire model or selected domains, boundaries, edges (3D only), or points. Variables can make a model easier to understand by using short and descriptive names for complicated expressions. 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 built-in variable for the density in a Solid Mechanics interface, `solid.rho`, represents a density ρ as $\rho(x, y, z, t)$ in 3D. You can visualize it in a surface or volume plot, for example, but you cannot use it in an ODE or a global evaluation, even if it should happen to be defined as a constant value. In such a situation, use an operator that computes an average quantity or a point probe to obtain a quantity with global evaluation scope.

NAMING CONVENTIONS, RESERVED NAMES, AND ERRORS

Parameter names and variable names are case sensitive and must begin with a lowercase or uppercase letter (a–z or A–Z). All other characters in the name must be a lowercase or uppercase letter, a number 0–9, or an underscore _.

It is good practice to use descriptive names that are different from the names of built-in functions and constants. Some fundamental built-in mathematical and numerical constants and built-in variables have reserved names; defining a variable using a reserved name is not recommended because it may cause unexpected results. If you use a variable name that is a reserved name (see [Summary of Built-In Variables With Reserved Names](#)), the name appears in orange and if you move the cursor to the name, a tooltip such as **j is a reserved name** appears. The following names are reserved: `eps`, `nan`, `NaN`, `inf`, `Inf`, `i`, `j`, and `pi`. Also, when used in a model, errors about duplicate

variable names occur if you defined parameter names using names of built-in variables for the geometry, mesh, and physics (`h`, `dom`, and similar names of built-in variables as well as the names of dependent variables and spatial coordinates in the model).

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.

User-Defined Parameters

Under the **Global Definitions** node (), right-click **Global Definitions** and select **Parameters** () if a **Parameters** node is not already created to create user-defined parameters. Parameters are useful in the following context:

- As parameters in dimensions for geometric primitives or other geometry operations
- 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
- As mesh-size parameters
- In physics settings, expressions, and coupling operators
- In expressions when evaluating results

PARAMETERS

Enter values in the **Parameters** table to define parameters used throughout the model. In the **Parameters** section you can enter parameters manually or import them from a text file.

- In the **Parameters** table or the field under the table, enter a parameter **Name**.
- In the **Expression** column or field, enter an expression that defines the parameter value—a scalar number, including a unit if applicable.

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

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

You can save the parameters to a text file to reuse in other models. Click the **Save to File** button () and enter a **File name** in the **Save to File** dialog box, including the extension .txt. Click **Save** to store the parameters in a text file or in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®. The information

is saved in space-separated columns in the same order as displayed on screen. When saving to Excel, an **Excel Save** dialog box appears where you can specify the sheet and range and whether to overwrite existing data, include a header, or use a separate column for units.

You can import or load data in files from a spreadsheet program, for example, with the **Load from File** button () and the **Load from File** dialog box that appears. Data must be separated by spaces or tabs. 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. If the license includes LiveLink™ for Excel® you can also load parameters from a Microsoft Excel Workbook spreadsheet. Then an **Excel Load** dialog box appears where you can specify the sheet and range and whether to overwrite existing data or to declare if the data is stored using a separate column for units.

User-Defined Variables

Right-click **Global Definitions** or **Definitions** to add the **Variables** () node, which enables you to create user-defined *variables* using an expression. Variables can be applied both at the global and local levels. 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. Global variables are primarily useful for expressions involving parameters that do not depend on the geometry, such as time, or dependent variables in an ODE or algebraic equation. Whenever possible, define variables locally to avoid making use of the global namespace.

The variables that you define, including their descriptions, become available as predefined variables (with their descriptions, if defined) when you have solved the model or updated an existing solution by right-clicking the **Study** node () and selecting **Update Solution** ().

GEOMETRIC ENTITY SELECTION (LOCAL DEFINITIONS ONLY)

Select the scope from the **Geometric entity level** list—**Entire model**, **Domain**, **Boundary**, **Edge** (3D only), and **Point**. For all levels except Entire model, you can define a selection manually, using any user-defined selections, or by selecting **All boundaries**, for example, from the **Selection** list.

VARIABLES

In the **Variables** table or the fields under the table, enter variables by defining a **Name**, an **Expression** (see [About Parameters, Variables, and Expressions](#)), and (optionally) **Descriptions** manually or importing them from a text file. You can save the variables to

a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®) to reuse in other models with the **Save to File** button (). The information is saved in space-separated columns in the same order as displayed on screen. When saving to Excel, an **Excel Save** dialog box appears where you can specify the sheet and range and whether to overwrite existing data, include a header, or use a separate column for units. You can import or load data in text files from a spreadsheet program, for example, with the **Load from File** button (). Data must be separated by spaces or tabs. If the license includes LiveLink™ for Excel® you can also load variables from a Microsoft Excel Workbook spreadsheet. Then an **Excel Load** dialog box appears where you can specify the sheet and range and whether to overwrite existing data or to declare if the data is stored using a separate column for units.



For an example of global variables, see [Effective Diffusivity in Porous Materials](#): Model Library path
COMSOL_Multiphysics/Diffusion/effective_diffusivity



For examples of local variables see:

- [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

Common Settings for the Definitions Nodes

When a node is added under **Global Definitions** and **Definitions**, many sections on the settings windows are similar or the same, used for other nodes, or generally found throughout COMSOL. [Table 4-1](#) provides cross references to the information relevant to these nodes.

COMMON BUTTONS ON THE SETTINGS WINDOWS

The following buttons are available on many of the settings windows and are self explanatory. These are not explicitly described or explained for every node.

- In general, use the **Move Up** (↑), **Move Down** (↓), and **Delete** (✖) buttons and the fields under tables to edit the table contents. Or right-click a table cell and select **Move Up**, **Move Down**, or **Delete**.
- When you see the **Plot** button (📈) click it to generate a preview plot (of a function, for example) and set ranges for all the arguments. Click the **Create Plot** button (➕📈) to create a plot group and a plot.
- Click the **Add to Selection** (➕), **Remove from Selection** (➖), and **Clear Selection** (✖) buttons, such as with model couplings, when required.
- For Selections, click the **Add** button (➕) to open an **Add** dialog box that contains all existing selections for the same geometric entity level. Add the selections for which to form a selection of adjacent entities.
- Click the **Save to File** button (💾) and enter a **File name** in the **Save to File** dialog box, 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.
- Import or load data in text files from a spreadsheet program, for example, with the **Load from File** button (CSV) and the **Load from File** dialog box that appears. Data must be separated by spaces or tabs (or be in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®).



- [About Selecting Geometric Entities](#)
- [Key to Nodes and Toolbar Buttons](#)
- [Named Selections](#)

LINKS TO COMMON SETTINGS WINDOW DESCRIPTIONS

TABLE 4-1: DETAILS FOR THE COMMON SETTINGS SECTIONS

SECTION NAME ON THE SETTINGS WINDOW	LINK TO MORE INFORMATION
Coordinate System Identifier	Coordinate System Identifier
Derivatives	Derivatives
Function Name	Function Name
Plot Parameters	Plot Parameters
Smoothing	Smoothing
Table and Window Settings	Table and Window Settings (for Probes)

FUNCTION NAME AND COORDINATE SYSTEM IDENTIFIER

Function Name

Enter a name for the function in the **Function name** field.

Coordinate System Identifier

Enter or edit an **Identifier** or use the default identifier name.



- [The Node Settings Windows](#)
- [Editing Node Properties, Names, and Identifiers](#)

DERIVATIVES

Enter data for derivatives of functions with respect to their input arguments in this section based on the following:

- In the **Derivatives** table, each row contains a partial derivative of a function with respect to one of its arguments.
- The entries in the **Function name** 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.
- The **Partial derivative** column contains expressions for the partial derivatives. Partial derivatives that are not specified in the table default to 0.

PLOT PARAMETERS

Use these parameters to generate a preview plot of the function and set ranges for all arguments. For each row in the **Plot Parameters** table, enter the **Argument** name, a

Lower limit, and an **Upper limit**. The default lower and upper limits are 0 and 1, respectively. If you want to use a constant value for one parameter (when the function has more than one argument), it is sufficient to enter a value for the **Lower limit**.

SMOOTHING

For the [Ramp](#), [Step](#), [Triangle](#), and [Rectangle](#) functions, 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. The smoothed functions can have continuous first and second derivatives. The default setting uses two continuous derivatives.

For the [Ramp](#) function and to turn on the smoothing function at the start or the cutoff, select the **Smooth at start** or **Smooth at cutoff** check boxes.



The [Waveform](#) function also supports smoothing for some of the waveform types.

TABLE AND WINDOW SETTINGS (FOR PROBES)

By default, COMSOL uses a probe table (typically **Probe Table 1**) under **Tables** and a probe table plot (typically **Probe Table Plot 1**) in a **Probe ID Plot Group** node, which appears in a separate plot window for probe plots (typically **Probe Plot 1**). To organize and group multiple probes, control the table and plot window to use for the probe results:

From the **Output table** list, select **Default**, **New table**, or any existing probe table. If an existing probe table is selected, click the **Go to Source** button () to move to the selected **Probe Table** node under **Tables**.

From the **Plot window** list, select **Default**, **New window**, or any existing plot window. Click the **Add Plot Window** button () to create a new plot window and make it the default for this list.

By default for both the **Output table** and **Plot window**, COMSOL uses a probe table or probe table plot that is created automatically. If **Default** is selected, COMSOL updates the list to show the name of the default probe table or probe plot window after the solution process.

Operators, Functions, and Constants

Many built-in 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 4-2 and Table 4-3), special operators (Table 4-9), mathematical functions and constants (Table 4-7), and physical constants predefined as variables (Table 4-8) that are available in COMSOL Multiphysics®. See also [Model Couplings](#) for information about coupling operators.

Unary, Binary, and List Operators and Their Precedence Rules

TABLE 4-2: UNARY OPERATORS

OPERATOR	DESCRIPTION
+	Unary plus
-	Unary minus
!	Logical not

The binary operators include arithmetic and logical operations.

TABLE 4-3: 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

The following operators are used for precedence, grouping, lists, and unit definitions:

TABLE 4-4: GROUPING, LIST, AND UNIT OPERATORS

OPERATOR	DESCRIPTION
<code>()</code>	Parentheses for controlling precedence in expressions
<code>{}</code>	Vector and tensor expressions
<code>,</code>	Element separator in lists
<code>.</code>	Scoping operator
<code>[]</code>	Unit

The following list shows the precedence order for the operators above:

TABLE 4-5: PRECEDENCE LEVELS

PRECEDENCE LEVEL	SYMBOL	DESCRIPTION
1	<code>() {} .</code>	Grouping, lists, scope
2	<code>^</code>	Power
3	<code>! - +</code>	Unary: logical not, minus, plus
4	<code>[]</code>	Unit
5	<code>* /</code>	Multiplication, division
6	<code>+ -</code>	Addition, subtraction
7	<code>< <= > >=</code>	Comparisons: less than, less than or equal, more than, more than or equal
8	<code>== !=</code>	Comparisons: equal, not equal
9	<code>&&</code>	Logical and
10	<code> </code>	Logical or
11	<code>,</code>	Element separator in lists

Mathematical and Numerical Constants

The following table includes the built-in mathematical and numerical constants. The names of these constants are reserved names that you cannot use when creating user-defined variables and parameters.

TABLE 4-6: MATHEMATICAL AND NUMERICAL CONSTANTS

NAME	DESCRIPTION
<code>eps</code>	Floating point relative accuracy (machine epsilon, 2^{-52} or about $2.2204 \cdot 10^{-16}$, for double floating point numbers).
<code>i, j</code>	Imaginary unit, $\sqrt{-1}$.

TABLE 4-6: MATHEMATICAL AND NUMERICAL CONSTANTS

NAME	DESCRIPTION
inf, Inf	Infinity, ∞ . A value larger than what can be handled with floating-point representation.
NaN, nan	Not-a-number. An undefined or unrepresentable value such as the result of $0/0$ or inf/inf .
pi	Pi (about 3.141592653589793).

Mathematical Functions

The following list includes the built-in mathematical functions that you can use to when defining variables or directly in expressions in the physics settings, for example. The function names are reserved names that cannot be used for user-defined functions, but they can be used for variable and parameter names.

TABLE 4-7: MATHEMATICAL FUNCTIONS

NAME	DESCRIPTION	SYNTAX EXAMPLE
abs	Absolute value	abs(x)
acos	Inverse cosine (in radians)	acos(x)
acosh	Inverse hyperbolic cosine	acosh(x)
acot	Inverse cotangent (in radians)	acot(x)
acoth	Inverse hyperbolic cotangent	acoth(x)
acsc	Inverse cosecant (in radians)	acsc(x)
acsch	Inverse hyperbolic cosecant	acsch(x)
arg	Phase angle (in radians)	arg(x)
asec	Inverse secant (in radians)	asec(x)
asech	Inverse hyperbolic secant	asech(x)
asin	Inverse sine (in radians)	asin(x)
asinh	Inverse hyperbolic sine	asinh(x)
atan	Inverse tangent (in radians)	atan(x)
atan2	Four-quadrant inverse tangent (in radians)	atan2(y,x)
atanh	Inverse hyperbolic tangent	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)

TABLE 4-7: MATHEMATICAL FUNCTIONS

NAME	DESCRIPTION	SYNTAX EXAMPLE
ceil	Nearest following integer	ceil(x)
conj	Complex conjugate	conj(x)
cos	Cosine	cos(x)
cosh	Hyperbolic cosine	cosh(x)
cot	Cotangent	cot(x)
coth	Hyperbolic cotangent	coth(x)
csc	Cosecant	csc(x)
csch	Hyperbolic cosecant	csch(x)
erf	Error function	erf(x)
exp	Exponential function e^x . That is, exp(1) is the mathematical constant e (Euler's number).	exp(x)
floor	Nearest previous integer	floor(x)
gamma	Gamma function	gamma(x)
imag	Imaginary part	imag(u)
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)
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)
tan	Tangent	tan(x)
tanh	Hyperbolic tangent	tanh(x)



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

Physical Constants

Physical constants are fundamental, universal constants that represent physical quantities. COMSOL Multiphysics includes the most widely used physical constants as built-in constants. [Table 4-8](#) lists all supported physical constants with their names, symbol (variable name), value, and SI unit. The values are taken from Ref. 1 and include the SI unit.

TABLE 4-8: PHYSICAL CONSTANTS

NAME	SYMBOL	VALUE
Acceleration of gravity	<code>g_const</code>	<code>9.80665[m/s^2]</code>
Avogadro constant	<code>N_A_const</code>	<code>6.02214129e23[1/mol]</code>
Boltzmann constant	<code>k_B_const</code>	<code>1.3806488e-23[J/K]</code>
Characteristic impedance of vacuum (impedance of free space)	<code>Z0_const</code>	<code>376.730313461...[ohm]</code> <code>(mu0*c)</code>
Electron mass	<code>me_const</code>	<code>9.10938291e-31[kg]</code>
Elementary charge	<code>e_const</code>	<code>1.602176565e-19[C]</code>
Faraday constant	<code>F_const</code>	<code>96485.3365[C/mol]</code>
Fine-structure constant	<code>alpha_const</code>	<code>7.2973525698e-3</code>
Gravitational constant	<code>G_const</code>	<code>6.67384e-11[m^3/(kg*s^2)]</code>
Molar volume of ideal gas (at 273.15 K and 1 atm)	<code>V_m_const</code>	<code>22.413968e-3[m^3/mol]</code>
Neutron mass	<code>mn_const</code>	<code>1.674927351e-27[kg]</code>
Permeability of vacuum (magnetic constant)	<code>mu0_const</code>	<code>4*pi*1e-7[H/m]</code>
Permittivity of vacuum (electric constant)	<code>epsilon0_const</code>	<code>8.854187817e-12[F/m]</code>
Planck's constant	<code>h_const</code>	<code>6.62606957e-34[J*s]</code>
Planck's constant over 2 pi	<code>hbar_const</code>	<code>1.054571726e-34[J*s]</code>
Proton mass	<code>mp_const</code>	<code>1.672621777e-27[kg]</code>

TABLE 4-8: PHYSICAL CONSTANTS

NAME	SYMBOL	VALUE
Speed of light in vacuum	c_const	299792458[m/s]
Stefan-Boltzmann constant	sigma_const	5.670373e-8[W/(m^2*K^4)]
Universal gas constant	R_const	8.3144621[J/(mol*K)]
Wien displacement law constant	b_const	2.8977721e-3[m^K]

REFERENCE

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

Built-In Operators

There are special built-in operators available for modeling and for evaluating results; these operators are similar to functions but behave differently. Many physics user interfaces use these operators to implement equations and special functionality. See [Table 4-9](#) and the detailed descriptions that follow.

TABLE 4-9: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
<code>adj(expr)</code>	Evaluate expression using the adjoint sensitivity.	adj
<code>at</code>	Access the solution at any time.	at
<code>ballint(r,expr), ballavg(r,expr), circint(r,expr), circavg(r,expr), diskint(r,expr), diskavg(r,expr), sphint(r,expr), sphavg(r,expr)</code>	Evaluates the integral or average of the expression on the specified shape with radius <code>r</code> .	ball , circle , disk , and sphere
<code>bdf(expr,i)</code>	Apply backward differentiation formula of order <code>i</code> on expression.	bdf
<code>bndenv(expr)</code>	Evaluates the expression <code>expr</code> at the coordinates of a particle at a boundary.	env and bndenv
<code>centroid(expr)</code>	For simplex meshes it evaluates the expression <code>expr</code> in the centroid of the mesh element to which the point belongs.	centroid

TABLE 4-9: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
<code>circumcenter(expr)</code>	Evaluates the expression <code>expr</code> in the circumcenter of the mesh element to which the point belongs.	<code>circumcenter</code>
<code>d(f,x)</code>	Differentiation operator. Differentiation of <code>f</code> with respect to <code>x</code> .	Differentiation Operators: <code>d</code> , <code>pd</code> , and <code>dtang</code>
<code>depends(expr)</code> <code>depends(expr,var)</code>	True if expression <code>expr</code> depends on the solution or <code>var</code> , respectively.	<code>depends</code>
<code>dest(expr)</code>	Evaluate parts of an integration coupling expression on destination side.	<code>dest</code>
<code>down(expr)</code>	Evaluate expression as defined in adjacent downside.	<code>up</code> and <code>down</code>
<code>dtang(f,x)</code>	Tangential differentiation of an expression <code>f</code> defined on a boundary with respect to a spatial dimension <code>x</code> .	Differentiation Operators: <code>d</code> , <code>pd</code> , and <code>dtang</code>
<code>env(expr)</code>	Evaluates the expression <code>expr</code> at the coordinates of a particle in a domain.	<code>env</code> and <code>bndenv</code>
<code>error('string')</code>	Generates an error with error message <code>string</code> .	<code>error</code>
<code>fsens(expr)</code>	Evaluate expression using the functional sensitivity.	<code>fsens</code>
<code>if(cond,expr1,expr2)</code>	Conditional expression evaluating the second or third argument depending on the value of the condition.	<code>if</code>
<code>integrate(expr,var,lower,upper)</code>	Evaluate integral of general expression with respect to an integration variable over a real interval specified by <code>lower</code> and <code>upper</code> limits.	<code>integrate</code>
<code>isdefined(variable)</code>	Returns one where the variable is defined and zero where it is not defined.	
<code>isinf(expr)</code>	True if expression evaluates to infinity.	<code>isinf</code> and <code>isnan</code>
<code>islinear(expr)</code>	True if expression is a linear function of the solution.	<code>islinear</code>
<code>isnan(expr)</code>	True if expression evaluates to NaN (not-a-number).	<code>isinf</code> and <code>isnan</code>

TABLE 4-9: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
<code>jacdepends(expr)</code> <code>jacdepends(expr, var)</code>	True if the derivative of the expression <code>expr</code> with respect to the solution depends on the solution or <code>var</code> , respectively.	jacdepends
<code>lindev</code>	Evaluate an expression linearized at the linearization point (when a linearization point is stored in the solution).	lindev
<code>linper</code>	Marks a load term to be used in a Linear perturbation solver.	linper
<code>linpoint</code>	Access the linearization point (when a linearization point is stored in the solution).	linpoint
<code>linsol</code>	Access the standard solution (for example inside <code>linpoint</code> or <code>lintotal</code>).	linsol
<code>lintotal</code>	Access the sum of the linearization point and linear perturbation.	lintotal
<code>lintotalavg</code>	Evaluate average of <code>lintotal(expr)</code> over all phases.	lintotalavg
<code>lintotalpeak</code>	Evaluate maximum of <code>lintotal(expr)</code> over all phases.	lintotalpeak
<code>lintotalrms</code>	Evaluate RMS of <code>lintotal(expr)</code> over all phases.	lintotalrms
<code>linzero</code>	Evaluate expression with zero solution.	linzero
<code>mean(expr)</code>	Mean value of expression as evaluated on adjacent boundaries.	mean
<code>nojac(expr)</code>	No contribution to the Jacobian.	nojac
<code>pd(f, x)</code>	Differentiation operator. Differentiation of <code>f</code> with respect to <code>x</code> . No chain rule for dependent variables.	Differentiation Operators: d, pd, and dtang
<code>ppr</code>	Accurate derivative recovery.	ppr and pprint
<code>pprint</code>	Accurate derivative recovery within each domain group.	ppr and pprint
<code>prev(expr, i)</code>	Evaluate expression at the <code>i</code> th previous time step.	prev
<code>reacf</code>	Accurate evaluation of reaction forces and fluxes.	reacf

TABLE 4-9: BUILT-IN OPERATORS

OPERATOR	DESCRIPTION	CROSS REFERENCE
<code>realdot(a,b)</code>	Treat complex numbers a and b as real 2-vectors and return their dot product.	realdot
<code>scope.ati(<i>coordinate exprs</i>,<i>expr</i>)</code>	Evaluates the expression <i>expr</i> at an i -dimensional entity in the point with coordinates given by the coordinate expressions <i>coordinate exprs</i> .	spatial at
<code>sens(<i>expr</i>,<i>i</i>)</code>	Evaluate expression using the forward sensitivity for the parameter given by the second argument.	sens
<code>shapeorder(<i>variable</i>)</code>	The element order used for discretization of a variable.	shapeorder
<code>side(<i>entity</i>,<i>expr</i>)</code>	Evaluate expression as defined in the adjacent entity.	side
<code>subst(<i>expr</i>, <i>expr1_orig</i>, <i>expr1_subst</i>,...)</code>	Substitute variables in an expression with other variables or expressions.	subst
<code>test(<i>expr</i>)</code>	Test function operator.	test
<code>timeint</code> , <code>timeavg</code>	Integrate or compute the average of a time-dependent expression over a time interval.	timeint and timeavg
<code>try_catch(<i>tryExpr</i>, <i>catchExpr</i>)</code>	Attempts to evaluate <i>tryExpr</i> , but if this fails for any point, <i>catchExpr</i> is evaluated instead.	try_catch
<code>up(<i>expr</i>)</code>	Evaluate expression as defined in adjacent upside.	up and down
<code>var(<i>expr</i>,<i>fieldname1</i>, <i>fieldname2</i>, ...)</code>	Variation operator.	var
<code>with</code>	Access any solution.	with

ADJ

- When you apply the adjoint sensitivity operator `adj` 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.

AT

- The `at` operator can access a solution to a time-dependent problem at any time. COMSOL 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.

BALL, CIRCLE, DISK, AND SPHERE

- The `ballint(r , $expr$)` operator integrates the expression $expr$ on a ball with radius r around the point in which it is evaluated. The `ballint` operator can be evaluated on all entities in 3D.
- The `ballavg(r , $expr$)` operator is defined as `ballint(r , $expr$)/ballint(r , 1)`.
- 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 , $expr$)` is similar to the `ballint` operator but calculates the integral on a sphere instead.
- The `sphavg(r , $expr$)` operator is defined as `sphint(r , $expr$)/sphint(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 , $expr$)`.

BDF

- 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`.

CENTROID

- The `centroid(expr)` operator evaluates the expression `expr` in the centroid of the mesh element to which the point belongs for simplex meshes.
- Note that the operator is context sensitive in the sense that it chooses the mesh element of the same dimension (a point often belongs to several different mesh elements) as the context in which the evaluation is performed.

CIRCUMCENTER

- The `circumcenter(expr)` operator evaluates the expression `expr` in the circumcenter of the mesh element to which the point belongs. This point is in general only well defined for simplices, but for other mesh elements a natural generalization is available.
- Note that the operator is context sensitive in the sense that it chooses the mesh element of the same dimension (a point often belongs to several different mesh elements) as the context in which the evaluation is performed.
- You can add a suffix (“`_frameId`”) that specifies the frame in which the evaluation is done—for example, `circumcenter_spatial(expr)`.

DIFFERENTIATION OPERATORS: D, PD, AND DTANG

- All differentiation operators (`d`, `pd`, and `dtang`) can be used both in model settings and in results evaluation.
- 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 . Some space derivatives are also available using predefined variables. For example, `uxx`, `d(ux, x)`, and `d(d(u, x), x)` are equivalent for a dependent variable u when evaluated in a domain. On a boundary, however, `d(u, x)` is 0, while `ux` is the average of the values from the adjacent domains. The expression `d(E, TIME)` computes the reference time derivative of the expression E .

- The `pd` operator works in a similar way to the `d` operator. The main difference is that `pd(u, x)` is 0 rather than `ux` (no chain rule is applied for dependent variables).
- Use the `dtang` operator to compute derivatives in the tangential direction along a boundary. The `dtang` operator can be applied to 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.

DEPENDS

- The `depends(expr)` operator returns 1 if the expression `expr` that it operates on depends on the solution; otherwise it returns 0.
- `depends(expr, var)` returns 1 if `expr` depends on `var`; otherwise it returns 0.

Use this operator to check user-defined expressions for dependency on the solution.

DEST

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 :

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

ENV AND BNDENV



These operators can only be evaluated on particles in a particle tracing simulation. Thus, the Particle Tracing Module is required.

- Evaluating `env(expr)` on a particle evaluates `expr` at the point in the domain where the particle is. When evaluating a variable `var` on a particle, if the variable is not defined on the particle it is automatically replaced by `env(var)`. Therefore the `env` operator can often be omitted.
- Evaluating `bndenv(expr)` on a particle, evaluates `expr` at the point on the boundary where the particle is. If the particle is not on a boundary, the evaluation fails. Use this operator instead of `env` when evaluating expressions that are only defined on boundaries.

ERROR

- The `error(string)` operator generates an error with error message `string`.
- You may for instance use this operator to make assertions on how your solution should behave. If you write `if(cond,expr,error('cond is false'))` you will calculate `expr` when `cond` is true and get an error message including the text `cond is false`, when `cond` is false.

FSENS

- When you apply the functional sensitivity operator `fsens` to an expression, COMSOL uses the functional sensitivity solution for the evaluation. See also [subst](#).
- 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.

IF

- The `if(cond,expr1,expr2)` operator implements a conditional expression.

- The first argument is a condition that COMSOL 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 $x = 0$.

INTEGRATE

- `integrate(expr,var,lower,upper)` computes the integral of `expr` for the integration variable `var` over an interval specified by expressions `lower` for the lower limit and `upper` for the upper limit. The expressions for lower and upper limits do not have to be constants but are required to evaluate to real values.
- `integrate(expr,var,lower,upper,tol)` sets the relative tolerance in the numerical integration to `tol`. The default value of the relative tolerance (used when the fifth argument is omitted) is `1e-3`. The tolerance must be a real constant between 0 and 1.

ISINF AND ISNAN

- 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.

ISLINEAR

- 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.

JACDEPENDS

- The `jacdepends(expr)` operator returns 1 if the derivative of the expression `expr`, with respect to any part of the solution, depends on the solution; otherwise it returns 0.
- `jacdepends(expr,var)` returns 1 if `expr` if the derivative of the expression `expr`, with respect to any part of the solution, depends on `var`; otherwise it returns 0.

LINDEV

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.

LINPER

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.

LINPOINT

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.

LINSOL

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.

LINTOTAL

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.

LINTOTALAVG

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.

LINTOTALPEAK

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.

LINTOTALRMS

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

LINZERO

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

MEAN

- COMSOL can evaluate expressions on entities of different dimensions and this may or may not affect the result. For instance, a point may lie on an edge, which may be an edge of a square, which may be part of the boundary of a cube. If you now want to access the numbering of the entity it is obvious that you should get different results for the point, edge, square and cube. If you on the other hand want to know the value of the dependent variable, this should be the same if you think of the point as part of a point, edge, square, or cube.
- The `mean` operator may be called on any entity which has lower dimension than the model. The dimension of the entity from where the call is made is called n .
- The `mean` operator determines the smallest integer $m > n$ for which there are adjacent entities of dimension m . It then evaluates the expression at the point one time for each m dimensional adjacent entity, regarding the point as a point in the entity, and takes the average of the calculated values.

See also [up](#) and [down](#) and [side](#) for similar operators.

NOJAC

- 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, such as when using a nonlocal coupling. The use of the `nojac` operator can then significantly

lower the memory requirements by avoiding fill-in of the Jacobian matrix, but its use might also slow down the convergence of the solution.

- The k - ε turbulence model is an example where a built-in use of the `nojac` operator improves performance.

PPR AND PPRINT

- 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. For example, if $e = ux + vy$, then
$$\text{ppr}(e^2) = (\text{ppr}(ux) + \text{ppr}(vy))^2.$$



Accurate Derivative Recovery

- 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.



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.

PREV

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 $(u - \text{prev}(u,1)) / \text{timestep}$. 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.

REACF

The reaction force operator (`reacf`) evaluates the reaction force at each node point where a constraint is applied. The reaction force at a node is equal to the corresponding component of the negated residual vector $-L$ computed while solving the model. The reaction forces are stored together with the solution vector by the solvers.

- The reaction force operator (`reacf`) is useful 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. The integration for the reaction force is a summation over the nodes, so the integration method must be summation rather than integration. The automatic integration method in the integrations available under **Results>Derived Values** detects the use of the `reacf` operation and then uses the summation method.
- Storing of the reaction forces can be disabled by clearing the **Reaction forces** check box in the **Output** section in the solver's settings window. This saves some computational time and memory. It is then not possible to use the reaction force operator.
- When using weak constraints, the residual vector is always 0, so reaction forces are not available.

REALDOT

- 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 $\text{conj}(b)$ and $\text{conj}(a)$, respectively, while for the latter expression, the partial derivatives are $\text{real}(\text{conj}(a))$ and $\text{real}(a)$.



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 $|a|^2 = \text{realdot}(a, a)$.

SENS

- 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.

SHAPEORDER

- 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, 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.

SIDE

- COMSOL can evaluate expressions on entities of different dimensions and this may or may not affect the result. For instance, a point may lie on an edge, which may be an edge of a square, which may be part of the boundary of a cube. If you now want to access the numbering of the entity it is obvious that you should get different results for the point, edge, square, and cube. If you, on the other hand, want to know the value of the dependent variable, this should be the same if you think of the point as part of a point, edge, square, or cube.
- The `side` operator may be called on an entity that has lower dimension than the model. The dimension of the entity from where the call is made is called `n`.
- The `side` operator is an operator that evaluates an expression, not on the entity where it is called but instead on one of the adjacent entities of dimension `n+1`. You choose which entity by giving its number (this is the number displayed, for instance, in the selection fields) as the first argument to the operator.
- It may happen that the entity you choose is adjacent to the evaluation point more than once. For instance, a boundary can have the same domain on both sides. In such cases the `side` operator takes the average of the different values.

See also [up and down](#) and [mean](#) for similar operators.

SPATIAL AT

- 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)`.

SUBST

- 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.

TEST

- 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, $\text{test}(F(u, \nabla u))$, the test operator is equivalent to:

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

for all dependent variables u_i .

TIMEINT AND TIMEAVG

- The `timeint` and `timeavg` operators integrate and compute the average of a time-dependent expression over a time interval, respectively. `timeint(t1,t2,expr)` and `timeavg(t1,t2,expr)` compute the integral and average of `expr` over the interval `t=t1` to `t=t2`, respectively. The first two arguments must be real scalars. The integral is computed by numerical integration, subdividing the interval until the required accuracy is reached. The `timeavg` operator numerically integrates the expression in the same way as `timeint` and then divides the result by `t2-t1`.
- `timeint(t1,t2,expr,tol)` and `timeavg(t1,t2,expr,tol)` set 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)` and `timeavg(t1,t2,expr,tol,minlen)` set 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 last argument is omitted) is `1e-6`.

- The `timeint` and `timeavg` operators can only be used during results evaluation, so they should not be used when setting up the model.

TRY_CATCH

- The `tryCatch(tryExpr, catchExpr)` operator attempts to evaluate the expression `tryExpr`, but if this fails for any point, the operator evaluates `catchExpr` instead.
- Note that the result might depend on how the mesh elements are partitioned into blocks during evaluation, which can be rather arbitrary. As soon as evaluation of the first argument fails in some part of the block, the second argument gets evaluated in the entire block. Also, during postprocessing the behavior might change because NaN (Not-a-Number) values in a subset of the evaluation points are then accepted, so then the second argument's expression might not be evaluated even at points where the first argument fails.
- Consider using the `if` operator (see [if](#)) with a suitable condition as the first argument instead of the `tryCatch` operator, if the `if` operator is applicable.

UP AND DOWN

- COMSOL can evaluate expressions on both sides of a boundary. One way to do this is by using the `up` and `down` operators. These operators are available only on boundaries (that is, geometric entities of dimension one less than the dimension of the model).
- For an expression or a variable that is discontinuous across a boundary, the value is different on either side, and COMSOL normally displays the mean values on the boundary.
- Use the `up` and `down` operators to evaluate an expression on the upside or downside of 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](#). See also `side` and `mean` for similar operators.

VAR

- 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, $\text{var}(F(u, \nabla u, v, \nabla v), a)$, where the dependent variable u is in the field named a and the dependent variable v is not, the `var` operator is equivalent to:

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

for all dependent variables u_i .

WITH

- 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, you can 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.

Predefined and Built-In Variables

This section provides information about available predefined and built-in variables that represent properties of the physics, geometry, mesh, and other parts of the model, including some tips on how you can use them in models.

Predefined Physics Variables

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

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



The **Equation View** subnode is available for all physics nodes and contains a table with the names, expressions, units, and descriptions for the physics 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. 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 user interface, use its interface identifier, such as `solid1`.
- 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, to access the density ρ in Material 1 use `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 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 user 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`.

Variable Classification and Geometric Scope

COMSOL provides a set of variables that 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 user 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 user 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:

- User-defined *variables*
- [Built-In Global Variables](#)
- [Physical Constants](#) (predefined universal constants)
- [Geometric Variables and Mesh Variables](#) (variables that characterize geometric properties).
- [Field variables](#) (dependent variables and variables derived from them). For example, [Shape Function Variables](#) and [Predefined Physics Variables](#).
- [Model Couplings](#)
- [Solver Variables](#) (available only during the solution process)

Built-In Global Variables

The following variables represent time (t), frequency ($freq$), eigenvalue ($lambda$), and the number of degrees of freedom ($numberofdofs$).

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 Analysis and Plots](#) 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 nodes 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 Analysis and Plots](#) section.
- For many physics user interfaces, the default is to use an eigenfrequency study and compute and display the eigenfrequencies rather than the eigenvalues.

THE NUMBER OF DEGREES OF FREEDOM VARIABLE

The variable `numberofdofs` returns the total number of degrees of freedom (DOFs), which is the number of DOFs solved for plus any internal DOFs that the solver might add. The number of DOFs solved for plus the number of internal DOFs are reported in the **Messages** window when you compute the solution.

Geometric Variables and Mesh Variables

The variables that characterize geometric properties and the mesh are listed in [Table 4-10](#), with detailed descriptions for some of the variables following the table.

TABLE 4-10: GEOMETRIC VARIABLES AND MESH VARIABLES

VARIABLE	DESCRIPTION
<code>curv</code>	The curvature of a boundary in 2D is called <code>curv</code> .
<code>curv1,curv2</code>	A boundary in 3D has two principal curvatures corresponding to the minimal and maximal normal curvatures. They are called <code>curv1</code> and <code>curv2</code> , respectively. See Curvature Variables for details.
<code>dom</code>	The domain number, the boundary number, the edge number, or the vertex (point) number (all are integer values).

TABLE 4-10: GEOMETRIC VARIABLES AND MESH VARIABLES

VARIABLE	DESCRIPTION
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.
meshelment	The mesh element number.
meshvol	Volume of the (linearized) mesh element.
nx, ny, nz	See Normal Variables .
qual	A mesh quality measure.
reldetjac reldetjacmin	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 <i>inverted mesh element</i> .
s, s1, s2	See Parameterization Variables .

TABLE 4-10: GEOMETRIC VARIABLES AND MESH VARIABLES

VARIABLE	DESCRIPTION
tcurvx, tcurvy (2D)	Tangential directions for the corresponding curvatures. See Curvature Variables for more information.
tcurv1x, tcurv1y, tcurv1z, tcurv2x, tcurv2y, tcurv2z (3D)	
tx and ty (2D) t1x, t1y, t1z (3D edges) t2x, t2y, t2z (3D surfaces)	See Tangent Variables .
x, y, z r, z	See Spatial Coordinate Variables .

! 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 *i*th 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 *curv*, *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, *curv_spatial*, *dvol_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) 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 $x\text{TIME}, y\text{TIME}, z\text{TIME}$ 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:



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 edges when in the boundary or edge selection mode and if you have selected the **Show edge direction arrows** check box in the **View** node's () settings window. You can use s on boundaries in 2D when specifying boundary conditions.

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 (faces). 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

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

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 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 Variables



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



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



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:

In 1D:



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

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.
-

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—`nxmesh`, `unxmesh`, and `dnxmesh`, 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.

CURVATURE VARIABLES

The curvature variables are defined on boundaries in 2D and 3D.

In 2D the curvature is denoted `curv`. Positive curvature is toward the normal (`nx,ny`).

In 3D there are two principal curvatures named `curv1` and `curv2`, where `curv1` is less than `curv2` seen as real numbers. These correspond to the minimal and maximal values for the curvature of a curve you get by intersecting the boundary with a plane in which the normal lies. Positive curvature is toward the normal (`nx,ny,nz`).

The components of the normalized tangential directions for the corresponding curvatures are called `tcurvx`, `tcurvy` in 2D and `tcurv1x`, `tcurv1y`, `tcurv1z`, `tcurv2x`, `tcurv2y`, and `tcurv2z` in 3D. The tangents (`tcurv1x,tcurv1y,tcurv1z`) and (`tcurv2x,tcurv2y,tcurv2z`) are orthogonal.

Note also that you can choose on which frame you evaluate all these variables by using the general instructions found below [Table 4-10](#).

Shape Function Variables

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 you have selected a physics user interface is selected, you can enter names 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.

 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 i th 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 4-11](#) 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 4-11: LAGRANGE ELEMENT VARIABLE NAMES

ENTITY TYPE/ SPACE DIMENSION	ID	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, uyx, uyyt, utt, uxtt, uytt, uxxtt, uxyt, uyxtt, uyyt	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 4-11](#) 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 first-order elements all these variables evaluate to zero.
- 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 ut is the time derivative of the dependent variable u . You can also form mixed space-time derivatives as $ux_i t$, for example, uxt ,

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

!

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 adds variables for the Lagrange multipliers (one for each dependent variable) by adding $_1m$ as a suffix to the dependent variable name. For example, for a dependent variable u , the corresponding Lagrange variable is u_1m . 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 [Computing Accurate Fluxes](#)).



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 [up and down](#)).

Solver Variables

The following table lists global solver 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 4-12: SOLVER VARIABLES

VARIABLE	DESCRIPTION
niterCMP	This variable contains 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 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 variable contains 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 to control artificial stabilization.
timestep	This variable contains the current time step used by the time-dependent solver. It is used by some physics 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, and 3.0. The step size is 1 if you provide only start and end values and skip the step value.



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 $a > b$.
- $10^{\text{range}(-3,3)}$ gives the exponentially increasing sequence $10^{-3}, 10^{-2}, \dots, 10^3$.
- $1^{\text{range}(1,10)}$ gives a sequence of length 10 where all elements equal 1. Multiplying the vector $1^{\text{range}(1,n)}$ by a constant value *a* gives a vector of *n* elements all equal to *a*.
- $0^{\text{range}(1,5)}$ 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.

For ranges that contain integer values only, an **Integer Range** dialog box opens instead of the normal **Range** dialog box. The **Integer Range** dialog box only contains **Start**, **Step**, and **Stop** fields, all of which must contain integer values.

SUPPORT FOR RANGES AND VECTOR-VALUED EXPRESSIONS

The following modeling settings 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** transforms 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 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.

Summary of Built-In Variables With Reserved Names

This section is an overview of the built-in elements of the following categories as defined by the underlying COMSOL language:

- Constants
- Variables
- Functions

These language elements are built-in or user-defined. In addition there are *operators* that cannot be user-defined, and *expressions*, which are always user-defined.

ABOUT RESERVED NAMES

Built-in elements have reserved names, names that cannot be redefined by the user. If you try to use a reserved name for a user-defined variable, parameter, or function, the text where you enter the name turns orange and you get a tooltip error message if you select the text string. *Reserved function names are reserved only for function names, which means that such names can be used for variable and parameter names, and vice versa.* The following tables list the most commonly used built-in elements and hence those reserved names.

CONSTANTS AND PARAMETERS

There are three different types of constants: built-in *mathematical and numerical constants*, built-in *physical constants*, and *parameters*. Parameters are user-defined constants that can vary over parameter sweeps. Constants are scalar valued. The tables below list the built-in mathematical and numerical constants as well as built-in physical constants. Constants and parameters can have units.

BUILT-IN PHYSICAL CONSTANTS

NAME	DESCRIPTION
g_const	Acceleration of gravity
N_A_const	Avogadro constant
k_B_const	Boltzmann constant
Z0_const	Characteristic impedance of vacuum (impedance of free space)
me_const	Electron mass
e_const	Elementary charge
F_const	Faraday constant
alpha_const	Fine-structure constant
G_const	Gravitational constant
V_m_const	Molar volume of ideal gas (at 273.15 K and 1 atm)
mn_const	Neutron mass
mu0_const	Permeability of vacuum (magnetic constant)
epsilon0_const	Permittivity of vacuum (electric constant)
h_const	Planck's constant
hbar_const	Planck's constant over 2 pi
mp_const	Proton mass
c_const	Speed of light in vacuum
sigma_const	Stefan-Boltzmann constant
R_const	Universal gas constant
b_const	Wien displacement law constant

BUILT-IN MATHEMATICAL FUNCTIONS

These functions do not have units for their input or output arguments.

NAME	DESCRIPTION
abs	Absolute value
acos	Inverse cosine (in radians)
acosh	Inverse hyperbolic cosine
acot	Inverse cotangent (in radians)
acoth	Inverse hyperbolic cotangent
acs	Inverse cosecant (in radians)
acsch	Inverse hyperbolic cosecant

NAME	DESCRIPTION
arg	Phase angle (in radians)
asec	Inverse secant (in radians)
asech	Inverse hyperbolic secant
asin	Inverse sine (in radians)
asinh	Inverse hyperbolic sine
atan	Inverse tangent (in radians)
atan2	Four-quadrant inverse tangent (in radians)
atanh	Inverse hyperbolic tangent
besselj	Bessel function of the first kind
bessely	Bessel function of the second kind
besseli	Modified Bessel function of the first kind
besselk	Modified Bessel function of the second kind
ceil	Nearest following integer
conj	Complex conjugate
cos	Cosine
cosh	Hyperbolic cosine
cot	Cotangent
coth	Hyperbolic cotangent
csc	Cosecant
csch	Hyperbolic cosecant
erf	Error function
exp	Exponential
floor	Nearest previous integer
gamma	Gamma function
imag	Imaginary part
log	Natural logarithm
log10	Base-10 logarithm
log2	Base-2 logarithm
max	Maximum of two arguments
min	Minimum of two arguments
mod	Modulo operator
psi	Psi function and its derivatives

NAME	DESCRIPTION
range	Create a range of numbers
real	Real part
round	Round to closest integer
sec	Secant
sech	Hyperbolic secant
sign	Signum function
sin	Sine
sinh	Hyperbolic sine
sqrt	Square root
tan	Tangent
tanh	Hyperbolic tangent

The following tables summarize the built-in variables and functions that are generally available in all COMSOL models. Some are only available in certain geometries or in time-dependent models, for example. These variable names are reserved names and appear in orange in the settings windows for parameters and variables.

TABLE 4-13: BUILT-IN VARIABLES

NAME	DESCRIPTION	TYPE
t	Time	Scalar
freq	Frequency	Scalar
lambda	Eigenvalues	Scalar
phase	Phase angle	Scalar
numberofdofs	Number of degrees of freedom	Scalar
x, y, z, r, X, Y, Z, R	Position	Field
s, s1, s2	Edge/surface parameters	Field
n, nx, ny, nz, nr	Edge/surface normals	Field
tx, ty, tz, tr	Edge tangents	Field
t1x, t1y, t1z, t2x, t2y, t2z	Surface tangents	Field
un, unx, uny, unz	Edge/surface upward normals	Field
dn, dnx, dny, dnz	Edge/surface downward normals	Field
eps, i, j, pi, inf, Inf, nan, NaN	Numerical constants	Scalar

TABLE 4-13: BUILT-IN VARIABLES

NAME	DESCRIPTION	TYPE
h	Local mesh element size (length of the longest element edge)	Field
dom	The domain number, boundary number, edge number, or point number	Field
meshtype	Mesh type index for the mesh element; this is the number of edges in the element.	Field
meshelement	Mesh element number	Field
meshvol	Volume of the (linearized) mesh element	Field
dvol	Volume scale factor variable; this is the determinant of the Jacobian matrix for the mapping from local (element) coordinates to global coordinates.	Field
qual	A mesh quality measure between 0 (poor quality) and 1 (perfect quality)	Field
reldetjac	Determinant of the Jacobian matrix for the mapping from the straight mesh element to the possibly curved element used when solving	Field
reldetjacmin	The minimum value of the reldetjac variable in each element	Field
linearizedelem	One inside elements that have been linearized; zero otherwise	Field
niterCMP	Iteration number for nonlinear iterations	Scalar
gmg_level	Geometric multigrid level	Scalar
timestep	Current time step	Scalar



The suffixes *x*, *y*, *z*, and *r* in some of the variables are the default names for the spatial coordinates, which you can change if desired.

The following user-defined variables generate built-in variables such as space and time derivatives. See [Shape Function Variables](#) for information about those built-in variables.

TABLE 4-14: USER-DEFINED VARIABLES THAT GENERATE BUILT-IN VARIABLES

DEFAULT NAME	DESCRIPTION	TYPE
x, y, z	Spatial coordinates (Cartesian)	Field
r, z	Spatial coordinates (cylindrical)	Field
u, T, and so on	Dependent variables (solution)	Field

User-Defined Functions

User-defined functions are available under **Global Definitions** or, in any **Model** branch, under **Definitions** on the **Functions** submenu.

About User-Defined Functions

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

SMOOTHING

Many user-defined function types have smoothing active by default or available as an optional choice. Smoothing makes a function more well-behaved and suitable for modeling. It replaces jumps with smooth transitions that eliminate discontinuities and can represent the inertia present in real applications.

PLOTTING FUNCTIONS

Click the **Plot** button () in the upper-right corner of the settings window to create plots of all user-defined 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.

NAMING FUNCTIONS

Function names for built-in mathematical functions such as `abs`, `cos`, and `test` are reserved function names, and naming a user-defined function using one of the reserved function names is not recommended because it may cause unexpected results. If the name that you type in the **Function name** text field is a reserved function name, a warning about this appears by changing the function name’s color to orange. If you move the cursor to a function name in orange, a tooltip **is a reserved name** appears.

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, you can define 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.

Analytic

An **Analytic** function () is defined by a symbolic expression. Analytic functions have the ability to bind arguments during function calls. In other words, they do not require the actual argument names in an expression when writing the function. For example, you can define a function $f(x) = x^2$ with the input argument `x` and the expression `x^2` and the call it as `f(T)`, where `T` is the temperature in a heat transfer model. Select an **Analytic** node from the **Functions** submenu under **Global Definitions** or, in any **Model** branch, **Definitions**. The default **Function name** is `an1`.

-
- 
 - Go to [Common Settings for the Definitions Nodes](#) for links to information about the **Function Name** and **Plot Parameters** section.
 - See [Units for Function Inputs and Outputs](#) for information about the **Unit** section.
-

DEFINITION

In the **Expression** field, enter the mathematical expression that defines the function, such as `sin(x)*cos(y)+g_const` or `a+b*cos(c)`. Enter **Arguments** to the analytic function as comma-separated entries (`x, y` and `a, b, c` for the functions above). In addition to the arguments that are defined, analytic functions also recognize global parameters and physical constants.

From the **Derivatives** list, **Automatic** is selected by default and computes the derivatives symbolically. COMSOL uses the derivatives of a function if a variable that depends on the solution is used in a function argument. Select **Manual** to specify the function derivatives with respect to its arguments in a table. If **Manual** is selected, enter the derivatives with respect to the function's arguments. For undefined derivatives, COMSOL uses 0 as the value of the derivative. In the second example above, enter `a`,

b , and c in the top three rows of the **Argument** column, and 1 , $\cos(c)$, and $-b*\sin(c)$ in the associated text fields in the **Partial derivative** column.

PERIODIC EXTENSION

Select the **Make periodic** check box to make the function periodic and extend its definition within an interval to the whole real axis. Then define the interval by entering values in the **Lower limit** (default is 0) and **Upper limit** (default is 1) fields.

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 a real-valued input.

-
- If you have the AC/DC Module, see [Geoelectrics: Model Library path ACDC_Module/Other_Industrial_Applications/geoelectrics](#).
 - If you have the RF Module, see [Second Harmonic Generation of a Gaussian Beam: Model Library path RF_Module/Tutorial_Models/second_harmonic_generation](#).
-

Elevation

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. You select an **Elevation** function node from the **Functions** submenu. The default **Function name** is `elev1`.

FILE

Enter the path and name of the elevation file in the **Filename** text field, or click **Browse** to select a DEM file with elevation data in the **Elevation Data** dialog box. When a DEM file is open, 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 references the elevation data on your file system. When the elevation data is imported, the **File** section (under **Data imported into model**) contains information about the filename and the location for the data. Click **Export** to save the

elevation data to a file and reference from that file instead of including it in the model. Click the **Discard** button to delete the imported data from the model.

INTERPOLATION AND EXTRAPOLATION

For interpolation in the elevation data, select an **Interpolation** method from the list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the elevation data, select an **Extrapolation** method from the list—**Constant** (the default), **Linear**, **Nearest function** (which evaluates the function from the closest grid point at the actual point where a value is requested), or **Specific value**. For a **Specific value**, enter a value to **Replace missing data with** field (SI unit: m). The default is 0 m.

On the settings window toolbar, click the **Create Surface** button () to add a **Parametric Surface** node to represent the elevation function as a parametric surface in the **Geometry** branch for 3D models.

External



The **External** function is only available with the **Global Definitions** node. See also `model.func()` in the *COMSOL API Reference Manual* for details about the interface to external functions, including an example and information about compiling the function on the supported platforms.

An **External** function () 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. For example, use it for a user-created shared library (depending on the platform, a DLL, .so, or .dylib file). You select an **External** function node from the **Functions** submenu.



Go to [Source and Destination Vertices and Boundaries](#) for links to information about the **Derivatives** and **Plot Parameters** sections.

FUNCTIONS

Enter a **Library** path and name (the complete network path), or click **Browse** to locate a library to import. For each row in the table, enter a **Function name** (myfun, for

example) and a space-separated or comma-separated list of the names of its input **Arguments** (*x y*, for example).

ADVANCED

Enter a value in the **Initialization data** field. The value is sent to the library when it is loaded. 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). Select this check box to then improve performance.

Gaussian Pulse

The **Gaussian Pulse** function () is the common bell-shaped curve (Gaussian function). It has a shape that is similar to a Gaussian (normal) distribution. You select a **Gaussian Pulse** function node from the **Functions** submenu. The default **Function name** is *gp1*.

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:

$$y(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-x_0)^2}{2\sigma^2}}$$

In the equation above, *x* is the input variable, *x₀* is the location (mean), and *σ* is the standard deviation. This function is a function of one variable (the time *t*, for example).

PARAMETERS

Enter a **Location** value for the Gaussian pulse mean *x₀* (the default location is 0). Enter a **Standard deviation** *σ* of the normal distribution. The default is 1.

-
- | | |
|---|--|
|  | <ul style="list-style-type: none">• If you have the RF Module, see Transient Modeling of a Coaxial Cable: Model Library path
RF_Module/Verification_Models/coaxial_cable_transient. |
|---|--|

- | | |
|---|---|
|  | <ul style="list-style-type: none">• 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. |
|---|---|

Image

The **Image** function () makes it possible to import an image (in BMP, JPEG, PNG, or GIF format) and map the image's RGB (red, green, blue) data to a scalar (single channel) function output value. By default the function's output uses the mapping $(R+G+B)/3$. You select an **Image** function node from the **Functions** submenu. The default **Function name** is `im1`. An image is defined on a two-dimensional domain, and you typically call the image function using spatial coordinates: `im1(x,y)`.

-
- 
 - Go to [Source and Destination Vertices and Boundaries](#) for links to information about the **Function Name** section.
 - See [Units for Function Inputs and Outputs](#) for information about the **Units** section.
-

FILE

Enter the path and name of the image file in the **Filename** 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 references the image on your file system. When you have imported the image, the **File** section, under **Data imported into model**, contains information about the image's filename and size. Click **Export** to save the image to a file and reference it from there instead of keeping it in the model. Click the **Discard** button to delete the imported image data from the model.

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. Click to clear the check box to define the image coordinates explicitly using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields. Select the **Flip horizontally** check box to flip the image horizontally from left to right and vice versa. Select the **Flip vertically** check box to flip the image vertically from up to down and vice versa.

COLOR SCALING

From the **Scaling** list, select **Automatic** (the default) to use the default scaling, which outputs the mean of the RGB values for each pixel in the image. Select **Manual** to specify a custom expression for the scalar image function output 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.

For interpolation in the image, select an **Interpolation** method from the list—**Nearest neighbor** or **Linear** (the default).

For extrapolation of values that are outside the range in the image, select an **Extrapolation** method from the list—**Constant** (the default), **Linear**, **Nearest function** (which evaluates the function from the closest grid point at the actual point where a value is requested), or **Specific value**. If **Specific value** is selected, enter a value in the **Value outside range** field (default value: 0).

CLIPPING

Apply clipping to create a box-shaped region inside of the original image where the image is rendered. From the **Clipping** list, select **None** (the default) for no clipping, or **Manual** to define the box-shaped region using the **x minimum**, **x maximum**, **y minimum**, and **y maximum** fields (unit: px). The default **x maximum** and **y maximum** values are both 1000 px; the default minimum values are 0.

Interpolation

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). You select an **Interpolation** function node from the **Functions** submenu.

DEFINITION

Select a **Data source**—**Table** (the default) or **File** to define the interpolation function by entering values in a table or by importing interpolation data from a file, respectively.

- If **Table** is selected, enter a **Function name** and enter coordinates t and function values $f(t)$ into the table cells. 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. The default **Function name** is **int1**.
- *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:* Import or 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**. If the license includes LiveLink™ for Excel® you can also load interpolation

data from a Microsoft Excel Workbook spreadsheet. 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 define the inverse function f^{-1} for the interpolation function f , select the **Define inverse function** check box and enter a function 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 you want to plot the inverse function instead of the interpolation function itself, first select the **Plot the inverse function** check box in the **Plot Parameters** sections, which is only available when you have chosen to define the inverse function.



The inverse function only exists if the function is strictly monotonic.

If **File** is selected to import interpolation data from a file select a **Data format**—**Spreadsheet**, **Grid**, or **Sectionwise**. If the license includes LiveLink™ for Excel®, you can also import interpolation data from a Microsoft® Excel Workbook spreadsheet. COMSOL then uses the spreadsheet format and the **Data format** list is not available.

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

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 (typically spatial coordinates); the following columns can contain one or more functions, and the positions entered are the relative position for each function’s data column.

When the data format is specified, enter the path and name of the interpolation data file in the **Filename** text field, or click **Browse** to select a text or data file with interpolation data in the **Interpolation Data** dialog box. Then click **Import** to import the interpolation data into the model; otherwise COMSOL references the interpolation data on your file system. When you have imported the interpolation data, the

Parameters section, under **Data imported into model**, contains information about the filename, data type, and dimension for the data. Click **Export** to save the interpolation data to a file and reference from there instead of including it in the model. Click the **Discard** button to delete the imported interpolation data from the model.

For the common case where the data source contains function values that are functions of the spatial coordinates, select the **Use space coordinates as arguments** check box.

Then select the frame to which the spatial coordinates are attached from the **Frame** list (the default is **Spatial** for the spatial frame). Then the function can be called without arguments when used in the model; the spatial coordinates are added as function arguments automatically. The **Use space coordinates as arguments** check box is available for **Interpolation** nodes in a **Model** branch when the **Data source** is **File** or when using a **Table** in 1D models.



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.

An 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 spatial coordinates (or other function arguments) where it is defined. In this example with two arguments (spatial 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 (see below).

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 adding the input arguments.

Examples of Spreadsheet, Sectionwise, and Grid File Formats

Spreadsheet File 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.

%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.

Sectionwise File A *sectionwise* file has coordinates and function values.

%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.

Grid File 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.

 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.

INTERPOLATION AND EXTRAPOLATION

The interpolation and extrapolation settings control 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.

Select an **Interpolation** method:

- For functions of one variable select **Nearest neighbor**, **Linear** (the default interpolation method), **Piecewise cubic**, or **Cubic spline**.
 - **Nearest neighbor** interpolation selects the value of the nearest point where the function is defined.
 - **Linear interpolation** uses a linear polynomial to interpolate the function between the points where it is defined.
 - **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.

Select an **Extrapolation** method 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 function evaluates the polynomial from the closest grid point at the actual point where a value is requested. This is the default extrapolation method.
- **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.
- **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.



Unstructured extrapolation supports a constant or a specific value only.

PLOT PARAMETERS

If you have selected to define the inverse function, this section becomes available.

Select the **Plot the inverse function** check box to plot the inverse function instead of the interpolation function itself when you click the **Plot** button () to generate a preview 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, which show how the selected extrapolation extends the interpolation function on both sides. In addition, a point graph shows the interpolation points (interpolation nodes).

	<p>Units for Function Inputs and Outputs for information about the Units section.</p>
	<p>If you have the:</p> <ul style="list-style-type: none">• 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 Transfer Module, see Temperature Field in a Cooling Flange: Model Library path Heat_Transfer_Module/Thermal_Processing/cooling_flange.• 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.

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

Use **MATLAB** functions node () from COMSOL to interface to functions written in MATLAB. These functions can be used as any other function defined in COMSOL. MATLAB functions are evaluated in MATLAB. You select a **MATLAB** function node from the **Functions** submenu.

FUNCTIONS

Under **Functions**, add the names of the MATLAB functions in the **Function name** column, and 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). Click the **Clear Functions** button to force a reload of the functions if they have been edited.

Alternatively, you can select the **Clear functions automatically before solving** check box.



- Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Derivatives** section.
- The *LiveLink™ for MATLAB® User's Guide*

Piecewise

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. You select a **Piecewise** function node from the **Functions** submenu. The default **Function name** is `pw1`.

DEFINITION

Enter a name for the argument to the piecewise function in the **Argument** field.

Select an **Extrapolation** method to control what happens when the function is evaluated in a point that does not belong to any interval—**Constant** (the default), **None**, **Nearest function**, **Specific value**, or **Periodic**.

- **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.** Also enter a value in the **Value outside range** field. If selecting the specific-number method, assign a single value (usually zero or NaN) to all points outside the intervals.
- **Periodic.** The function becomes periodic by repeating its values in the interval where it is defined in regular intervals of the same size.

Select a **Smoothing: No smoothing** (the default), **Continuous function** (to make the function continuous but not its derivatives), **Continuous first derivative**, or **Continuous second derivative**. For any selection (except **No smoothing**), enter a value in the **Relative size of transition zone** field (dimensionless). The default is 0.1. Relative size means relative in relation to the size of the intervals on both sides of the border.



The functions in contiguous intervals need not evaluate to the same value where the intervals meet. If the values differ the function has a discontinuity, and it is recommended that you apply smoothing to make the piecewise function more well-behaved numerically. Functions from neighboring intervals are then blended close to where the intervals meet.

For each cell in the **Intervals** table, enter **Start** and **End** interval limits. The intervals must not overlap, and there cannot be any holes between intervals. Enter an expression defining the **Function**.



The intervals must be contiguous.

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.



See [Units for Function Inputs and Outputs](#) for information about the **Units** section.

- 
- If you have the Heat Transfer Module, see [Radiative Heat Transfer in a Utility Boiler](#): Model Library path **Heat_Transfer_Module/Thermal_Radiation/boiler**.
 - If you have the Wave Optics Module, see [Nanorods](#): Model Library path **Wave_Optics_Module/Optical_Scattering/nanorods**.

Ramp

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). You select a **Ramp** function node from the **Functions** submenu. The default **Function name** is `ramp1`.

PARAMETERS

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. Enter a **Slope k** (dimensionless) of the ramp. The default is 1.

To ensure that the value never exceeds a certain point, select the **Cutoff** check box and enter a value. The default is 1. 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 $s \geq s_0$.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Function Name** and **Smoothing** section.

Random

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. You select a **Random** function node from the **Functions** submenu. The default **Function name** is `rn1`.

To generate a random function of the spatial 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.

PARAMETERS

Enter a **Number of arguments** to the random function (the default is 1). Select a **Distribution** method—**Uniform** (the default) or **Normal**. If **Uniform** is selected, enter a **Mean** and **Range**. The default mean and range is 0 and 1, respectively. 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**. The defaults are 0 and 1, respectively.

Rectangle

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; you can 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). You select a **Rectangle** function node from the **Functions** submenu. The default **Function name** is `rect1`.

LIMITS

Enter a **Lower limit** (the default is -0.5) and **Upper limit** (the default is 0.5) 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 the interval from the lower limit to the upper limit. Outside the interval it evaluates to 0.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Function Name** and **Smoothing** section.



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.

Step

A **Step** function (J) 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). You select a **Step** function node from the **Functions** submenu. The default **Function name** is `step1`.

PARAMETERS

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. In the **From** field, enter a start level (L_{start}). 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 $L_{\text{end}} - L_{\text{start}}$.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Function Name** and **Smoothing** sections.



[Fluid Valve](#): Model Library path
COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve

Thermodynamics Package

COMSOL provides a **Thermodynamics Package** feature to enable the linking to external physical and thermodynamic property calculations for use in the Chemical Reaction Engineering Module.



For more information see the *Chemical Reaction Engineering Module User's Guide*.

Triangle

A **Triangle** function () is a linear increase and linear decline within an interval and 0 everywhere else. You can use the triangle function for band-pass filtering, for example; that is, use it to select values within an interval. The triangle function is a function of one variable (the time t , for example). You select a **Triangle** function node from the **Functions** submenu. The default **Function name** is tri1.

LIMITS

Enter a **Lower limit** (the default is -0.5) and **Upper limit** (the default is 0.5) 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.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Function Name** and **Smoothing** sections.

Waveform

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). You select a **Waveform** function node from the **Functions** submenu. The default **Function name** is wv.

PARAMETERS

Select a waveform **Type:** **Sawtooth**, **Sine** (the default), **Square**, or **Triangle**. For any selection, enter **Angular frequency** (default is 1), **Phase** (unit: radians; the default is 0), and **Amplitude** (default is 1) values.



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_Id_for_thermal_models**.

Specifying Discontinuous Functions

To specify a discontinuous function, such as a step in space or time, you can use logical functions that evaluate to 1 if true and 0 otherwise. For instance, the following function defines a sine wave that exists for 10 seconds and afterward takes the value 0:

```
sin(2*pi*t)*(t<10)
```

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 or 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. It includes smoothing by default. The Piecewise, Ramp, Rectangle, and Triangle functions also include smoothing (active by default for Rectangle and Triangle functions).

Smoothed Step and Rectangle functions are defined by piecewise 5th-degree polynomials, smoothed Ramp functions by piecewise 4th-degree polynomials, and smoothed Triangle functions by piecewise 3rd-degree polynomials. None of these functions have any overshoot or undershoot.

ADDITIONAL SMOOTHED FUNCTIONS

In addition, the following smoothed functions are available:

- **f1smhs**, 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=f1smhs(x,scale)` approximates the logical expression `y = (x>0)` by smoothing the transition within the interval `-scale < x < scale`. **f1dsrhs** is the derivative of the smoothed Heaviside function.
- **f1smsign**, a smoothed sign function with a continuous first derivative. `y = f1smsign(x,scale)` approximates the function `y = sign(x)` by smoothing

the transition within the interval $-scale < x < scale$. `f1dsmhs` is the derivative of the smoothed sign function.

- `f1c1hs`, a smoothed Heaviside function with a continuous first derivative without overshoot. Its syntax is similar to the functions just described.
- `f1c2hs`, 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 `f1smhs` and `f1smhs` 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 `f1c1hs` 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` with a value of 0.1 and then an **Analytic** node where an analytic function `HeatCapacity` is defined using the following expression with an argument `T: 1+f1c1hs(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

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 used at a point in the destination, the value is computed by evaluating the argument in the source. You can use a coupling operator to compute several quantities that use the same source and destination by calling it with different arguments. All types of model couplings have a source and a destination.



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 (such as some domains or boundaries) where the coupling operator evaluates the supplied expression (as an integration over the source, for example), while the *destination* is the part of the geometry where the result of the coupling operator is defined. The destination can, depending on the type of coupling, be a subset of one or several models or a “global destination” (for a scalar value that is available everywhere). The coupling operator’s value is computed by evaluating the expression given as an argument at one or several points in the source. The source and destination are both geometrical objects, but the source is limited to one geometry whereas the destination is often global.

To add model couplings to the **Model Builder**, right-click **Definitions** in a Model branch and select a **Model Coupling**.

ABOUT COUPLING OPERATORS

Coupling operators are useful for modeling coupled problems and are generalizations of expressions. They are defined by first selecting the source, where the argument of the operator is evaluated, and, in some cases, a destination. An expression to evaluate is not required when you define a coupling operator; instead you can use coupling operators in different modeling contexts, passing the expression to evaluate as an input argument.

There are three categories of coupling operators:

- *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. When the source and destination are of the same space dimension, it is typically a pointwise mapping. When the destination has higher dimension than the source, the mapping is done by extruding pointwise values to the higher dimensions. For some examples of the use of extrusion coupling operators, see [Examples of Extrusion Couplings](#).
- *Projection*. These operators—General Projection and Linear Projection—evaluate a series of line or curve integrals on the source, where the line or curve positions depend on the positions of the evaluation points in the destination. 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, you can obtain the average along the y direction of a variable u defined on some 2D domain in the xy -plane by computing the integral

$$\bar{u}(x) = \int u(x, y) dy$$

COMSOL uses a method whereby it first applies a one-to-one map to the mesh of the source. It then carries out the integrals in the source over curves that correspond to vertical lines in the transformed source mesh. You can define the map between source and destination in two ways: as a *linear projection* or as a *general projection*. For some examples of the use of projection coupling operators, see [Examples of Projection Couplings](#)

- *Scalar*. These operators—Integration, Average, Maximum, and Minimum—define a scalar value such as an integration, the average over a set of geometric entities, or

the maximum or minimum value of an expression and have a “global destination” (that is, they are available everywhere in the model):

- An *Integration coupling operator* is the value of an integral of an expression over the source, which is a set of geometric entities (domains, for example).
- An *Average coupling operator* computes the average of an expression over the source.
- A *Maximum* or *Minimum coupling operator* computes the maximum or minimum, respectively, of an expression over the source.

These operators can be evaluated anywhere in a model, and the value does not depend on where in the model the evaluation occurs. Integration couplings are useful for evaluating integrated quantities. To evaluate the total current across a boundary in a 2D Electric Currents model, for example, define an integration coupling operator `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). For some other examples of the use of integration coupling operators, see [Examples of Integration Couplings](#)

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 for results evaluation and visualization purposes.
- Define nonlocal couplings including mesh transformations, integrals over domains and boundaries, and projections.

ABOUT SOURCE AND DESTINATION MAPPINGS

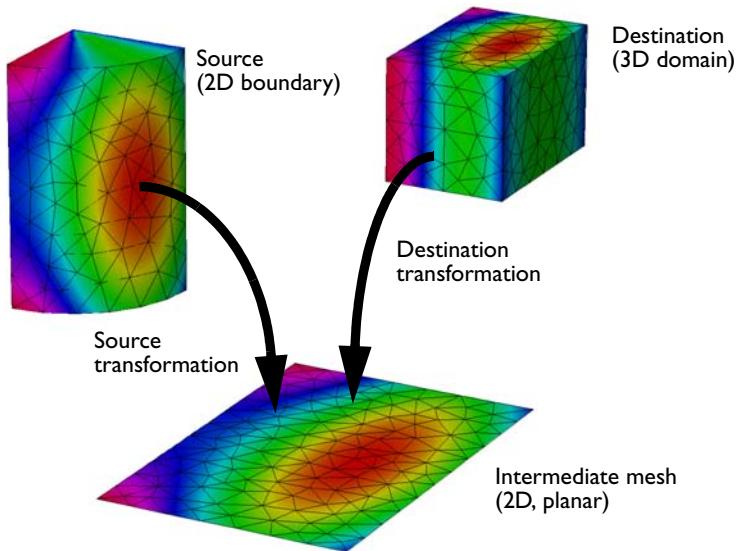


Figure 4-1: An example of a general extrusion mapping.

The definition of any extrusion model coupling involves two mesh maps. 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` \geq `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.

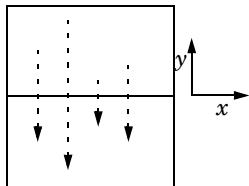
When the value of the coupling operator is requested somewhere in the destination, the software transforms the destination points using the destination map. It compares the resulting coordinates to the elements in the intermediate mesh to find corresponding locations in the physical source. This means that the source map must be inverted but not the destination map. The latter can in fact be noninvertible, 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, COMSOL 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.

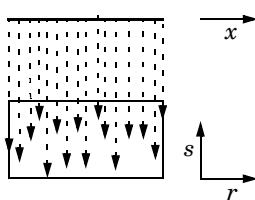
EXAMPLES OF EXTRUSION COUPLINGS



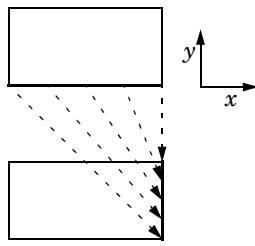
All the graphics in these examples use **General Extrusion** model coupling.



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 ($\text{srcedim} = \text{idim} = \text{dstedim}$). 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 $\text{srcedim} = \text{idim} = 1$, $\text{dstedim} = 2$.

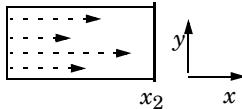


Another example maps values on the lower boundary of a rectangle that extends from $x = -1$ to $x = 1$ and from $y = 0$ to $y = 1$, to the right boundary on the same rectangle. The source map is $(x+1)/2$ and the destination map is y . Both maps have a single component since $\text{srcedim} = \text{idim} = \text{dstedim}$. 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.

$$\begin{aligned}-\Delta u &= \frac{\partial}{\partial n} u(x_2, y) && \text{on } \Omega \\ u &= xy && \text{on } \partial\Omega\end{aligned}$$



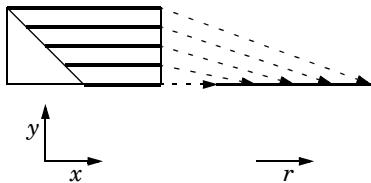
The figure to the left illustrates the extrusion process. The values of the influx on the boundary become available throughout the domain by extrusion along the y -axis. The source map is y , and the destination map is y .

EXAMPLES OF PROJECTION COUPLINGS



All the graphics in these examples use the **General Projection** model coupling. These examples can also be done using **Linear Projection**.

Projection Coupling—Example 1

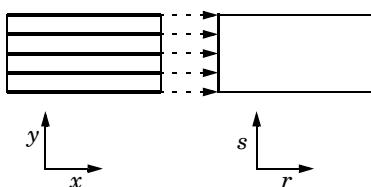


For each point r , the coupling operator returns the integral

$$v(r) = \int_{y=r/2}^{y=1} u(x,y) dx$$

$$(x,y) \in S_2$$

The source map is y , x , and the destination map is $r/2$.

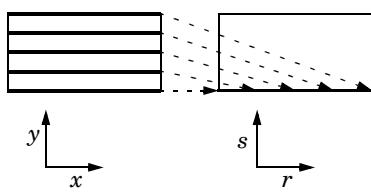


For each point $(0,s)$, the coupling operator returns the integral

$$v(0,s) = \int_{y=s}^{y=1} u(x,y) dx$$

$$(x,y) \in S_2$$

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 $r/2$.



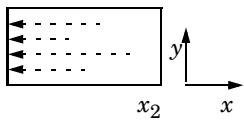
The integration can also sweep non-rectangular domains. The integrals include only the source domains; they exclude other domains and the external area.

Projection Coupling—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.

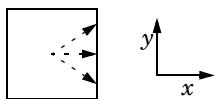
$$-\Delta u = 1 \quad \text{on } \Omega$$

$$u = 0 \quad \text{on } \partial\Omega$$



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)`.

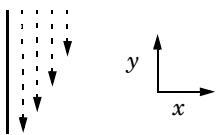
EXAMPLES OF INTEGRATION COUPLINGS



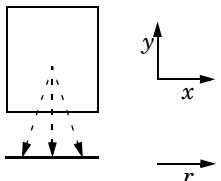
Consider Poisson's equation on a rectangular domain. The integral of the solution squared serves as the influx in a Neumann boundary condition on the right boundary. There is a Dirichlet boundary condition on the left boundary, and the top and bottom boundaries have zero influx.

$$\begin{aligned} u &= x && \text{on } \partial\Omega_1 \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega_{2,3} \\ \frac{\partial u}{\partial n} &= -\int_{\Omega} u^2 d\Omega && \text{on } \partial\Omega_4 \end{aligned}$$

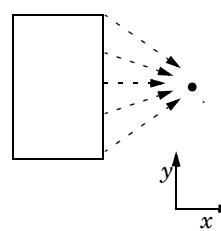
For example, define an integration coupling operator called `inttop1`, with the rectangular domain as source. The influx for the Neumann boundary condition is obtained as `inttop1(u^2)`.



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.

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; they 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. For example, defining a heat source based on an integration operator over the whole domain that is also a function of temperature (the dependent variable) leads to a coupling between all the degrees of freedom in the model creating a full Jacobian matrix instead of the sparse limited-bandwidth matrices typical of finite element models.



You can prevent the fill-in of the Jacobian matrix using the **nojac** operator, which forces COMSOL to exclude the expression that it encloses when forming the Jacobian. Using the **nojac** operator can slow down the convergence of the solution. Another possible solution is to add a single degree of freedom that represent the value of an expression with a scalar coupling operator.

General Extrusion

A **General Extrusion** coupling operator () maps an expression defined on a source to an expression that can be evaluated on any destination geometry 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. Specifically, when the destination has more space dimensions than the source, the operator performs extrusion of values. The default **Operator name** is genext1.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, **Source**, and **Advanced** sections.

DESTINATION MAP

Specify the general extrusion destination map by entering an 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.



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.

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/Scattering_and_RCS/radar_cross_section.
- Subsurface Flow Module, see [Aquifer Characterization](#): Model Library path **Subsurface_Flow_Module/Fluid_Flow/aquifer_characterization**.

Linear Extrusion

A **Linear Extrusion** coupling operator () maps an expression defined on a source to an expression that can be evaluated in the destination. 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. The Linear Extrusion operator defines a linear extrusion that maps between geometric parts of the same dimension. The parts can exist in geometries of different space dimensions. For example, you can couple edges (boundaries) in 2D to edges in 3D; or couple 2D domains to 3D faces. In these cases geometries of different space dimensions are needed for the source and

destination. You define the linear extrusion by specifying points in both source and destination. The default **Operator name** is `linext1`.

 Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, **Source**, **Source Vertices**, **Destination Vertices**, and **Advanced**, sections.

DESTINATION

The mapping from destination to source is defined as the following:

- First, the destination is orthogonally projected onto the linear space spanned by the destination vertices.
- Then this linear space is mapped linearly to the source, so that each destination vertex is mapped to the corresponding source vertex.

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. Select an option from the **Destination frame** to evaluate the destination vertex coordinates in the specified frame.

-
- 
 - If you have the CFD Module, see [Turbulent Flow Over a Backward Facing Step](#): Model Library path **CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**.
 - If you have the Heat Transfer Module, see [Turbulent Flow Over a Backward Facing Step](#): Model Library path **Heat_Transfer_Module/Verification_Models/turbulent_backstep**.
-

Boundary Similarity

The **Boundary Similarity** coupling operator () maps an expression defined on a part of a boundary to another part of a boundary with the same shape. This operator is slightly different for 2D and 3D models:

- 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. By default, the algorithm automatically chooses a map when symmetries make several maps possible. To control this choice in 3D, add a [One-Point Map](#), [Two-Point Map](#), or

Edge Map subnode ().

- *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.

The default **Operator name** is `bndsim1`.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Boundaries**, and **Destination Boundary** sections. For the rest of the **Advanced** settings, see [Advanced Settings for Model Couplings](#).

ADVANCED



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.



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.

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.



Only one map node is allowed per boundary similarity coupling and only one source or destination point per field is allowed.

One-Point Map

Use a **One-Point Map** to control a [Boundary Similarity](#) model coupling mapping in 3D.

POINTS

To select a single **Point on source** and a single **Point on destination** adjacent to the source and destination selection, click the **Activate Selection** button () under the fields. Then click in the **Graphics** window and select the point or use the **Selection List** window.

Two-Point Map

Use a **Two-Point Map** to control a [Boundary Similarity](#) model coupling mapping in 3D.

SOURCE POINTS

To select a single **First point on source** and a single **Second point on source**, click the **Activate Selection** button () under the field (one at a time) and then click in the **Graphics** window and select the points adjacent to the source selection or use the **Selection List** window.

DESTINATION POINTS

To select a single **First point on destination** and a single **Second point on destination** adjacent to the destination selection, click the **Activate Selection** button () under the field (one at a time) then click in the **Graphics** window and select the point or use the **Selection List** window.

Edge Map

Use an **Edge Map** to control a [Boundary Similarity](#) model coupling mapping in 3D.

EDGES

To select a single **Source edge** and a single **Destination edge**, click the **Activate Selection** button () under the field (one at a time) and then click in the **Graphics** window and select the edge or use the **Selection List** window. The destination edge is mapped to the source edge by the similarity mapping from destination to source.

ADVANCED

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.

Identity Mapping

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. The default **Operator name** is `idmap1`.

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 (x,y,z)** frame can be used in both cases. Or choose **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics. Choose a **Destination geometry** from the list.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, and **Advanced** sections.



If you have the CFD Module, see [Solar Panel in Periodic Flow: Model](#) Library path **CFD_Module/Single-Phase_Tutorials/solar_panel**.

General Projection

Use a **General Projection** model coupling (srecdim, and the destination to the subspace of dimension `srecdim-1` obtained by setting the last coordinate to 0 (`srecdim` is the dimension of the source selection). To every point in the destination, there is 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 that 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. The default **Operator name** is `genproj1`.



It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Source Selection** section.

SOURCE MAP

Specify the general projection source map by entering expressions in the **x-expression**, **y-expression**, and **z-expression** fields.



The dimension of the intermediate space equals the dimension `srecdim` of the source. If the selection has lower dimension than the source geometry, specify only the first `srecdim` expressions.

Use expressions containing spatial coordinates in the source geometry when defining the map. The map must be approximately linear within each mesh element. Select a **Source frame** from the list.

DESTINATION MAP

Enter an **x-expression** and, depending on the 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 you can use expressions containing spatial coordinates in the destination geometry. The destination mapping can be highly nonlinear or noninvertible.

- If the selection has lower dimension than the source geometry, specify only the first `srcdim-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.

ADVANCED

Enter an **Integration order** of the numerical integration method (default: 4).

Linear Projection

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 a source and a destination of the nearest lower dimension. The source and destination can exist in geometries of different space dimensions. For example, you can couple domains in 2D to edges in 3D or couple 3D domains to 2D domains. You define the linear projection by specifying points in both the source and destination. Select a **Linear Projection** node from the **Model Couplings** submenu. The default **Operator name** is `linproj1`.



It is only possible to use projection model coupling with simplex elements such as triangles and tetrahedra.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, **Source Vertices**, and **Destination Vertices** sections.

SOURCE

Select a **Source frame** from the list to evaluate the coordinates of the source vertices in the selected frame.

Then specify the linear projection by giving a set of points in the source and in the destination. The order of the vertices is significant. COMSOL constructs a linear projection from the source to the destination using the subspaces spanned by the vertices. Denote the map rank by n , denote the source vertices by x_0, x_1, \dots, x_n , and denote the destination vertices x'_0, x'_1, \dots, 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*:

$$\begin{aligned}x'_0 &= Tx_0 + V \\x'_1 - x'_0 &= T(x_1 - x_0) \\&\dots \\x'_n - x'_0 &= T(x_n - x_0)\end{aligned}$$

For the projection model coupling there must be one more vertex in the source than in the destination.

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**.

ADVANCED

Enter an **Integration order** of the numerical integration method (default: 4).

Integration

An **Integration** model coupling (*fdu*) integrates an expression over the source (some selected geometric entities like domains, boundaries, or edges). You can also use it with a point as the source to make the value of an expression at that point available globally. The integral is evaluated by integrating the expression (integrand) in the argument over the source (or, in some cases, by summing the expression over the node points in the source). Integration coupling operators have global destination, so they can be

evaluated anywhere in the model. Because it is an operator, you can 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. The default **Operator name** is `intop1`.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, and **Advanced** sections.

-
- 
- **Acoustics of a Muffler:** Model Library path:
[COMSOL_Multiphysics/Acoustics/automotive_muffler](#)
 - **Fluid Valve:** Model Library path
[COMSOL_Multiphysics/Fluid_Dynamics/fluid_valve](#)
-

Average

An **Average** model coupling (`av`) computes the average of an expression over the source (some selected geometric entities). It can be evaluated anywhere in any model. 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 type of geometric entities in the source. The default **Operator name** is `aveop1`.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name**, **Source Selection**, and **Advanced** sections.



Effective Diffusivity in Porous Materials: Model Library path
[COMSOL_Multiphysics/Diffusion/effective_diffusivity](#)

Maximum and Minimum

The **Maximum** (`MAX`) and **Minimum** (`MIN`) coupling operators compute the maximum or minimum of an expression over selected geometric entities in the source and gives the maximum or minimum value of the expression in the argument over the source. The operator can be evaluated anywhere in any model. Two arguments can be given, and the returned value is then 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)`. The default **Operator name** is `maxop1` or `minop1`.



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).

ADVANCED

Select a **Point type**—**Node points** (the default), **Integration points**, or **Lagrange points**. 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.

- If **Integration points** is selected, enter an **Integration order**. The default is 4.
- Select **Lagrange points** to compute the maximum or minimum by evaluating the expression in the argument at a finite set of points in the source and taking the maximum or minimum of these values. If it is selected, enter a **Lagrange order**. The default is 2.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Operator Name** and **Source Selection** sections.



[Deformation of a Feeder Clamp](#): Model Library path
`COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp`

Common Settings for Model Couplings

The following sections in the settings windows for the model coupling nodes are similar or the same for some of the model coupling nodes and are described in this section.

OPERATOR NAME

Enter a name for the operator in the **Operator name** field or use the default name. This is the name that is used to access the operator in the model, so use a name that describes it well.

SOURCE SELECTION

The source selection defines the source for the model coupling—the part of the geometry where the coupling operator evaluates the supplied expressions.

From the **Geometric entity level** list, select **Domain**, **Boundary**, **Edge** (3D only), or **Point**. 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 to the **Selection** list.

SOURCE AND DESTINATION VERTICES AND BOUNDARIES

Click the **Activate Selection** button () to define the source vertices and destination vertices, or source boundaries and destination boundaries. Select and add one single source vertex or source boundary for each destination vertex or destination boundary, respectively.

SOURCE VERTICES AND DESTINATION VERTICES



The selection of **Source Vertices** and **Destination Vertices** define the linear mapping from the destination to the source.

Select a single source vertex for each of **Source vertex 1**, **Source vertex 2**, **Source vertex 3**, and **Source vertex 4**. Then select a single destination vertex for each of **Destination vertex 1**, **Destination vertex 2**, **Destination vertex 3**, and **Destination vertex 4** (vertex 4 is available for [Linear Extrusion](#) only).

- For [Linear Extrusion](#): The number of source vertices must be at least one and not more than $1+\min(\text{sresdim}, \text{dstsdim})$, where sresdim and dstsdim are the dimensions of the source and destination geometries, respectively. The number of

destination vertices entered should be the same as the number of source vertices. If not all destination vertex selections are used, the empty selections must be last.

- For **Linear Projection**, select `srecdim+1` source vertices where `srecdim` 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. 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 `srecdim` destination vertices where `srecdim` 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.

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 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.

SOURCE BOUNDARIES AND DESTINATION BOUNDARIES

Boundary Similarity Coupling

Select **Manual** or **All boundaries** from the **Selection** list to define the source selection. If **Manual** is selected, select boundaries in the **Graphics** window. Select **All boundaries** to add all boundaries to the **Selection** list.

There can only be one destination boundary. Click the **Activate Selection** button () and then choose the boundary in the **Graphics** window or from the **Selection List**.

SOURCE FRAME AND SOURCE MAP

Select a **Source frame** to use in the source. In most cases the **Source** section default settings can be used. 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.

For the **General Extrusion** model coupling, 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 nonempty source and destination map expressions, which must be the same, and must also satisfy `srcedim ≤ idim ≤ srctdim`, where `srcedim` is the dimension of the source selection, and `srctdim` is the dimension of the source geometry.

ADVANCED SETTINGS FOR MODEL COUPLINGS

General Extrusion, Linear Extrusion, Boundary Similarity, and Identity Mapping

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.

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.

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.

Integration and Average

Select **Integration** or **Summation over nodes** from the **Method** list. 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).

Coordinate Systems

About Coordinate Systems

COMSOL 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 coordinate 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*.

The spatial Cartesian coordinate system coordinates default to the following names in 2D and 3D (in 2D axisymmetric geometries, COMSOL uses cylindrical coordinates):

GEOMETRY	DEFAULT NAME OF SPATIAL COORDINATES
2D	x y
3D	x y z
Axial symmetry 2D	rφ z

In 3D, an image displays in the lower-left corner of the **Graphics** window to indicate the orientation of the global coordinate system.



User-defined coordinate systems can be used on all geometric entity levels to simplify the modeling process. In some of the physics user interfaces, these coordinate systems can be used to define orthotropic and anisotropic material properties that are not aligned with the global Cartesian coordinate system. See [Table 4-15](#) 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.

COORDINATE SYSTEM DESCRIPTIONS:

TABLE 4-15: COORDINATE SYSTEM DESCRIPTIONS

NAME AND LINK	ICON	TYPE	DESCRIPTION
Base Vector System		relative	1D, 2D, and 3D. Define this using a set of base vectors to form a coordinate system, which are declared as orthonormal.
Boundary System		absolute	2D and 3D. A local base vector system on 2D boundaries (t, n) and on 3D boundaries (t_1, t_2, 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.
Cylindrical System		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.
Mapped System		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.
Rotated System		relative	2D and 3D. Use this to define rotation about the out-of-plane direction in 2D and Euler angles in 3D.
Spherical System		absolute	3D only. Use this when a field or property using spherical coordinates is to be specified.
Scaling System		absolute	For physics 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.



- Sorting Nodes by Space Dimension and Type
- Spatial Coordinate Variables

Base Vector System

Define a **Base Vector System** (blue icon) 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.

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:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3] \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$
$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3]^{-1} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \left\{ \begin{array}{l} |\mathbf{u}_i| = 1 \\ \mathbf{u}_i \cdot \mathbf{u}_j = \delta_{ij} \end{array} \right\} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3]^T \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

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 coordinate 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.



The Curvilinear Coordinates user interfaces can create special base vector systems in **Curvilinear System** nodes (blue icon). See [Curvilinear Coordinates](#).

SETTINGS

Coordinate Names

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. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

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).



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.



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.



Simplifications

Set base vector system properties that help simplify the coordinate transformations. Select the **Assume orthonormal** check box if the coordinate system is orthonormal.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.



- If you have the Nonlinear Structural Materials Module, see [Pressurized Orthotropic Container](#): Model Library path **Nonlinear_Structural_Materials_Module/Plasticity/orthotropic_container**.
- If you have the Structural Mechanics Module, see [Piezoelectric Shear-Actuated Beam](#): Model Library path **Structural_Mechanics_Module/Piezoelectric_Effects/shear_bender**.

Boundary System

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.



For 3D and 2D models, a **Boundary System** node is automatically added under **Definitions**.



Common applications for this coordinate system include specifying pressure or normal displacement on a surface.

To specify the boundary coordinate system, you 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, but you can reverse the normal direction. The general definition of the normal 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](#).

- 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. 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.
- 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. 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.

SETTINGS

Frame Type

Select a **Frame type**—**Deformed configuration** (the default), **Geometry configuration**, or **Reference configuration**. The deformed configuration follows the material whereas the reference configuration is attached to the spatial frame. The geometry configuration is used to specify normal and tangential components of boundary conditions and refers to the undeformed geometry when using a Deformed Geometry interface.

Coordinate Names

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. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses. To reverse the direction of the normal for the boundary system, select the **Reverse normal direction** check box.

Select an option from the **Create first tangential direction from** list: **Global Cartesian** (the default) or **Manual**. If **Global Cartesian** is selected, select **1**, **2**, or **3** (that is, *x*, *y*, or *z*) from the **Axis** list. 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.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.



Many Model Library examples use this coordinate system. For one example, see [Electric Sensor](#): Model Library path [COMSOL_Multiphysics/Electromagnetics/electric_sensor](#).

Cylindrical System

A **Cylindrical System** () can be used in 2D and 3D where rotational symmetry about the axis is required. The cylindrical coordinate system is not applicable in geometries with 2D axial symmetry. 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, **a**, and

the radial base vector, \mathbf{e}_r ($\phi = 0$), 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

$$\begin{bmatrix} r \\ \phi \\ a \end{bmatrix} = \begin{bmatrix} \left| \mathbf{r} - (\mathbf{r}_0 + \mathbf{a}(\mathbf{a} \cdot (\mathbf{r} - \mathbf{r}_0))) \right| \\ \text{atan} \frac{(\mathbf{a} \times \mathbf{e}_r) \cdot (\mathbf{r} - \mathbf{r}_0)}{\mathbf{e}_r \cdot (\mathbf{r} - \mathbf{r}_0)} \\ \mathbf{a} \cdot (\mathbf{r} - \mathbf{r}_0) \end{bmatrix}$$

SETTINGS

Frame

Select with respect to which **Frame**—**Spatial** (the default), **Mesh**, **Material**, or **Geometry**—the coordinate system is cylindrical as defined by the above transformations. Note that the actual coordinate names—typically **(x, y, z)** or **(X, Y, Z)**—are displayed for each frame, indicating which frames actually differ from each other in the current model.



A coordinate system with **Frame** set to **Spatial** is orthonormal only in the spatial frame. Similarly, a **Material** system is orthonormal only in the material frame. Some physics require that coordinate systems used are orthonormal in a particular frame. For example, choose the **Material** frame if you want to use the coordinate system in a structural mechanics model.

Coordinate Names

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. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Origin

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.

Longitudinal Axis

For 3D models, enter the **Longitudinal axis direction**. The default is the z direction in the global system.

Direction of Axis

For 3D models, specify the **Direction of axis** $\varphi = 0$, where φ is the azimuthal angle. The default direction is the x direction in the global system.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.



If you have the AC/DC Module and Particle Tracing Module, see [Magnetic Lens: Model Library path ACDC_Module/Particle_Tracing/magnetic_lens](#).

Mapped System

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:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1(x, y, z) \\ u_2(x, y, z) \\ u_3(x, y, z) \end{bmatrix}$$

SETTINGS

The **Frame** list is the same as for the [Cylindrical System](#).

Coordinate Names

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. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Coordinate Mapping

Under **Coordinate mapping**, the **Coordinate** column displays the **Coordinate names** with the **Expression** column displaying the associated mapped coordinate.

Simplifications

If required, select the **Assume orthonormal** check box. The program then uses the assumption that the settings define an orthonormal coordinate system.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.

Rotated System

Use a **Rotated System** () to define rotation about the out-of-plane axis in 2D and Euler angles in 3D.

In the **Rotated System** node's settings window, define the rotation relative to the global Cartesian coordinate system. In 3D models, you specify the local coordinate system (x_l, y_l, z_l) using three consecutive Euler angles (rotation angles) α , β , and γ . See [Figure 4-2](#).

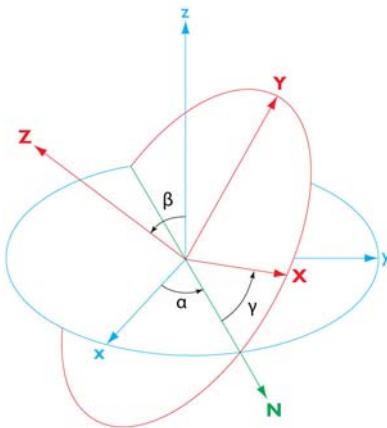


Figure 4-2: 3D Euler angles in a rotated coordinate system.

The transformation matrix for the 3D case is then

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\cos\beta\sin\gamma & -\cos\alpha\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \sin\beta\sin\alpha \\ \sin\alpha\cos\gamma + \cos\alpha\cos\beta\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\beta\cos\gamma & -\sin\beta\cos\alpha \\ \sin\beta\sin\gamma & \sin\beta\cos\gamma & \cos\beta \end{bmatrix}^T \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

In 2D models, you 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.

SETTINGS

Coordinate Names

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. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

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.

Out-of-plane Axis (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.

Euler Angles (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.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.



If you have the MEMS Module, see [Gecko Foot: Model Library path](#)
[**MEMS_Module/Actuators/gecko_foot**](#).

Spherical System

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, **Z**), and the axis $\theta = \pi/2$, $\varphi = 0$ (the azimuth axis, **A**). The direction vectors are automatically normalized.

This is a mapped normalized coordinate system using the following transform in global coordinates

$$\begin{aligned} r &= |\mathbf{r} - \mathbf{r}_0| \\ \theta &= \arccos\left(\frac{\mathbf{Z} \cdot (\mathbf{r} - \mathbf{r}_0)}{|\mathbf{r} - \mathbf{r}_0|}\right) \\ \varphi &= \text{atan2}(\mathbf{r}_{\perp} \cdot (\mathbf{Z} \times \mathbf{A}), \mathbf{r}_{\perp} \cdot \mathbf{A}) \end{aligned}$$

where \mathbf{r}_0 is the position of the origin, **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

$$\mathbf{r}_{\perp} = (\mathbf{r} - \mathbf{r}_0 - \mathbf{Z}(\mathbf{Z} \cdot (\mathbf{r} - \mathbf{r}_0)))$$

SETTINGS

The **Frame** list is the same as for the [Cylindrical System](#).

In the **Coordinate names** table, the default **Coordinate names** are entered—**r**, **theta**, and **phi**. The labels for each coordinate name—**First**, **Second**, and **Third**—include the default name in parentheses.

Enter the location of the **Origin** in the global Cartesian coordinate system. The default is an origin coinciding with that of the global system.

Enter the **Direction of axis θ=0** (the *zenith axis*). The default axis direction is the z direction in the global Cartesian system.

Define the **Direction of axis** $\theta=\pi/2$, $\varphi=0$ (the *azimuth axis*). The default direction is the x direction in the global Cartesian system.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.

Scaling System

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 node.



The Scaling System is only available for physics that support infinite elements or perfectly matched layers. See [Infinite Element Domain](#).

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 are 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 a “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 user 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.

SETTINGS

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.



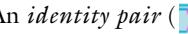
Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.

Identity and Contact 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—identity and contact.

About Identity and Contact Pairs

IDENTITY PAIRS

An *identity pair* () is a pair that by default make the fields across two connected boundaries (one from each connecting part in an assembly) 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 provide special boundary conditions for identity pairs to model “slit conditions” such as resistive layers. 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. The nodes in the **Model Builder** that represent pair boundary conditions use an icon with a pair symbol in the lower-left corner: .

CONTACT PAIRS

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.

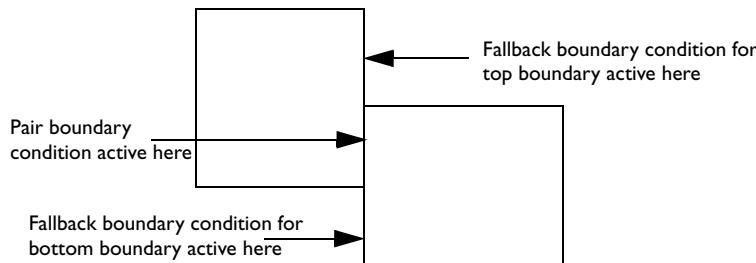


Contact pair modeling requires the Structural Mechanics Module or MEMS Module. Details about this pair type can be found in the respective user guide.

FALLBACK BOUNDARY CONDITIONS ON NON-OVERLAPPING PARTS

For 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 (*fallback 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 is added 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 (**Continuity**, for example) and select any of the standard boundary conditions from the **Fallback Features** submenu. In the settings window, the selection includes all applicable boundaries by default, but a separate boundary condition can be added for only a subset of the pair boundaries. 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. You can right-click the pair's boundary condition node and, from the **Fallback Features** submenu, add one fallback boundary condition for the top boundary and another fallback boundary condition for the bottom boundary if desired.



The options for the available fallback conditions are based on the physics user interface and the license type (see [Figure 4-3](#)).



With only a few exceptions for the Solid Mechanics user interface or other physics user interfaces using Solid Mechanics functionality, all subnodes to pairs are fallback nodes.

When additional fallback feature nodes are added, the node has an indicator in the lower-left corner () identifying it as a fallback feature node.

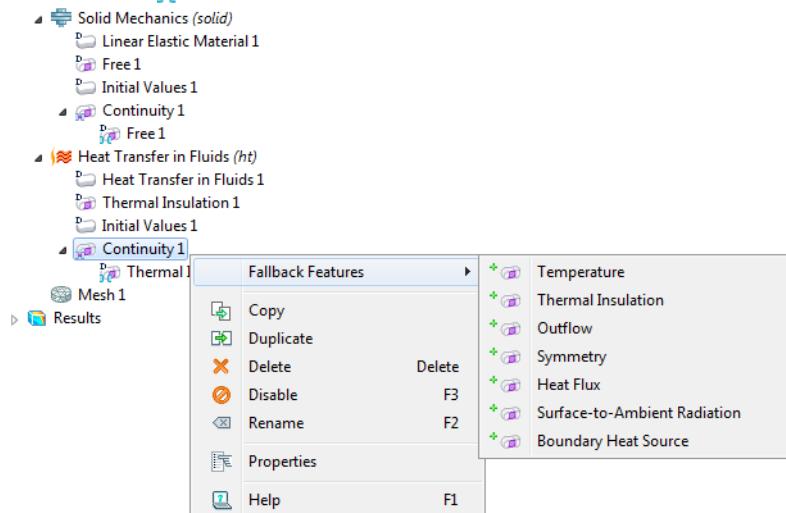


Figure 4-3: An example of the Fallback Features submenu for a Continuity pair added to the Heat Transfer in Fluids interface.

Identity Pair

Use an **Identity Pair** node (for an identity boundary 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 user interface. Identity pairs connect overlapping boundaries in different connecting parts of an assembly.

The **Identity Boundary Pair** () is the most commonly used. For 3D models, the **Identity Edge Pair** is available (, which can be useful for connecting two edges in a shell model, for example. You can also choose the **Identity Point Pair** ().



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Source Boundaries** and **Destination Boundaries** sections.

GENERAL

Enter a **Pair name**. It is used as a suffix in names of operators and variables defined by the pair. The default **Pair type** is **Identity pair**, or select **Contact pair**. Identity and Contact pairs are both available for boundary pairs, and Identity pairs are also available on edge and point levels.

Manual Control of Selections

If the pair was created automatically when forming an assembly, the **Manual control of selections** check box is visible. Click to clear this check box to be able to make manual changes to the **Source Boundaries** and **Destination Boundaries** selections. Pairs in manual mode do not have their selections updated when the geometry sequence is rebuilt.

The operator mapping an expression E on the source side to the destination side is denoted $\text{src2dst}_{pn}(E)$, where $_{pn}$ is the pair name.

For an **Identity Pair**, 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} .

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} .

SOURCE BOUNDARIES AND DESTINATION BOUNDARIES

The destination 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.

For **Source Boundaries**, click the **Activate Selection** button () to define the source boundaries, and similarly for **Destination Boundaries**.

Select **Manual** or **All boundaries** for the boundaries on the source side or the destination side. If **Manual** is selected, click in the **Graphics** window to add boundaries to the

Selection section. If required, click the **Swap Source and Destination** button () to swap the source boundaries and the destination boundaries.



For **Identity Edge Pairs** and **Identity Point Pairs**, edges and points, respectively, replace boundaries in the selections of the pair's source and destination.

FRAME

If there are several frames in the model, the **Frame** section is visible. Select the **Source frame** and the **Destination frame**. Source and destination points are connected if their coordinates in their respective frames are equal.



Thin-Layer Diffusion: Model Library path
COMSOL_Multiphysics/Diffusion/thin_layer_diffusion

Contact Pair

Use a **Contact Pair** node () to specify two selections of boundaries that cannot penetrate each other under deformation. The contact pairs define 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.

GENERAL

This section is described for the **Identity Pair** except that the default **Pair type** is **Contact pair**.

SOURCE BOUNDARIES AND DESTINATION BOUNDARIES

The contact algorithm constrains the destination boundaries so that they do not penetrate the source boundaries.

Click the **Activate Selection** button () to define the source or destination boundaries. Select **Manual** or **All boundaries** for the boundaries on the source or destination side. If **Manual** is selected, click in the **Graphics** window to add boundaries to the **Selection** section. If required, click the **Swap Source and Destination** button () to swap the source boundaries and the destination boundaries.

ADVANCED

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.

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. If the **Manual control of search distance** check box is selected, enter a different value in the **Distance** field.



For a contact pair, the fallback boundary condition is applied to all parts of the boundaries currently not in contact.

- 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**.

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. You can probe while solving, as a monitor and diagnostic tool, and probe after the computation is finished for results analysis. On top of this functionality, a probe variable with entire model scope is also defined. This variable can be used as any other variable in, for example, equations, boundary conditions, or 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 4-16](#) 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 **Table** 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 give access to 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 4-16: PROBE TYPES AND ICONS BY SPACE DIMENSION

PROBE TYPE	3D	2D	1D
Domain			
Boundary			
Edge		—	—
Domain Point			—
Boundary Point		—	—
Global Variable			



Getting Results While Solving

Domain Probe, Boundary Probe, and Edge Probe

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).



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Source Selection**, **Expression**, and **Table and Window Settings** sections.

PROBE SETTINGS

Select an option from the **Type** list—**Average** (the default), **Maximum**, **Minimum**, or **Integral** depending on what type of value takes over the domain, boundary, or edge that you want the probe to compute and output. If needed, enter or edit a name for

the **Probe variable**. The defaults are `dom1` for a Domain Probe, `bnd1` for a Boundary Probe, and `edge1` for an Edge Probe.

INTEGRATION SETTINGS

If you have selected **Average** or **Integral** from the **Type** list, the **Integration Settings** section contains the following settings:

- From the **Method** list, select **Integration** (the default) or **Summation**. Only reaction forces use the summation method.
- Enter an integer value in the **Integration order** field (default: 4).

If you have selected **Maximum** or **Minimum** from the **Type** list, the **Integration Settings** section contains an **Element refinement** field, where you can enter the element refinement (number of partitions of an element edge) to control the accuracy of the maximum or minimum value (default value: 4).

When working with multiple frames for any type of probe, you can also select a **Frame—Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics, for the volume element to be used in the integration.

-
- 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 and for a boundary probe example, see [Snap Hook](#): Model Library path **Nonlinear_Structural_Materials_Module/Plasticity/snap_hook**.
-

Domain Point Probe

Use a **Domain Point Probe** () to monitor the development of a real or complex-valued number from a dynamic simulation (a time-dependent, frequency-domain, or parametric study). By default a **Point Probe Expression** () subnode is added, or right-click **Domain Point Probe** to add additional nodes.

POINT SELECTION

When working with multiple frames, select a **Frame**—**Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics.

For 3D models, select a **Line entry method**—**Point and surface normal** (the default), **Point and direction**, **Two points**, or **None**.

For all models, enter **Coordinates**—Enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D). Also select the **Snap to closest boundary** check box to snap the selected points to the grid.

For **Point and surface normal**, **Point and direction**, or **Two points**, enter a **Depth along line** or use the slider to select a value between 0 and 1 to determine the probe location along the line anywhere from the starting point (0) to the ending point (1).

- For **Point and surface normal** 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.
- For **Point and direction**, 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).

For **Two points**, from the **Point being modified** list, also 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.



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

Boundary Point Probe

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). By default a **Point Probe Expression** () subnode is added, or right-click **Boundary Point Probe** to add additional nodes.

BOUNDARY SELECTION

Select a single boundary to add to the **Selection**.

POINT SELECTION

When working with multiple frames, select a **Frame—Spatial (x,y,z)**, **Material**, **Geometry**, or **Mesh**, followed by the coordinate names: **(x, y, z)** or **(X, Y, Z)** depending on the physics.

Enter the **Coordinates**. A red dot indicates the position of the point on the selected surface in the **Graphics** window. Click 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.

Point Probe Expression

A **Point Probe Expression** () is automatically added as a subnode to a **Domain Point Probe** and a **Boundary Point Probe**. Right-click the main node to add additional **Point Probe Expression** subnodes. Under **Probe Settings**, edit or enter a name for the **Probe variable**. The default name is **ppb1**.

-
- 
 - Go to [Source and Destination Vertices and Boundaries](#) for information about the **Expression** and **Table and Window Settings** sections.
 - [Domain Probe, Boundary Probe, and Edge Probe](#)
-

Global Variable Probe

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). Under **Probe Settings**, edit or enter a name for the **Probe variable**. The default name is **var1**.

-
- 

Go to [Source and Destination Vertices and Boundaries](#) for links to information about the **Expression** and **Table and Window Settings** sections.
-

If you have the Plasma Module or MEMS 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**.
 - See [Transient Response of a Biased Resonator—2D](#): Model Library path **MEMS_Module/Actuators/biased_resonator_2d_transient**.
-



Infinite Element Domains and Perfectly Matched Layers

Simulation of Infinite Domains

Simulation of unbounded or infinite domains is a challenge encountered in many types of physics. Normally, any physics simulates a process within a bounded domain represented by the geometry drawn in, or imported into, COMSOL. But in many cases the domain is delimited 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 of infinite extent.

Artificial truncation of the domain can be handled in a number of different ways. Some physics user 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 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 toward 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 so-called *perfectly matched layers* (PMLs).

Because of their common background as coordinate stretching, infinite elements and PMLs in COMSOL share a number of important properties. 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 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.



It is important that separate, normally quadratic, domains are drawn at the corners.

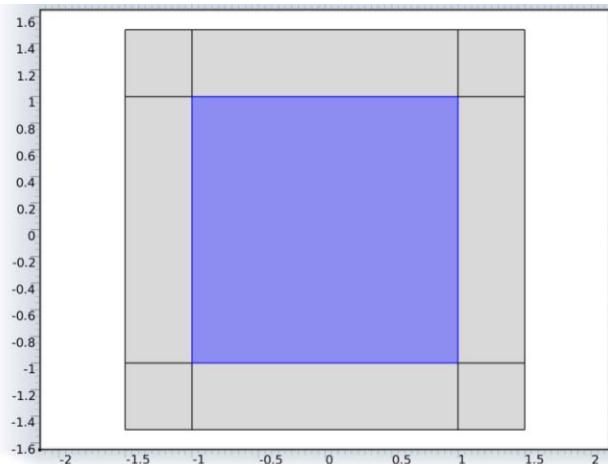


Figure 4-4: Typical Cartesian scaling configuration. Note the distinction between edge and corner domains.

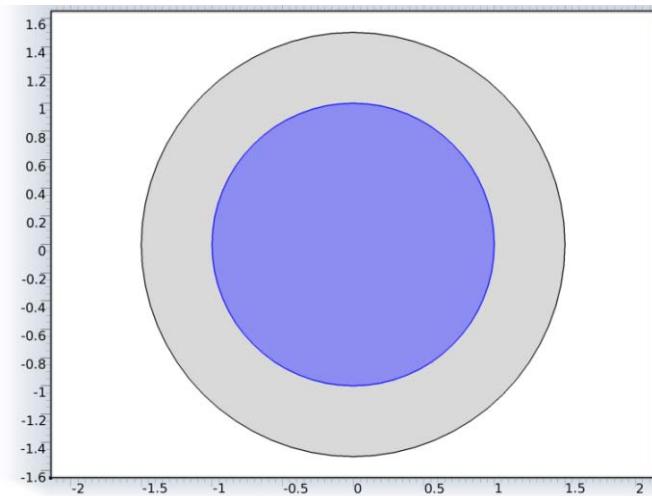


Figure 4-5: Example of cylindrical scaling configuration in plane 2D. You must specify the center point of the model when different from the origin of the coordinate system.

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, behaves identically to the plane 2D Cartesian option. Similarly, the axisymmetric spherical

scaling is similar to plane 2D cylindrical scaling, except that it is always centered on the axis.

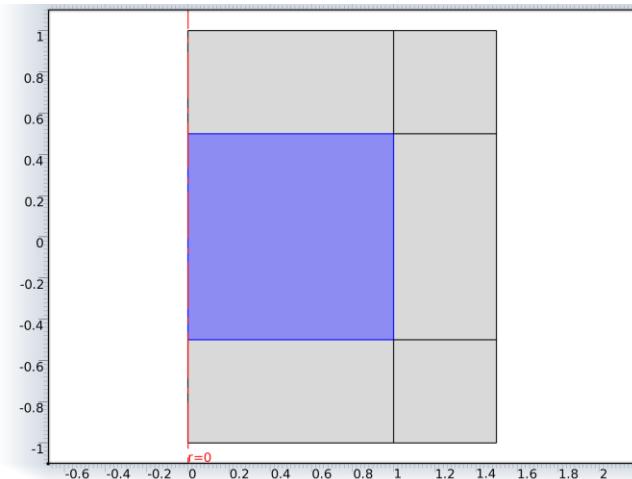


Figure 4-6: Axisymmetric cylindrical scaling uses domains of three distinct types: with radial stretching, with axial stretching, and with both radial and axial stretching. The latter are the corner zones, which must be drawn as distinct domains.

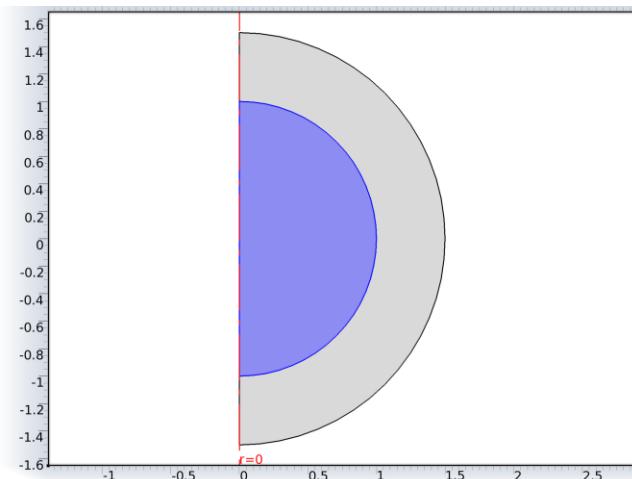


Figure 4-7: Axisymmetric spherical scaling assumes radial stretching in an annulus centered at a point on the axis. If the centerpoint is not the origin of the coordinate system, you must specify its axial position.

3D MODELS

The available scaling types in 3D are Cartesian and spherical. The Cartesian scaling domains are of three different types. Depending on 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.

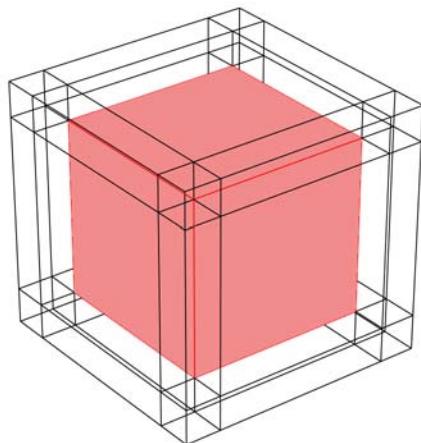


Figure 4-8: Typical Cartesian scaling system configuration. Note the distinction between face, edge, and corner domains.

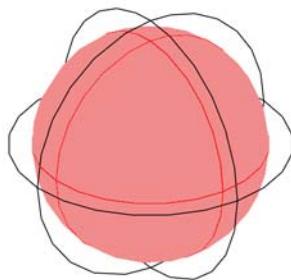


Figure 4-9: Typical spherical scaling system configuration.

Note on Availability

Infinite elements and perfectly matched layers are available only for some physics and when COMSOL is used together with certain add-on modules. If you have not added any physics that is compatible with infinite elements or perfectly matched layers under the available licenses, you cannot add such features to the model.

Further, after turning a domain into a PML or infinite element, that domain is not allowed in the active selection of physics user interfaces and individual nodes that 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.

The Challenge of Open Boundaries for PMLs in Problems



The **Perfectly Matched Layer** node is added under the **Model>Definitions** node. It is not specific to a certain physics interface, but introduces the coordinate transform in that domain for all physics.

For the RF Module and Structural Mechanics Module, one of the challenges in finite element modeling is how to treat open boundaries in radiation problems (RF) wave-propagation (Structural Mechanics). One option is to use *perfectly matched layers* (PMLs). This module offers two closely related types of *absorbing boundary conditions*, the *scattering boundary condition* and the *port boundary condition*. The scattering boundary condition is a first order absorbing boundary condition for a plane wave or (optionally) a cylindrical or spherical wave, whereas a port boundary condition is a perfectly absorbing condition for general modes of a known shape, provided that the correct mode shape and the propagation constant are supplied.

In many scattering and waveguide-modeling problems, you cannot describe the incident radiation as a plane wave with a well-known direction of propagation. In such situations, consider using *perfectly matched layers* or PMLs.

A PML is strictly speaking not a boundary condition but an additional domain that absorbs the incident radiation without producing reflections. It provides good performance for a wide range of incidence angles and is not particularly sensitive to the shape of the wave fronts. The PML formulation can be deduced from Maxwell's equations by introducing a complex-valued coordinate transformation under the additional requirement that the wave impedance should remain unaffected ([Ref. 3](#)).

This section describes how to use semiautomatic frequency-domain PMLs in COMSOL to create planar, cylindrical, and spherical PMLs. Transient PMLs are not supported.

For the RF Module, several port boundary conditions representing an expansion into mutually orthogonal modes are also allowed and can be used to account for higher diffraction orders from a grating or to truncate a waveguide operated in the multimode regime. However, in many scattering and antenna-modeling problems, the outgoing radiation cannot be described as a plane wave with a well-known direction of propagation or as a known, finite modal expansion.

For the Acoustics Module, MEMS Module, or Structural Mechanics Module, the PMLs damp a certain wavelength existing in the system. This wavelength is deducted from the frequency and a reference wave speed c_{ref} . The wave speed is defined under **Typical Wave Speed** in the settings window for the main physics node. For the Acoustics Module, the speed c_{ref} should be set equal to the compressional speed of sound of the material in the PML. If the PML spans both a solid and a fluid domain try to set an average value. In most acoustic interfaces the reference speed is set to 343 m/s, while in the Elastic Waves and Poroelastic Waves user interfaces, the compressional speed of sound for the solid material is used as an initial guess.

PML Implementation

For the RF Module and Structural Mechanics Module, the PML implementation uses the following coordinate transform for the general coordinate variable t :

$$t' = \left(\frac{t}{\Delta_w} \right)^n (1 - i) \lambda F \quad (4-1)$$

The coordinate, t , and the width of the PML region, Δ_w , are geometrical parameters that are automatically extracted for each region. The other parameters are the PML scaling factor F and PML order n , which can be modified in the PML (both default to unity).

To avoid a nonlinear dependence in the eigenvalue, the wavelength, λ , is removed from the scaling expression when computing an eigenfrequency study.

The software automatically computes the value for Δ_w and the orientation of the transform for PML regions that are Cartesian, cylindrical, or spherical.

 There is no check that the geometry of the region is correct, so it is important to draw a proper geometry and select the corresponding region type.

For the MEMS Module and Acoustics Module, the *rational* coordinate stretching (Acoustics) and *polynomial* (MEMS) coordinate stretching type PMLs, which absorb waves in the coordinate direction ξ , use the following coordinate transformation inside the PML domain:

$$\xi' = \text{sign}(\xi - \xi_0) |\xi - \xi_0|^n \frac{L}{\delta\xi^n} (1 - i) \quad (4-2)$$

For each orthogonal absorbing coordinate direction, the software automatically determines the coordinate of the inner PML boundary, ξ_0 , and the (actual) width of the PML, $\delta\xi$. The PML scaling factor, $L/\delta\xi$, and the PML order, n , are input parameters for each PML domain.

The default value for L is one wavelength, $\lambda = c_{\text{ref}}/f$, which is appropriate for acoustic waves propagating along the absorbing coordinate direction, ξ . To preserve the attenuation level for obliquely incident waves, adjust the scaled PML width; for a wave with wave vector \mathbf{k} , the optimal value for L in the coordinate direction ξ is $2\pi/|\mathbf{k} \cdot \mathbf{e}_\xi| = \lambda/|\cos \theta|$, where \mathbf{e}_ξ is a unit vector in the ξ direction, and θ is the angle between \mathbf{k} and \mathbf{e}_ξ . Thus, multiply the default value for the scaled PML width by the factor $|\cos \theta|^{-1}$.

 If you increase L , make sure that the mesh resolution is sufficient to resolve the number of wavelengths that fit inside the adjusted scaled width.

The default value of the scaling exponent is 1, which gives a linear scaling that works well in most cases. For scattering problems and models where different wavelengths should be absorbed (outside waveguides, for example) increase the exponent somewhat; the useful range for n is roughly 1 to 2. Increasing the exponent uses fewer mesh elements to resolve wavelengths much smaller than the scaled PML width.

The parameters ξ_0 and $\delta\xi$ have default settings that the software deduces from the drawn geometry and stores in variables referred to as guess variables.

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

- *Cartesian*—PMLs absorbing in Cartesian coordinate directions. It is available in 2D and 3D ([Figure 4-10](#)).
- *Cylindrical*—PMLs absorbing in cylindrical coordinate directions from a specified axis. It is available in 3D, 2D, and 2D axisymmetry. In axisymmetry, the cylinder axis is the z -axis ([Figure 4-11](#)).
- *Spherical*—PMLs absorbing in the radial direction from a specified centerpoint. It is available in 2D axisymmetry and 3D ([Figure 4-12](#)).
- *General*—PMLs or domain scaling with user-defined coordinate transformations.

GENERAL SCALING

With manual control of the scaling, the geometric parameters that define the stretching are added as Manual Scaling subnodes. These subnodes have no effect unless the type of the Perfectly Matched Layers node is set to General. Each Manual Scaling subnode has three parameters: the scaling direction \mathbf{a}_{rot} , the geometric width Δ_r , and the coordinate at interface \mathbf{r}_I . The first parameter sets the direction from the interface to the outer boundary, the second parameter sets the width of the region, and the last parameter sets an arbitrary coordinate at the interface. When going from any of the other types to the general type, Manual Scaling subnodes are automatically added to represent stretching of the previous type.

For each of the these PML types, choose the coordinate directions in which the PML absorbs waves, that is, for which directions a coordinate transformation is applied.

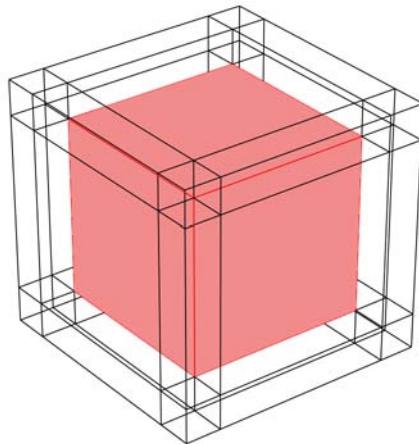


Figure 4-10: A cube surrounded by typical PML regions of the type “Cartesian.” Notice that you need separate PML domains for each direction, including separate PML or infinite element domains at the corners. This is necessary for achieving the correct scaling in the different directions.

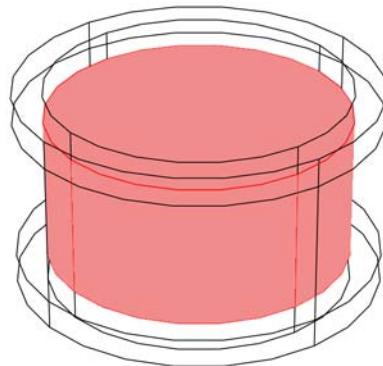


Figure 4-11: A cylinder surrounded by typical cylindrical PML or infinite element domain regions.

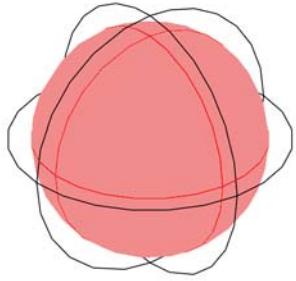


Figure 4-12: A sphere surrounded by a typical spherical PML or infinite element domain region.

The same requirements for the PML domain partitioning apply to planar and axisymmetric 2D geometries. Keep in mind that the scaling directions cannot change within a PML domain, so it is necessary to partition the PML domain in parts in which the scaling direction is constant.

Perfectly Matched Layer

Right-click the **Definitions** node to add a **Perfectly Matched Layer** node () and 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 an **Infinite Element Domain**. 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). The default **Coordinate System Identifier** is pm11. Except for the **Scaling** section, the rest of the settings are the same as for **Infinite Element Domain**.

SCALING

Select a **Coordinate stretching type**—**Polynomial** (the default), **Rational**, **Elliptic**, or **Logarithmic**.

Select an option from the **Typical wavelength from** list—**Physics interface** (the default) or **User defined**. If **Physics interface** is selected, select an option from the **Physics** list. If

User defined is selected, enter a value or expression for the **Typical wavelength**. The default is 1.

Enter a value or expression for the **PML scaling factor** and the **PML scaling curvature parameter**. The defaults are 1 for both.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.



If you have the Acoustics Module, see the Model Library:

- [Cylindrical Subwoofer](#): Model Library path
Acoustics_Module/Tutorial_Models/cylindrical_subwoofer
- [Acoustic Scattering off an Ellipsoid](#): Model Library path
Acoustics_Module/Tutorial_Models/acoustic_scattering



If you have the RF Module, see the Model Library:

- Tutorial Models/[Radar Cross Section](#) (2D, cylindrical PML).
- Tutorial Models/[RF Coil](#) (3D, spherical PML with swept mesh).

Known Issues When Modeling Using PMLs

When modeling with PMLs be aware of the following:

USE OF ONE SINGLE INFINITE ELEMENTS NODE

A separate Perfectly Matched Layers node must be used for each isolated PML domain. That is, to use one and the same Perfectly Matched Layers node, all PML domains must be in contact with each other. Otherwise the PMLs do not work properly.

ELEMENT QUALITY

The coordinate scaling resulting from PMLs also yields an equivalent scaling of the mesh that may effectively result in a poor element quality. (The element quality displayed by the mesh statistics does not account for this effect.) This typically happens when the geometrical thickness of the PML deviates much from one wavelength (local wavelength rather than free space wavelength). The poor element quality causes poor convergence for iterative solvers and make the problem ill-conditioned in general.

For the RF Module, especially *vector element* formulations (the ones using two or more components of a vector field variable) are sensitive to low element quality. For this reason, it is strongly recommended to use swept meshing in the PML domains. The sweep direction should be selected the same as the direction of scaling. For Cartesian PMLs and regions with more than one direction of scaling it is recommended to first sweep the mesh in the domains with only one direction of scaling, then sweep the domains with scaling in two directions, and finish by sweeping the mesh in the domains with PML scaling in all three directions.

COMPLICATED EXPRESSIONS

The expressions resulting from the stretching get quite complicated for spherical PMLs. This increases the time for the assembly stage in the solution process. After the assembly, the computation time and memory consumption is comparable to a problem without PMLs. The number of iterations for iterative solvers might increase if the PML regions have a coarse mesh.

ERRONEOUS RESULTS

PML regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the PML parameter to give erroneous result. Enter the parameter values manually if necessary.

USE THE SAME MATERIAL PARAMETERS OR BOUNDARY CONDITIONS

The PML region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an PML region. Also the PML scaling does probably not work for complex shapes that deviate significantly from the shapes shown here.

Modeling Unbounded Domains

Many environments are unbounded or open, meaning that the fields extend toward infinity. The easiest approach to modeling an unbounded domain is to extend the simulation domain “far enough” that the influence of the terminating boundary conditions at the far end becomes negligible. This approach can create unnecessary mesh elements and make the geometry difficult to mesh due to large differences between the largest and smallest object.

Another approach is to use *infinite elements*. There are many implementations of infinite elements available, and the elements used in this module are often referred to as *mapped infinite elements* (see Ref. 1). This implementation maps the model

coordinates from the local, finite-sized domain to a stretched domain. The inner boundary of this stretched domain coincides with the local domain, but at the exterior boundary the coordinates are scaled toward infinity.

The principle can be explained in a one-coordinate system, where this coordinate represents Cartesian, cylindrical, or spherical coordinates. Mapping multiple coordinate directions (for Cartesian and cylindrical systems only) is just the sum of the individual coordinate mappings.

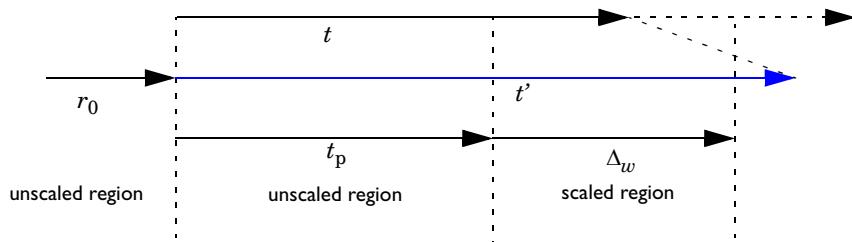


Figure 4-13: The coordinate transform used for the mapped infinite element technique. The meaning of the different variables are explained in the text.

Figure 4-13 shows a simple view of an arbitrary coordinate system. The coordinate r is the unscaled coordinate that COMSOL draw the geometry in (reference system). The position r_0 is the new origin from where the coordinates are scaled, t_p is the coordinate from this new origin to the beginning of the scaled region also called the *pole distance*, and Δ_w is the unscaled length of the scaled region. The scaled coordinate, t' , approaches infinity when t approaches $t_p + \Delta_w$. To avoid solver issues with near infinite values, it is possible to change the infinite physical width of the scaled region to a finite large value, Δ_{pw} . The true coordinate that the PDEs are formulated in is given by $r' = r_0 + t'$ where t' comes from the formula

$$t' = t_p \frac{\Delta_w}{\Delta_p - \gamma(t - t_p)}$$

$$\gamma = 1 - \frac{t_p}{\Delta_{pw} - t_p}$$

$$t' = t_0 \frac{\delta t}{t_0 + \delta t - t}$$

The inner coordinate, t_0 , and the width of the infinite element region, δt , are input parameters for each region. COMSOL uses default values for these properties for

geometries that are Cartesian, cylindrical, or spherical. However, for complex geometries, it might be necessary to define other parameters using a general mapping.

The pole distance, t_p , and the physical width of the infinite element region, Δ_{pw} , are input parameters for the region. The software automatically computes the transform for infinite element regions that are Cartesian, cylindrical, or spherical.

 There is no check that the geometry of the region is correct, so it is important to draw a proper geometry and select the corresponding region type.

The same requirements for the infinite-element domain partitioning apply to planar and axisymmetric 2D geometries. Keep in mind that the scaling directions cannot change within an infinite-element domain, so it is necessary to partition the infinite-element domain in parts in which the scaling direction is constant.

If other shapes are used for the infinite element regions not similar to the shapes in the previous figures, you might have to define the infinite element parameters manually.

GENERAL STRETCHING

With manual control of the stretching, the geometrical parameters that defines the stretching are added as **Manual Scaling** subnodes. These subnodes have no effect unless the type of the **Infinite Elements** node is set to **General**. Each **Manual Scaling** subnode has three parameters:

- Scaling direction, which sets the direction from the interface to the outer boundary.
- Geometric width, which sets the width of the region.
- Coordinate at interface, which sets an arbitrary coordinate at the interface.

When going from any of the other types to the General type, subnodes that represent stretching of the previous type are added automatically.

Infinite Element Domain

Right-click the **Definitions** node to add an **Infinite Element Domain** node () and 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, and create a **Perfectly Matched Layer** (PML). The default

Coordinate System Identifier is ie1.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

GEOMETRY

Select a **Type**—**Cartesian** (the default), **Spherical**, or **Cylindrical**.

- If **Spherical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)** in the table. For axisymmetric models, only the *z* coordinate is required.
- If **Cylindrical** is selected, enter **Center coordinate** values for **x (m)**, **y (m)**, and **z (m)**. For 3D models, also enter **Center axis direction** values for **x**, **y**, and **z** in each table.

SCALING

Enter expressions for the following:

- **Physical width** (SI unit: m). The default variable is `1e3*dGeomChar`.
- **Pole distance** (SI unit: m). The default variable is `dGeomChar`.



Go to [Source and Destination Vertices and Boundaries](#) for a link to information about the **Coordinate System Identifier** section.

Known Issues When Modeling Using Infinite Elements

Be aware of the following when modeling with infinite elements:

USE OF ONE SINGLE INFINITE ELEMENTS NODE

Use a separate Infinite Elements node for each isolated infinite element domain. That is, to use one and the same Infinite Elements node, all infinite element domains must be in contact with each other. Otherwise the infinite elements do not work properly.

ELEMENT QUALITY

The coordinate scaling resulting from infinite elements also yields an equivalent stretching or scaling of the mesh that effectively results in a poor element quality. (The element quality displayed by the mesh statistics does not account for this effect.)

The poor element quality causes poor or slow convergence for iterative solvers and make the problem ill-conditioned in general. For the AC/DC Module, especially

vector element formulations like the ones using two or more components of the magnetic vector potential are sensitive to low element quality.

For this reason, it is strongly recommended to use swept meshing in the infinite element domains. Select the sweep direction should to be the same as the direction of scaling. For Cartesian infinite elements in regions with more than one direction of scaling it is recommended to first sweep the mesh in the domains with only one direction of scaling, then sweep the domains with scaling in two directions, and finally sweep the mesh in the domains with infinite element scaling in all three direction.

C O M P L I C A T E D E X P R E S S I O N S

The expressions resulting from the stretching get quite complicated for spherical infinite elements in 3D. This increases the time for the assembly stage in the solution process. After the assembly, the computation time and memory consumption is comparable to a problem without infinite elements. The number of iterations for iterative solvers might increase if the infinite element regions have a coarse mesh.

E R R O N E O U S R E S U L T S

Infinite element regions deviating significantly from the typical configurations shown in the beginning of this section can cause the automatic calculation of the infinite element parameter to give erroneous result. Enter the parameter values manually if this is the case. See [General Stretching](#).

U S E T H E S A M E M A T E R I A L P A R A M E T E R S O R B O U N D A R Y C O N D I T I O N S

The infinite element region is designed to model uniform regions extended toward infinity. Avoid using objects with different material parameters or boundary conditions that influence the solution inside an infinite element region.

References for PMLs and Infinite Element Domains

1. O.C. Zienkiewicz, C. Emson, and P. Bettess, “A Novel Boundary Infinite Element,” *International Journal for Numerical Methods in Engineering*, vol. 19, no. 3, pp. 393–404, 1983.
2. J.P. Bérenger, “A Perfectly Matched Layer for the Absorption of Electromagnetic Waves,” *J. Comput. Phys.*, vol. 114, pp. 185–200, 1994.
3. Jianming Jin, *The Finite Element Method in Electromagnetics*, 2nd ed., Wiley-IEEE Press, 2002.

5

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.

Selecting and Visualizing in Models

About Viewing and Selecting in the Graphics Window

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 ways to select each part of the geometry and geometric entities, and to 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 [Named Selections](#) to reuse throughout the model when assigning material properties, boundary conditions, and other model settings.
- 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 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 [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 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*. Those entities are volumes, surfaces, curves, or points. [Table 5-1](#) summarizes the terms used in COMSOL Multiphysics.

Geometric entities of the maximum dimension are called *domains*, while those of the next highest dimension 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 can 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.

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 5-1](#)). This enables visualization of a cube by displaying one or more of these four types. For instance, you can create a wireframe plot by rendering only the cube's edges.

TABLE 5-1: 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.

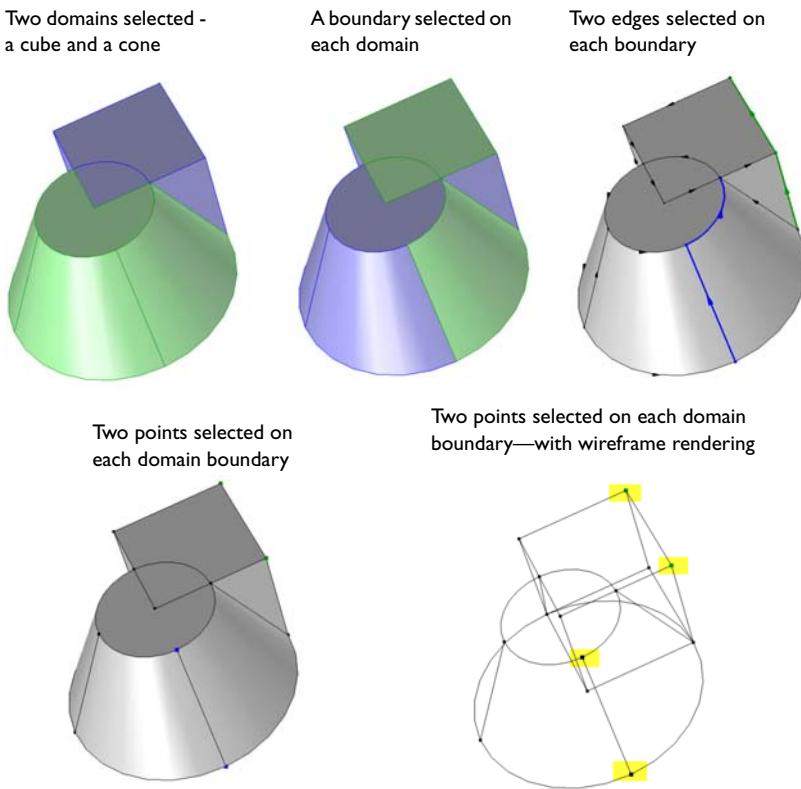


Figure 5-1: A 3D geometry comprises domains, boundaries, edges, and points.

About Selecting Geometric Entities

Throughout COMSOL Multiphysics there are many lists of selected geometric entities, all based on the same principle—pick a domain, boundary, edge, or point and use methods to add or remove these geometric entities to create *selections* that define, for example, the parts of the geometry where a material or boundary condition is active. Such lists appear in settings 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.

THE GEOMETRY ENTITY SELECTION SECTIONS

The name of the section where you manage the list of selected geometric entities depends on the geometric entity level:

- **Geometric Entity Selection:** For **Material** and **Variables** nodes, where you first select the level (domain, boundary, and so on), from a **Geometric entity level** list.
- **Domain Selection:** For nodes that define, for example, material models, sources, and body loads in domains.
- **Boundary Selection:** For nodes that define, for example, boundary conditions.
- **Edge Selection:** For nodes that define, for example, conditions and forces on edges.
- **Point Selection:** For nodes that define, for example, point sources and point loads.

From the **Selection** list you can choose one of the following options:

- **Manual** (the default): You can pick the geometric entities directly in the **Graphics** window, using the **Selection List** window, or using the **Paste Selection** button. See below for more information about those selection methods.
- **All domains, All boundaries, All edges, All points:** Depending on the geometric entity level, you can choose one of these options to select all entities.
- Defined named selections: Selection nodes that you add in the geometry sequence or under **Definitions** (as well as selections created from Boolean operations, for example) are available in the **Selection** lists for nodes that define model properties for the same geometric entity level. You can rename such selection nodes to better reflect what the selected entities represent. A named selection can consist of, for example, the domains where a volume force acts, the boundaries where an inflow occurs, or points that are grounded. Names selections are useful for reusing selections in a model and to clearly indicate what parts of geometry that the selected entities include or represent. See [Creating Named Selections](#).



For physics nodes that are default nodes (see [Physics Default Nodes](#)) the selection defaults to all geometric entities on the applicable level (all domains or all boundaries, for example), and the **Selection** list is not available. You can add other nodes that override the default nodes for some or all entities. Those entities are then marked (**overridden**) in the list.

In the lists of selected entities, (**overridden**) and (**not applicable**) can appear next to the label (number) of a selected entity. See [Physics Node Status](#) for more information about these status indicators. There is also an **Override and Contribution** section in all

physics nodes. It provides an overview of how the physics nodes and their selections interact. See [Physics Exclusive and Contributing Node Types](#).

PAIR SELECTION

If the geometry is an assembly you have access to specific pair conditions (typically on boundaries) that you choose from a **Pairs** submenu on the main physics nodes' context menus. In the settings window for such pair nodes, a **Pair Selection** section contains a list of the applicable pairs (typically identity pairs). The **Boundary Selection** section (or another standard selection section) is then unavailable and shows the entity numbers for the boundaries, for example, that the selected pairs include. You specify for which pairs the pair condition is applied by selecting one or more pairs from the **Pairs** list.

TOOLS FOR VIEWING AND SELECTING GEOMETRIC ENTITIES

There are toolbar buttons, mouse click options, window 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. All levels of geometry can be treated individually. See [Table 5-5](#) for a list of different ways to complete the same task.

[Table 5-2](#) lists the buttons that display on every settings window that has a geometric entity selection list as displayed in [Figure 5-5](#).

TABLE 5-2: 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 model properties. See Pasting Lists of Geometric Entities to Create Selections .
	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 modeling instruction. See Pasting Lists of Geometric Entities to Create Selections .
	Zoom Selected	Use this button to zoom into the selected geometric entity.

TABLE 5-2: 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 5-5 .
	Remove from Selection	Use this button to remove a geometric entity from the Selection list. See Table 5-5 .
	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 highlights geometric entities at different stages of selection. Selected lines also have a thicker line width and selected points are larger. In [Figure 5-1](#), 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).

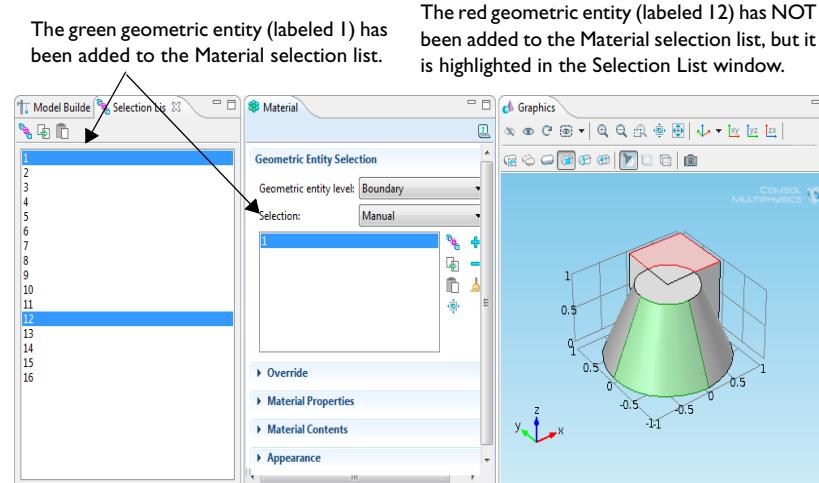


Figure 5-2: Selecting geometric levels, highlighted to indicate status.

Red The geometric entity is highlighted but not added to the selection list. Left-click once with the mouse. If you right-click it locks all red geometric entities. See [Figure 5-2](#) for an example.

Blue The geometric entity has been added to the model selection list. The selection is added to the selection list and highlighted in blue after you right-click. See [Figure 5-1](#) for an example.

Green After the geometric entity is added to the selection list, when you click anywhere (directly in the Graphics window, in the settings window, or in the Selection List window) it is highlighted green and the name (number) of the geometric entity in the selection lists is also highlighted. See [Figure 5-1](#) and [Figure 5-2](#) 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 entity the farthest away.

Pasting Lists of Geometric Entities to Create 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 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 5-3](#), [Figure 5-7](#), and [Figure 5-8](#)). 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 such as COMSOL Multiphysics model documentation (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.

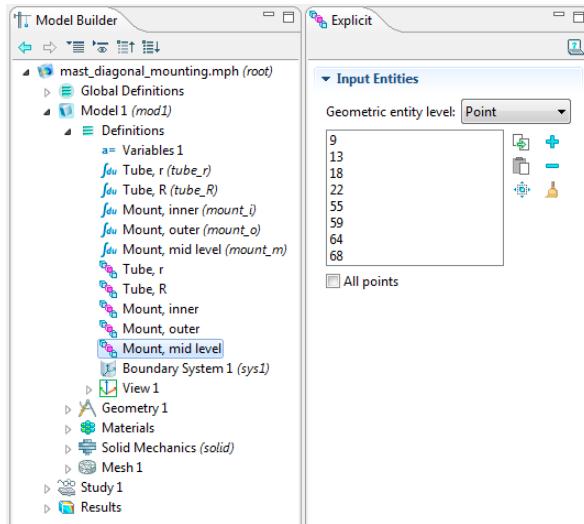


Figure 5-3: An example of an *Explicit* selection window for the *Mount, mod* level node. The Selection nodes under Definitions are renamed by the user.

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, which are also based on the model space dimension as shown in [Table 5-3](#). The buttons in [Table 5-4](#) are available in any space dimension.



Physics Nodes by Space Dimension

TABLE 5-3: GRAPHICS TOOLBAR BUTTONS BY SPACE DIMENSION

BUTTON NAME	3D	2D AND 2D AXISYMMETRIC	ID AND ID AXISYMMETRIC
Select Boundaries			—
Select Box			—
Select Domains			—
Select Objects			—
Select Points			—
Select Edges		—	—
Go to XY View		—	—
Go to YZ View		—	—
Go to ZX View		—	—
Scene Light		—	—
Transparency		—	—
Wireframe Rendering		—	—
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 a 2D Work Plane in 3D](#).

TABLE 5-4: GRAPHICS TOOLBAR BUTTONS AVAILABLE FOR ALL SPACE DIMENSIONS

BUTTON	NAME	SEE ALSO
	Hide Selected	Hiding and Showing Geometric Entities
	Show Selected	
	Reset Hiding	
	View All	
	View Hidden Only	
	View Unhidden	
	Zoom In	Zooming In and Out in the Graphics Window
	Zoom Out	
	Zoom Selected	
	Zoom Extents	
	Go to Default View	Changing Views in the Graphics Window
	Image Snapshot	Capturing and Copying Screenshots

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 Multiphysics 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 selection lists in settings windows, which contain lists of the selected entities only. Click on any item to see it highlighted in the Graphics window—except if the item is hidden, which is indicated in the Selection List by **(hidden)**—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, `b1k1 (solid)` and `cone1 (solid)`.
- When in the Model Builder under the **Mesh** node ()**,** the list includes also information on which entities are meshed by adding **(meshed)** to the right of the meshed entities. If the Geometry has Mesh Construction entities, the list also specifies if a construction entity has been removed; see [Mesh Control Entities](#). This is indicated next to the entity in the list by **(meshed and removed)**.
- 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 5-4](#).
- When working in the Geometry node with *Chamfer* or *Fillet* nodes 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**.

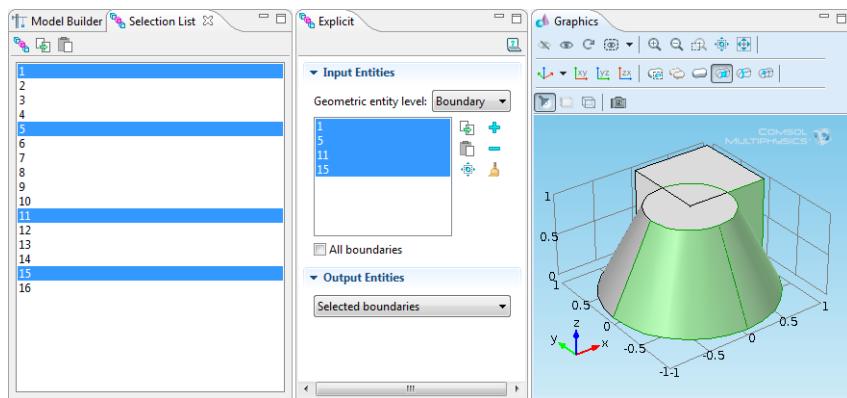


Figure 5-4: Using a combination of the Selection List and Selection window to choose boundaries on a 3D model.

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 5-5](#) 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 5-4](#) above and [Figure 5-3](#) in the [Named Selections](#) section for more examples and comparisons of the different Selection types.

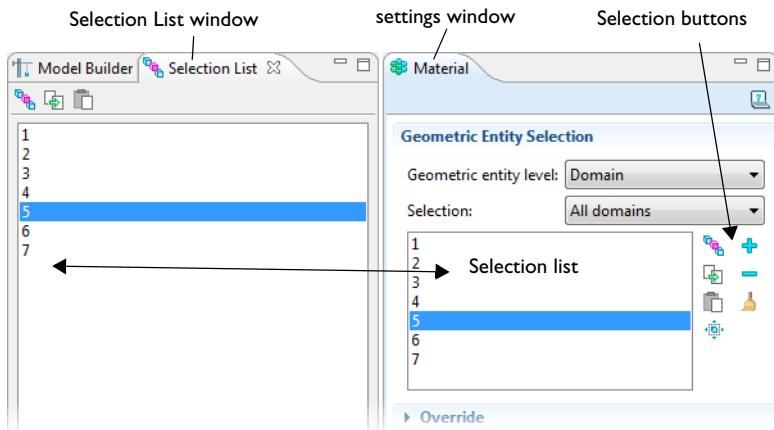


Figure 5-5: Selection List window compared to the Selection list in the settings window.

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 1** 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 boundaries in the geometry. There are many ways to select the boundaries:

- Click on any boundary number 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



The same methods are used when working with 2D geometric entities except there are no edges.

You can 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 overlapping objects. When an object is locked (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 boundaries. If you make any changes to this list, for example, remove a boundary, the **Selection** list reverts to **Manual**.

See Table 5-5 for the many different ways to select geometric entities using toolbar buttons, mouse click options, page settings, and keyboard shortcuts.

TABLE 5-5: 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 settings 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 node to select all entities that the node 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 edges that are next to that edge. Hold the Ctrl key down while clicking to confirm selection of all adjacent geometric entities.
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 ID )
Lock parts of the geometry during selection:	On some settings windows, click the Select all (domains, boundaries, edges, or points) check box. See Pasting Lists of Geometric Entities to Create Selections for an example.

TABLE 5-5: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Add items to a selection list:	<p>For user-defined selections this action must be completed on the selection page. See Pasting Lists of Geometric Entities to Create Selections for information.</p> <p>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.</p> <p>In the Selection List window, select the entity names to add, and click the Add to Selection button (+) in the settings window.</p> <p>You can also paste selections from a file. See Pasting Lists of Geometric Entities to Create Selections for information.</p>
Select multiple parts of the geometry:	<p>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.</p> <p>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.</p>
Use an -OR operation with the current selection	<p><i>2D only:</i> In the Graphics window, right-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.</p>
Select only objects:	<p>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.</p>
Select only domains:	<p>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.</p> <p>In the settings window, select Domain from a Geometry entity level list and then click in the Graphics window. Only domains are highlighted.</p>

TABLE 5-5: DIFFERENT WAYS TO SELECT GEOMETRIC ENTITIES

TASK	ACTION
Select only boundaries:	<p>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.</p> <p>In the settings window, select Boundary from a Geometry entity level list and then click in the Graphics window. Only boundaries are highlighted.</p>
Select only edges:	<p>In the Graphics window, click the Select Edges button (). To select more than one at a time, press Ctrl+click or Shift+click.</p> <p>In the settings window, select Edge from a Geometry entity level list and then click in the Graphics window. Only edges are highlighted.</p>
Select only points:	<p>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.</p> <p>In the settings window, select Point from a Geometry entity level list and then click in the Graphics window. Only points are highlighted.</p>
Remove a geometric entity from a selection list:	<p>For user-defined selections this action must be completed on the selection page. See Pasting Lists of Geometric Entities to Create Selections for information.</p> <p>In the Graphics window, highlight or select the geometric entity and click the Remove from Selection button () in the settings window.</p> <p>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.</p> <p>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.</p>
Clear all items from a selection list:	<p>For user-defined selections this action must be completed on the selection page. See Pasting Lists of Geometric Entities to Create Selections for information.</p> <p>Click the Clear Selection button () in the settings window.</p>

Zooming In and Out in the Graphics Window

TASK	ACTION
Zoom in and out:	<p>Click the Zoom In button () to zoom in.</p> <p>Click the Zoom Out button () to zoom out.</p> <p><i>3D only:</i> 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.</p>
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 the Zoom Selected button () to zoom in on selected geometric entities. This button is also available in connection with the selection lists for domains, boundaries, edges, and points.
Zoom out to 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):	<p>Click the Go to View XY (), YZ (), or ZX () buttons.</p> <p>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.</p>
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.

Moving Around and Rotating 3D Geometry in the Graphics Window

TASK	ACTION AND RESULT	OPERATION ORDER
Rotate the geometry about the axes:	<p>Left-click and hold down the mouse button while dragging it in any direction.</p> <p>This rotates the scene around the axes parallel to the screen X- and Y-axes with origin in the scene rotation point.</p>	left-click
Move the visible frame on the image plane in any direction:	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):	<p>Press Ctrl and left-click. While holding down both buttons, drag the mouse in any direction.</p> <p>This places the rotation coordinate system in the camera and rotates around the axes parallel to the screen X- and Y-axes.</p>	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 about 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

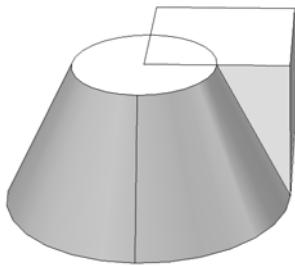
Lighting, Transparency, and Wireframe Rendering



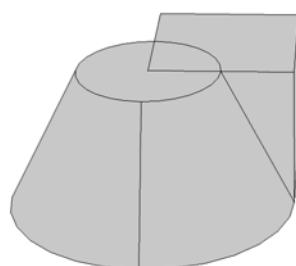
The following nodes are available for the 3D View.

TASK	ACTION
Turn Scene Light on or off:	In the Graphics window or any other plot window, click the Scene Light button (). See Figure 5-6 . 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 5-6 . 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 5-6 . When creating a View , this action toggles the Wireframe Rendering check box on the View page. See also Preferences Settings to set the level of graphic detail to Wireframe and speed up the rendering of complex models or to improve visual appearance.

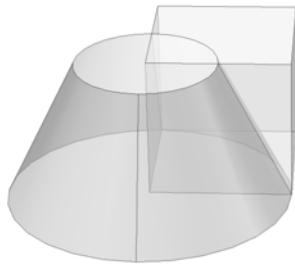
Scene Light on and Transparency off



Scene Light off



Transparency on



Wireframe rendering on

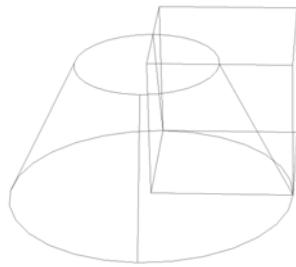


Figure 5-6: Scene light, transparency, and wireframe rendering examples.

Hiding and Showing Geometric Entities



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 and geometry objects:	<p>In the Graphics window, select any geometric entity (domain, boundary, edge, or point) and click the Hide Selected button ().</p> <p>When creating a View, right-click the View node and select Hide Geometry Objects. Select a Geometric entity level from the list to hide.</p> <p>When creating a View, right-click the View node and select Hide Geometric Entities. Select a Geometric entity level from the list to hide.</p>
Show selected geometric entities:	Click the Show Selected button () to display a hidden geometric entity that is selected in a selection list.
Reset all hidden entities and objects to the default:	<p>In the Graphics window, click the Reset Hiding button () to reset all hidden domains, boundaries, edges, or points to the default.</p> <p>This removes any Hide Geometry Objects or Hide Geometric Entities subnode added to a View node.</p>
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.



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.

Named Selections

Creating Named Selections

You can 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 properties. These user-defined selections can be reused during modeling and named using descriptive titles—for example, “Tube,” “Wall,” or “Fluid.” Changes to the selection (for example, by adding or removing a boundary) updates all nodes in the model that use that particular selection.

There are different types of selections—**Explicit** selections, selections by enclosing part of the geometry by a bounding ball, box, or cylinder (**Ball**, **Box**, and **Cylinder** selections), Boolean selections, and selections of adjacent geometric entities (**Adjacent** selections). To add selection nodes, right-click a **Definitions** node and choose from the **Selections** options as listed in [Table 5-6](#). You can also right-click the Geometry node and choose from **Selections** options similar to those in [Table 5-6](#) for defining selections based on the geometry objects in the geometry sequence. See [Creating Named Selections in the Geometry Sequence](#).

In the selection settings windows (except explicit selections) click the **Zoom Selected** () button to zoom in on the selected geometric entities.

TABLE 5-6: SELECTION TYPE DESCRIPTIONS

TYPE	ICON	DESCRIPTION
Explicit		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.
Ball		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.
Cylinder		Use the Cylinder node to create the selection by enclosing part of the geometry by a bounding cylinder to select geometric entities that are partially or completely inside the cylinder.
Union		Use the Union node to create the selection as the union (addition) of two or more selections.

TABLE 5-6: SELECTION TYPE DESCRIPTIONS

TYPE	ICON	DESCRIPTION
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		Use the Adjacent node to create the selection as the adjacent geometric entities (boundaries, for example) to one or more selections.
		<ul style="list-style-type: none"> • Sorting Nodes by Space Dimension and Type • Creating Selections From Geometric Primitives and Operations • Pasting Lists of Geometric Entities to Create Selections • About Selecting Geometric Entities • Selecting and Deselecting Geometric Entities

Explicit

Use an **Explicit** node () to create the selection using the selection tools for individual geometric entities (boundaries, for example) on the chosen geometric entity level.

INPUT ENTITIES

Based on space dimension, select a **Geometric entity level—Domain, Boundary, Edge, or Point** for the geometric entities to add or remove from the selection list.

Select and add geometric entities in the **Graphics** window, using other selection methods, or by selecting the **All domains**, **All boundaries**, **All edges**, or **All points** check box. The selected items are highlighted in the **Graphics** window. Selecting the check box for all geometric entities locks all entities of this type as selected even if the geometry changes.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at

their junctions to select all faces that make up a continuous sheet, for example. When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).



To deselect one or more of the faces or edges that make up the group with continuous tangents, first clear the **Group by continuous tangent** check box.

OUTPUT ENTITIES

Define the geometry objects that the selection contains. The options available and defaults depend on the selection in the **Geometric entity level** list as well as the space dimension of the model.

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** or **Edge** are chosen as the input, the default output is the **Selected boundaries** or **Selected edges**, respectively. As with the **Domain** input, select other options as required. The **Point** output is the same as the selection input (that is, the selected points).



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.

Ball, Box, and Cylinder

Another way to select geometric entities is to define an enclosing **Ball** (), **Box** (), or **Cylinder** () to select geometric entities that are completely or partially inside the ball, box, or cylinder.

GEOMETRIC ENTITY LEVEL

Select the **Level** for the geometric entities —**Domain**, **Boundary**, **Edge**, or **Point**.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions (to select all faces that make up a continuous sheet, for example).

When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).

See **Output Entities** for details about how the behavior depends on the condition for which the selection considers the group of entities to be enclosed.

INPUT ENTITIES

The **Entities** list defaults to **All**, which bases the selection on all entities of the selected type. Select **From selections** to base the selection on other defined selections. Then, in the **Selections** list, add the selections for which you want to create a selection of geometric entities from those selections that are located within the ball, box, or cylinder that you define for the resulting selection.

BALL CENTER/BALL RADIUS

Position the center of the ball by entering the center position in the **x**, **y**, and (3D only) **z** fields (the unit is the length unit for the geometry). Enter the radius for the ball (disk in 2D) in the **Radius** field. The default is 0.

BOX LIMITS

Define the dimensions of the box by entering the maximum and minimum values in all directions in the **x minimum**, **x maximum**, **y minimum**, **y maximum**, and (for 3D) **z minimum** and **z maximum** fields. The unit is the length unit for the geometry. The default is **-Inf** and **Inf** for the minimum and maximum values, respectively; that is, the box encloses the entire geometry. Use **-Inf** or **Inf** in some of these settings to make the box only partially bounded.

SIZE AND SHAPE (CYLINDER)

Define the dimensions of the cylinder by entering the radius and the positions of the upper and lower faces on the cylinder axis in the **Radius**, **Top distance**, and **Bottom distance** fields. The unit is the length unit for the geometry. The default is 0, **Inf**, and **-Inf** for these settings, respectively.

POSITION (CYLINDER)

Position the cylinder by entering the center position in the **x**, **y**, and (for 3D) **z** fields. The default is 0 for all coordinates.

AXIS (CYLINDER)

Set the cylinder axis by choosing an **Axis type—z-axis** (the default), **x-axis**, **y-axis**, **Cartesian**, or **Spherical**. If **Cartesian** is selected, enter coordinates for **x**, **y**, and **z**. If **Spherical** is selected, enter angles for **theta** and **phi** (unit: deg).

OUTPUT ENTITIES

For the selections made under **Input Entities**, define the dimension of the ball, box, or cylinder and select the condition for the geometric entities to be selected (from the **Include entity if** list under **Output Entities**):

Entity Intersects (Ball, Box, or Cylinder)

The default is **Entity intersects ball**, **Entity intersects box**, or **Entity intersects cylinder** and selects all geometric entities that intersect the enclosing ball, box, or cylinder; that is, the selection includes all entities that are partially or completely inside the ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, all entities in each group are selected if any entity in the group intersects the ball, box, or cylinder.

Entity Inside (Ball, Box, or Cylinder)

Select **Entity inside ball**, **Entity inside box**, or **Entity inside cylinder** to select all geometric entities that are completely inside the enclosing ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, the entities in each group are selected only if all entities in the group are completely inside the ball, box, or cylinder.

Some Vertex Inside (Ball, Box, or Cylinder)

Select **Some vertex inside box**, **Some vertex inside ball**, or **Some vertex inside cylinder** to select all geometric entities where at least some vertex is inside the enclosing ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, all entities in each group are selected if any entity in the group has at least some vertex inside the ball, box, or cylinder.

All Vertices Inside (Ball, Box, or Cylinder)

Select **All vertices inside ball**, **All vertices inside box**, or **All vertices inside cylinder** to select all geometric entities where all vertices are inside the enclosing ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, the entities in each group are selected only if all entities in the group have all vertices inside the ball, box, or cylinder. This selection might differ slightly compared to when selecting **Entity inside ball**, **Entity inside box**, or **Entity inside cylinder** if the geometric entity is outside the ball, box, or cylinder at some points between the vertices.



The ball, box, and cylinder 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 **Visualization**).

Union, Intersection, Difference, or Complement

Boolean selections—**Union** (), **Intersection** (), **Difference** (), and **Complement** ()—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 a selection is needed that consists of all boundaries except one or a few. Then it is easier to first create a selection (just the one boundary, for example) and then use a **Complement** node to create its complement. All Boolean selection nodes' settings windows have similar sections.

GEOMETRIC ENTITY LEVEL

Based on space dimension, select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Selections to add** (**Selections to intersect**, **Selections to subtract**, **Selections to invert**) list.

INPUT ENTITIES

Use the buttons in this section to move, add, or delete selections in the **Selections to add** (Union), **Selections to intersect** (Intersection), or **Selections to invert** (Complement) lists. For the **Difference** selection node also choose **Selections to subtract**. Click the **Add** button () to open an **Add** dialog box that contains all existing selections for the same geometric entity level.

Adjacent

The **Adjacent** () selection 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.

INPUT ENTITIES

Based on space dimension, select a **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Input selections** list and to create a selection of adjacent geometric entities. Click the **Add** button () to open an **Add** dialog box that contains selections of the chosen geometric entity level that appear earlier in the geometry sequence.

OUTPUT ENTITIES

In the **Geometry entity level** list, choose the type of output entities—**Adjacent domains**, **Adjacent boundaries**, **Adjacent edges** (3D only), or **Adjacent points**. If the output entities have a lower dimension than the input entities, there are also two check boxes that you can use to select exterior and interior entities of the union of the input selections. By default, only exterior entities are selected. For example, if the input selections are domains selections, and the output is adjacent boundaries, the **Exterior boundaries** (selected by default) and **Interior boundaries** check boxes appear.

Opening an Example Model

Figure 5-3 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.

Creating a Selection Node from the Selection List Window

- 1** From the main menu, select **Options>Selection List** ().

COMSOL Multiphysics 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 5-7](#).
- 4** In the **Model Builder** the new **Selection** node is added under **Definitions**.

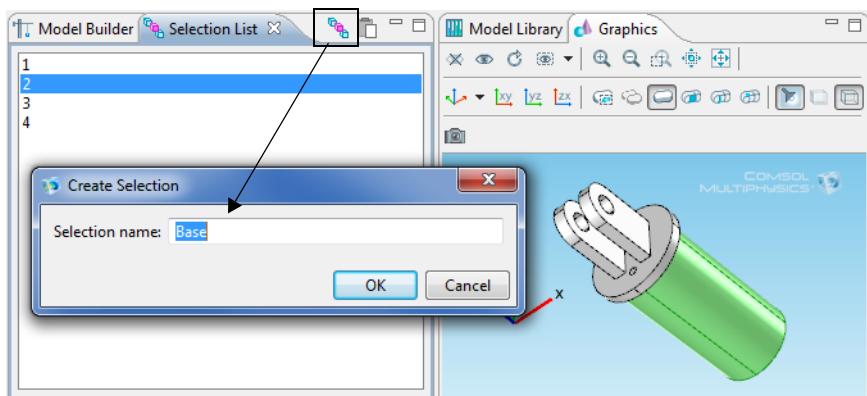


Figure 5-7: Creating a selection from the Selection List window.

Creating a Selection Node from a Settings Window

- 1** At any time during model creation, click a node that has the option to add a geometric entity to a selection (for example, under the **Materials** node or the **Solid Mechanics** physics 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 5-8](#).
- 4** In the **Model Builder** the new **Selection** node is added under **Definitions**.

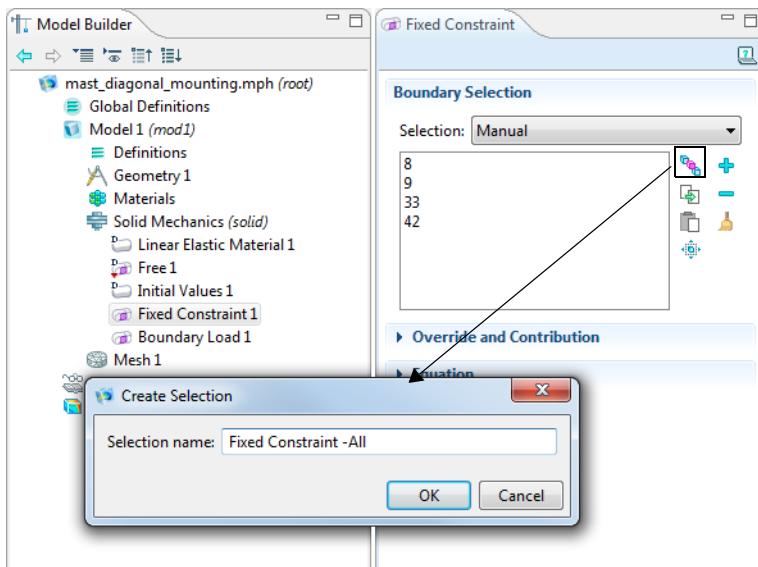


Figure 5-8: Creating a selection from the settings window for a Fixed Constraint node.

Creating Selections From Geometric Primitives and Operations

For all geometric primitives in 3D, 2D, and 1D geometries—for example, blocks, spheres, squares, polygons, and intervals—as well as for all geometry operations—for example, union, intersection, fillet, array, and mirror—you can 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 3D: domains, boundaries, edges, and points
- In 2D: domains, boundaries, and points
- In 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 nodes, for example. There are no explicit Selection nodes for these selections. For example, for a 3D model with a single **Block** node, the **Selection** list contains the

selection **Block 1**, which for a domain selection is the single domain, and for a boundary selection consists of the six faces of the block.

Creating Named Selections in the Geometry Sequence

You can also create selection nodes in the geometry sequence for user-defined named selections of all or a few of the geometric entities at a specific geometric entity level based on one or more of the nodes above the 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 and also create selections based on entire 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 during a parametric sweep, for example. That is, the selection nodes in the **Geometry** branch can provide better associativity when changing or updating the geometry than the corresponding selection nodes under **Definitions**. See the following sections for details on the selection nodes in the geometry sequence, which you choose from the **Selections** submenu in the **Geometry** node's context menu.

BUILDING SELECTION NODES

If the current node in the geometry sequence is before the node preceding the selection node, or after the selection node, the selection is not visualized (because the selected object or entities might not be visible in this state). In this case, the **Build Preceding State** button appears instead of the selection list (this also applies for nonexplicit selections in some selection nodes if **From selections** is selected under **Input Entities**). To visualize a state where the selection can be visualized, either click **Build Preceding State** or click the **Build Selected** button (). To rebuild the entire geometry, click the **Build All** button ().

Explicit Selection

Use an **Explicit Selection** node () to create the selection using the selection tools for individual geometry objects or geometric entities (boundaries, for example).

ENTITIES TO SELECT

Based on space dimension, select a **Geometric entity level—Object, Domain, Boundary, Edge**, or **Point** for the geometric objects or entities to add to the selection list.

Select and add geometric entities in the **Graphics** window or using other selection methods. The selected items are highlighted in the **Graphics** window.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions to select all faces that make up a continuous sheet, for example. When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).



To deselect one or more of the faces or edges that make up the group with continuous tangents, first clear the **Group by continuous tangent** check box.



If a particular selection is used elsewhere in the model, including in other selections, it is not possible to change the geometric entity level, for example, from domains to boundaries.

Ball Selection, Box Selection, and Cylinder Selection

Another way to select geometry objects or geometric entities is to use an enclosing **Ball Selection** (), **Box Selection** (), or **Cylinder Selection** () to select objects or entities that are completely or partially inside the ball, box, or cylinder.

G E O M E T R I C E N T I T Y L E V E L

Select the **Level** for the geometric entities —**Object**, **Domain**, **Boundary**, **Edge**, or **Point**.

If **Boundary** (for 2D and 3D models) or **Edge** is selected, also select the **Group by continuous tangent** check box to extend the selection to all adjacent faces or edges that have continuous tangents (an angle less than the value in the **Angular tolerance** field) at their junctions (to select all faces that make up a continuous sheet, for example).

When the **Group by continuous tangent** check box is selected, set the tolerance on the continuity condition in the **Angular tolerance** field as the maximum angle between two faces or edges that are considered as having continuous tangents (a value between 0 and 180 degrees; the default value is 5 degrees).

See **Output Entities** for details about how the behavior depends on the condition for which the selection considers the group of entities to be enclosed.

INPUT ENTITIES

The **Entities** list defaults to **All**, which bases the selection on all entities of the selected type. Select **From selections** to base the selection on user-defined selections earlier in the geometry sequence. Then, in the **Selections** list, add the selections that you want to use as input for creating a selection of the geometric entities that are located within the ball, box, or cylinder.

BALL CENTER/BALL RADIUS

Position the center of the ball by entering the center position in the **x**, **y**, and (3D only) **z** fields (the unit is the length unit for the geometry). Enter the radius for the ball (disk in 2D) in the **Radius** field. The default is 0.

BOX LIMITS

Define the dimensions of the box by entering the maximum and minimum values in all directions in the **x minimum**, **x maximum**, **y minimum**, **y maximum**, and (for 3D) **z minimum** and **z maximum** fields. The unit is the length unit for the geometry. The default is **-Inf** and **Inf** for the minimum and maximum values, respectively; that is, the box encloses the entire geometry. Use **-Inf** or **Inf** in some of these settings to make the box only partially bounded.

SIZE AND SHAPE (CYLINDER)

Define the dimensions of the cylinder by entering the radius and the positions of the upper and lower faces on the cylinder axis in the **Radius**, **Top distance**, and **Bottom distance** fields. The unit is the length unit for the geometry. The default is 0, **Inf**, and **-Inf** for these settings, respectively.

POSITION (CYLINDER)

Position the cylinder by entering the center position in the **x**, **y**, and (for 3D) **z** fields. The default is 0 for all coordinates.

AXIS (CYLINDER)

Set the cylinder axis by choosing an **Axis type**—**z-axis** (the default), **x-axis**, **y-axis**, **Cartesian**, or **Spherical**. If **Cartesian** is selected, enter coordinates for **x**, **y**, and **z**. If **Spherical** is selected, enter angles for **theta** and **phi** (unit: deg).

OUTPUT ENTITIES

For the selections made under **Input Entities**, define the dimension of the ball, box, or cylinder and select the condition for the geometric entities to be selected (from the **Include entity if** list under **Output Entities**):

Entity Intersects (Ball, Box, or Cylinder)

The default is **Entity intersects ball**, **Entity intersects box**, or **Entity intersects cylinder** and select all geometric entities that intersect the enclosing ball, box, or cylinder; that is, the selection includes all entities that are partially or completely inside the ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, all entities in each group are selected if any entity in the group intersects the ball, box, or cylinder.

Entity Inside (Ball, Box, or Cylinder)

Select **Entity inside ball**, **Entity inside box**, or **Entity inside cylinder** to select all geometric entities that are completely inside the enclosing ball, box, or cylinder; that is, the selection includes all entities that are completely inside the ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, the entities in each group are selected only if all entities in the group are completely inside the ball, box, or cylinder.

Some Vertex Inside (Ball, Box, or Cylinder)

Select **Some vertex inside box**, **Some vertex inside ball**, or **Some vertex inside cylinder** to select all geometric entities where at least some vertex is inside the enclosing ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, all entities in each group are selected if any entity in the group has at least some vertex inside the ball, box, or cylinder.

All Vertices Inside (Ball, Box, or Cylinder)

Select **All vertices inside ball**, **All vertices inside box**, or **All vertices inside cylinder** to select all geometric entities where all vertices are inside the enclosing ball, box, or cylinder. If the **Group by continuous tangent** check box is selected, the entities in each group are selected only if all entities in the group have all vertices inside the ball, box, or cylinder. This selection might differ slightly compared to when selecting **Entity inside ball**, **Entity inside box**, or **Entity inside cylinder** if the geometric entity is outside the ball, box, or cylinder at some points between the vertices.



The ball, box, and cylinder 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 **Visualization**).

Union, Intersection, Difference, or Complement Selection

Boolean selections—**Union Selection** () , **Intersection Selection** () , **Difference Selection** () , and **Complement Selection** () —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 a selection is needed that consists of all boundaries except one or a few. Then it is easier to first create a selection (just the one boundary, for example) and then use a **Complement** node to create its complement. All Boolean selection nodes' settings windows have similar sections.

GEOMETRIC ENTITY LEVEL

Based on space dimension, select a **Geometric entity level**—**Object**, **Domain**, **Boundary**, **Edge** (3D only), or **Point** for the selections to add or remove from the **Selections to add** (**Selections to intersect**, **Selections to subtract**, **Selections to invert**) list.

INPUT ENTITIES

Use the buttons in this section to move, add, or delete selections in the **Selections to add** (Union), **Selections to intersect** (Intersection), or **Selections to invert** (Complement) lists. For the **Difference** selection node also choose **Selections to subtract**. Click the **Add** button () to open an **Add** dialog box that contains selections for the same geometric entity level that appear earlier in the geometry sequence.

Adjacent Selection

The **Adjacent Selection** () node selects all entities of a given dimension that are adjacent to entities in a given set of selections (having another dimension). For example, it can select all boundaries adjacent to a given domain selection.

INPUT ENTITIES

From the **Geometric entity level** list, choose the type of the input entities—**Domain**, **Boundary**, **Edge** (3D only), or **Point**. Use the buttons below the **Input selections** list to add or delete input selections. Click the **Add** button () to open an **Add** dialog box that contains selections of the chosen geometric entity level that appear earlier in the geometry sequence.

OUTPUT ENTITIES

In the **Geometry entity level** list, choose the type of output entities—**Adjacent domains**, **Adjacent boundaries**, **Adjacent edges** (3D only), or **Adjacent points**. If the output entities have a lower dimension than the input entities, there are also two check boxes that you can use to select exterior and interior entities of the union of the input selections. By default, only exterior entities are selected. For example, if the input selections are domains selections, and the output is adjacent boundaries, the **Exterior boundaries** (selected by default) and **Interior boundaries** check boxes appear.

User-Defined Views

Views provide the camera setting, grid, rendering, arrows, lighting, and transparency in the **Graphics** window. You can use several user-defined views to highlight and display the geometry in different ways.

The View Nodes

By default a **View** node is added to all 2D (), 2D axisymmetric () , and 3D () models.

 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.

 For 3D models, a **Camera** subnode () and three **Directional Light** nodes () , including default settings, are also added.

 The **View** nodes and their 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, lighting attributes, and camera settings.

Also right-click the **View** node to add **Hide Geometric Entities** and **Hide Geometry Object** nodes for any space dimension.

Right-click **Definitions** () and select **View** to create additional **View** nodes and then experiment by switching views to find the best way to illustrate a model. Views can be selected from the list of views () on the **Graphics** toolbar.

 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** ().

View 2D

For 2D and 2D axisymmetric models, the **View 2D** node () controls the settings to display or hide geometry labels and lock the axis. An **Axis** node is added by default.

Also right-click the **View** node to add **Hide Geometry Objects** and **Hide Geometric Entities** nodes. To add additional **View** nodes, in the **Model Builder**, right-click **Definitions** and select **View**.

VIEW

Select the **Show geometry labels** check box to display the geometry object labels and the geometric entity labels (numbering) in the **Graphics** window. The labels appear for geometry objects or geometric entities (domain, boundary, edge, or point numbers), depending on what part of the model tree you display and the selection mode you are using.

Select the **Show edge direction arrows** check box to display the direction arrows on boundaries (edges) in the **Graphics** window. The direction arrows indicate the directions in which the boundary parameterization's value increases.

Select the **Lock axis** check box to store the current axis limits so that the zoom tools can be temporarily 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.

Axis

For 2D and 2D axisymmetric models, the **View 2D** node has an **Axis** () subnode where the axis coordinates are set, manual spacing is defined, and the aspect ratio preserved.

AXIS

Enter **x minimum**, **x maximum**, **y minimum**, and **y maximum** values for the axis limits (**r** and **z** replace **x** and **y** in 2D axial symmetry). 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.

GRID

Select the **Manual spacing** check box and enter **x spacing** and **y spacing** values to control the grid spacing manually. **r** and **z** replace **x** and **y** in 2D axial symmetry. Enter **Extra x** and **Extra y** values directly or click the **Range** button () as required.



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.

View 3D

The **View 3D** 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 in the **Graphics** window. See Figure 5-9.



[About the 3D View Light Sources and Attributes](#)

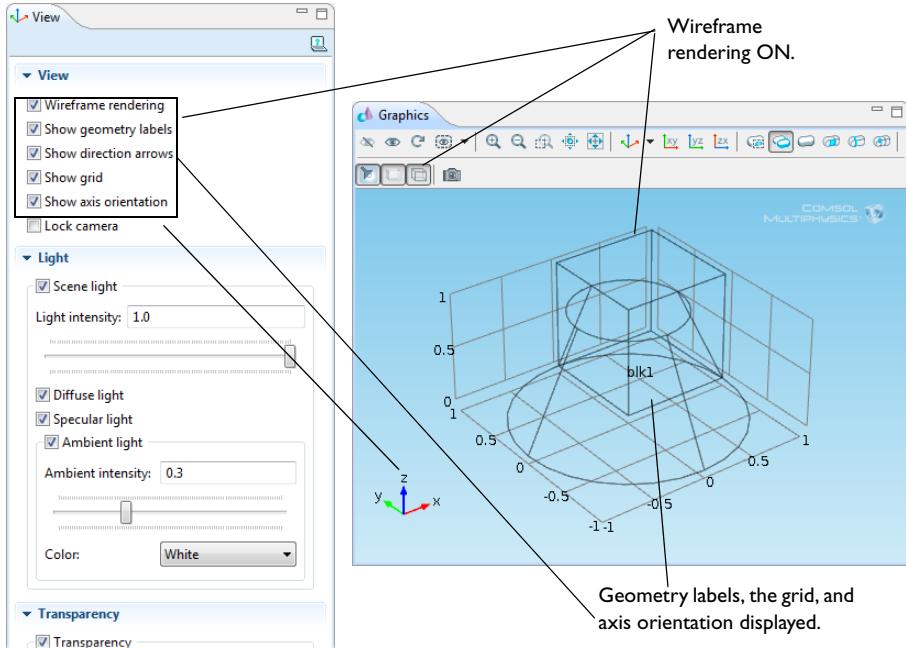


Figure 5-9: An example of a 3D View settings window with all View option check boxes selected. The Graphics window shows what the check boxes represent.

The **View 3D** node () has the following sections:

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 object names and geometric entity labels (numbers) in the **Graphics** window. The labels appear for geometry objects or geometric entities (domain, boundary, edge, or point numbers), depending on what part of the model tree you display and the selection mode you are using.
- Select the **Show edge 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 then revisiting the **View** node restores the camera settings to the values in the view at the time the **Lock camera** check box was 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 between 0 and 1 for the **Ambient intensity** (default value: 0.3) 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.

Camera

Use the **Camera** node () to orient the camera view in 3D models. In the **Model Builder**, under **Definitions** click to expand the **View** node where you want to define the camera position.

CAMERA

From the **Projection** list, select **Perspective** (the default) or **Orthographic** (parallel) 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.

Enter a **Zoom angle** (in degrees) or use the **Zoom** buttons on the **Graphics** toolbar.

By default, the **Preserve aspect ratio** check box is selected to make the increments equal in the *x*-, *y*-, and *z*-directions. Clear this check box to make the geometry fill the graphics window with unequal increments. This can be useful when working with slender geometries.

POSITION

In the **Graphics** window, left-click and hold the mouse to orient the geometry on the axes, or enter **x**, **y**, and **z** coordinates.

TARGET

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 for the camera target location.

The **Position** is the location of the camera and the **Target** default is 1 length unit in front of the camera position.

UP VECTOR

In the **Graphics** window (or other plot windows), left-click and hold the mouse to orient the geometry on the axes, or enter **x**, **y**, and **z** coordinates for the camera's up vector, which determines which direction is up in the plot window.

CENTER OF ROTATION

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.

VIEW OFFSET

Right-click the mouse and move 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. For example, an offset of $x = 0.5$ moves the projection 0.5 screen widths to the left. 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.

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 5-10](#)). Combined, these attribute and source settings enable the software to render the 3D model to look realistic.

LIGHT ATTRIBUTES

Ambient light + Diffuse light + Specular light == Scene light

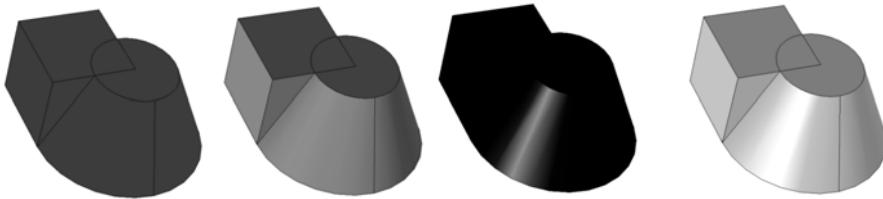


Figure 5-10: Scene light is a combination of ambient, diffuse, and specular light attributes. The default Scene light color is white and in this example the attributes display as different shades of black and gray. The Scene light is further enhanced using the various light source nodes.

Scene Light

For all geometry, scene light is applied by default and is a combination of the different light attributes. 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 [View 3D](#)). 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 5-10](#) 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 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



For 3D models, you can 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. Each View can have a maximum of eight light sources (nodes) in any combination. See [Directional Light](#), [Point Light](#), [Spotlight](#), and [Headlight](#) below.

Each light source has a unique light marker displayed in the Graphics window. [Figure 5-12](#) 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. 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.

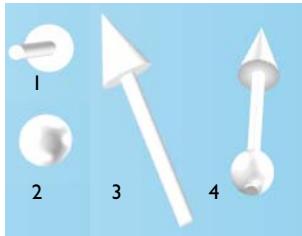


Figure 5-11: The light markers for each type of light source: (1) Headlight, (2) Point light, (3) Directional light, and (4) Spotlight.

Directional Light

By default, a **View** contains three **Directional Light** (💡) nodes. 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. The direction of the light, the light and specular intensity levels, and the color can be adjusted as required.

[Figure 5-12](#) 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.

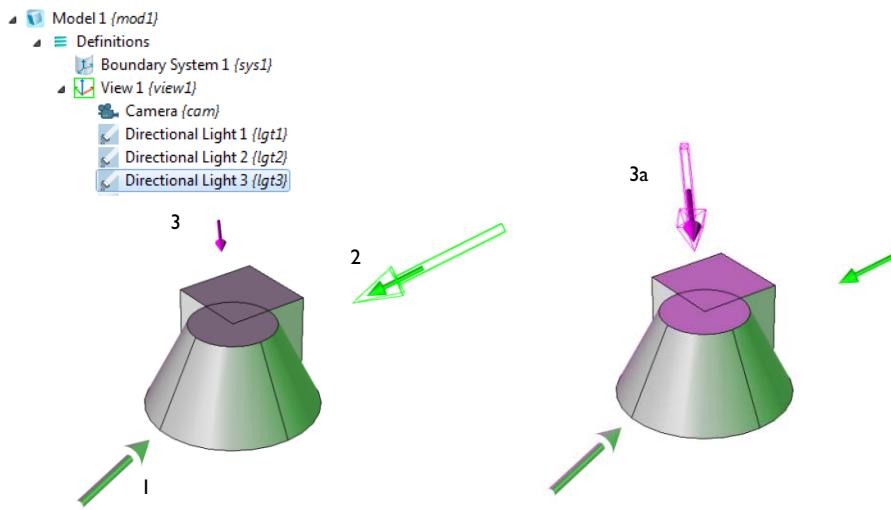


Figure 5-12: Examples of directional light markers and the location of each directional light. The markers and light can be color coded to see which areas need adjustment. The markers indicate which node is selected in the Model Builder (a wireframe around the arrow in 2 and 3a), and the level of light intensity applied (the size of the arrow; compare 3 and 3a).

DIRECTION

The **x**, **y**, and **z** coordinates define the direction in which the light falls on the objects in the scene. 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.



Change the color to something other than white to observe where the light hits the object.

SETTINGS

Adjust some of the settings such as the intensity and color of the directional light.

- Enter a **Light Intensity** or use the slider to select values. 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.

- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and 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 5-10](#) for an example of specular light.
- Select a **Color—White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
- 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.



To lock the camera settings, see [Camera](#).

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See [Figure 5-11](#) for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Point Light

The **Point Light** () has a position and emits the light equally in all directions; it is like a light bulb. Point light have therefore no direction. The position of the light, the light and specular intensity levels, and the color can be adjusted as required.

[Figure 5-13](#) is an example of two **Point Light** () nodes added to the **View** and all other nodes disabled.

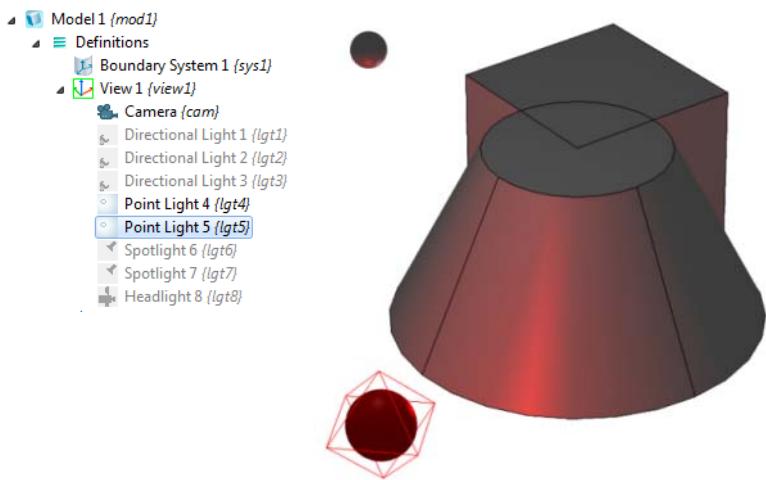


Figure 5-13: An example of two Point Light nodes added to a View. All other light nodes are disabled to show only the Point Light effects to the geometry. Point Light 5 is selected in the Model Builder as indicated by the wireframe around the light marker.

POSITION

Enter **x**, **y**, and **z** coordinates for the point light's position.

SETTINGS

Adjust some of the settings such as the intensity and color of the point light.

- Enter a **Light Intensity** or use the slider to select values. 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.
- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and 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 5-10](#) for an example of specular light.
- Select a **Color—White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.

- 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..



To lock the camera settings, see [Camera](#).

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See [Figure 5-11](#) for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Spotlight

The **Spotlight** () acts like a flashlight and has both a position and a direction.

[Figure 5-14](#) 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.

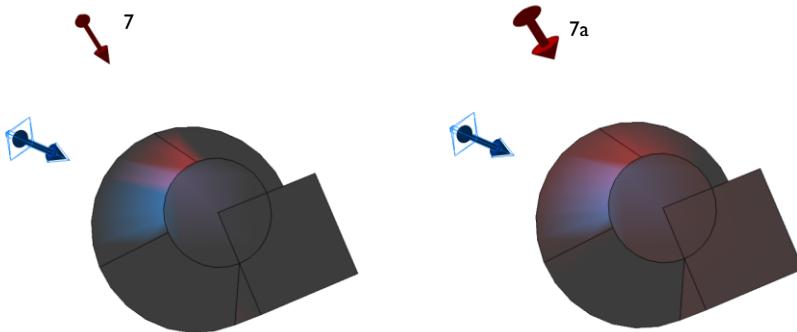


Figure 5-14: Two Spotlight nodes displayed with all other nodes disabled showing where the Spotlight is focused. When the spread angle is changed from 20 to 100 for Spotlight 7 (as indicated by the change in marker width between 7 and 7a), a corresponding change is made to the model.

POSITION

Enter **x**, **y**, and **z** coordinates for the position of the spotlight.

DIRECTION

The **x**, **y**, and **z** coordinates define the direction in which the spotlight falls on the objects in the scene. The arrow points in the direction of the light.



Change the color to something other than white to observe where the light hits the object.

SETTINGS

Adjust some of the settings such as the spread, intensity, and color of the spotlight.

- 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 5-14](#) for an example.
 - Enter a **Light Intensity** or use the slider to select values. Watch the changes in the **Graphics** window at the same time to help choose a level. The light marker's length changes as the corresponding light intensity changes.
 - Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and used to define spheres and curves). You can turn specular light on and off in the View node's settings window (default: 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 5-10](#) for an example of specular light.
 - Select a **Color—White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
 - 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.
-



To lock the camera settings, see [Camera](#).

- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See [Figure 5-11](#) for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Headlight

A headlight is a directional light that 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 5-15](#) 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.

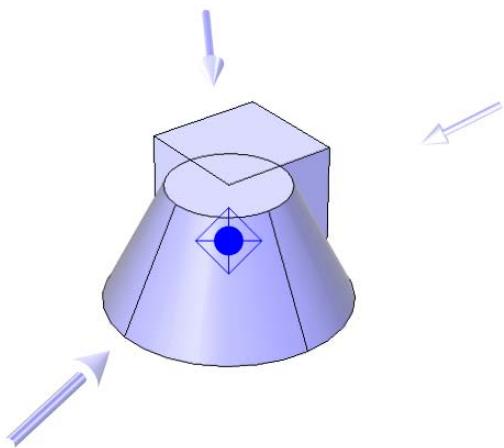


Figure 5-15: An example of a blue Headlight source with three Directional light sources set to white. If the geometry is rotated, you can adjust and view the effects of the Headlight source on the geometry based on the shades of blue and white.

SETTINGS

Adjust the settings such as the intensity and color of the headlight.

- Enter a **Light Intensity** or use the slider to select values. 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.

- Enter a **Specular intensity** (the intensity of the light that reflects off the surface of the geometry and 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 5-10](#) for an example of specular light.
- Select a **Color—White** (default), **Custom**, **Black**, **Blue**, **Cyan**, **Green**, **Magenta**, **Red**, or **Yellow**. If you select **Custom**, click the **Color** button to choose a color from the **Custom color** palette.
- By default, the **Show light marker** is selected. The light marker is associated to the type of light applied to the object. See [Figure 5-11](#) for examples. Click to clear the check box as required and remove the marker from the **Graphics** window.

Hide Geometry Objects and Hide Geometric Entities

- 5 In the **Model Builder**, under **Definitions**, right-click any **View** node and select **Hide Geometry Objects** () to hide a set of geometry objects or some of their geometric entities—in the created or imported geometry. Right-click **View** and select **Hide Geometric Entities** () to hide geometric entities (boundaries, for example)—in the analyzed (finalized) geometry used for modeling—at a specific geometric entity level for any View.



The **Hide Geometry Objects** node hides geometry objects or parts of such objects and affects the view in the Geometry branch only. The **Hide Geometric Entities** node hides geometric entities (domains, boundaries, edges, or points) in the finalized geometry and affects the view in the Materials, physics, Mesh, studies, and Results branches.

SELECTION

Hide a set of geometry objects at a specific **Geometric entity level—Object** (the default), **Domain**, **Boundary**, **Edge** (3D only), or **Point**. Then select the geometry objects or some of their geometric entities in the **Graphics** window. The list under the **Geometric entity level** list contains the selected objects or entities to hide in the **Geometry** branch. For example, for **Hide Geometry Objects**, hide a geometry objects, or hide a boundary of a 3D geometry object to make it possible to view the inside of the object. Or, for **Hide Geometric Entities**, 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 under **Plot Settings** in the corresponding plot group's settings window.

The **Hide Geometric Entities** settings window contains the following section:

GEOMETRIC ENTITY SELECTION

Hide a set of geometric entities at a specific **Geometric entity level**—**Domain** (the default), **Boundary**, **Edge** (3D only), or **Point**. From the **Selection** list, select **Manual** (the default) or **All domains**, **All boundaries**, **All edges** (3D only), or **All points**. If **Manual** is selected, go to the **Graphics** window and select the geometric entities that you want to hide. The selected entities appear in the list under the **Selection** list. Use the **Add to Selection** (), **Remove from Selection** () , **Clear Selection** () , and **Zoom Selected** () 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 (��片) on the Graphics toolbar to capture an image snapshot of a plot. To do so, follow these steps:

- 1 In the **Graphics** window (视窗) or any other plot window, click the **Image Snapshot** button (照相) 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** (1D graphs) or **Color legend** (2D plots), **Axes**, and **Logotype** check boxes (1D and 2D images) or the **Title**, **Color 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 or excluding these parts of the plot.
- 9 Enter a **Font** size in points (pt). The default value is 9 pt. This font size overrides the system font size used in the COMSOL Desktop®.
- 10 Select a **Background**—**Current**, **Color**, or **Transparent**. 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. The **Transparent** option is only available for the PNG file format..



Transparent image support includes two parts: Raw data of the png image and an external renderer (image viewer). COMSOL can provide a correct png image itself but cannot control the external renderer.



When importing an image with a transparent background to another Windows application, first save the image as a file rather than saving it to the clipboard. In some cases the transparent background is not preserved if you copy an image via the clipboard.

II Under **Output**, select the target: **Clipboard** (the default) copies the image to the clipboard. **File** saves the image to a file.

I2 If **File** is selected:

- a** Select a file **Format**—**BMP**, **EPS** (1D plots only), **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.

I3 Click **OK** to generate the image snapshot.



Printing from the COMSOL Desktop

6

Geometry Modeling and CAD Tools

The CAD tools in COMSOL Multiphysics® include many geometric primitives and operations for modeling the geometry 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 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.

Working with Geometry Sequences

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](#)). You create the model's geometry by adding nodes to the sequence and building them. The following section describes how to add geometry nodes and edit and work with them in a geometry sequence. A 2D example illustrates how to work with the geometry sequence.

Overview of Geometry Modeling Concepts

In COMSOL Multiphysics you can 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.

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.

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 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.



See [Import](#) for details of how to import these CAD file formats.

Combining several objects where at least one is imported from an STL/VRML file is not supported.

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 optional LiveLink™ products offer bidirectional links to 3D CAD software. Using these, you can run parametric geometry sweeps driven from the COMSOL environment but operating directly on the geometries in the respective CAD package.

The following sections provide an overview of available geometric primitives and operations.



- [Opening a Context Menu to Add Nodes](#)
- [Moving Nodes in the Model Tree](#)
- [Copying, Pasting, and Duplicating Nodes](#)
- [Expanding and Collapsing All Nodes](#)
- [Geometry in the *COMSOL API Reference Manual*](#)

Adding a Model (Geometry)

To add a new model, right-click the root node of the **Model Builder**, and select **Add Model**. This opens the **Model Wizard** window with the heading **Select Space Dimension**. This window also shows up when you launch the COMSOL Desktop or click the **New** button () in the main toolbar and select a 3D, 2D axisymmetric, 2D, 1D axisymmetric, 1D, or 0D geometry in the **Model Wizard**. Click the **Finish** button () on the toolbar to close the **Model Wizard** and create a new, empty model. A new **Model** node displays in the **Model Builder** containing a geometry sequence node **Geometry 1** (if it is the first model) with a geometry sequence only containing a **Form Union** node.

Alternatively, in the **Model Wizard**, clicking **Next** takes you to the **Add Physics** page, where you can add the physics you want to use.

Adding Geometry Nodes

You can add and build geometry nodes in two ways:

- Right-click a **Geometry** node in the Model Tree and then select one of the available nodes. Enter the node's properties in its settings window. In numerical fields you can enter expressions that contain parameters defined in **Parameters** under **Global Definitions** in the Model Tree to parameterize the geometry. Click the **Build Selected** button () in the settings window to see the geometry objects that result. All geometry nodes can be added using this method.
- Use the buttons in the Geometry toolbar. This method is available for some drawing primitives in 2D and for some operations and conversions acting on geometry objects.



The Geometry Toolbars

EXAMPLE OF ADDING AND BUILDING A 2D GEOMETRY

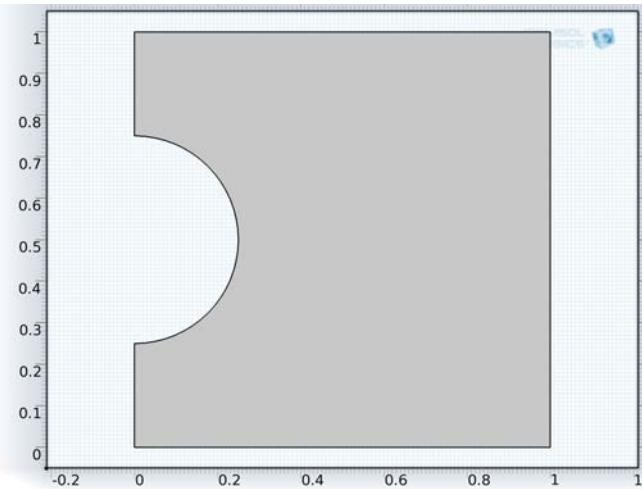
The following example assumes that you have created a new 2D model:

- 1 In the **Model Builder** under **Geometry**, right-click **Geometry 1** and select **Square** ().
- 2 Click the **Build Selected** button () in the settings window to create a unit square.
- 3 Right-click **Geometry 1** and select **Circle** ().
- 4 In the **Size and Shape** section, enter a **Radius** of 0.25. In the **Position** section enter a value of 0.5 for **y**.
- 5 Right-click **Geometry 1** and from the **Boolean Operations** menu select **Difference** ().



When you add the Difference node, the software builds the Circle node (and all its preceding nodes) before showing its settings.

- 6 In the **Graphics** window, 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 **cl** to its list.
- 8 Click **Build Selected** () to build the node and to subtract the circle from the square.

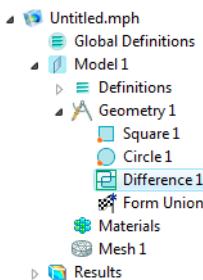


Also notice that when selecting a node that represents a geometry object in the Model Builder highlights that geometry object in the Graphics window.



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** adds the nodes to the sequence.



Editing and Building Geometry Nodes

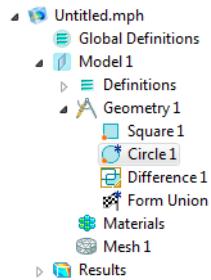
To edit a node, select it in the model tree and make changes in the settings window that appears. Nodes that you have edited display with an asterisk (*) at the upper-right corner of their icons in the **Model Builder** window. Nodes that depend on the edited node display with a yellow triangle at the lower-right corner of the node's icon to indicate that they need to be rebuilt. To see the result of your edits in the **Graphics** window, you need to build the node. You can do this in two ways:

- Click the **Build Selected** button () in the settings window, or right-click the node in the tree and select **Build Selected**. This builds all nodes (if needed) from the first up to the selected node.
- Click the **Build All** button () in the settings window, or right-click the **Geometry** node in the tree and select **Build All**. This builds all nodes in the sequence above the **Finalize (Form Union/Form Assembly)** node (if needed).

EXAMPLE OF EDITING AND BUILDING GEOMETRY NODES

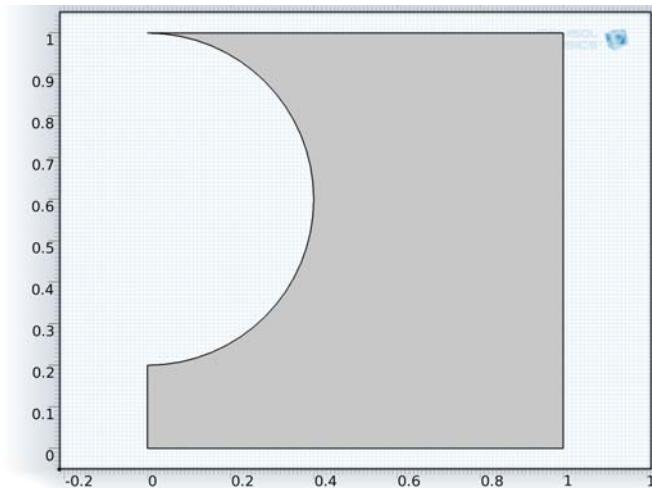
- 1 In **Model Builder** click **Circle 1**.
- 2 In the **Circle** settings window in the **Size and Shape** section, enter a **Radius** of **0.4**.
- 3 In the **Position** section enter **0.6** for **y**.

In the **Model Builder** an asterisk and a yellow triangle display in the icons for **Circle 1** and **Difference 1**, respectively.



- 4 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 displays in the **Graphics** window.



The **Circle 1** and **Difference 1** icons in **Model Builder** now display without the asterisk and the yellow triangle, respectively.

EDITING 2D GEOMETRY OBJECTS

Use the **Edit Object** () node to adjust the edges and vertices for a 2D geometry object or to add or delete edges and vertices in the object. In the **Model Builder**, right-click a 2D **Geometry** and select **Edit Object** ().



See [Edit Object](#) for details.

The **Edit Object** function can also be started using the **Edit Object** button () in the **Geometry** toolbar:

- 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 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.

AUTOMATIC BUILDING OF GEOMETRY NODES

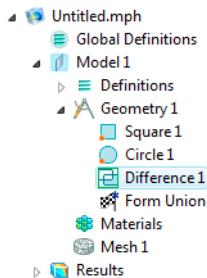
In some situations, COMSOL Multiphysics builds geometry nodes directly, such as when you:

- Add a node, the software builds the current node (and all its preceding nodes) before showing the settings window for the new node.
- Generate a mesh or solve the model, the software builds the finalized geometry; that is, it builds all nodes if needed.

- Select a node that uses the finalized geometry, like physics and mesh nodes, the software builds the finalized geometry provided that **Automatic rebuild** is selected in the geometry node settings.
- Enter a work plane, the software builds all nodes preceding the work plane node.
- Open a CAD Defeaturing tools settings, the software builds the current node (and all its preceding nodes).

The Current Node in Geometry Sequences

When you add a geometry node, it is inserted in the sequence after the *current node*. To indicate the current node, it displays with a quadratic frame around its icon. When you have added a node, it becomes the current node, but COMSOL Multiphysics does not build it automatically. 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 the **Model Builder** a quadratic frame appears 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 is yellow. See [Dynamic Nodes in the Model Builder](#) for examples that show these icon additions for visual feedback about a node's status.



Key to Nodes and Toolbar Buttons

ADDING A NODE AT AN ARBITRARY POSITION

To add a node after an existing node, first select the existing node and then click **Build Selected** (), or right-click the existing node and select **Build Selected**. The selected node then becomes current. Then add the new node.

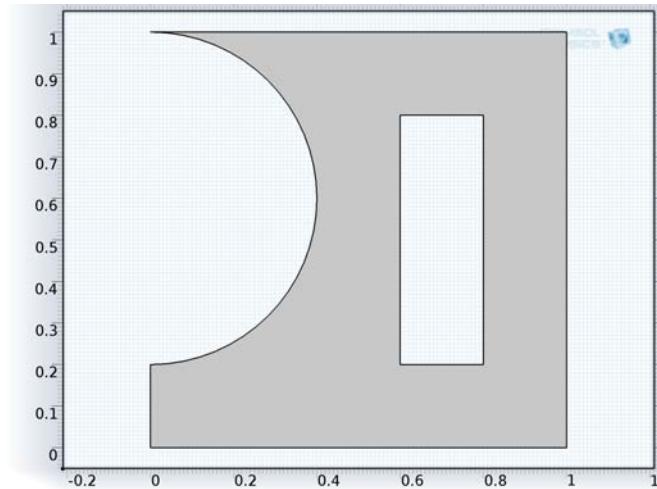
To add a node before an existing node, first select the existing node, right-click the existing node and select **Build Preceding** (). Then add the node.

EXAMPLE OF ADDING A NODE AT AN ARBITRARY POSITION

Add a Rectangle node between Circle 1 and Difference 1.

- 7 In the **Model Builder** right-click **Difference 1** and select **Build Preceding** ().
- 8 Right-click **Geometry 1** and select **Rectangle** ().
- 9 In the **Rectangle** settings window, in the **Size** section, enter a **Width** of 0.2 and a **Height** of 0.6.
- 10 In the **Position** section enter 0.6 for **x** and 0.2 for **y**.
- 11 Click the **Build Selected** button () in the settings window to build the node and to create the rectangle.
- 12 In the **Model Builder**, click the **Difference 1** node.
- 13 In the **Difference** settings window in the **Objects to subtract** section, click the **Activate Selection** button (). In the **Graphics** window, select **r1** to add it to the selection. Click **Build Selected** ().

The following geometry displays in the **Graphics** window.



Deleting, Disabling, and Enabling Geometry Features

- To delete a node, right-click it and select **Delete** ().

- 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. You cannot use the **Undo** command.
- To delete geometry objects or entities, in the **Model Builder**, right-click **Geometry** and select **Delete Entities** (). Under **Entities or Objects to Delete**, select a **Geometric entity level** to delete—**Object**, **Domain**, **Boundary**, **Edge** (in 3D only), or **Point**. Select the geometry in the **Graphics** window. Use the **Add to Selection** () , **Remove from Selection** () , and **Clear Selection** () buttons as required. Alternatively, select objects in the **Graphics** window, and click the **Delete** button () in the **Graphics** toolbar.
- To disable a node, right-click it and select **Disable** (). The disabled node does not affect the finalized geometry. This is indicated with a gray icon in the model tree.
- To enable a disabled node, right-click it and select **Enable** (.

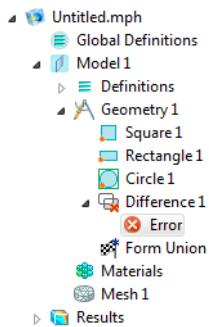
If you use the **Delete** button () to delete objects, COMSOL Multiphysics deletes the selected objects that correspond to primitive features by deleting their nodes from the geometry sequence. If you delete objects that do not correspond to primitive features or if you delete geometric entities a **Delete Entities** node appears in the sequence.

Errors and Warnings in a Geometry Sequence

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 node's icon displays with a red cross in the lower-left corner.

- 1 In the **Model Builder** click the **Difference I** node.
- 2 In the **Difference** settings window in the **Objects to subtract** section, click the **Activate Selection** button (). Click the **Clear Selection** button () and then the **Build Selected** button ().

An **Error** node displays in the **Model Builder**.



After a successful build of a node, a warning message can sometimes display as a **Warning** subnode (⚠). If a warning message exists, the node's icon displays with a yellow triangle in the lower-left corner.



Key to Nodes and Toolbar Buttons

The Geometry and CAD Environment

The following sections provide an overview of the available geometric primitives and operations.

Geometric Primitives

- [Creating Selections From Geometric Primitives and Operations](#)
- [The Geometry Toolbars](#)
- [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 [Geometric Primitives](#) section. 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 nodes for the resulting geometry objects from such operations.

For some geometric primitives (blocks and spheres, for example) you can add layers for creating sandwich structures or, for example, layers of concentric spheres.:.



Right-click the **Geometry** node to add the geometric primitives in [Table 6-1](#) to the geometry sequence. In 3D, some options are found on the **More Primitives** submenu.

TABLE 6-1: 1D, 2D, AND 3D GEOMETRY PRIMITIVES

ICON	NAME AND LINK	DIMENSION
	Bézier Polygon	2D, 3D
	Block	3D

TABLE 6-I: 1D, 2D, AND 3D GEOMETRY PRIMITIVES

ICON	NAME AND LINK	DIMENSION
	Circle	2D
	Cone	3D
	Cylinder	3D
	Eccentric Cone	3D
	Ellipse	2D
	Ellipsoid	3D
	Helix	3D
	Hexahedron	3D
	Interpolation Curve	2D, 3D
	Interval	1D
	Parametric Curve	2D, 3D
	Parametric Surface	3D
	Point	1D, 2D, 3D
	Polygon	2D, 3D
	Pyramid	3D
	Rectangle	2D
	Sphere	3D
	Square	2D
	Tetrahedron	3D
	Torus	3D

Using Work Planes

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 to create 3D solid objects from 2D sections or faces. In addition, you can use a 2D section as an “embedded” surface in the 3D geometry.



For an example of how to create a 3D geometry using work planes, see [Creating a 3D Geometry Model](#).

WORK PLANE OPERATIONS

The following table lists operations available with work planes in 3D geometries:

TABLE 6-2: WORK PLANE RELATED OPERATIONS

ICON	NAME	DESCRIPTION
	Work Plane	Create a work plane for drawing 2D objects that are embedded in 3D
	Extrude	Extrude planar objects from a work plane into 3D
	Revolve	Revolve planar objects from a work plane into 3D
	Sweep	To sweep a face along a spine curve

DRAWING ON A 2D WORK PLANE IN 3D

When using a **Work Plane** () node to define 2D objects in 3D (for example, to extrude into a 3D object), the 3D projection settings enable you to draw on the work plane in 3D. These instructions provide an example. When the **Draw on work plane in 3D** check box is selected, two additional buttons are 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 2-1) is also available for use..



Some computer graphic cards may not be able to run the work plane rendering. In these cases, the work plane is rendered as a blue plane. It is possible to go to the 3D work plane but not to draw on the plane.



If you prefer to draw on the work plane in 3D, you can change the default 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 2-2 and Figure 2-3 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 2-3 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.



Drawing on a work plane works just as drawing in 2D. The 3D work plane adapts its size to the drawn geometry.

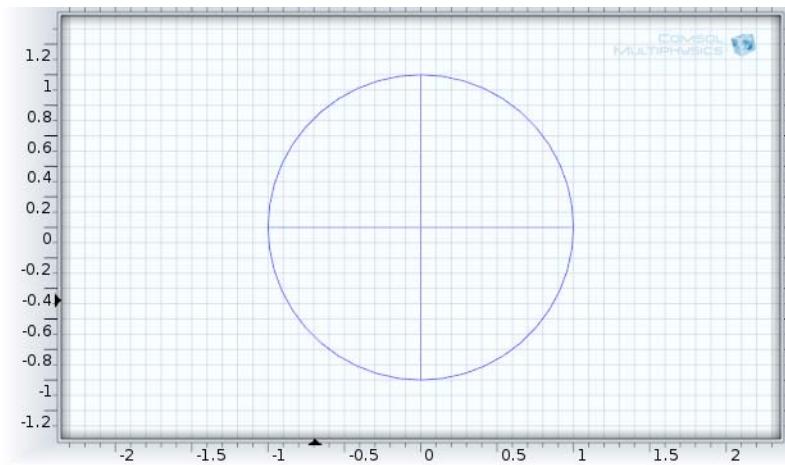


Figure 6-1: The Graphics window displaying the Work Plane Geometry without the Draw on work plane in 3D check box selected.

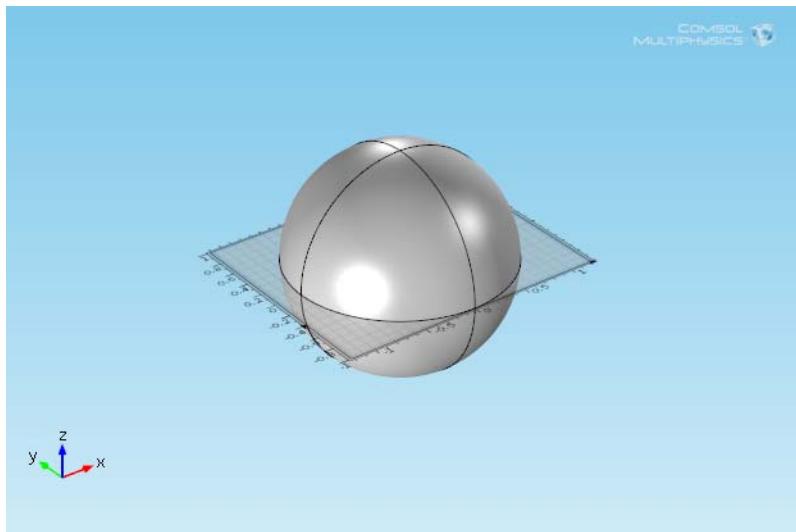


Figure 6-2: The Graphics window displaying the Work Plane Geometry with the Draw on work plane in 3D check box selected.

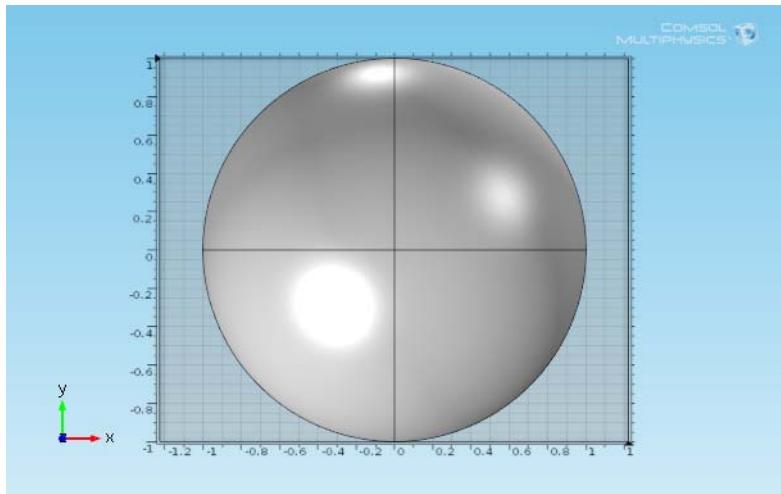


Figure 6-3: Click the Align with Work Plane button to display the geometry from the top down.

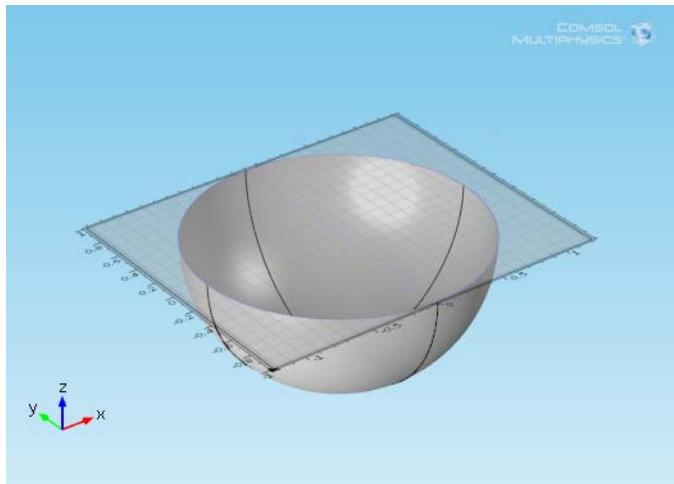


Figure 6-4: Click to turn on the Work Plane Clipping button and cut away all geometries above the work plane.

Boolean Operations

Use Boolean operations to create a *composite geometry object* by forming unions, set differences, and set intersections—and combinations of those operations—of existing geometry objects.

The following Boolean operations are available in all space dimensions:

TABLE 6-3: BOOLEAN OPERATIONS

ICON	NAME AND LINK	DESCRIPTION
	Union	Create the union of the selected objects.
	Intersection	Create the intersection of the selected objects.
	Difference	Create a set difference by taking the selected object with the largest volume (area, length) and subtract the others.
	Compose	Compose a geometry object by specifying Boolean operations using a set formula.
	Partition	Partition a geometry object using a work plane or another geometry object (tool object).

In addition the **Cross Section** operation is available in **Plane Geometry** sequences for work planes and in 2D and 2D axisymmetric geometries for creating cross sections of 3D geometry objects.

To add Boolean operations to the **Model Builder**, select geometry objects in the **Graphics** window for which you want to apply a Boolean operation. Then click the **Union** () , **Intersection** () , or **Difference** () buttons on the Geometry toolbar. Alternatively, right-click **Geometry** and select **Boolean Operations** then select **Union** () , **Intersection** () , **Difference** () , or **Compose** () .

REMOVING INTERIOR BOUNDARIES

To remove interior boundaries, clear the **Keep interior boundaries** check box 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.

	<p>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 original interior boundaries for mesh control. The interior boundaries are then not part of the geometry for defining physics but are present during meshing to define areas where you want to use a finer mesh, for example. See Mesh Control Entities for more information.</p>
---	--

Geometry Object Transforms

You can use the transforms to create rectangular and linear arrays of identical geometry objects and to move, rotate, mirror, and scale geometry objects. Mirroring, moving, rotating, and scaling are *affine transformations* applied to geometry objects. All transforms are available in all space dimensions, except Rotate, which is not applicable for 1D geometries.

TABLE 6-4: GEOMETRY OBJECT TRANSFORMS

ICON	NAME AND LINK	DESCRIPTION
	Array	Create an array of geometry objects.
	Copy	Copy a geometry object.
	Mirror	Mirror a geometry object in a plane (in 3D), a line (in 2D), or a point (in 1D).
	Move	Move a geometry object.
	Rotate	Rotate a geometry object about a centerpoint.
	Scale	Scale a geometry object about a centerpoint.

To add a geometry object transform node in the **Model Builder**, right-click the **Geometry** node () and select an option from the **Transforms** menu.

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. You can transform one or more geometry objects into a different type of object by *converting* it, for example, from a solid to a curve. The object is then called a composite object, which becomes a solid, face, curve, or point object depending on the target type. You can perform the following geometry object conversions:

	<p>For 2D models:</p> <ul style="list-style-type: none">• 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).
	<p>For 3D models:</p> <ul style="list-style-type: none">• 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).

The following geometry object conversion operations are available:

TABLE 6-5: GEOMETRY OBJECT CONVERSION OPERATIONS

ICON	NAME AND LINK	DESCRIPTION
	Convert to Solid	Convert a geometry object to a solid.
	Convert to Surface	Convert a geometry object to a surface (3D only).
	Convert to Curve	Convert a geometry object to a curve (2D, 3D).

TABLE 6-5: GEOMETRY OBJECT CONVERSION OPERATIONS

ICON	NAME AND LINK	DESCRIPTION
	Convert to Point	Convert a geometry object to a point.
	Convert to COMSOL . See the <i>CAD Import Module User's Guide</i> .	Available with the CAD Import Module. Converts an object to a COMSOL kernel representation (3D).

To add a conversion operation node in the **Model Builder**, click the **Geometry** node and select an object to convert in the **Graphics** window. Then click the **Convert to Solid** () , **Convert to Surface (3D)** () , **Convert to Curve** () , or **Convert to Point** () buttons on the Geometry toolbar. Alternatively, right-click **Geometry** and select an option from the **Conversions** menu.

General Geometry Operations

For 2D geometry objects, you can use geometry operations such as fillets and tangents to construct the geometry. There are also general geometry operations in all space dimensions for splitting geometry objects and to delete geometry objects or geometric entities (domains, boundaries, edges, and points). See [Geometry Operations](#) for details about all geometry operations. The following table provides an overview of available general geometry operations:

TABLE 6-6: GENERAL GEOMETRY OPERATIONS

ICON	NAME AND LINK	DESCRIPTION
	Chamfer	Create a chamfer at a corner of a 2D geometry object.
	Fillet	Create a fillet at a corner of a 2D geometry object.
	Tangent	Create a tangent from an edge to another edge or point in a 2D geometry.
	Cross Section	Create a 2D cross section from an intersection between a 3D geometry and a work plane.
	Split	Split one or several objects into their entities.
	Delete Entities	Delete entities (domains, boundaries, edges, or points) from the objects they belong to or to delete entire geometry objects,

TABLE 6-6: GENERAL GEOMETRY OPERATIONS

ICON	NAME AND LINK	DESCRIPTION
	Edit Object	Edit the properties of a 2D object.
	Import	Import geometry objects from a file or from another geometry.

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).

To split a geometry object, in the **Model Builder**, click the **Geometry** node, select objects in the **Graphics** window. Then click the **Split** button (). Alternatively, right-click **Geometry** and select **Split**.

The Geometry Toolbars

The Geometry toolbars are part of the main toolbar and have buttons to perform geometry operations. In addition, the Mesh toolbar is available when a geometry sequence node is selected in the **Model Builder**. This means that the Geometry toolbar disappears if you, for example, select a geometry node in the **Model Builder**. Use the **Toolbar** button () in the Main toolbar to make the Geometry toolbar for a specific geometry node available; click the associated menu arrow and choose the menu item corresponding to the geometry node. If you click the button associated with this menu button, COMSOL Multiphysics selects the node in the **Model Builder** corresponding to the last selected item in the menu associated with the menu button.

To use the buttons on the toolbar, first select geometry objects in the **Graphics** window. When you click one of the buttons, COMSOL performs the associated operation on the selected objects and creates they resulting objects. In addition, the software adds a node for the operation to the geometry sequence. If you want to modify the operation, you can edit and rebuild this node. You do not need to confirm the selections in the

Graphics window when working with the Geometry toolbar; it is sufficient to highlight them, a state that is indicated by the color red.

In 2D there are also buttons for drawing geometric primitives. To use these, first click the button and then draw the object in the **Graphics** window using the mouse. Then the primitive geometry object appears in the graphics. In addition, the software adds a node for the primitive to the geometry sequence. If you want to modify the primitive, you can edit and rebuild this feature.



Figure 6-5: The 3D, 2D, and 1D geometry toolbars.

TABLE 6-7: GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
●	Draw Point	ID, 2D	To draw a point, first click the Draw Point button, then click in the Graphics window.
○ + ↘ ↙	Draw Interval	ID	To draw an interval in 1D, first click the Draw Interval button, then click the start and end points in the Graphics window.

TABLE 6-7: GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
	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 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 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 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 node appears in the geometry sequence.
	Draw Rectangle (Center)	2D	
	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 node appears in the geometry sequence.
	Draw Circle (Center)	2D	

TABLE 6-7: GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
	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 node appears in the geometry sequence.
	Draw Ellipse (Center)	2D	
	Snap Coordinates	ID, 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 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	ID, 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	ID, 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.
	Difference	ID, 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	ID, 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.

TABLE 6-7: GEOMETRY TOOLBAR BUTTONS

BUTTON	NAME	DIMENSION	ACTION AND NOTES
	Convert to Point	ID, 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	ID, 2D, 3D	To split one or several objects into their entities, select them in the Graphics window, and then click the Split button.
	Delete	ID, 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	ID, 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.

The Measurements Page

The **Measurements** page is a tool to measure geometry objects and entities. You access it by right-clicking the Geometry node and selecting **Measurements** () from the context menu. You can, for example, measure the volume, area, or length of a selected domain, face, or edge.

GEOMETRY TYPE

Select to measure geometry objects or the finalized geometry.

SELECTION

From the **Geometric entity level** list, select **Object**, **Domain**, **Boundary**, **Edge**, or **Point**. Then select some objects or entities of the selected type. Under **Measurements** you find information about these objects or entities. For objects, this section contains their total number of domains, boundaries, edges, and points, and the geometry representation. For domains, boundaries, and edges, their total volume, area, or length appears. If you select a point, its coordinates are shown. If you select two points, their distance is shown.

You can also click the **Measure** () button on the geometry toolbar. The results then appear 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 (requires a license for the CAD Import Module) of an object.

Copying and Pasting Geometry Objects

There are two methods to copy and paste geometry objects.



Duplicate () is a convenient way to copy and paste in one step. In other words, it combines the **Copy** and **Paste** functions.

METHOD I

When using the copy/paste functionality, the copy 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 object (a rectangle or sphere, 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.



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** (for example, **Rectangle**, **Circle**, **Sphere**, or **Ellipse**) (). A copy of the node is added to the end of the geometry sequence.
- 3 Click the **Build Selected** button ().



It is also possible to copy/paste and duplicate nodes corresponding to operation features, such as the Union node.

METHOD 2—DISPLACED AND LINKED COPIES

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.



- [Copying, Pasting, and Duplicating Nodes](#)
- [Copy](#)

The Geometry Node

Under a **Geometry** node () you find the sequence of nodes that define the geometry for the model. The **Geometry** node also contains some general settings for the geometry such as the length unit. To open the settings window for a geometry, click the **Geometry** node in the Model Builder.

UNITS

Select the **Scale values when changing units** check box to scale the values for the geometric dimensions so that the geometric objects keep their physical size. The default setting is to not scale the values when changing units; the program then interprets the values for the geometric dimensions using the new units for length and angle. The values themselves do not change.

From the **Length unit** list select the length unit to use in fields for lengths and for visualization of the geometry. You can override the unit using the unit syntax to specify the length unit (for example, 13[mm]). When solving the model, all lengths are converted to the base unit for length. If you change the unit, COMSOL Multiphysics converts all pure numeric values in fields for lengths to the new unit, if you have selected the **Scale values when changing units** check box (see above).



For information about available length units and prefixes, see [Specifying Model Equation Settings](#).

Angular Unit

From the **Angular unit** list choose to use radians or degrees as the angular unit to use in fields for angles. You can override the unit by entering, for example, 0.3[rad]. The program assumes that numeric inputs and outputs of trigonometric functions are in radians. If you change the unit, all pure numeric values in fields for angles are converted to the new unit, if you have selected the **Scale values when changing units** check box (see above).

ADVANCED

Geometry Representation (3D Only)

The **Geometry representation** list controls which kernel (geometric modeler) that COMSOL uses to represent and operate on the geometry objects: COMSOL's own kernel (COMSOL kernel) or the CAD Import Module's kernel (Parasolid).

- If you choose **COMSOL kernel**, all objects are represented using COMSOL's kernel.
- If you choose **CAD Import Module kernel** (requires the CAD Import Module), all objects and operations that support the CAD Import Module's kernel use it. For example, Work Plane, Extrude, and Revolve operations do not support this kernel.

The default geometry representation is controlled by the preference setting

Geometry>Default geometry representation.

If the chosen geometry representation cannot be used—for example, because a feature requires the CAD Import Module’s kernel or because no CAD license is available—the behavior is controlled by the preference setting **Geometry>Switch kernel automatically**. If automatic switching is enabled, the software automatically switches to the other geometry kernel when needed. If automatic switching is disabled, an error message is generated if the selected kernel cannot be used.

When you change the **Geometry representation** setting, all nodes that support the CAD Import Module’s kernel are marked as edited with an asterisk (*) in the upper-right corner the node’s icon. To rebuild the geometry using the new kernel, click the **Build All** button () or the **Build Selected** button ().

Default Relative Repair Tolerance

This is the default value that is used when you add a feature that has a **Relative repair tolerance** field (for example, Boolean operations and conversions). Changing the **Default relative repair tolerance** does not affect the tolerances in existing features. Adjust the **Relative repair tolerance** if you experience problems with a Boolean 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.

Automatic Rebuild

The **Automatic rebuild** check box controls if the geometry sequence is automatically rebuilt when clicking on a node in the model tree outside the geometry sequence. The default value is controlled by the preference setting **Geometry>Automatic rebuild>Default in new geometries**. Select the **Automatic rebuild** check box to always rebuild the geometry, or clear it to prevent any automatic rebuilding of the geometry.

The Finalize Node—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*, which is used for meshing and analysis. 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 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 on interior boundaries between domains in the geometry. By default, COMSOL ensures continuity in the physics fields across interior boundaries. See also [Removing Interior Boundaries](#) below).



When forming a union for axisymmetric models, COMSOL removes all parts of the geometry from the $r < 0$ half plane.

- 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 it 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.

The **Finalize** node () ends each geometry sequence in 1D. In 2D and 3D, it is possible to add virtual operation nodes after the **Finalize** node. In the Model Tree, its label is **Form Union** or **Form Assembly** depending on its settings. By default, it unites all geometry objects into a single geometry object (this is the **Form Union** variant). You cannot delete or disable the **Finalize** node. When you leave the geometry sequence to define materials or physics, the **Messages** window provides information about the finalization of the geometry (forming a union or an assembly) and about the number of geometric entities (domain, boundaries, and so on) in the finalized geometry.

FINALIZE

The default finalization method, **Form a union**, forms a union of all geometry objects. Select **Form an assembly** from the **Finalization method** list if you do not want the geometry objects to be united. The program then forms the finalized geometry by collecting the objects as parts in an *assembly object*. If you form an assembly, select the **Create imprints** check box to get imprints of the parts of the assembly that touch each other. Select the **Create pairs** check box to generate pairs corresponding to the parts of the assembly that touch each other. From the **Pair type** list, select **Identity pair** (the default) to generate identity pairs or **Contact pair** to generate contact pairs. The contact

pairs are only useful for contact modeling in structural mechanics and require a license for the Structural Mechanics Module or the MEMS Module.

Adjust the value in the **Relative repair tolerance** field (default value: $1 \cdot 10^{-6}$) if you experience problems with the finalization operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of all geometry objects. Geometric entities with a distance less than the absolute repair tolerance are merged.



In 2D and 3D, virtual operation nodes can appear after the **Finalize** node.

Insert Sequence from File

To insert a geometry sequence from an MPH-file, right-click the **Geometry** node and select **Insert Sequence from File** () from the context menu. Browse to a file name and click **Open**. The file is scanned for geometry sequences having the right space dimension. If there is just one such sequence, its nodes are inserted into the geometry sequence after the current node. If the file contains more than one such sequence, a dialog box opens. Select the geometry sequence from the list of available sequences. Finally click **OK**. The nodes in the selected sequence are inserted into the geometry sequence after the current node.

If the geometry sequence contains references to functions or parameters, those functions and parameters are also inserted in the model under **Global Definitions**, or in case of nonglobal functions, under **Definitions** in the same model the geometry sequence is located.



Functions and parameters are inserted even if a function or parameter with the same name already exists in the model. You have to manually resolve any conflicts before the geometry sequence can be built.

Exporting Geometry to File

You can export geometry objects to a COMSOL binary file (.mphbin) or text file (.mphtxt).



2D geometry objects can also be exported to a DXF file.



3D geometry objects can also be exported to an STL file. If you have a license for the CAD Import Module, 3D geometry objects can also be exported to Parasolid files (.x_t, or .x_b) or ACIS files (.sat, or .sab).

To open the **Export** settings window, right-click the **Geometry** node, and select **Export to File** () from the menu. Then 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).

For STL file export you can select objects, domains, or boundaries to export. For the other file types you select the objects to export by first select Export selected objects and then add the objects to export to the **Selected object** list, or use the option **Export entire finalized geometry** to export the finalized geometry object.



To export a geometry to use in an earlier version of COMSOL in the COMSOL format, select a version from the **Compatible with version** list.

Click the **Export** button to export the selected geometry to the specified file. A confirmation message appears in the **Messages** window.



The .mphbin and .mphtxt formats do not contain unit information. When the exported file is imported into a geometry with a different length unit, you may need to use a **Scale** () feature to scale the imported objects to the correct size.

Composite Object (Backward Compatibility)

If you open a model created in the 3.5a version of COMSOL one **Composite Object** node () appears for each nonprimitive geometry object in the model. The **Composite Object** node contains the follow sections:

COMPOSITE OBJECT

If you save the model as a .java file, COMSOL uses the filename specified in the **Filename** field to determine the path to a geometry file, containing the geometry object, that appears together with the .java file. The software uses this geometry file when you run the resulting .java file. By default, the filename has the prefix \$FILENAMES\$. If the filename starts with this prefix, COMSOL stores the geometry file in the same directory as the .java file. It is also possible to remove this prefix and specify the full path to the geometry file.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the composite object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Creating a Geometry for Analysis

Techniques for Creating Geometries

Several techniques can ensure that a geometry results in a good mesh and gives reasonable solution times for the 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 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

Sometimes the modeling domain is unbounded or too large for successful analysis. For those cases a suitable boundary condition can replace the exterior of the domain.



It is important that the geometry is 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, 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 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 3D model you can also remove small details and prepare the geometry for efficient meshing 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. The stress value for a sharp corner is finite in the stress analysis, but refinement of the mesh increases the stresses in the corner without limit. To avoid a singularity, round sharp corners using fillets.

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 during the analysis. 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. It is then better to create the solid object by extruding a cross section or to use geometric 3D primitives.

Associative Geometry and Selections of Geometry Objects

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 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. COMSOL 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.

User-defined named *selection nodes* in the geometry sequence are useful to improve associativity compared to other selection nodes (see [Creating Named Selections in the Geometry Sequence](#)). You can refer to such selections defined in following geometry nodes (for example, as input objects). This applies both to selections created by the **Create selections** check box and selections created by selection nodes.

Selecting the Space Dimension

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. 2D models are easier to modify and generally solve much faster, so modeling mistakes are much easier to find when working in 2D. Once the 2D model is verified, you are in a better position to build a 3D model.



Not all physics are available in all space dimensions. See the documentation for the physics user interfaces in COMSOL and its modules for the supported space dimensions.

1D PROBLEMS



The following is a guide for 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 that represents the only space dimension where there is spatial (or other) variation.

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



The following is a guide for some of the common approximations made for 2D problems. Modeling in 2D often represents a 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).

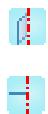
There are usually two approaches that lead to a 2D cross-sectional view of a problem:

- When 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 user 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)



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 user 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). COMSOL provides this is an option during postprocessing.

3 D PROBLEMS



This section discusses 3D geometry modeling practices.

Although COMSOL 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.

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.

THE COORDINATE SYSTEMS AND THE SPACE DIMENSION

COMSOL uses a global Cartesian or cylindrical (axisymmetric) coordinate system. You select the geometry dimension and coordinate system in the Model Wizard when starting a new model. The default 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 models.

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 , y (2D and 3D), 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 labeled r , and represented by the x -axis.

For axisymmetric cases the geometry model must fall in the positive half plane ($r \geq 0$).

-
- 
 - About Cylindrical Coordinate Systems
 - Coordinate Systems
 - 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:

TABLE 6-8: SPACE DIMENSION ICONS IN THE MODEL BUILDER

ICON	SPACE DIMENSION
	3D
	2D axisymmetric
	2D
	1D axisymmetric
	1D
	0D (space-independent models for chemical reactions and other ODEs/DAEs)

The nodes under a physics in the Model Builder and Plot Groups and Plots use different icons and colors to help visualize a model.

-
- 
 - Physics Nodes by Space Dimension
 - Color Coding for Plot Groups and Plot Types
-

Geometry Modeling Examples

The following short examples provide step-by-step instructions for creating 1D, 2D, and 3D geometries and for working with virtual geometry operations.

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 Multiphysics icon to launch COMSOL.
- 2 In the **Model Wizard** 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 Multiphysics. 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 6-6](#). 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.

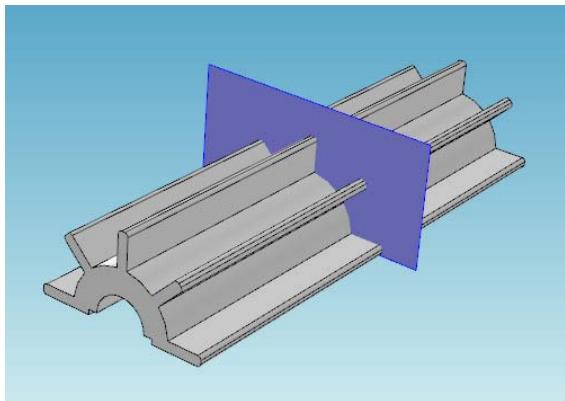


Figure 6-6: Example of a 3D heat sink model with cross section.

CREATING A BASIC 2D GEOMETRY MODEL

- 1 Double-click the COMSOL Multiphysics icon to launch COMSOL.
- 2 In the **Model Wizard** on the **Select Space Dimension** page, click the **2D** button.
- 3 Click the **Finish** button ().



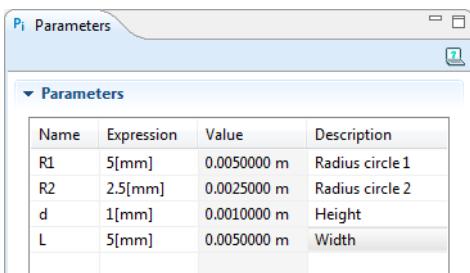
This model geometry can be used in a future example in the [Free Meshing](#) and [Structured Meshes](#) sections of this guide. It is recommended you save the model for this purpose.

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

The following steps explain how to create two circles to form the core of the heat sink in [Figure 6-6](#). 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** ().

- 2** In the **Parameters** table, enter, or copy and paste, the **Name**, **Expression**, and **Description**. The **Value** column automatically displays the **Expression** value.



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

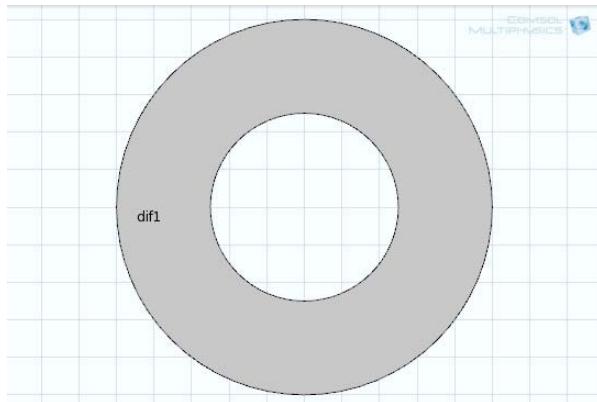
- 1** In the **Model Builder**, under **Model 1**, right-click **Geometry 1** and select **Circle** ().
- 2** Under **Size and Shape**, enter R1 in the **Radius** field.
- 3** Click the **Build Selected** button ().
- A circle (c1) with radius R1 displays in the **Graphics** window.
- 4** Right-click **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 ().

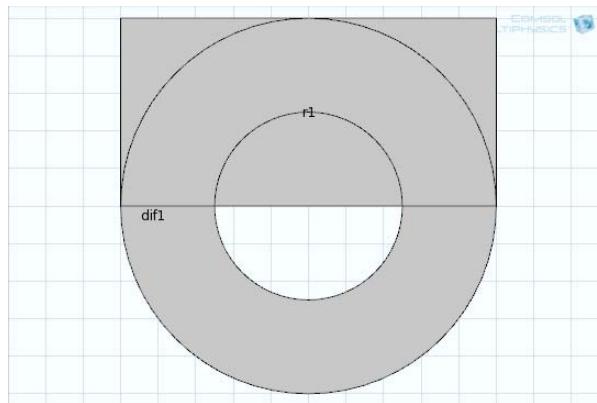
The object **dif1** is created by subtracting the smaller circle from the larger circle.



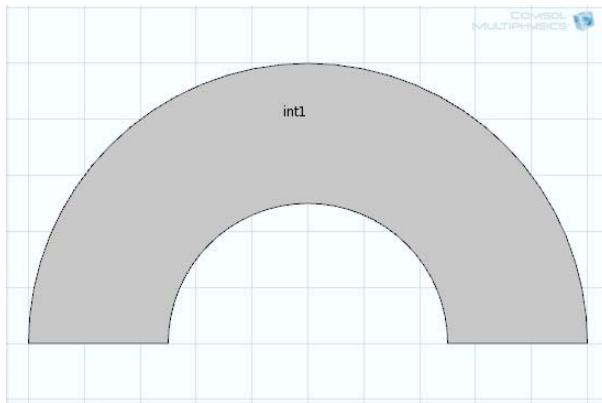
INTERSECTING WITH RECTANGLE

- 1 In the **Model Builder**, right-click **Geometry 1** and select **Rectangle** ().
- 2 Under **Size**: In the **Width** field enter $2*R1$, and 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**.

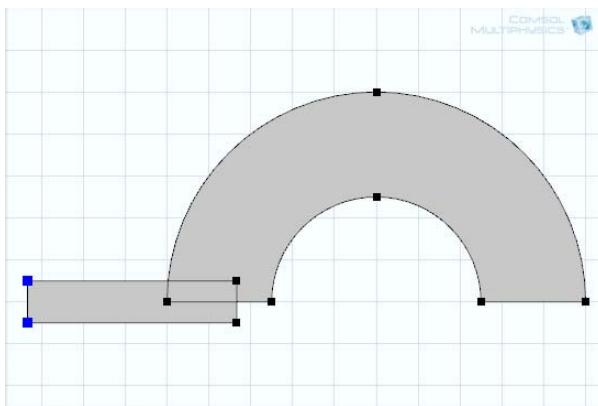


ADDING A RECTANGLE TO CREATE A FLANGE

- 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**: In the **x** field enter $-(2/3*R1+L)$, and in the **y** field enter $-d/2$.

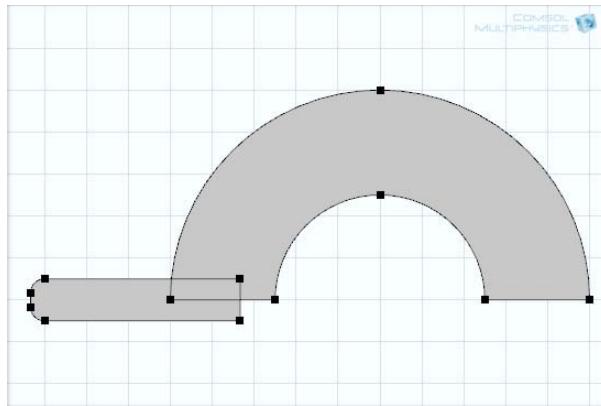
- 4 Click the **Build Selected** button ().

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 $fil1$.

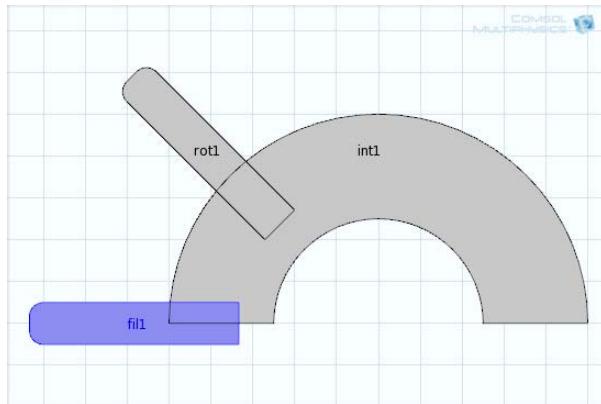


ADDING ROTATE OPERATIONS TO CREATE FIVE FLANGES

Rotate the flange 45 degrees and keep the original input object to create five flanges on top of the heat sink.

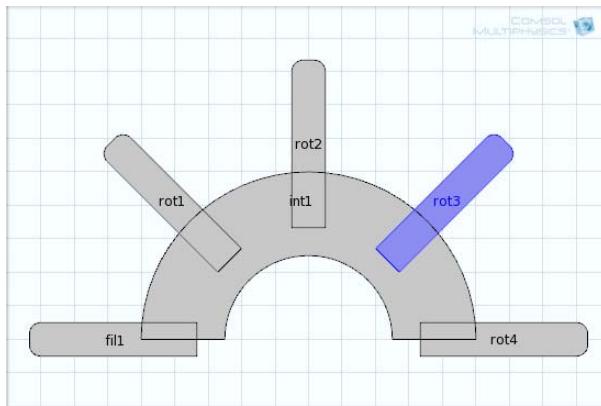
Adding Rotate 1 to Create Object Rot1

- 1 In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ().
- 2 Select object **fill1** (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 **Build Selected** () to create the object **rot1**. Click **Zoom Extents**().



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 define separate parts with different materials, for example.
- 4 Click the **Build All** button (). Click the **Zoom Extents** button (). The final geometry is shown in [Figure 6-7](#).

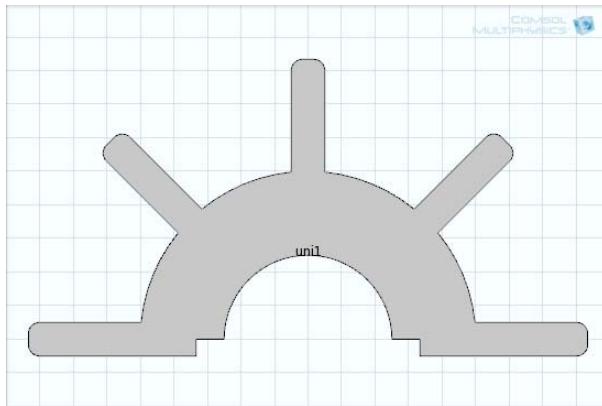


Figure 6-7: Final 2D object created in the Model Builder.

VIEWING THE GEOMETRY SEQUENCE

Figure 6-8 shows the geometry sequence used to create Figure 6-7. 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.

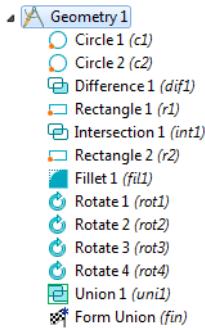


Figure 6-8: An example of a 2D geometry sequence.

RE-RUNNING THE GEOMETRY SEQUENCE WITH DIFFERENT PARAMETERS

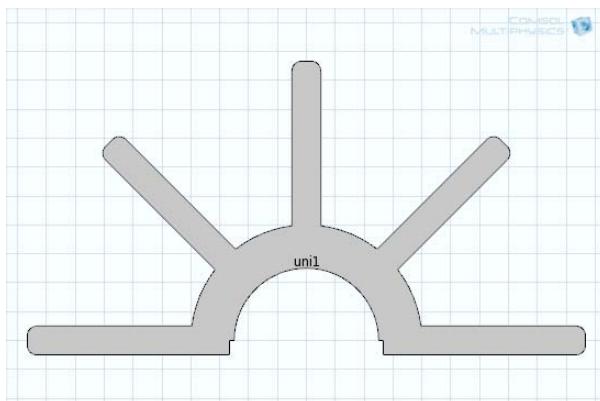
- | In the **Model Builder**, under **Global Definitions**, click **Parameters** ().

- 2** Under **Parameters**, enter the following settings in the table. Replace the previous data::.

NAME	EXPRESSION	VALUE	DESCRIPTION
R1	4 [mm]	0.0040 m	Radius Circle 1
R2	2.5 [mm]	0.0025 m	Radius Circle 2
d	1 [mm]	0.0010 m	Height
L	7e [mm]	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 6-9 shows the geometry of a heat sink used for cooling microprocessors. The following sections describe the steps to create this geometry and introduces 3D drawing tools and techniques.

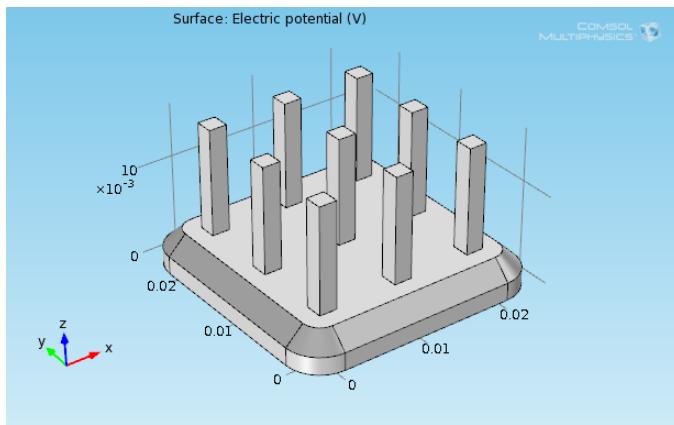


Figure 6-9: Example of a 3D heat sink model.

CREATING A BASIC 3D GEOMETRY MODEL

- 1 Double-click the COMSOL Multiphysics icon to launch COMSOL.
- 2 In the **Model Wizard** on the **Select Space Dimension** page, click the **3D** button.
- 3 Click the **Finish** button ().

CREATING PARAMETERS FOR GEOMETRY PARAMETERIZATION

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

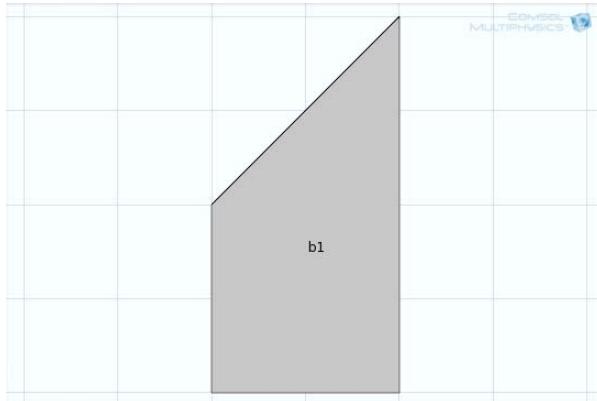
USING WORK PLANES TO CREATE A BÉZIER POLYGON

Use work planes to create 2D geometries that you can 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**: In row **1**, enter $-2e-3$ in the **xw** field, and 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. In row **2**, enter $-2e-3$ in the **xw** field, and 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.
- II Click **Close Curve** then click the **Build Selected** button () and the **Zoom Extents** button ().



REVOLVING A 2D OBJECT TO CREATE A 3D OBJECT

- I 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.

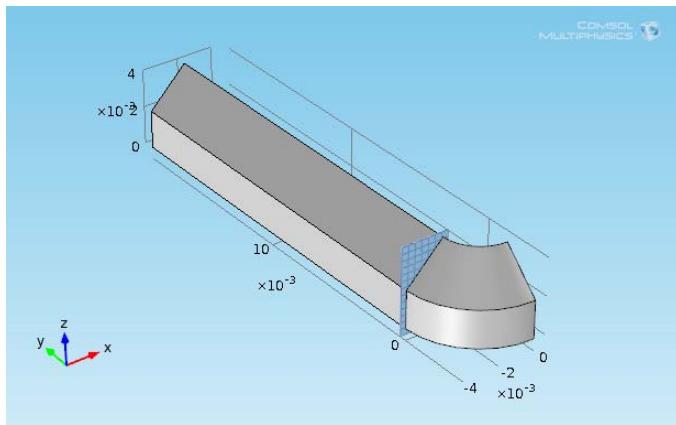


The **Revolution Axis** corresponds to the position of the *y*-axis in the work plane's 2D coordinate system.

- 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**.

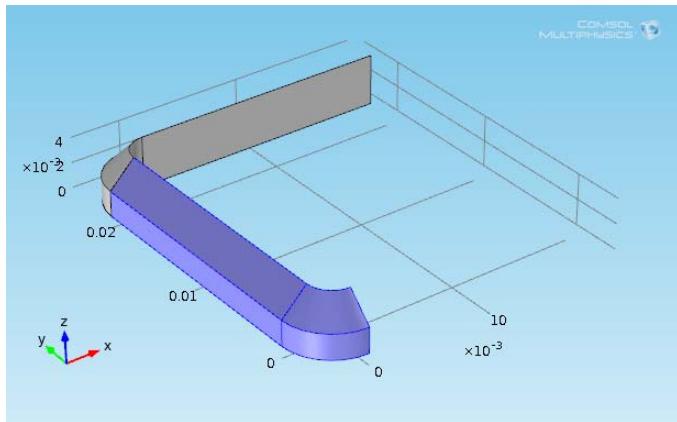


- 4 In the **Model Builder**, under **Model 1**, right-click **Geometry 1** and select **Boolean Operations>Union** ().
- 5 Select the objects **rev1** and **ext1** and add them to the **Input objects** section.
- 6 On the **Union** page, under **Union**, click to clear the **Keep interior boundaries** check box to remove the interior boundary between the corner section and the edge section.

- 7 Click the **Build Selected** button (). Objects **rev1** and **ext1** are combined to create object **unil**.

ADDING A ROTATION TO THE 3D OBJECT

- 1 In the **Model Builder**, right-click **Geometry 1** and select **Transforms>Rotate** ().
- 2 Select the object **unil** 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**: In the **x** field, enter **1e-2**, and in the **y** field, enter **1e-2**.
- 6 Click the **Build Selected** button () and the **Zoom Extents** button () to view the object **rot1**.

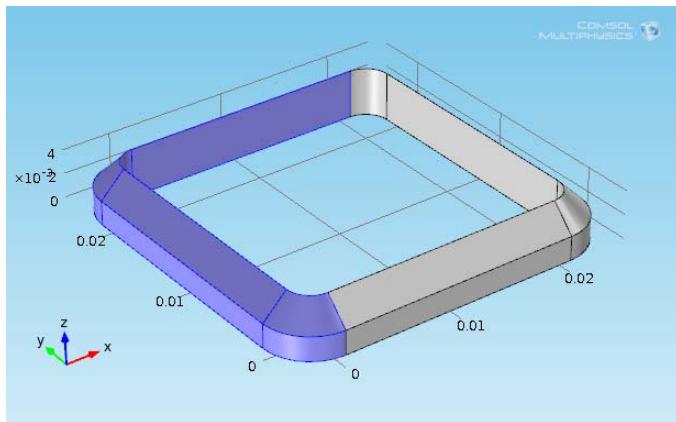


CREATING UNION 2

- 1 In the **Model Builder**, right-click **Geometry 1** and select **Boolean Operations>Union** ().
- 2 Select the objects **unil** and **rot1** 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 **uni2**.

ADDING A SECOND ROTATION

- 1 In the **Model Builder**, right-click **Geometry 1** 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**: In the **x** field, enter 1e-2, and in the **y** field, enter 1e-2.
- 6 Click the **Build Selected** button ().



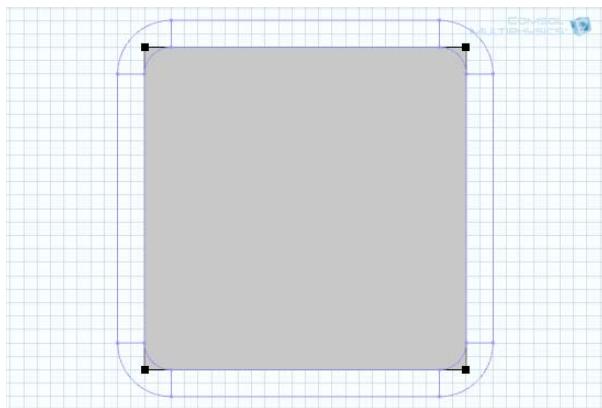
CREATING UNION 3

- 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

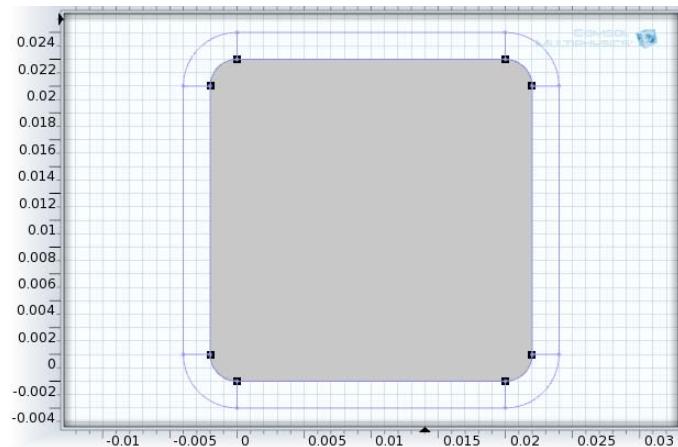
- 1 In the **Model Builder**, right-click **Geometry 1** and select **Work Plane**. The **Work Plane** page opens in the settings window and a **Work Plane 2** node is added in the **Model Builder**.
- 2 Click the **Build Selected** button ().

- 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**: Select **Center** from the **Base** list. Then in the **xw** field, enter $1e-2$, and in the **yw** field, enter $1e-2$.
- 7 Click the **Build Selected** button () and the **Zoom Extents** button ().



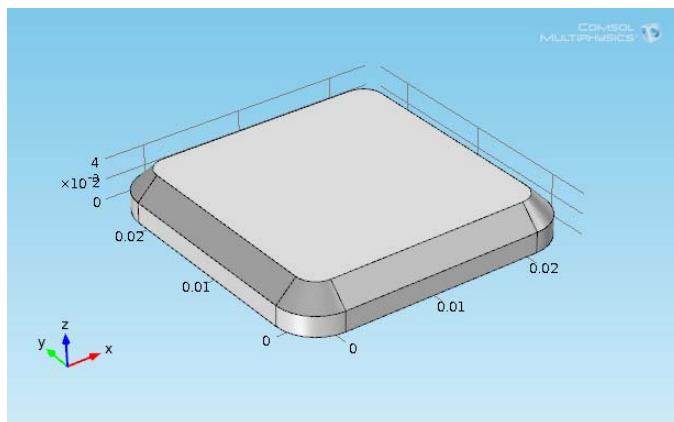
TRIMMING THE SQUARE TO FIT USING THE FILLET OPERATION

- 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.



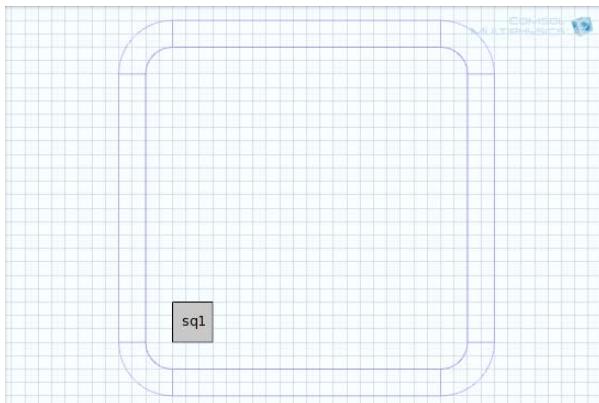
DRAWING THE UPPER PART OF THE HEAT SINK

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**.

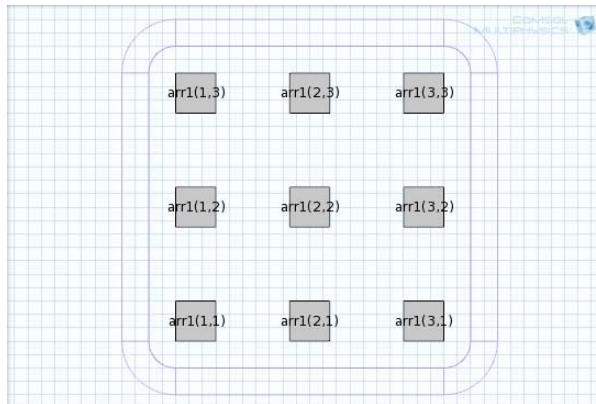
- 2 Under **Work Plane**, enter $4e-3$ in the **z-coordinate** field.
- 3 From the **3D projection** list, **Entire 3D geometry** is selected by default, and this visualizes the projected edges of the heat sink's base in the work plane.
- 4 Click the **Build Selected** button ().
- 5 To the right of 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 AN ARRAY OF PILLARS

- 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**: In the **xw size** field, enter 3, and in the **yw size** field, enter 3.

- 4 Under **Displacement**: In the **xw** field, enter $1e-2-L2/2$, and 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 6-10 shows the list of the geometry in the **Model Builder** used to create Figure 6-9. 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.

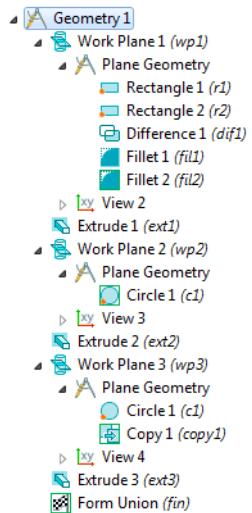


Figure 6-10: Sample 3D Model Builder Geometry Sequence. From the Busbar model in the COMSOL Multiphysics Model Library.

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.

Forming Composite Edges and Faces by Ignoring Vertices and Edges

This example of how to use virtual geometry operations 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.

MODEL WIZARD

1 Click **New** to open the **Model Wizard** and start a new model.

2 Click **Finish** ().

GEOMETRY I

Import I

1 In the **Model Builder**, right-click **Model I>Geometry I** and choose **Import** ().

2 Go to the settings window for Import. Under the **Import** section, click **Browse**.

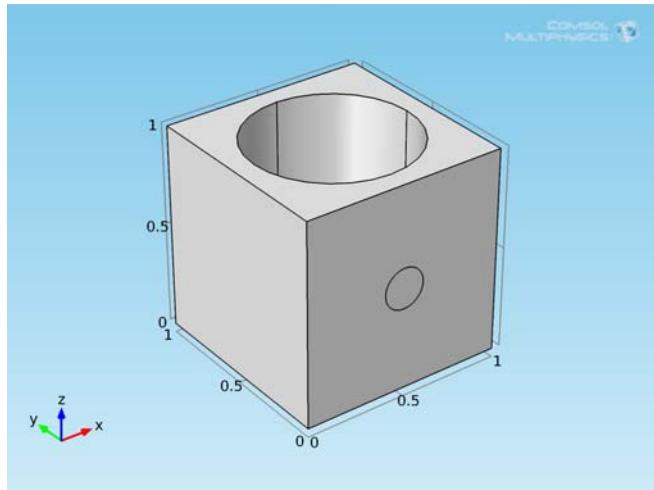
3 In the COMSOL installation directory navigate to the folder `models/COMSOL_Multiphysics/Tutorial_Models` and double-click `virtualgeom_demo_1.mphbin`.



The location of the file varies based on the installation. For example, if the installation is on your hard drive, the file path might be similar to `C:\Program Files\COMSOL43b\models`.

4 Click **Import**.

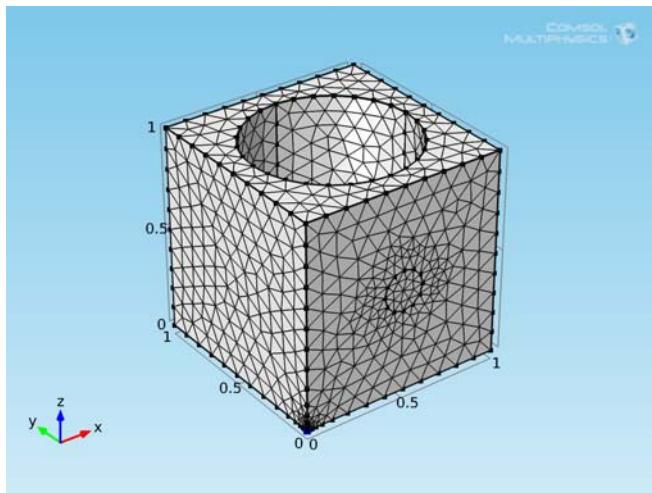
The imported geometry displays in the **Graphics** window.



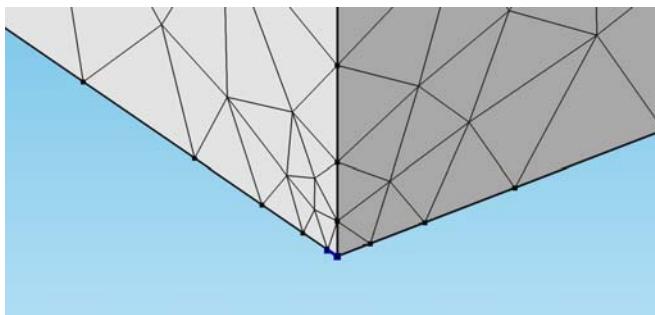
MESH I

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ().

The resulting mesh displays in the **Graphics** window.



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.



Eliminate the short edge by ignoring the vertex between this edge and its adjacent longer edge.

GEOMETRY I

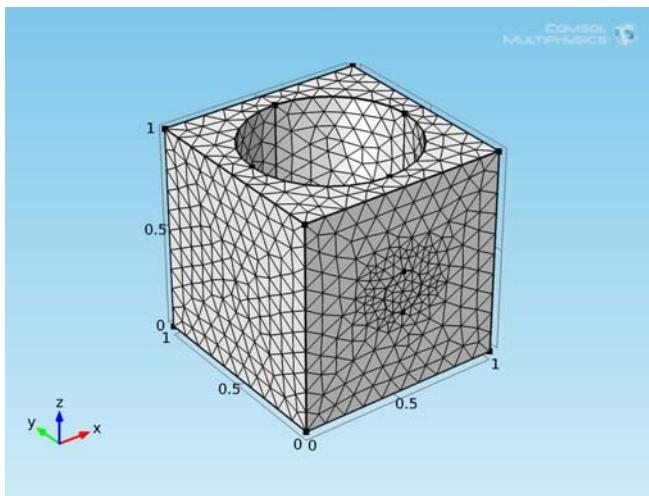
Ignore Vertices /

- 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 I

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** (.

The mesh displays in the **Graphics** window.



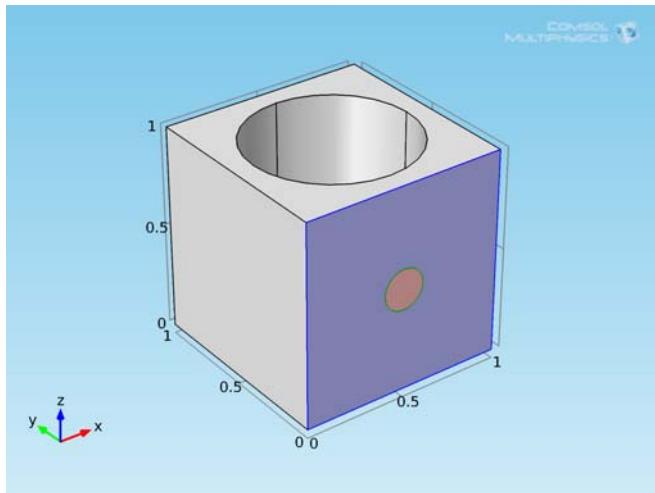
The geometry's domain is well suited for swept meshing.

Swept /

- 1 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 In 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 **COMSOL Error** window. COMSOL fails to create a swept mesh due to the circular imprint on one of the linking faces of the sweep.



Use the **Ignore Edges** operation to remove this imprint.

GEOMETRY I

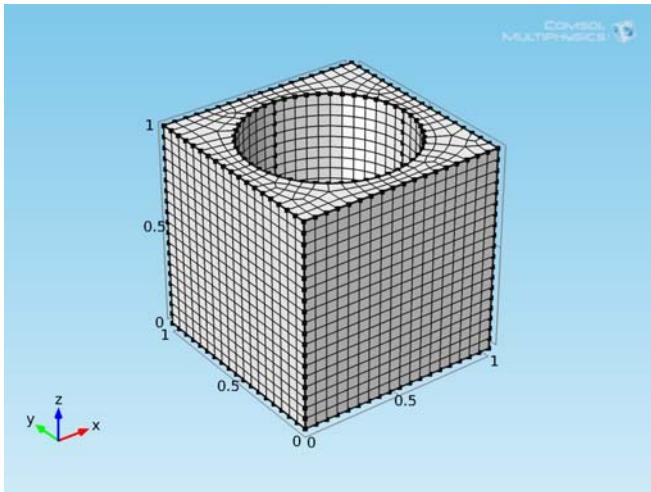
Ignore Edges /

- 1 In the **Model Builder**, right-click **Model 1>Geometry 1** and choose **Virtual Operations>Ignore Edges** ().
Make sure to select the **Ignore adjacent vertices** check box such that the vertices of the imprint disappear as well.
- 2 Select Edges 14 and 15.
3 Click the **Build Selected** button ().

MESH I

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ().

The swept mesh displays in the **Graphics** window.



You can achieve the same virtual geometry using **Form Composite Edges** and **Form Composite Faces** operations.

GEOMETRY I

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 I

In the **Model Builder**, right-click **Model I>Mesh I** and choose **Build All** ().

Merging Vertices by Collapsing Edge

This example of virtual geometry operations illustrates how you can use the Collapse Edges operation (or the Merge Vertices operation) to prepare the geometry for efficient meshing.

MODEL WIZARD

- 1 Go to the **Model Wizard**.
- 2 Click **Finish** ().

GEOMETRY I

Import I

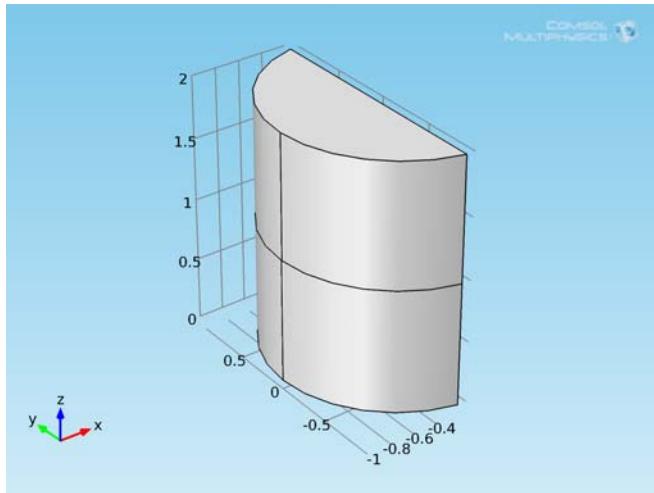
- 1 In the **Model Builder**, right-click **Model I>Geometry I** and choose **Import** ().
- 2 Go to the settings window for Import. Locate the **Import** section. Click **Browse**.
- 3 In the COMSOL installation directory navigate to the folder `models/`
`COMSOL_Multiphysics/Tutorial_Models` and double-click
`virtualgeom_demo_2.mphbin`.



The location of the file varies based on the installation. For example, if the installation is on your hard drive, the file path might be similar to
`C:\Program Files\COMSOL43b\models`.

- 4 Click **Import**.

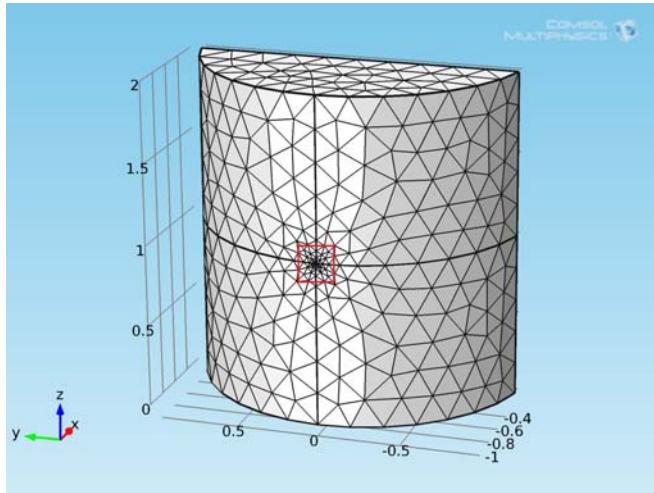
The imported geometry displays in the **Graphics** window.



GEOMETRY I

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 at the junction of the four curved faces.



Remove this short edge by collapsing it into a vertex.

Collapse Edges /

- 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 ().

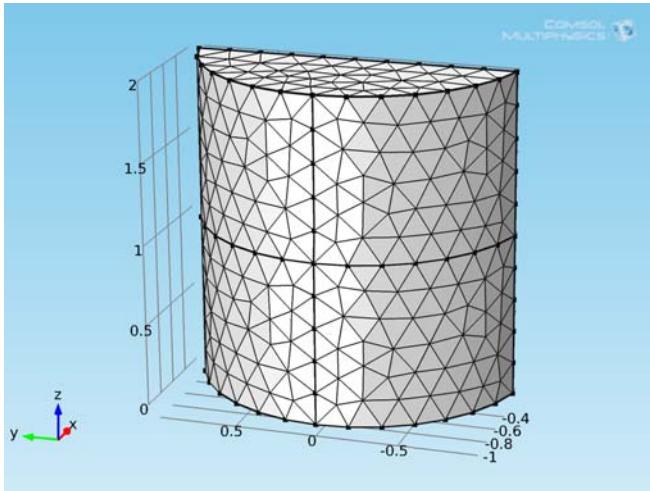


You can also remove this short edge by an **Ignore Edges** operation forming a composite face of the two adjacent faces. However, then the small distance between the vertices of the ignored edge remains.

MESH 1

In the **Model Builder**, right-click **Model 1>Mesh 1** and choose **Build All** ().

The mesh displays in the **Graphics** window.



You can achieve the same virtual geometry using a **Merge Vertices** operation.

GEOMETRY I

Right-click **Collapse Edges I** and choose **Disable** ().

Merge Vertices I

1 In the **Model Builder**, right-click **Model I>Geometry I** 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 I

In the **Model Builder**, right-click **Model I>Mesh I** and choose **Build All** ().

Geometric Primitives

The following sections describe the nodes for creating geometric primitives.

Bézier Polygon

A Bézier polygon consists of a sequence of connected line segments, quadratic Bézier curves (for example circular arcs), and cubic Bézier curves. See [About Rational Bézier Curves](#) below for some information about Bézier curves in general.

THE BÉZIER POLYGON NODE

To create a Bézier polygon, right-click a 2D **Geometry** node and select **Bézier Polygon** () or right-click a 3D **Geometry** node and select **More Primitives>Bézier Polygon**.

Then enter the properties of the Bézier polygon.

GENERAL

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the Bézier polygon is a solid object (only available in 2D) or a closed or open curve object. If you choose **Solid** or **Closed curve**, the software automatically adds a line segment if needed to close the polygon.



When using the geometry toolbar, the **Draw Solid** button () is available to toggle between drawing a solid object or an outline (or curve) instead of selecting a **Type**.

POLYGON SEGMENTS

Define the Bézier polygon by adding curve segments to the list of segments. Choose linear segments, quadratic segments, or cubic segments. Delete segments by selecting them and clicking **Delete**. To edit a segment, select it in the list. When editing the last segment, click **Close Curve** to make the last control point coincide with the first control point of the first segment.

Linear Segments

To add a linear segment, click **Add Linear**. Specify the start of the linear segment on the first row of coordinates under **Control points**. Specify the end of the linear segment on the second row of coordinates.

Quadratic Segments

To add a quadratic segment, click **Add Quadratic**. Specify the coordinates of the three control points on rows under **Control points**. Add the weights of the control points under **Weights**. The default weights— -1 , $1/(\sqrt{2})$, and 1 —correspond to a circular arc if the control points are three corners of a square.

Cubic Segments

To add a cubic segment, click **Add Cubic**. Specify the coordinates of the four control points on each row under **Control points**. Add the weights of the four control points under **Weights**. Cubic segments with self-intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries and points) that the Bézier polygon consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

ABOUT RATIONAL BÉZIER CURVES

A rational Bézier curve is a parameterized curve of the form

$$\mathbf{b}(t) = \frac{\sum_{i=0}^p \mathbf{b}_i w_i B_i^p(t)}{\sum_{i=0}^p w_i B_i^p(t)}, \quad 0 \leq t \leq 1$$

where the functions

$$B_i^p(t) = \binom{p}{i} t^i (1-t)^{p-i}$$

are the *Bernstein basis* functions of *degree p*; $\mathbf{b}_i = (x_1, \dots, x_n)$ are the control points of the n -dimensional space; and w_i are the weights, which should always be nonnegative numbers. The end-point interpolation property corresponds to $\mathbf{b}(0) = \mathbf{b}_0$ and $\mathbf{b}(1) = \mathbf{b}_p$. Another useful property of the rational Bézier curves is that the direction of the tangent vector at $t = 0$ and $t = 1$ is determined by the vectors $\mathbf{b}_1 - \mathbf{b}_0$ and $\mathbf{b}_p - \mathbf{b}_{p-1}$, respectively. That is, the curve is always tangent to the line connecting the control points \mathbf{b}_0 and \mathbf{b}_1 and the line connecting \mathbf{b}_{p-1} and \mathbf{b}_p . When joining curves at end

points, aligning the (nonzero) tangent vectors assures tangential continuity. This technique produces visually smooth transitions between adjacent curves.

Quadratic Curves (Conic Sections)

Rational Bézier curves of degree 2 can represent all conic sections: circles, ellipses, parabolas, and hyperbolas. Elliptical or circular curve segments are often called arcs. The conic sections are also called quadric curves or *quadrics*. Because the parameter t is constrained to be in the interval $[0, 1]$, only a segment of the conic section is represented. A 2nd degree curve consists of three control points and three weights. There is a simple rule for classifying a 2nd degree curve if the end point weights are set to 1, only allowing the central weight w_1 to vary: if $w_0 = w_2 = 1$, then $0 < w_1 < 1$ gives ellipses, $w_1 = 1$ gives parabolas, and $w_1 > 1$ gives hyperbolas. For a fixed control polygon, at most one value of w_1 (among the ellipses generated by letting $0 < w_1 < 1$) gives a circle segment. For example, a quarter of a full circle is generated by a control polygon with a right angle and with a central weight of $1/\sqrt{2}$.

Cubic Curves

Rational Bézier curves of degree 3 (cubic curves) have more dynamic properties than conic section curves. A cubic curve has four control points and four weights, making it possible to create a self-intersecting control polygon or a zigzag control polygon. A self-intersecting polygon may give rise to a self-intersecting curve (loop). Self-intersecting curves and cusps are not supported.

A zigzag control polygon generates an S-shaped curve containing a point of inflection where the tangent line lies on both sides of the curve.

Block

To create a block (box), right-click a 3D **Geometry** node and select **Block** (). Then enter the properties of the block using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the block is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the edge lengths in the **Width**, **Depth**, and **Height** fields. With the default axis (representing the z -axis) and no rotation, the width, depth, and height correspond to the dimensions in the x -, y -, and z -directions, respectively.

POSITION

Enter the position of the block using the **x**, **y**, and **z** fields. From the **Base** list, choose **Center** if the block is centered about the position, or choose **Corner** if the block has one corner in this position.

AXIS

Specify the direction of the block's third axis—that is, the direction of the edges corresponding to the height. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector in the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the block's third axis in the **Rotation** field. When this angle is zero (the default), the block's second axis is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. Specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes under **Layer position** to specify where to apply the layers (see the graphics to the right of the check boxes to see the definitions of the left, right, front, back, bottom, and top sides of the block).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the block consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Circle

To create a circle or disk, right-click a 2D geometry and select **Circle** . Then enter the properties of the circle using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the circle is a solid object (disk) or a curve object.



When using the geometry toolbar, the **Draw Solid** button () is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**.

SIZE AND SHAPE

Define the circle's radius in the **Radius** field. Enter a sector angle (in degree) for a circle sector in the **Sector angle** field. The default value is 360 degrees for a full circle.

POSITION

Enter the position of the circle using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the circle is centered about the position, or choose **Corner** if a surrounding box has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the position in the **Rotation** field. The default angle is 0 degrees.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric circles. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the circle consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Cone

To create a right circular cone or cone frustum (conical frustum, truncated cone), right-click a 3D **Geometry** and select **Cone** (). By adding a **Cone** feature you can create the part of a cone contained between two circular bases without going through an apex. Enter the properties of the cone using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the cone is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the cone in the **Bottom radius**, **Height**, **Specify top size using**, and **Semi-angle** or **Top radius** fields. From the **Specify top size using** list select **Radius** to specify the cone top size using the **Top radius** field. The top radius must be a positive value or 0 for a cone with a sharp apex. From the **Specify top size using** list select **Angle** (the default setting) to specify the cone top size using the **Semi-angle** field. The semi-angle is the angle a cone makes with the vertical axis. For the default cone with a bottom radius and height of 1, the default semi-angle is roughly 26.565 degrees, which makes the radius of the second (top) basis $1/2$. For the default radius and height the maximum semi-angle is 45 degrees (for a cone with a sharp apex). The maximum semi-angle depends on the values for the radius and height. The semi-angle must be larger than -90 degrees. Setting the semi-angle to 0 makes the cone into a cylinder.

POSITION

Enter the position of the cone using the **x**, **y**, and **z** fields. This is the center of the bottom circle.

AXIS

Specify the direction of the cone's axis. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cone's local coordinate system is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers to one or more sides of the cone. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cone consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Cylinder

To create a right circular cylinder, right-click a 3D **Geometry** node and select **Cylinder** (). Then enter the properties of the cylinder using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the cylinder is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the cylinder in the **Radius** and **Height** fields.

POSITION

Enter the position of the cylinder using the **x**, **y**, and **z** fields. This is the center of the bottom circle.

AXIS

Specify the direction of the cylinder's axis. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis.

Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose

Spherical to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cylinder's local coordinate system is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cylinder consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Eccentric Cone

To create an eccentric (oblique) cone or cone frustum with elliptic base, right-click a 3D **Geometry** node and select **More Primitives>Eccentric Cone** (). Then enter the properties of the eccentric cone using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the eccentric cone is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the eccentric cone in the **a-semiaxis**, **b-semiaxis**, **Height**, **Ratio**, **Top displacement 1**, and **Top displacement 2** fields. The bottom of the cone is an ellipse with semiaxes given in the **a-semiaxis** and **b-semiaxis** fields. The **Height** field determines the height of the cone frustum. The **Ratio** field controls the ratio between the perimeters of the top and bottom ellipses. To get an oblique cone, use the **Top displacement** fields to specify the displacement of the top ellipse's center relative to the bottom ellipse's center, in the cone's local coordinate system.

POSITION

Enter the position of the eccentric cone using the **x**, **y**, and **z** fields. This is the center of the bottom ellipse.

AXIS

Specify the direction of the third axis of the cone's local coordinate system—that is, the normal to the base ellipse. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the cone's local coordinate system is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the eccentric cone consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Ellipse

To create an ellipse, right-click a 2D geometry and select **Ellipse** (). Then enter the properties of the ellipse using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the ellipse is a solid object or a curve object.



When using the geometry toolbar, the **Draw Solid** button () is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**.

SIZE AND SHAPE

Define the ellipse's semiaxes in the **a-semiaxis** and **b-semiaxes** fields. Enter a sector angle (in degree) for an ellipse sector in the **Sector angle** field. The default value is 360 degrees for a full ellipse.

POSITION

Enter the position of the ellipse using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the ellipse is centered about the position, or choose **Corner** if a surrounding box has one corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the base point in the **Rotation** field. The default angle is 0 degrees.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric ellipses. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, edges, and points) that the ellipse consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the Model Tree.

Ellipsoid

To create an ellipsoid, right-click a 3D **Geometry** node and select **More Primitives>Ellipsoid** (). Then enter the properties of the ellipsoid using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the ellipsoid is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the semiaxes of the ellipsoid in the **a-semiaxis**, **b-semiaxis**, and **c-semiaxis** fields.

POSITION

Enter the position of the ellipsoid's center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the ellipsoid's third axis—that is, the principal axis corresponding to **c-semiaxis**. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector in the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the ellipsoid's third axis in the **Rotation** field. When this angle is zero (the default), the ellipsoid's second axis is parallel to the *xy*-plane.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric ellipsoids. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the ellipsoid consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Helix

To create a helix (coil), right-click a 3D **Geometry** node and select **More Primitives>Helix** (). Then enter the properties of the helix using the sections in the **Helix** node's settings window described below. The **Helix** node creates a helix with a circular cross section.



To create a helix with a noncircular cross section, define the cross section using a work plane. Define the helix centerpoint as a 3D curve using a **Parametric Curve** node, and then use a **Sweep** node to sweep the cross section from the work space along the parametric curve to create the helix.

OBJECT TYPE

From the **Type** list, select **Solid** (the default) to create a solid helix, or select **Surface** to create a hollow helix that consists of surfaces only.

SIZE AND SHAPE

This section contains a number of properties that determine the size and shape of the helix.

The **Number of turns** field contains a positive number. The default value is 3 turns.

There are two radii:

- The **Major radius** (SI unit: m) field is the radius from the center of the helix (the default is 1 m).
- The **Minor radius** field (SI unit: m) is the radius of the cross section (the default is 0.1 m). The **Minor radius** can be zero, in which case a curve object is created. You

can use this together with the **Sweep** feature to create helices with noncircular cross sections.

There are two pitches:

- The **Axial pitch** field (SI unit: m) determines the axial distance between similar positions on two consecutive turns of the helix (the default is 0.3 m).
- The **Radial pitch** field (SI unit: m) determines the radial distance between similar positions on two consecutive turns of the helix (the default is 0, which means that each turn has the same radius).

Select **Right handed** or **Left handed** from the **Chirality** list. The *chirality* or handedness of the helix can be either right handed (the default) or left handed. For a right handed helix, a clockwise screwing motion moves the helix away from the observer; for a left handed helix, a clockwise screwing motion moves it toward the observer.

From the **End caps** list, select an option to create the end caps of the helix:

- Select **Parallel to axis** (the default) to create end caps that are parallel to the helix axis.
- Select **Perpendicular to axis** to create end caps that are perpendicular to the helix axis.
- Select **Parallel to spine** to create end caps that are parallel to the spine of the helix.

POSITION

This is the center position for the starting turn of the helix. Enter the coordinates in the **x**, **y**, and **z** fields. The default position is the origin.

AXIS

Select the **Axis type—x-axis, y-axis, z-axis, Cartesian, or Spherical**.

- Select **x-axis**, **y-axis**, or **z-axis** (the default) to define the axis direction parallel to one of the coordinate axes.
- Select **Cartesian** to define the axis direction using Cartesian coordinates in the **x**, **y**, and **z** fields. The default axis is in the *z*-direction (0, 0, 1).
- Select **Spherical** to define the axis direction using spherical coordinates θ and ϕ (angles of inclination and azimuth, respectively) in the **theta** and **phi** fields. The default angles are 0.

ROTATION ANGLE

Rotate the helix around its axis by entering an angle in the **Rotation** field. The default value is 0 degrees.

ADVANCED SETTINGS

By default, the **Twist compensation** check box is selected, which prevents the twisting that would otherwise occur due to nonzero torsion for curves that do not belong to a fixed plane. Twist compensation rotates the base circle during the sweep along the helix curve by an amount equal to the integral of the curve torsion.



Twist compensation affects the position of the vertices on the top side of the helix. Clear the **Twist compensation** check box to turn it off. See [Figure 2-6](#) below.

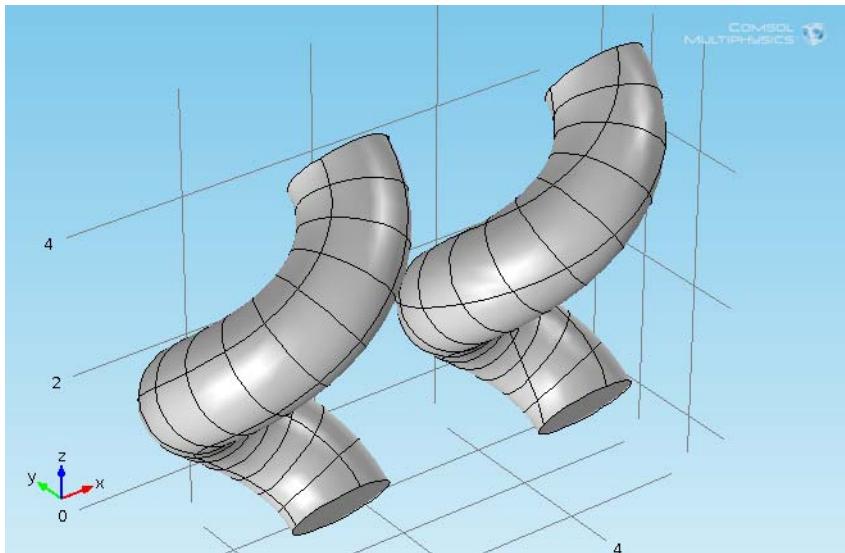


Figure 6-11: The helix on the left has twist compensation (the default). For the helix on the right, twist compensation has been turned off.

From the **Geometry representation** list, select **Spline** (the default) to represent the helix using splines, or **Bézier**, to represent the helix using Bézier curves. The difference is that using Bézier curves, the intersections between the surfaces that form the helix are visible edges, whereas they are hidden when using splines.

The values in the **Relative tolerance** field is a relative tolerance that controls the accuracy of the geometric representation of the helix. The geometric representation is an approximation, which is necessary because it is not possible to exactly represent a helix

using NURBS (nonuniform rational basis splines). The default value is 10^{-4} (or 0.01%).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the helix consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Hexahedron

To create a hexahedron bounded by bilinear faces, right-click a 3D **Geometry** node and select **Hexahedron** (). Then enter the properties of the hexahedron using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the hexahedron is a solid object or a (hollow) surface object.

VERTICES

Define the position, size, and shape of the hexahedron by specifying the coordinates of its vertices. Vertices 1–4 are the vertices of the bottom face in clockwise order. Vertices 5–8 are the vertices of the top face in clockwise order.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the hexahedron consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Interpolation Curve

An interpolation curve consists of a curve that interpolates or approximates a sequence of points. To create an interpolation curve, right-click a 2D **Geometry** node and select **Interpolation Curve** () or right-click a 3D **Geometry** node and select **More Primitives>Interpolation Curve**. Then enter the properties of the interpolation curve using the following sections:

INTERPOLATION CURVE

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the interpolation curve is a solid object (only available in 2D) or a closed or open curve object. If **Solid** or **Closed curve** is selected, a point is automatically added if needed to close the curve, and the curve has continuous first and second derivatives everywhere.

From the **Data source** list, select **Table** to specify the points to interpolate in a table in the settings window. This is the default data source.

From the **Data source** list, select **Vectors** to specify the points to interpolate as vectors (lists) in the fields **x**, **y**, and (3D only) **z**; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. Each field can contain a list of numbers or expressions containing parameters, separated with commas or spaces. Click the **Range** button () to use the **Range** dialog box for specifying the vector of values for each coordinate.

From the **Data source** list, select **File** to read the points to interpolate from a text file. Specify the file name in the **Filename** field or click the **Browse** button. If **Data format** is **Spreadsheet**, the file must be a text file with the number of columns equal to the dimension of the geometry sequence, and one row for each data point. The columns can be separated by space, tab, comma, or semicolon characters. If **Data format** is **Sectionwise**, the file must be in the sectionwise COMSOL postprocessing data format (see [Sectionwise Data Format](#)). Click the **Import to Table** button to copy the file contents into the data point table and change the **Data source** to **Table**.

If **Data source** is **File**, changes in the file do not automatically cause the interpolation curve feature to be rebuilt. To rebuild the feature after a change in the file, click the **Rebuild with Current File** button.

In the **Relative tolerance** field, enter the maximum allowed distance between the generated curve and the sequence of points. The default value 0 implies that the curve interpolates all points. If the relative tolerance is larger than 0, the curve does not necessarily interpolate all points, but the first and last points are interpolated.

Curves with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also if two different parts of the curve touch, even if they do not intersect.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of the following types: domains, boundaries, edges, and points) that the interpolation curve consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Interval

To create one or several intervals, right-click a 1D geometry and select **Interval** (). Then enter the properties of the interval using the following section:

INTERVAL

Enter the coordinates of the endpoints of the interval in the **Left endpoint** and **Right endpoint** fields. To get an object consisting of a sequence of connected intervals, change **Number of intervals** from **One** to **Many**, and enter a comma-separated list of coordinates in the **Points** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all domains and points that the interval consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Parametric Curve

A parametric curve is a curve in 2D and 3D where you use a parameter to define the coordinates of the curve. For example, the coordinates $(s \cdot \cos(s), s \cdot \sin(s))$ for a parameter s that runs from 0 to 10π defines a spiral in 2D. To create a parametric curve, right-click a 2D geometry and select **Parametric Curve** ()**,** or a 3D geometry and select **More Primitives>Parametric Curve** (). Then enter the properties of the parametric curve using the following sections:

PARAMETER

Define the parameter name in the **Name** field (default name: **s**). Also define the interval for the parameter values in the **Minimum** (default: 0) and **Maximum** (default: 1) fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each spatial coordinate in the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (3D only) **z** fields. To create the spiral described earlier with the parameter **s**, type **s*cos(s)** in the **x** field and **s*sin(s)** in the **y** field.



Self-intersecting curves are not supported, except for closed curves (that is, when the start and end points coincide).

By default, the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (in 3D) **z** expressions define the coordinates of points on the curve in the standard coordinate system. It is, however, possible to change this using the settings in the **Position**, **Axis** (3D only), and **Rotation Angle** sections. This is useful if you have created a parametric curve with the right shape but want to move it to another position or orientation. These settings can be thought of as defining a local coordinate system in which the parametric curve is defined.

Curves with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also if two different parts of the curve touch, even if they do not intersect. Simple closed curves are allowed though.

POSITION

Enter the position of the local coordinate system origin using the **x**, **y** (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes), and (3D only) **z** fields.

AXIS

In 3D, enter the axis that you want to rotate the local coordinate system about. The axis can be chosen parallel to one of the coordinate axes or entered in Cartesian or spherical coordinates. The *z*-axis of the local coordinate system is parallel to this axis.

ROTATION ANGLE

Enter the angle you want the local coordinate system to be rotated (default: 0 degrees). In 2D the local coordinate system is rotated about its origin. In 3D, the local coordinate system is rotated about its *z*-axis, which is parallel to the axis defined in the previous section.

ADVANCED SETTINGS

Internally, the software represents the parametric curve by a B-spline, which is computed to approximate the mathematical curve defined by the **x**, **y** in 2D, **r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes, and **x**, **y**, and **z** in 3D expressions. The number of knot points in the spline increases automatically until the curve approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field. The tolerance is measured relative to the space diagonal of the bounding box of the parametric curve.

If the coordinate expressions contain user-defined functions, changes in those functions do not automatically cause the parametric curve feature to be rebuilt. To

rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries or edges, and points) that the parametric curve consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Parametric Surface

A parametric surface is a surface in 3D where you use two parameters to define the coordinates of the surface. For example, the coordinates $(s_1 \cdot \cos(s_2), s_1 \cdot \sin(s_2), s_2)$ for a parameter s_1 that runs from 0 to π , and a parameter s_2 that runs from -1 to 1 define a “twisted rectangle.” To create a parametric surface, right-click a 3D geometry and select **More Primitives>Parametric Surface** (). Then enter the properties of the parametric surface using the following sections:

PARAMETERS

Define the parameter names in the **Name** fields (default names: **s1** and **s2**). Also define the intervals for the parameter values in the **Minimum** (default: 0) and **Maximum** (default: 1) fields.

EXPRESSIONS

Enter the expressions that define the functions of the parameter for each spatial coordinate in the **x**, **y**, and **z** fields. To create the twisted rectangle described earlier with the parameters s_1 and s_2 , type **s1*cos(s2)** in the **x** field, **s1*sin(s2)** in the **y** field, and **s2** in the **z** field.

By default, the **x**, **y**, and **z** expressions define the coordinates of points on the surface in the standard coordinate system. It is, however, possible to change this using the settings in the **Position**, **Axis**, and **Rotation Angle** sections. This is useful if you have created a parametric surface with the right shape but want to move it to another position or orientation. These settings can be thought of as defining a local coordinate system in which the parametric surface is defined.

Surfaces with self intersections might look correct when displayed but are not handled correctly by other geometry and meshing operations. This applies also to surfaces where one edge touches the surface or another edge, and to surfaces with singular points. If necessary, several parametric surfaces can be combined to overcome this

limitation. For example, constructing a cylindrical shell by typing `cos(s1)` in the **x** field, `sin(s1)` in the **y** field, and `s2` in the **z** field, where `s1` runs from 0 to 2π , and `s2` runs from 0 to 1, is incorrect because two edges of the parametric surface touch each other. Instead, use two parametric surfaces, with the same coordinate expressions, and where `s1` runs from 0 to π in the first surface and from π to 2π in the second one.

POSITION

Enter the position of the local coordinate system origin using the **x**, **y**, and **z** fields.

Axes

Enter the axis that you want to rotate the local coordinate system about. The axis can be chosen parallel to one of the coordinate axes or entered in Cartesian or spherical coordinates. The *z*-axis of the local coordinate system is parallel to this axis.

ROTATION ANGLE

Enter the angle you want the local coordinate system to be rotated (default: 0 degrees). The local coordinate system is rotated about its *z*-axis, which is parallel to the axis defined in the previous section.

ADVANCED SETTINGS

Internally, the software represents the parametric surface by a B-spline, which is computed to approximate the mathematical surface defined by the **x**, **y**, and **z** expressions. The number of knot points in the spline increases automatically until the surface approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field. The tolerance is measured relative to the space diagonal of the bounding box of the parametric surface.

If the coordinate expressions contain user-defined functions, changes in those functions do not automatically cause the parametric surface feature to be rebuilt. To rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries—that is, the parametric surfaces—, edges, and points) that the parametric surface consists of. These selections are available in all applicable selection lists (as, for example, **Parametric Surface 1 {geom1_psl_edg}** for a selection that includes the parametric surface's edges) but do not appear as separate selection nodes.

Point

To create one or several points, right-click a 1D, 2D, or 3D geometry and select **Point** (●) (in 3D: **More Primitives>Point**). Then enter the properties of the point using the following section:

POINT

Define the position of the point by entering its coordinates in fields labeled **x**, **y** (2D and 3D), and **z** (3D); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To get several points, enter a list of coordinates in each of these fields. Separate the coordinates with commas or blanks.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create a predefined selection for the point. This selection is available in all selection lists for points but does not appear as a separate selection node in the model tree.

Polygon

A polygon consists of a sequence of connected line segments. To create a polygon, right-click a 2D **Geometry** node and select **Polygon** (Polygon icon) or right-click a 3D **Geometry** node and select **More Primitives>Polygon**. Then enter the properties of the polygon using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid**, **Closed curve**, or **Open curve** to specify if the polygon is a solid object (only available in 2D) or a closed or open curve object. If you choose **Solid** or **Closed curve**, the program adds a line segment if needed to close the polygon.



When using the geometry toolbar, the **Draw Solid** button (Solid icon) is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**.

COORDINATES

From the **Data source list**, you can choose from three different data sources for the coordinates:

- Select **Vectors** (the default) to specify the coordinates of the vertices as vectors (lists) in the fields **x**, **y**, and **z** (3D only); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work

planes. Each field can contain a list of numbers or expressions containing parameters, separated with commas or spaces. Click the **Range** button () to use the **Range** dialog box for specifying the vector of values for each coordinate.

- Select **Table** to specify the coordinates of the vertices in a table directly in the settings window.
- Select **File** to read vertex coordinate data from a text file where each row represents the x , y , and (in 3D) z coordinates for a vertex in the polygon. Specify the filename in the **Filename** field, or click the **Browse** button. The file must be a text file with the number of columns equal to the dimension of the geometry sequence and one row for each data point. The columns can be separated by space, tab, comma, or semicolon characters. Click the **Import to Table** button to copy the file contents into the data point table and change the **Data source** to **Table**. Changes in the file do not automatically cause the polygon to be rebuilt. To rebuild the node after a change in the file, click the **Rebuild with Current File** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of the following types: domains, boundaries, edges, and points) that the polygon consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Pyramid

To create a rectangular pyramid or pyramid frustum, right-click a 3D **Geometry** node and select **More Primitives>Pyramid** (). Then enter the properties of the pyramid using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the pyramid is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the pyramid in the **Base length 1**, **Base length 2**, **Height**, **Ratio**, **Top displacement 1**, and **Top displacement 2** fields. The **Base length** fields determine (default: 1) the side lengths of the bottom rectangle. The **Height** field (default: 1) determines the height of the pyramid frustum. The **Ratio** field (default: 0.5) controls the ratio of the perimeters of the top and bottom rectangles. To get an oblique pyramid, use the **Top displacement** fields (default: 0) to specify the

displacement of the top rectangle's center relative to the bottom rectangle's center, in the pyramid's local coordinate system.

POSITION

Enter the position of the pyramid using the **x**, **y**, and **z** fields. This is the center of the bottom rectangle.

AXIS

Specify the direction of the third axis of the pyramid's local coordinate system—that is, the normal to the base rectangle. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the pyramid's local coordinate system (corresponding to **Base length 2**) is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the pyramid consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Rectangle

To create a rectangle, right-click a 2D geometry and select **Rectangle** (). Then enter the properties of the rectangle using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the rectangle is a solid object or a curve object.



When using the geometry toolbar, the **Draw Solid** button () is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**.

SIZE

Define the size of the rectangle in the **Width** and **Height** fields.

POSITION

Enter the position of the rectangle using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the rectangle is centered about the position, or choose **Corner** if the rectangle has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle (default: 0 degrees) about the position the **Rotation** field.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table, and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the rectangle consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Sphere

To create a sphere or ball, right-click a 3D **Geometry** node and select **Sphere** (). Then enter the properties of the sphere using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the sphere is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the radius of the sphere in the **Radius** field.

POSITION

Enter the position of the sphere's center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the third axis of the sphere's local coordinate system. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero (the default), the second axis of the sphere's local coordinate system is parallel to the **xy**-plane.

LAYERS

Layers can be used to create sandwich primitives by adding several concentric spheres. You specify the thicknesses and, optionally, names of each layer in the **Layers** table. The outermost layer comes first. The layers are positioned inside the sphere's radius.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the sphere consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Square

To create a square, right-click a 2D geometry and select **Square** (). Then enter the properties of the square using the following sections:

OBJECT TYPE

From the **Type** list, select **Solid** or **Curve** to specify if the square is a solid object or a curve object.



When using the geometry toolbar, the **Draw Solid** button () is available to toggle between drawing a solid object or an outline (or curve) instead of selecting an **Object Type**.

SIZE

Define the size of the square in the **Side length** field.

POSITION

Enter the position of the square using the **x** and **y** fields (**r** and **z** in 2D axial symmetry, **xw** and **yw** in work planes). From the **Base** list, choose **Center** if the square is centered about the position, or choose **Corner** if the square has a corner at the position.

ROTATION ANGLE

Specify the counterclockwise rotational angle about the position the **Rotation** field.

LAYERS

Layers can be used to create sandwich primitives by adding layers on one or more sides. You specify the thicknesses of layers in the **Layers** table and optionally a name for each layer. The outermost layer comes first. Select the check boxes to specify where to apply the layers.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, and points) that the square consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Tetrahedron

To create a tetrahedron, right-click a 3D **Geometry** node and select **More Primitives>Tetrahedron** ().

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the tetrahedron is a solid object or a (hollow) surface object.

VERTICES

Define the position, size, and shape of the tetrahedron by specifying the coordinates of its vertices. Vertices 1–3 are the vertices of the bottom face in clockwise order. Vertex 4 is the top vertex.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the tetrahedron consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Torus

To create a torus, right-click a 3D **Geometry** node and select **More Primitives>Torus** ().

OBJECT TYPE

From the **Type** list, select **Solid** or **Surface** to specify if the torus is a solid object or a (hollow) surface object.

SIZE AND SHAPE

Define the size and shape of the torus in the **Major radius**, **Minor radius**, **Revolution angle** and **Interior faces** fields. The **Major radius** (default: 1) field controls the radius of the directrix circle. The **Minor radius** (default: 0.5) field controls the radius of generatrix circle (the cross section). To get less than a full revolution (360 degrees, the default), use the **Revolution angle** field. Select the **Interior faces** check box to create cross-sectional faces that partition the domain of a **Solid** torus.

POSITION

Enter the position of the torus' center using the **x**, **y**, and **z** fields.

AXIS

Specify the direction of the third axis of the torus' local coordinate system—that is, the normal to the plane of directrix circle. From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

ROTATION ANGLE

Specify the rotational angle about the axis in the **Rotation** field. When this angle is zero, the second axis of the torus' local coordinate system is parallel to the *xy*-plane.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the torus consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Geometry Operations

This section describes the available geometry operations. See also [Virtual Geometry Operations](#) and [Mesh Control Operations](#) for additional operations that you can add to the finalized geometry.

Array

To create rectangular or linear array of identical geometry objects, right-click a geometry and select **Transforms>Array** (). Then enter the properties of the array operation using the following sections:

INPUT

Select the geometry objects that you want to duplicate in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Array** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

SIZE

In 2D and 3D, you get a rectangular (in 2D) or three-dimensional (in 3D) array by default. Enter the number of duplicates in each coordinate direction in the **x size**, **y size**, and **z size** fields; **r size** and **z size** in 2D axial symmetry; **xw size** and **yw size** in work planes.

To create a linear array of objects in 2D or 3D, change **Array type** to **Linear**. Enter the number of duplicates in the **Size** field.

In 1D, enter the number of duplicates in the **Size** field.

DISPLACEMENT

Set the displacement in each coordinate direction in the **x**, **y**, and **z** fields (not all fields are available in 1D and 2D).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Chamfer

To chamfer corners in 2D geometry objects, right-click a geometry and select **Chamfer** (). Then enter the properties of the chamfer operation using the following sections:

POINTS

Select the points (vertices) that you want to chamfer in the Graphics window. They then appear in the **Vertices to chamfer** list. If the geometry sequence includes user-defined selections above the **Chamfer** node, choose **Manual** to select points, or choose one of the selection nodes from the list next to **Vertices to chamfer**.

DISTANCE

In the **Distance from vertex** field, enter the distance from the vertex to the endpoints of the chamfer segment.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Compose

To create a composite geometry object from other geometry objects using Boolean operations given in a set formula, right-click a geometry and select **Boolean Operations>Compose** (). Then enter the properties of the compose operation as a set formula using the following section:

COMPOSE

Select the geometry objects that you want to compose in the Graphics window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Compose** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected solid geometry objects for further geometry operations.

In the **Set formula** field, enter a set formula involving the names of the selected geometry objects—for example, $r1+c1*(c2-r2)$ —to take the union of $r1$ and the

object that is the result of the intersection between c1 and the set difference where r2 is subtracted from c2. Use the binary operations +, *, and - for set union, set intersection, and set difference, respectively. The precedence of the operators + and - are the same. The operator * has higher precedence. You can override the precedence rules using parentheses. When you change the set formula, the **Input objects** selection is automatically updated.

To create a geometry object without interior boundaries, clear the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the compose operation. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the input objects. The value is relative to the overall dimensions of the geometry. For example, if the dimensions are in meters, the default repair tolerance makes the geometry repair heal gaps that are smaller than a micrometer (10^{-6} m). Geometric entities that have a distance less than the absolute repair tolerance are merged.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting composite geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Convert to Curve

To unite and convert geometry objects to single a curve object, right-click a geometry and select **Conversions>Convert to Curve** (). Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries or edges, and points) that the resulting curve object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Convert to Point

To unite and convert geometry objects to single a point object, right-click a geometry and select **Conversions>Convert to Point** (). Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create a predefined selection for the points that the resulting geometric object consists of. This selection is available in all selection lists for points but does not appear as a separate selection node in the model tree.

Convert to Solid

To unite and convert geometry objects to single a solid object, right-click a geometry and select **Conversions>Convert to Solid** (). Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains and all or some of boundaries, edges, and points) that the resulting solid object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Convert to Surface

To unite and convert geometry objects to single a surface object, right-click a geometry and select **Conversions>Convert to Surface** (). Then enter the properties of the convert operation using the following section:

INPUT

Select the geometry objects that you want to convert in the Graphics window. The objects appear in the **Input objects** list.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Adjust the **Relative repair tolerance** 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries, edges, and points) that the resulting surface object consists of. These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the Model Tree.

Copy

To create copies of geometry objects, right-click a geometry and select **Transforms>Copy** ().



To copy geometry objects without displacements, you can also right-click the geometry object (for example, **Rectangle** or **Sphere**) and select **Copy** () . Then right-click the **Geometry** node and select **Paste {Rectangle or Sphere}** ().

Enter the properties of the copy operation using the following sections:

INPUT

Select the geometry objects that you want to copy in the Graphics window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Copy** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Clear the **Keep input objects** check box to remove the input objects.

DISPLACEMENT

Set the displacement in each direction by entering **x**, **y**, and **z** (not all fields are available in 1D and 2D); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To create several copies, enter a comma-separated or space-separated list of displacements in these fields, or click the **Range** button () to use the **Range** dialog box for specifying a range of displacements for multiple copies.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.



- [Copying and Pasting Geometry Objects](#)
- [Copying, Pasting, and Duplicating Nodes](#)

Cross Section

In a **Work Plane** node's **Plane Geometry** sequence you can add a **Cross Section** node (). By default, this computes the cross section of all 3D objects generated by preceding nodes in the geometry sequence. You can also select specific 3D objects to intersect with the work plane. You can also add a new 2D or 2D axisymmetric **Model** and add the **Cross Section** node there. In that case you can select which **Work Plane** to use. For example, if you have a 3D geometry that is symmetric about an axis, you can add a work plane that contains the axis. In the axisymmetric 2D model you then get the cross section and can use a 2D axisymmetric physics, which is computationally efficient compared to a full 3D model. A **Cross Section** node can also be useful to extract a planar surface for modeling a thin flat 3D structure using shell elements, for example. To add a cross section, right click a **Plane Geometry** node under a **Work Plane** node or a 2D **Geometry** node and select **Cross Section** (). Then enter the properties of the cross section using the following sections:

CROSS SECTION

If you add the **Cross Section** node to a 2D or 2D axisymmetric geometry, first select the work plane to use for the cross section from the **Work plane** list.

From the **Intersect** list, choose **All objects** (the default) to intersect all 3D geometry objects with the work plane, or choose **Selected objects** to intersect only the geometry objects that you add to the **Objects to intersect** list that appears.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the cross section 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (domains, boundaries, edges, and points) that the cross-sectional geometry consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Deformed Configuration

To create a geometry from a (deformed) mesh, under **Results>Data Sets**, right-click a **Solution** data set and select **Remesh Deformed Configuration** (). The new geometry is added under **Meshes** as a **Deformed Configuration** node (). In the settings window of

this node, the **Time** or **Parameter value** list controls which solution is used to generate the deformed configuration. If you change the time or parameter value, or if the solution itself has changed, update the deformed configuration by clicking the **Update** button. The meshes that belong to the deformed configuration appear as child nodes under the deformed configuration node. Thus, to remesh the deformed configuration, right-click such a mesh node and select **Build All**.

-
- 
 - [Remeshing a Deformed Mesh](#)
 - [Solution \(Data Set\)](#)
-

Delete Entities

To delete geometry objects or geometric entities from objects, right-click a geometry and select **Delete Entities** (). Then enter the properties of the delete operation in the **Input** section.



ENTITIES OR OBJECTS TO DELETE

From the **Geometric entity level** list, choose the level of the entities to delete: **Object**, **Domain**, **Boundary** (that is, faces in 3D and edges in 2D), **Edge** (3D only), or **Point**. Then select the objects or entities that you want to delete in the **Graphics** window or use the **Selection List** window. The objects appear in the **Selection** list when you have confirmed (locked) the selection in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Delete Entities** node, choose **Manual** to select objects or entities, or choose one of the selection nodes from the list next to **Selection**.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Difference

To subtract geometry objects from other geometry objects to make a set difference, right-click a geometry and select **Boolean Operations>Difference** (). Then enter the properties of the difference operation using the following section:

INPUT

Activate the **Objects to add** list by selecting its **Activate Selection** button () , and select the objects that you want to add in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Difference** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Objects to add**.

Activate the **Objects to subtract** list by selecting its **Activate Selection** button () , and select the objects that you want to subtract in the **Graphics** window. If the geometry sequence includes user-defined selections above the **Difference** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Objects to subtract**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the difference 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Edit Object

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.



Editing 2D Geometry Objects

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.

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)**, **(xw (m), yw (m))** in work planes)—where the **m** in parentheses indicates the current geometry length unit; in this case the default unit: meter—and **Weights**, and under the table for the **Degree** list and the **Start vertex** and **End vertex** fields.

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. Click **Delete** to delete the current edge. Deleting an edge also deletes its adjacent vertices, if these vertices are not connected to other edges.

Click in the table cells to edit the **x (m)**, **y (m)**, **(xw (m), yw (m))** in work planes) 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.

Select the **Degree—Linear**, **Quadratic**, or **Cubic** to change the degree of the edge. When decreasing the degree, the control points are recalculated so as to approximate the old shape of the edge.

Under **Start vertex**, click the **Activate Vertices** 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 **Edge** 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.



When the **Edit Object** node is the current node, you can visualize the edited object by observing the edge or vertex numbers displayed in the **Graphics** window next to the edges or vertices.

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.

- Click **New** to create a new vertex; the coordinates for the new vertex are set to 0.
- 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.

Under **Coordinates** edit the **x** and **y** fields as required.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Extrude

To extrude planar objects, right-click a geometry or a work plane feature, and select **Extrude** (). Then enter the properties of the extrude operation.

GENERAL

From the **Extrude from** list, select **Faces** to extrude planar faces from the 3D geometry. Select the faces that you want to extrude in the **Graphics** window. They appear in the

Input faces list. All selected faces must lie in the same plane. Alternatively, from the **Extrude from** list, select **Work plane** to extrude objects from a work plane. In the **Work plane** list, select the work plane to extrude from. Select the objects that you want to extrude in the Graphics window. They appear in the **Input objects** list.

Select the **Unite with input objects** check box to unite the input objects with the extruded objects. Clear the **Unite with input objects** check box to keep the extruded objects separate from the input objects.

DISTANCES FROM PLANE

Specify one or several distances in the table. These determine the height above the work plane or planar faces for the top of each layer. The **Reverse direction** check box reverses the direction of the extrusion.

If you extrude several layers, remove the interior boundaries by clearing the **Keep cross-sectional faces** check box.

The direction arrow that appears in the Graphics window indicates the length of each extrusion distance.

SCALES

For each layer, specify a length scaling factor for the top of the layer relative to the work plane object or planar faces.

DISPLACEMENTS

For each layer, specify a displacement vector for the top of the layer in the work plane's coordinate system, or the local coordinate system defined by the first face to extrude. The first face is the face with smallest face number in the geometry object that comes first in the geometry sequence.

TWIST ANGLES

For each layer, specify a rotation angle for the top of the layer around the work plane's normal vector or the normal vector of the first face to extrude. The first face is the face with smallest face number in the geometry object that comes first in the geometry sequence.

POLYGON RESOLUTION OF EDGES

This setting determines how accurately the edges in the extrusion direction are represented.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Fillet

To fillet corners in 2D geometry objects, right-click a geometry and select **Fillet** (). Then enter the properties of the fillet operation using the following sections:

POINTS

Select the points (vertices) that you want to fillet in the **Graphics** window. They then appear in the **Vertices to fillet** list. If the geometry sequence includes user-defined selections above the **Fillet** node, choose **Manual** to select points, or choose one of the selection nodes from the list next to **Vertices to fillet**.

RADIUS

Enter the **Radius** of the circular fillet arc.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Import

To import geometry objects from a file or from another geometry, right-click a geometry and select **Import** (). Then enter the properties of the import feature in the **Import** section of the settings window for the **Import** node.

IMPORT

In the **Geometry import** list choose the type of data to import—**Any importable file**, **Geometry sequence**, and **COMSOL Multiphysics file** are always available. In addition, you can choose **DXF file** in 2D and **STL/VRML file** in 3D.



If you have license for the CAD Import Module, you also have **3D CAD file** in 3D (see the *CAD Import Module User's Guide* for more information about supported CAD file formats).



Use the ECAD Import Module (if you have a license for the AC/DC Module, MEMS Module, or RF Module) to use the alternative **ECAD file (GDS/NETEX-G)** and **ECAD file (ODB++)** import formats in 2D and 3D. See the *ECAD Import Module User's Guide* or go to <http://www.comsol.com/products/multiphysics/> for more information.



Combining several objects where at least one is imported from an STL/VRML file is not supported.

For file import, specify the file name in the **Filename** field or click the **Browse** button. For import from another geometry, select the geometry sequence from the **Source** list below. For DXF, STL/VRML, and 3D CAD import, you can change a number of properties when you have selected the file type. To import the file, click the **Import** button (). If you have changed some property, the software automatically re-imports the file when you click a build button.

Properties for DXF import

The repair tolerance specifies the largest distance between the end points of curves allowed in the imported geometry. You can specify this tolerance as an import option.

In the **Layer selection** list, select the layers to import.

Under **Import options**, select **Form solids** to unite and convert all objects in each layer to a solid object, select **Knit curves** to unite and convert all objects in each layer to a curve object, or select **Do not knit** to do nothing.

If the **Repair imported objects** check box is selected, enter a **Relative repair tolerance**. To create a geometry for mesh generation and finite element analysis, COMSOL requires a high degree of accuracy within the CAD drawing. Sometimes DXF geometries

contain small gaps and exceedingly short edges that make it impossible to create a valid 2D solid or a valid mesh. COMSOL provides repair tolerance settings to remove short edges and close small gaps during DXF file import. The absolute repair tolerance is the relative repair tolerance times the maximum coordinate of the imported objects (the default value is 10^{-5}). Geometric entities that have a distance less than the absolute repair tolerance are merged.

Properties for STL/VRML import

You can import 3D surface meshes into COMSOL with STL and VRML 1.0, two formats for describing solids as surface meshes:

- The STL format describes the boundary of a solid as a triangular mesh. STL files are either ASCII or binary, and COMSOL can import both ASCII and binary STL files. It is advisable to use binary files if possible to avoid a loss of precision.
- The VRML format describes scenes (collections of independent objects that reside within a single “frame” or logical space), and it can describe the geometry in a scene as a triangular mesh. COMSOL ignores all other information in a VRML file. COMSOL only imports VRML 1.0 files.

The imported objects are represented using COMSOL’s geometry kernel, so you cannot apply CAD defeaturig or repair on them.

These two formats do not contain any information about what face each triangle belongs. Instead, COMSOL uses heuristics to partition the triangles into faces. Usually the partition is reasonable for geometries containing sharp angles between faces, but the partitioning can appear arbitrary for curved geometries such as spheres and tori.

The mesh triangles are automatically partitioned into faces so that you get edges where neighboring triangles make a large angle. To control this algorithm, change **Face partitioning** to **Manual**. The angle between any two triangles in the same face is less than the **Maximum angle within face** value. The angle between neighboring triangles in the same face is less than the **Maximum face neighbor angle** value.

If the **Detect planar faces** check box is selected, (approximately) planar faces are detected. A planar face has an area larger than the total area of all faces times the value **Minimum relative area**. The angle between neighboring triangles in the same planar face is less than the value in the **Maximum neighbor angle** field.

A small face is removed if its area is less than the mean area of all faces times the value in the **Removal of small faces** field.

Advanced STL/VRML import parameters

If you select **Manual** from the **Advanced parameters** list, some additional controls for detection of extruded faces and faces with constant curvature become visible.

In an extruded face all triangles are approximately orthogonal to the extruded plane (work plane). The angle between such a triangle's normal and the extruded plane is less than the value in the **Maximum angle to extruded plane** field. An extruded face has an area larger than the total area of all faces times the value in the **Detect extruded faces** field.

In a face with constant curvature, the relative deviation of the curvature at neighboring triangles is at most the value in the **Maximum curvature deviation in face** field. A face with constant curvature has an area larger than the total area of all faces times the value in the **Detect constant curvature** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. For ECAD import, if more than one layer is imported, the **Create selections** check box additionally creates individual selections for each imported layer. For other types of import, if more than one object is imported, the **Create selections** check box creates individual selections for each imported object. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.



See the *ECAD Import Module User's Guide* or go to <http://www.comsol.com/products/multiphysics/> for more information.

Intersection

To create the intersection of geometry objects, right-click a geometry and select **Boolean Operations>Intersection** (). Then enter the properties of the intersection operation using the following section:

INPUT

Select the geometry objects that you want to intersect in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined

selections above the **Intersection** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

To create a geometry object without interior boundaries, clear the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default value: 10^{-6}) if you experience problems with the intersection 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Mirror

To mirror (reflect) geometry objects in a plane (3D), a line (2D), or a point (1D), right-click a geometry and select **Transforms>Mirror** (). Then enter the properties of the mirror operation using the following sections:

INPUT

Select the geometry objects that you want to reflect in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Mirror** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

POINT ON PLANE/LINE OF REFLECTION



Specify a point to be fixed during reflection by entering **x**, **y**, and **z**.



NORMAL VECTOR TO PLANE/LINE OF REFLECTION



Specify a vector in the direction to reflect by entering **x**, **y**, and **z**.



POINT OF REFLECTION



Specify the coordinate of the point of reflection in the **x** field.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Move

To move (translate) geometry objects, right-click a geometry and select

Transforms>Move (⊕). Then enter the properties of the move operation using the following sections:

INPUT

Select the geometry objects that you want to move in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined

selections above the **Move** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

DISPLACEMENT

Set the displacement in each direction by entering **x**, **y**, and **z** (not all fields are available in 1D and 2D geometries); **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. To create several copies, enter a comma-separated or space-separated list of displacements in these fields, or click the **Range** button () to use the **Range** dialog box for specifying a range of displacements for moving multiple copies.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Partition

The **Partition** node () provides a way to partition geometry objects as a Boolean operation. Partitioning a geometry objects can be useful to create separate domains or to introduce an interior boundary, for example. Using the **Partition** node you can partition a target object using a set of *tool objects* (geometry objects that are only used to partition—or tool—other geometry objects) or using an (infinite) plane defined by a **Work Plane** node (you do not need to draw anything in the work plane). To add it to a model, right-click a geometry and select **Boolean Operations>Partition** (). Then enter the properties of the partitioning operation using the following sections:

PARTITION

In the **Objects to partition** list, add the geometry objects that you want to apply a partition operation on. You might have to first click the **Activate Selection** button ().

From the **Partition with** list, select **Objects** (the default) to partition using the geometry objects that you add to the **Tool objects** list below, or select **Work plane** to partition using any of the added work planes.

- If you select **Objects**, add the geometry objects that you want to use as tool object to the **Tool objects** list. You might have to first click the **Activate Selection** button

(). Those geometry objects are only used to partition the geometry objects in the **Objects to partition** list and are not included in the analyzed (finalized) geometry used for defining materials and physics.

- If you select **Work plane**, select from the available work planes in the **Work plane** list. Click the **Go to Source** button () to move to the **Work Plane** node for the selected work plane.

Select the **Keep input objects** check box to use the selected geometry objects to for further geometry operations.

Adjust the **Relative repair tolerance** (default: 10^{-6}) if you experience problems with the partition 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Revolve

To revolve planar objects into 3D, right-click a geometry or a work plane node, and select **Revolve** (). Then enter the properties of the revolve operation.

GENERAL

From the **Revolve from** list, select **Faces** to revolve planar faces from the 3D geometry. Select the faces that you want to revolve in the Graphics window. They appear in the **Input faces** list. All selected faces must lie in the same plane. Alternatively, from the **Revolve from** list, select **Work plane** to revolve objects from a work plane. In the **Work plane** list, select the work plane to revolve from. Select the objects that you want to revolve in the Graphics window. They appear in the **Input objects** list.

Select the **Unite with input objects** check box to unite the input objects with the revolved objects. Clear the **Unite with input objects** check box to keep the revolved objects separate from the input objects.

REVOLUTION ANGLES

Click the **Angles** button to specify the start angle (default: 0 degrees) and end angle (default value: 360 degrees; that is, a full revolution) of the revolution. If you click the **Full revolution** button you get a full revolution. This selection also enables the **Keep original faces** check box that controls if the original faces are kept in the revolved 3D geometry. Keeping these faces is necessary if you want to create a swept mesh. By default, COMSOL keeps such faces.

REVOLUTION AXIS

Select **2D** in the **Axis type** list to specify the revolution axis in the local coordinate system. When revolving work plane objects, the local coordinate system is defined by the work plane's coordinate system. When revolving planar faces, the local coordinate system is defined by the face with the smallest face number in the first geometry object in the geometry sequence. Select **3D** in the **Axis type** list to specify the revolution axis in the 3D coordinate system.

POINT ON THE REVOLUTION AXIS

Specify a point on the revolution axis in the local coordinate system in the **xw** and **yw** fields. Alternatively, if **Axis type** is **3D**, specify a point on the revolution axis in the 3D coordinate system in the **x**, **y**, and **z** fields.

DIRECTION OF REVOLUTION AXIS

Specify a direction vector for the revolution axis in the local coordinate system in the **xw** and **yw** fields. Alternatively, if **Axis type** is **3D**, specify a direction vector for the revolution axis in the 3D coordinate system in the **x**, **y**, and **z** fields.

POLYGON RESOLUTION OF EDGES

This setting determines how accurately the edges in the revolution direction are represented.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Rotate

To rotate geometry objects, right-click a 3D or 2D geometry node and select **Transforms>Rotate** (). You can create one or multiple rotated copies with varying rotation angle. Then enter the properties of the rotate operation:

INPUT

Select the geometry objects that you want to rotate in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Rotate** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

ROTATION ANGLE

Specify the rotational angle in the **Rotation** field (default: 0 degrees). To get several rotated objects, enter a list of angles separated with commas or spaces or using the range function. Click the **Range** button () to define a range of angles using the **Range** dialog box. For example, `range(0,45,315)` creates eight objects, one at the original position and seven rotated copies at 45 degrees distance around a full 360 degrees circle.

CENTER OF ROTATION



Enter the center of the rotation in the **x** and **y** (for 2D) and **r** and **z** (in 2D axial symmetry) and 2D axisymmetric models for **xw** and **yw** in work planes fields.



POINT ON AXIS OF ROTATION



Enter a point on the rotation axis in the **x**, **y**, and **z** fields.

A X I S O F R O T A T I O N

 From the **Axis type** list, choose **x-axis**, **y-axis**, or **z-axis** (the default) to obtain an axis aligned with the specified coordinate axis. Choose **Cartesian** to enter a direction vector using the **x**, **y**, and **z** fields. Choose **Spherical** to enter the direction using the angles **theta** (polar, zenith) and **phi** (azimuth).

S E L E C T I O N S O F R E S U L T I N G E N T I T I E S

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

S c a l e

To scale geometry objects, right-click a geometry and select **Transforms>Scale** (). Then enter the properties of the scale operation using the following sections:

I N P U T

Select the geometry objects that you want to scale in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Scale** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

S C A L E F A C T O R

By default, you get an isotropic scaling. Specify the scaling factor in the **Factor** field.

To get an anisotropic scaling, change **Scaling** to **Anisotropic**, and specify separate scale factors for the coordinate directions in the **x**, **y**, and (3D only) **z** fields; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes.

C E N T E R O F S C A L I N G

Specify the center point of the scaling by specifying **x**, **y**, and (3D only) **z**; **r** and **z** in 2D axial symmetry; **xw** and **yw** in work planes. This is the point that stays fixed during the

scaling (that is, the point that the scaled geometry objects approach when the scale factor goes to zero).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Split

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).

To split geometry objects into their entities, right-click a geometry and select **Split** (). Then enter the properties of the split operation in the **Input** section.

INPUT

Select the geometry objects to split on in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Split** node, choose **Manual** to select objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric objects consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Sweep

To sweep a face along a curve, right-click a 3D **Geometry** node and select **Sweep** (). Then enter the properties of the sweep operation using the following sections:

CROSS SECTION

Select the face you want to sweep in the **Graphics** window. Only one face can be selected. The face appears in the **Face to sweep** list. You might have to first click the **Activate Selection** button ().

SPINE CURVE

Select the edges you want to sweep along in the **Graphics** window. More than one edge can be selected, but the selected edges must form a nonclosed connected chain. The edges appear in the **Edges to follow** list. You might have to first click the **Activate Selection** button ().

Select the **Reverse direction** check box to sweep in the negative edge direction.

KEEP INPUT

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Select the **Include all inputs in finalize operation** check box to force the objects in the **Face to sweep** and **Edges to follow** lists to be included in the finalize operation. If the **Include all inputs in finalize operation** check box is not selected, these objects are not included in the finalize operation if the **Face to sweep** list or **Edges to follow** list contains all faces or edges in the objects.

MOTION OF CROSS SECTION

This section contains a number of properties that determine how the face is transformed when swept along the spine curve.

A curve parameter name can be defined in the **Parameter name** field. Use this parameter in the expressions defining scale factor and twist angle.

The **Scale factor** field controls the size of the cross section face when swept along the spine curve.

The **Twist angle** field controls the rotation angle of the cross section face about the spine curve.

By default, twist compensation is active and prevents the twisting that would otherwise occur due to nonzero torsion for nonplanar curves. Clear the **Twist compensation** check box to turn off this compensation. When **Twist compensation** is active, it behaves as if a term was added to the **Twist angle** with a magnitude matching the integral of the torsion of the curve. This makes the edges in the sweep direction locally parallel to the spine curve. In the case of a noncircular cross section, twist compensation also affects the shape of the generated object.

From the **Face-spine alignment** list, select an option to align the cross section to the spine curve:

- Select **No adjustment** (the default) to sweep the face starting from its original position. Using this setting, it is possible to create sweeps where the face is not perpendicular to the spine curve, and where the face does not contain that start point of the spine curve.
- Select **Adjust spine** to adjust the spine curve so that it starts on the face to sweep and so that it is parallel to the face normal at the point where it touches the face. The first part of the spine curve is replaced by a cubic Bézier curve, with the length of the replaced part, measured in parameter values, controlled by the value in the **Adjustment parameter length** field.
- Select **Move face** to move the face to the start of the spine curve and orient the face perpendicularly to the spine curve. This setting is only allowed when the face is located in a work plane, and the movement is such that the work plane origin coincides with the spine curve.

ADVANCED SETTINGS

From the **Geometry representation** list, select **Spline** (the default) to represent the swept object using splines, or **Bézier**, to represent the swept object using Bézier curves. The difference is that using Bézier curves, the intersections between the surfaces that form the swept object are visible edges, whereas they are hidden when using splines.

The values in the **Relative tolerance** field is a relative tolerance that controls the accuracy of the geometric representation of the swept object. The geometric representation is an approximation, which is necessary because it is not possible to exactly represent a swept object using NURBS (nonuniform rational basis splines). The default value is 10^{-4} (0.01%).

Internally, the software represents the swept object by B-spline curves and surfaces, which are computed to approximate the mathematical definition of the swept surface. The number of knot points in the splines increases automatically until the

approximation satisfies the tolerance specified in the **Relative tolerance** field or until it reaches the number of knots specified in the **Maximum number of knots** field (default value: 1000).

If more than one edge is selected in the **Edges to follow** list, the **Direction-defining edge** controls which edge is used to define the positive sweep direction. The **Direction-defining edge** is automatically set when the first edge is added to the **Edges to follow** list, so usually it does not have to be changed manually.

If the expressions for scale or twist contain user-defined functions, changes in those functions do not automatically cause the sweep feature to be rebuilt. To rebuild the feature after a change in a user-defined function, click the **Rebuild with Updated Functions** button.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable **Selection** lists but do not appear as separate selection nodes in the Model Tree.

Tangent

To create a tangent (that is, a line segment tangent to a specified edge) right-click a 2D **Geometry** node and select **Tangent** (). Then enter the properties of the tangent.

TANGENT

Select the edge you want to tangent in the **Graphics** window. Only one edge can be selected. The edge appears in the **Edge to tangent** list.

Parameter start guess is a number between 0 and 1 that specifies where on the edge the expected point of tangency is located. The tangent returned is the first tangent found starting the search from the start guess.

Type of tangent specifies how the end point of the tangent is specified. There are three options: **Edge-edge**, **Edge-point**, and **Edge-coordinate**.

When the type is edge-edge, you use the **Graphics** window to select a second edge to tangent. The edge appears in the **Second edge to tangent** box. The line segment created is tangent to both edges. Use **Parameter start guess for second edge** to specify the start guess for the second edge, it is a number between 0 and 1.

When the type is edge–point, you use the **Graphics** window to select a point in the geometry as tangent’s endpoint. The point appears in the **Point** list.

When the type is edge–coordinate, manually specify the coordinates of the tangent’s endpoint.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (boundaries and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Union

To create the union of geometry objects, right-click a geometry and select **Boolean Operations>Union** (). Then enter the properties of the union operation.

INPUT

Select the geometry object that you want to unite in the **Graphics** window. The objects appear in the **Input objects** list. If the geometry sequence includes user-defined selections above the **Union** node, choose **Manual** to select geometry objects, or choose one of the selection nodes from the list next to **Input objects**.

Select the **Keep input objects** check box to use the selected geometry objects for further geometry operations.

Create a geometry object without interior boundaries by clearing the **Keep interior boundaries** check box. This can be useful to simplify a geometry where the interior boundaries do not separate domains with different physics or materials, for example.

Adjust the **Relative repair tolerance** (default: 10^{-6}) if you experience problems with the union 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.

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of domains, boundaries, edges, and points) that the resulting geometric object consists of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

Work Plane

To create a work plane for defining 2D objects in 3D (for example, to extrude a 2D object into a 3D object), right-click a 3D **Geometry** node and select **Work Plane** (). Then enter the properties defining the location of the work plane in the following sections of its settings window:

PLANE DEFINITION

Plane Type

The layout of the **Work Plane** section depends on the selection in the **Plane type** list, where you select how to define the work plane. Choose between the following alternatives:

- **Quick**—In the **Plane** list, select one of the coordinate planes *xy*, *yz*, *, *yx*, *zy*, or *xz*, denoting the first and second axes of the work plane’s local coordinate system. Specify an offset from the coordinate plane in the field **z-coordinate**, **x-coordinate**, or **y-coordinate**.*
- **Face parallel**—Select a planar face in the **Graphics** window that is parallel to the work plane you want to create. The **Planar face** list shows the selected face.

In the **Origin** list, you choose the location of the origin of the work plane’s coordinate system: **Center of face** or **Bounding box corner**.

By entering a value in the **Offset in local z direction** list, you can offset the work plane along the normal of the planar face. The work plane’s normal is the outward normal of the face in the **Planar face** list.

To reverse the direction of the *z*-axis of the work plane’s coordinate system, select the **Reverse direction of local axis** check box. This also swaps the coordinate axes in the work plane to preserve the positive orientation of the local coordinate system.

- **Edge angle**—Activate the **Straight edge** list by first selecting its **Activate Selection** button () and then selecting an edge in the **Graphics** window. Similarly, activate the **Face adjacent to edge** list by first selecting its **Activate Selection** button () and then selecting an adjacent face in the **Graphics** window. Also, specify a value in the **Angle between face and work plane** field. This results in a work plane through the given edge that makes the specified angle with the adjacent face.

By default, the origin of the local coordinate system coincides with the edge’s start vertex, and the direction of the local *x*-axis coincides with the direction of the edge. If you select the **Reverse direction of local axis** check box, the origin instead is at the end vertex, and the direction of the local *x*-axis is reversed.

- **Circular edge**—Activate the **Circular edge** list by selecting its **Activate Selection** button. Then select a circular edge in the **Graphics** window. This results in a work plane perpendicular to the given circular edge. The origin of the local coordinate system is at the circle's center. The local x -axis goes through the edge's start vertex. Thus, if the geometry is rotationally symmetric, the symmetry axis coincides with the local y -axis. The work plane can be used together with a Cross Section feature to get a 2D axisymmetric geometry corresponding to a rotationally symmetric 3D geometry.
- **Vertices**—In each of the lists **First vertex**, **Second vertex**, and **Third vertex**, select a vertex by first selecting the corresponding **Activate Selection** button () and then selecting a vertex in the **Graphics** window. This creates a work plane parallel to a plane through the three vertices.

Specify an offset in the **Offset in local z direction** field.

The origin of the local coordinate system is located above the first vertex, and the vector to the second vertex becomes the local x -axis. Reverse the directions of the local z -axis and y -axis by selecting the **Reverse direction of local axis** check box.

- **General**—This choice creates a work plane through the three points with the given coordinates. The origin of the local coordinate system coincides with **Point 1**. The x -axis of the local coordinate system is in the direction of the vector from **Point 1** to **Point 2**. The positive direction of the y -axis is determined by the condition that the vector from **Point 1** to **Point 3** has a positive y -component.

VISUALIZATION

Select the **Draw on work plane** in 3D check box to change the behavior when in the **Plane Geometry** node under the **Work Plane** node. When this check box is selected, the 3D scene with the work plane displays and the 2D drawing toolbar is activated. Two buttons also become available on the **Graphics** window—the **Align with Work Plane** button () to rotate and move the camera to see the work plane from the top down, and the **Work Plane Clipping** button (). When toggled 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.



Drawing on the 2D Work Plane in 3D

Under **In-plane visualization of 3D geometry**, specify how to visualize 3D objects in the work plane (as blue curves and points) by selecting one or more of the following check boxes:

- **Coincident entities**—Show edges and points (in a pure blue color) that lie in the work plane.
- **Intersection**—Show the intersection of 3D geometry and the work plane (in cyan).
- **Projection**—Show the projection of all edges and points onto the work plane (in light blue).

SELECTIONS OF RESULTING ENTITIES

Select the **Create selections** check box to create predefined selections for all entities (all or some of boundaries, edges, and points) that the 2D geometric objects in the work plane consist of. These selections are available in all applicable selection lists but do not appear as separate selection nodes in the model tree.

DRAWING IN THE WORK PLANE

To show the work plane, click the **Show Work Plane** button () in the settings window's toolbar, or click the **Plane Geometry** node that appears under the **Work Plane** node. To create 2D objects in the work plane, right-click the **Plane Geometry** node and create nodes like in a 2D geometry.

EMBEDDING THE WORK PLANE IN THE 3D GEOMETRY

To embed the 2D work plane geometry in the 3D geometry, build the Work Plane feature by either right-clicking the **Work Plane** node and choosing **Build Selected** or selecting the **Work Plane** node and then clicking **Build Selected** or **Build All**.

Virtual Geometry Operations

For a 2D or 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:

TABLE 6-9: VIRTUAL OPERATIONS (2D AND 3D)

ICON	NAME	DESCRIPTION
	Ignore Vertices	Virtually remove isolated vertices or vertices adjacent to precisely two edges
	Ignore Edges	Virtually remove isolated edges or edges adjacent to precisely two faces or between two domains
	Ignore Faces	Virtually remove isolated faces or faces between two domains (3D only)
	Form Composite Edges	Form virtual composite edges from sets of connected edges by ignoring the vertices between the edges in each set
	Form Composite Faces	Form virtual composite faces from sets of connected faces by ignoring the edges between the faces in each set (3D only)
	Form Composite Domains or 	Form virtual composite domains from sets of connected domains by ignoring the boundaries between the domains in each set
	Collapse Edges	Virtually collapse each edge into a vertex by merging its adjacent vertices
	Collapse Faces	Virtually collapse faces into edges or vertices by merging their adjacent opposite edges or collapsing all adjacent edges
	Merge Edges	Virtually merge opposite edges adjacent to a face
	Merge Vertices	Virtually merge one adjacent vertex of an edge with the other adjacent vertex

See also [Mesh Control Operations](#) for examples of how to use virtual geometry operations to remove a short edge and to ignore and collapse edges to prepare the geometry for efficient meshing.

You can use several of these operations to also control the mesh. Also see [Mesh Control Operations](#) for information about separate mesh control operations that you can use exclusively for controlling the meshing.

IMPORTING A VIRTUAL GEOMETRY

It is possible to import a virtual geometry to a sequence using the Import node. The source can be a COMSOL 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.

Ignore Vertices

To remove isolated vertices or vertices adjacent to two edges only from the geometry, right-click a geometry and select **Virtual Operations>Ignore Vertices** (). Then enter the properties of the operation using the following sections:

INPUT

Select the vertices (points) that you want to ignore in the **Graphics** window. They then appear in the **Vertices to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Vertices** node, choose **Manual** to select vertices, or choose one of the selection nodes from the list next to **Vertices to ignore**.

The operation removes the selected vertices that are isolated or that are adjacent to precisely two edges. However, it does not remove vertices if that would introduce closed or periodic *composite edges*.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected vertices disappear from the geometry but become available when you build the mesh. You can, for example, use a *mesh control vertex* to control the element size inside a domain. See also [Mesh Control Vertices](#).

Ignore Edges

To remove isolated edges or edges adjacent to precisely two domains or two faces from the geometry, right-click a geometry and select **Virtual Operations>Ignore Edges** (). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to ignore in the **Graphics** window. They then appear in the **Edges to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to ignore**.

Use the **Ignore adjacent vertices** check box to specify if the operation also removes the ignorable start and end vertices of the edges.

The operation removes the selected edges that are isolated or that are adjacent to precisely two faces or are between two domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected edges disappear from the geometry but become available when you build the mesh. You can, for example, use *mesh control edges* to control the element size inside a domain or to partition the geometry to use a mapped mesh. See also [Mesh Control Edges](#).

Ignore Faces



This operation is available for 3D models only.

To remove isolated faces or faces between two domains from the geometry, right-click a geometry and select **Virtual Operations>Ignore Faces** (). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to ignore in the **Graphics** window. They then appear in the **Faces to ignore** list. If the geometry sequence includes user-defined selections above the **Ignore Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to ignore**.

Use the **Ignore adjacent vertices and edges** check box to specify if the operation also removes the ignorable vertices and edges adjacent to the faces.

The operation removes the selected faces that are isolated or that are between two domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected faces disappear from the geometry but become available when you build the mesh. You can, for example, use a *mesh control face* to partition the geometry to make it possible to sweep a hexahedral mesh. See also [Mesh Control Faces](#).

Form Composite Edges

To form virtual composite edges, right-click a geometry and select **Virtual Operations>Form Composite Edges** (). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to composite in the **Graphics** window. They then appear in the **Edges to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to composite**.

The operation forms a composite edge for each connected edge component (of manifold type) of the selected edges by ignoring the vertices between the edges. However, it does not ignore vertices when that would introduce closed or periodic composite edges.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected edges are composed in the geometry but are available individually when you build the mesh.

Form Composite Faces

To form virtual composite faces, right-click a geometry and select **Virtual Operations>Form Composite Faces** (). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to composite in the **Graphics** window. They then appear in the **Faces to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to composite**.

Use the **Ignore adjacent vertices** check box to specify if the operation also removes the ignorable vertices on the boundary of each resulting composite face.

The operation forms a composite face for each connected face component (of manifold type) of the selected faces by ignoring the edges between the faces.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected faces are composed in the geometry but are available individually when you build the mesh.

Form Composite Domains

To form virtual composite domains, right-click a geometry and select **Virtual Operations>Form Composite Domains** ( or 

INPUT

Select the domains that you want to composite in the **Graphics** window. They then appear in the **Domains to composite** list. If the geometry sequence includes user-defined selections above the **Form Composite Domains** node, choose **Manual** to select domains, or choose one of the selection nodes from the list next to **Domains to composite**.

Use the **Ignore adjacent vertices** (2D) or **Ignore adjacent edges and vertices** (3D) check box to specify if the operation also removes the ignorable vertices (and edges in 3D) on the boundary of each resulting composite domain.

The operation forms a composite domain for each connected domain component of the selected domains by ignoring the boundaries between the domains.

MESH CONTROL

Use the **Keep input for mesh control** check box to specify that the selected domains are composed in the geometry but are available individually when you build the mesh. This gives you more control of the meshing. A well partitioned geometry is more efficient to mesh and can, for example, make it possible to create a high quality hexahedral mesh through the sweep operations.

Collapse Edges

To virtually collapse edges into vertices, right-click a geometry and select **Virtual Operations>Collapse Edges** (). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to collapse in the Graphics window. They then appear in the **Edges to collapse** list. If the geometry sequence includes user-defined selections above the **Collapse Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to collapse**.

Select the **Ignore merged vertices** check box to ignore the resulting merged vertices (if possible).

The operation collapses an edge by removing it, merging its adjacent vertices to the vertex with lowest index, and reconnecting the adjacent edges to the *merged vertex*.

The operation collapses the edges in the selection where no other edge shares the same adjacent vertices (unless this is in the selection).

Collapse Faces

To virtually collapse faces into edges or vertices, right-click a geometry and select **Virtual Operations>Collapse Faces** (). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to collapse in the **Graphics** window. They then appear in the **Faces to collapse** list. If the geometry sequence includes user-defined selections above the **Collapse Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to collapse**. The operation collapses a face by removing it, merging its adjacent opposite edges or collapsing all adjacent edges, and reconnecting the adjacent faces to the resulting merged edges.

Select the **Ignore merged entities** check box to ignore the resulting merged edges or vertices (if possible).

Merge Edges

To virtually merge opposite edges adjacent to a face, right-click a geometry and select **Virtual Operations>Merge Edges** (). Then enter the properties of the operation using the following sections:

EDGES TO KEEP

Select the edges that you want to keep in the **Graphics** window. They then appear in the **Edges to keep** list.

EDGES TO REMOVE

Select the edges that you want to remove in the **Graphics** window. They then appear in the **Edges to remove** list.

The operation merges the opposite edges by collapsing the face between the edges and reconnecting the faces adjacent to the removed edges to the resulting merged edges.

Merge Vertices

To virtually merge one vertex adjacent to an edge with the other adjacent vertex, right-click a geometry and select **Virtual Operations>Merge Vertices** (). Then enter the properties of the operation using the following sections:

VERTEX TO KEEP

Select the vertex that you want to keep in the **Graphics** window. It then appears in the **Vertex to keep** list.

VERTEX TO REMOVE

Select the vertex that you want to remove in the **Graphics** window. It then appears in the **Vertex to remove** list.

The operation merges the two vertices by collapsing the edge between the vertices and reconnecting the edges adjacent to the removed vertex to the resulting merged vertex.

Mesh Control Operations

The following able mesh control operations are available for including ignored geometric entities for mesh control purposes to, for example, make it possible to use a mapped mesh.

Mesh Control Vertices

To use isolated vertices, or vertices adjacent to precisely two edges, only for mesh control, right-click a geometry and select **Virtual Operations>Mesh Control Vertices** (). Then enter the properties of the operation using the following sections:

INPUT

Select the vertices (points) that you want to use for mesh control in the **Graphics** window. They then appear in the **Vertices to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Vertices** node, choose **Manual** to select vertices, or choose one of the selection nodes from the list next to **Vertices to include**.

The operation removes the selected vertices that are isolated or that are adjacent to precisely two edges. However, it does not remove vertices if that would introduce closed or periodic *composite edges*. The selected vertices are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control vertex* to control the element size inside a domain.



[Ignore Vertices](#)

Mesh Control Edges

To use isolated edges, or edges adjacent to precisely two domains (in 2D) or two faces (in 3D), only for mesh control, right-click a geometry and select **Virtual Operations>Mesh Control Edges** (). Then enter the properties of the operation using the following sections:

INPUT

Select the edges that you want to use for mesh control in the **Graphics** window. They then appear in the **Edges to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Edges** node, choose **Manual** to select edges, or choose one of the selection nodes from the list next to **Edges to include**.

Use the **Include adjacent vertices** check box to specify if the operation also include the ignorable start and end vertices of the edge.

The operation removes the selected edges that are isolated or that are adjacent to precisely two faces (in 3D) or two domains (in 2D). The edges are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control edge* to control the element size inside a domain or to partition the geometry to make use of a mapped mesh.



Ignore Edges

Mesh Control Faces

To use isolated faces or faces between two 3D domains only for mesh control, right-click a geometry and select **Virtual Operations>Mesh Control Faces** (). Then enter the properties of the operation using the following sections:

INPUT

Select the faces that you want to use for mesh control in the **Graphics** window. They then appear in the **Face to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Faces** node, choose **Manual** to select faces, or choose one of the selection nodes from the list next to **Faces to include**.

Use the **Include adjacent vertices and edges** check box to specify if the operation also includes the ignorable vertices and edges adjacent to of the faces.

The operation removes the selected faces that are isolated or that are between two domains. The faces are removed from the geometry but become available when you

build the mesh. You can, for example, use a *mesh control face* to partition the geometry to make it possible to sweep a hexahedral mesh.



Ignore Faces

Mesh Control Domains

To use a domain only for mesh control, right-click a geometry and select **Virtual Operations>Mesh Control Domains** (). Then enter the properties of the operation using the following sections:

INPUT

Select the domains that you want to use for mesh control in the Graphics window. They then appear in the **Domains to include** list. If the geometry sequence includes user-defined selections above the **Mesh Control Domains** node, choose **Manual** to select domains, or choose one of the selection nodes from the list next to **Domains to include**.

The operation removes the selected domains by composing them with adjacent domains. The faces (3D only) and edges in between are removed from the geometry but become available when you build the mesh. You can, for example, use a *mesh control domain* to partition the geometry to make it possible to sweep a hexahedral mesh.

7

Meshing

This chapter describes meshing capabilities, meshing techniques, and meshing operations available for meshing the geometry.

Creating Meshes

The **Mesh** (grid icon) nodes enable the discretization of the geometry model into small units of simple shapes, referred to as *mesh elements*.

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 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.

Adding Meshing Sequences

When you add a new model to the Model Tree, a meshing sequence is added by default in the node **Mesh 1**. You can add more meshing sequences to the model by right-clicking the model node and selecting **Mesh**. When a model has more than one meshing sequence, they are collected under a **Meshes** node. To add new meshing sequences to such a model, you can alternatively right-click **Meshes**.

You create a mesh by building a meshing sequence, which contains a number of meshing operations as nodes in the sequence. For the default physics-controlled meshes, the software sets up the meshing sequences automatically.

Adding and Building Meshing Operations

A mesh is the result of building a meshing sequence. A meshing sequence corresponding to a geometry consists of [Meshing Operations](#) and [Mesh Attributes](#). The attribute nodes store properties that are used by the operation nodes when creating the mesh.

Building an operation node creates or modifies the mesh on the part of the geometry defined by the selection of the operation node. Some of the operation node use properties defined by attribute nodes; for example, the Free Tetrahedral node reads properties from the Distribution and Size attribute nodes. For some operation nodes it is possible to add local attribute nodes as subnodes. You can add a local attribute node from the context menu of an operation node. Properties defined in local attribute nodes of an operation node override the corresponding properties defined in global attribute nodes (on the same selection).

An attribute node contains properties defined on a selection. You can add an attribute as a node in the meshing sequence (this is referred to as a *global attribute node*) or

add it as a node under an operation node (a *local attribute node*). Global attribute nodes are used by subsequent operation nodes when building the meshing sequence. Local attribute nodes are only used by the owning operation node.

If you choose to import a mesh you have access to a different set of operations (see [Operations on Imported Meshes](#)).

You can add and build meshing nodes in two ways:

- By right-clicking a mesh node in the Model Tree and then selecting one of the available options from the context menu. Enter the node's properties in the settings window that appears. In numerical fields you can enter expressions that contain global parameters. Click the **Build Selected** button () in the settings window to see the mesh that results. All meshing nodes can be added and built in this way.
- By using the buttons in the Meshing toolbar. Most nodes can be added and built in this way.



For a description of how to use the buttons in the Meshing toolbar see [The Meshing Toolbar](#).

Editing Meshing Nodes

To edit a meshing node, select it in the tree, 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. Following nodes are marked with a yellow triangle at the lower-right corner of the node's icon to indicate that they need to be rebuilt. To see the result of your edits in the graphics, you need to build the node. You can do this in two ways:

- Click the **Build Selected** button () in the settings window, or right-click the node in the tree and select **Build Selected** (or press F7). This builds all nodes (if needed) from the first up to the selected node.
- Click the **Build All** button () in the settings window, or right-click the main **Mesh** node () in the tree and select **Build All** (or press F8). This builds all nodes in the meshing sequence (if needed).

The Graphics window shows the resulting mesh from the nodes that have been built. The result of subsequent nodes is not visible. The last built node becomes the *current node* and appears with a quadratic frame around the node's icon. The frame is green if

the node and all preceding nodes are built; that is, the mesh in the Graphics window is up to date. The frame is yellow if the node or some preceding node has been edited since the node was built and needs to be rebuilt.

Errors and Warnings in Meshing Sequences

If a problem occurs when you build a node, the build continues if it is possible to avoid the problem in the corresponding meshing operation, otherwise the build stops.

CONTINUING THE BUILD

When you build a **Free Triangular**, **Free Quad**, or **Free Tetrahedral** node where problems are encountered, you can avoid the problems related to meshing of faces and domains by leaving the corresponding faces and domains unmeshed. The operation corresponding to the node continues meshing the remaining entities and stores information on the encountered problems in subnodes of the node. A node that encountered this type of problems during the build gets a Warning status. The node's icon is decorated with a yellow triangle in the lower-right corner. If you build several nodes in a sequence the build does not stop by a node that gets a Warning status.

STOPPING THE BUILD

When you build other nodes than **Free Triangular**, **Free Quad**, or **Free Tetrahedral** the build stops if a problem is encountered. This means that no changes are made to the mesh. The node gets an Error status, which the program indicates by adding a red cross in the lower-right corner of the node's icon. You find information about the error in a subnode of the node where the error occurred. If the node is part of a sequence build, the build stops and the preceding node becomes the current node.



[Dynamic Nodes in the Model Builder](#)

Deleting, Clearing, Disabling, and Enabling Meshes

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



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.

DELETING, DISABLING, AND ENABLING NODES

- To delete a node, right-click it and select **Delete** ().
- To disable a node, right-click it and select **Disable** (). The disabled node does not affect the mesh. This is indicated with a gray icon around the node name in the model tree.
- To enable the node, right-click it and select **Enable** ().

Using Several Meshing Sequences of Imported Mesh Type

You can define several meshing sequences for the same geometry (see [Adding Meshing Sequences](#)). If the geometry sequence is empty (a necessary condition for the **Imported mesh** sequence type), the first **Mesh** node under the **Meshes** node is referred to as the *master sequence*. All the other **Mesh** nodes should define a geometry similar to the one defined by the master sequence. Two geometries are considered to be similar if they have the same number of geometric entities and their points have the same coordinates.

When you build a non-master sequence, COMSOL Multiphysics first builds the master sequence. If the build of the master sequence fails or if the geometries defined by these two sequences are not similar, an error occurs.

The Meshing Toolbar

This section describes the meshing tools accessible from the Meshing toolbar, which is a part of the main toolbar.

Instead of creating and modifying a mesh by adding and building nodes from the right-click menu of a mesh node you can use the tools available in the Meshing toolbar. Using the buttons in the toolbar it is possible to add and build meshing nodes in a one-click fashion.

The Meshing toolbar is available when you have selected a meshing sequence node in the **Model Builder**. This means that the Meshing toolbar disappears if you, for example, select a meshing node in the **Model Builder**. You can use the **Toolbar** button () in the main toolbar to make the Meshing toolbar for a specific mesh node available; click the associated menu arrow and choose the menu item corresponding to the mesh node. If you click the button associated with this menu button COMSOL Multiphysics selects the node in the **Model Builder** corresponding to the last selected item in the menu associated with the menu button.

The Meshing toolbar for a 3D meshing sequence contains the following buttons:

GEOMETRIC SCOPE

Choose the menu items—**Select Domains**, **Select Boundaries**, **Select Edges**, and **Select Points**—to switch between domain, boundary, edge, and point selection mode. The selection mode determines which tools in the Meshing toolbar that are enabled.

DOMAIN MESHING TOOLS

- Use the **Free Tetrahedral** button () in domain selection mode to create an unstructured tetrahedral mesh. To create an unstructured tetrahedral mesh for a domain selection, select the domains in the **Graphics** window, click the menu arrow associated with the menu button, and select the menu item corresponding to the desired predefined element size, for example, **Normal**. The software creates the resulting tetrahedral mesh by adding and building a Free Tetrahedral node, using the selected domains, with a Size node, using the selected predefined element size, added as a subnode. Alternatively, you can click the button associated with the menu button. Then COMSOL uses the last selected menu item (or **Free Tetrahedral**

(**Normal**), as indicated by the tooltip. If you use this menu button with an empty selection the software meshes the remaining, unmeshed geometry.

- Use the **Swept** button () to create a swept mesh. In domain selection mode this button works in the same way as the **Free Tetrahedral** button. In boundary selection mode the software creates a swept mesh on the remaining domains using the selected boundaries as source faces.

BOUNDARY MESHING TOOLS

- Use the **Free Triangular** button () in boundary selection mode to create an unstructured triangular mesh. This menu button works in the same way as the **Free Tetrahedral** button.
- Use the **Free Quad** button () in boundary selection mode to create an unstructured quadrilateral mesh. This menu button works in the same way as the **Free Tetrahedral** button.
- Use the **Mapped** button () in boundary selection mode to create a structured quadrilateral mesh. This menu button works in the same way as the Free Tetrahedral menu button.
- Use the **Copy Face** button () in boundary selection mode to copy a mesh between boundaries. If you select both the boundaries to copy the mesh from and the boundaries to copy the mesh to in the **Graphics** window and click this button the software copies the mesh by adding and building a Copy Face node with the source boundaries set to the selected boundaries with a mesh and the destination boundaries set to the selected boundaries without a mesh.

EDGE MESHING TOOLS

Use the **Edge** button () in edge selection mode to create an edge mesh. This menu button works in the same way as the **Free Tetrahedral** button.

OTHER TOOLS

- Use the **Boundary Layers** button () in boundary selection mode to create a boundary layer mesh. If you select the boundaries where you want to insert boundary layer elements in the **Graphics** window and click this button the software inserts boundary layer elements for the selected boundaries by adding and building a Boundary Layers node, with the meshed domains adjacent to the selected

boundaries as domain selection, with an added Boundary Layer Properties node, using the selected boundaries, as a subnode.

- Use the **Convert** button (▣▣) in domain or boundary selection mode to convert a mesh. If you select the domains or boundaries in the **Graphics** window for which you want to convert the elements and click this button the software converts the mesh on the selected entities by adding and building a Convert node using the selection.

PREDEFINED MESH ELEMENT SIZES

The following table shows the icons on the drop-down menus for selecting predefined mesh element sizes:

TABLE 7-1: PREDEFINED ELEMENT SIZE ICONS

ICON	NAME	ICON	NAME
	Extremely Fine		Normal
	Extra Fine		Coarse
	Finer		Coarser
	Fine		Extra Coarse
			Extremely Coarse

GEOMETRIC MEASUREMENTS

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 a geometry object.

The Meshing toolbar in 1D and 2D contains a subset of the tools in the 3D toolbar.

You do not need to confirm the entity selections in the **Graphics** window when working with the Meshing toolbar; it is sufficient to highlight them, a state that is indicated by the color red.



Key to Nodes and Toolbar Buttons

Meshing Techniques

Physics-Controlled Meshing

If you select **Physics-controlled mesh** in the **Sequence type** list in the settings window of a mesh node and build the meshing sequence, COMSOL Multiphysics creates a mesh that is adapted to the current physics settings in the model. The default is to use physics-controlled mesh. 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 from the **Element size** list 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 creates a new mesh adapted to the new physics settings.



A physics-induced mesh is not adapted by numerical error estimates—that type of adaptive meshing is provided by mesh adaptation in the solver sequence.

To edit a physics-induced meshing sequence, or to see the errors and warnings of a failing mesh build, select **User-controlled mesh** in the **Sequence type** list or 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.



By doing this the sequence is no longer updated according to changes that applied to the physics settings in the model.

If you right-click the Mesh node and select **Reset to the Physics-Induced Sequence** () , the sequence is reset to the physics-induced sequence. However, the type of the sequence is still **User-controlled mesh**. To switch back to physics-controlled meshing, select **Physics-controlled mesh** in the **Sequence type** list in the settings window of the mesh node. If you add a node to the sequence the type of the sequence automatically switches to **User-controlled mesh**.

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.



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.

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.



- [Free Meshing](#)
- [Generating a 2D Free Mesh](#)

3D Meshing Options



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.



- [Free Meshing](#)
 - [Generating a 3D Free Tetrahedral Mesh](#)
 - [Generating a 3D Swept Mesh](#)
-

Free Meshing

About Free Meshing

The free mesher is available in all space 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 nodes 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.

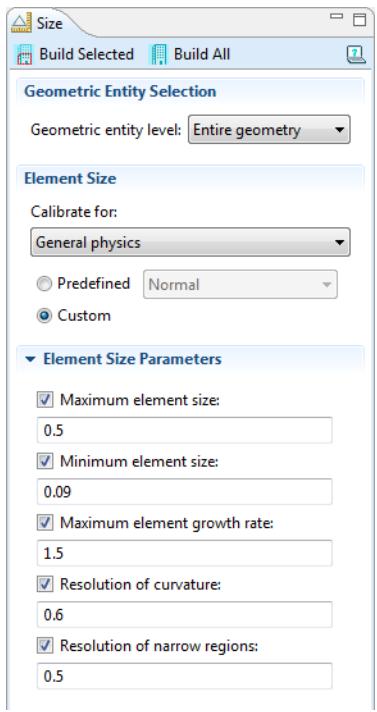


Figure 7-1: An example of custom element mesh sizes. You can also select Predefined element sizes.

-
- 
 - 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.

-
- 
 - Free Triangular and Free Quad
 - About Free Meshing
 - Generating a 3D Free Tetrahedral Mesh
-



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**.

Generating a 3D Free Tetrahedral Mesh

Use the free mesher in 3D to create an unstructured mesh with tetrahedral elements.



Free Tetrahedral



For a 3D free tetrahedral mesh example see the Busbar model: Model Library path **COMSOL_Multiphysics/Multiphysics/busbar**.

SPECIFYING THE MESH SIZE



To adapt the mesh for the selected domains, add a **Size** node 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 7-2](#) 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 meshing 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 Meshing toolbar.

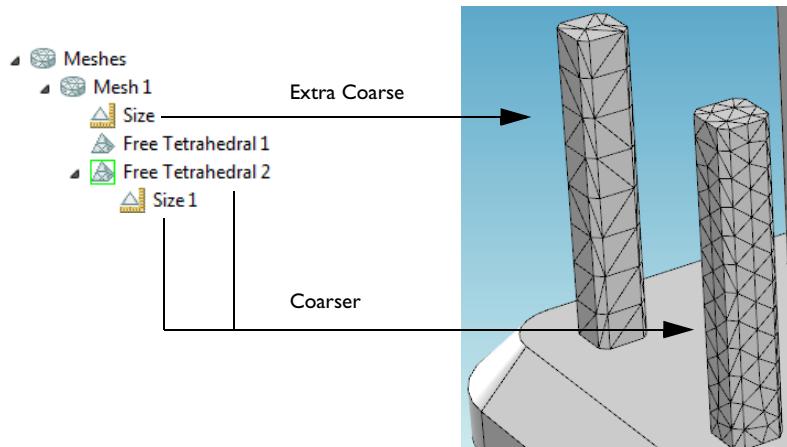
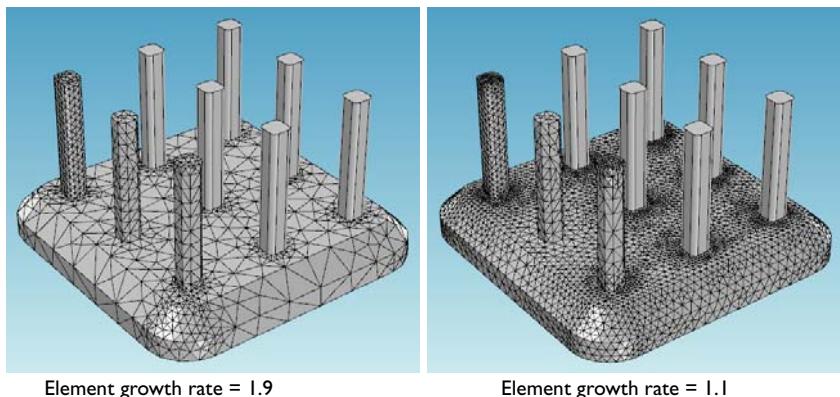


Figure 7-2: An example of two free tetrahedral meshing sizes.

The difference between the two meshes is based on the following. The following figure, [Figure 7-3](#), shows how you can control the mesh distribution using the element growth rate: A higher growth rate means that elements quickly become larger outside of small regions.



Element growth rate = 1.9

Element growth rate = 1.1

Figure 7-3: 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 node to a Free Tetrahedral node. In the settings for this Distribution, constrain the number of elements for the selected edges. [Figure 7-4](#) shows three heat sink pillars with different mesh sizes and distribution.



The settings in the **Distribution** attribute overrule the settings in the **Size** node. See [Distribution](#).

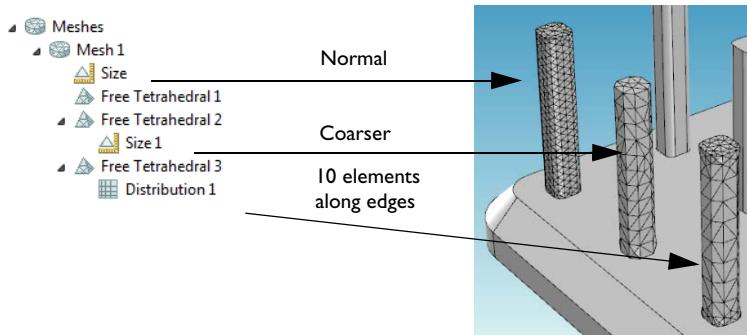


Figure 7-4: Specifying element distribution in heat sink pillars.

Troubleshooting Free Tetrahedral Mesh Generation

This section gives you some suggestions about how to solve problems that 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 get warnings when building the node. 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.



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 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 [Virtual Geometry Operations](#)). If you have a license for the CAD Import Module, you can also use CAD defeaturering 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 your 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 partition 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](#)).



To split a solid geometry object into parts using a **Work Plane**, place the work plane where you want to cut the domain. Then add a **Partition** node from the **Boolean Operations** submenu, and select **Work plane** from the **Partition with** list in the **Partition** node's settings window.

Structured Meshes

About 2D Mapped Meshing

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, for example, by splitting it into simpler domains.



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 sensitive to the topology of the model geometry. If you get an error when trying to build a Boundary Layers node, 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 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.



To turn off boundary layer splits, see [Boundary Layers](#).



- 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**](#).
 - If you have the CFD Module, see [Turbulent Flow Over a Backward Facing Step \(2D\)](#): Model Library path [**CFD_Module/Single-Phase_Benchmarks/turbulent_backstep**](#).
 - If you have the Heat Transfer Module, see [Turbulent Flow Over a Backward Facing Step](#): Model Library path [**Heat_Transfer_Module/Verification_Models/turbulent_backstep**](#).
-

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 corresponds 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 7-5](#)). 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.

	Swept
<hr/>	
	<ul style="list-style-type: none">• Thin-Layer Diffusion: Model Library path COMSOL_Multiphysics/Diffusion/thin_layer_diffusion• See Deformation of a Feeder Clamp: Model Library path COMSOL_Multiphysics/Structural_Mechanics/feeder_clamp• Joule Heating of a Microactuator: Model Library path COMSOL_Multiphysics/Multiphysics/thermal_actuator_jh

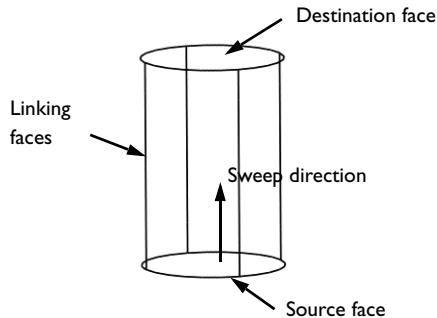


Figure 7-5: Classification of the boundaries about a domain used for swept meshing.

Generating a 3D Swept Mesh

[Figure 7-6](#) 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.

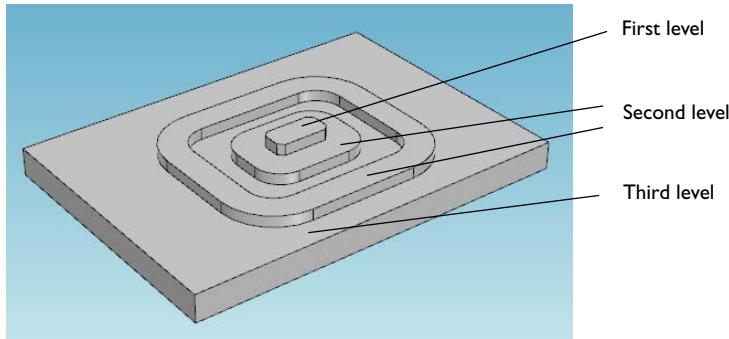
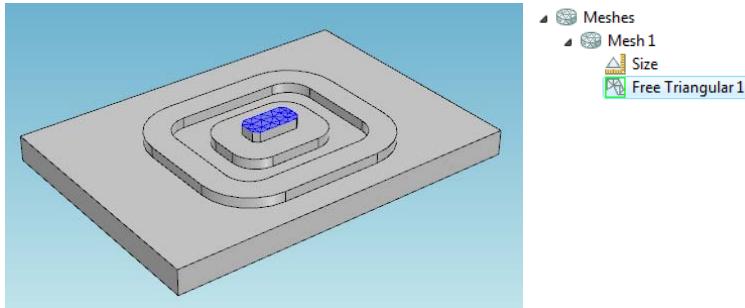


Figure 7-6: An example of the layered geometry used for creating a swept mesh.

- 1 Right-click **Mesh** () and select **More Operations>Free Triangular** ().
- 2 Add the first level boundary to the selection list (see [Figure 7-6](#) for an example of a suitable geometry).

3 Click **Build Selected** (). The mesh below displays.



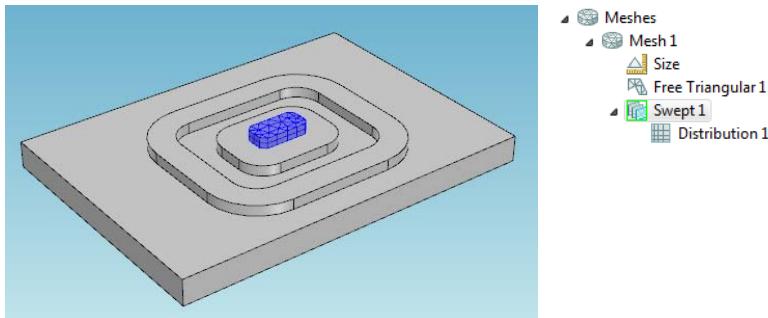
4 Right-click **Mesh** () and select **Swept** ().

5 Select the domain in the first level.

6 Right-click **Swept 1** and select **Distribution** ().

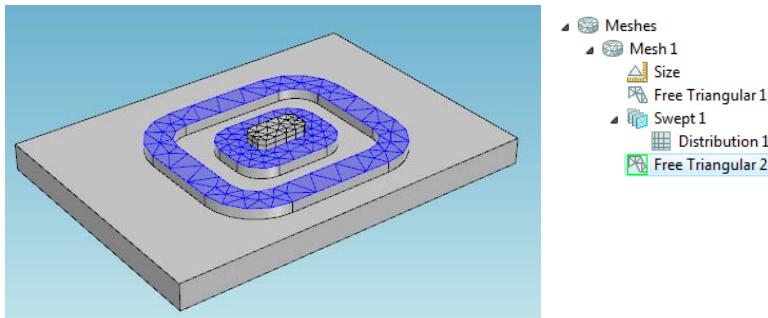
7 Enter the **Number of elements** in the field (for example, 2).

8 Click **Build Selected** ().

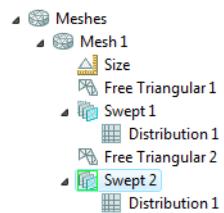
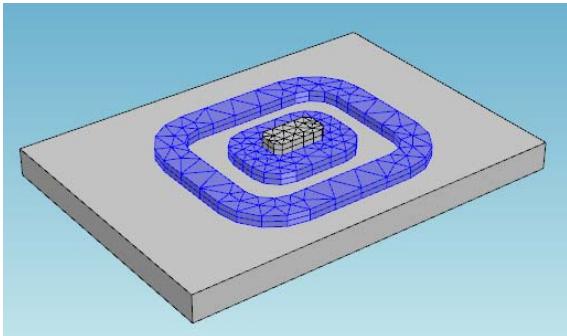


9 Right-click **Mesh** () and select **More Operations>Free Triangular** ().

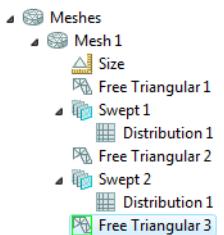
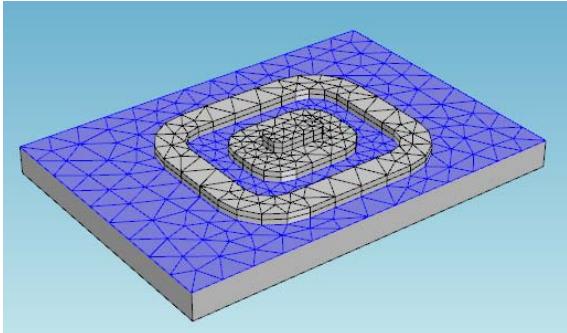
10 Select the boundaries at the second level and click the **Build Selected** button().



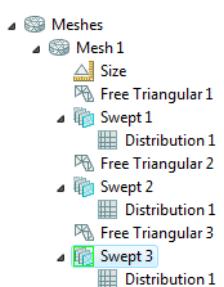
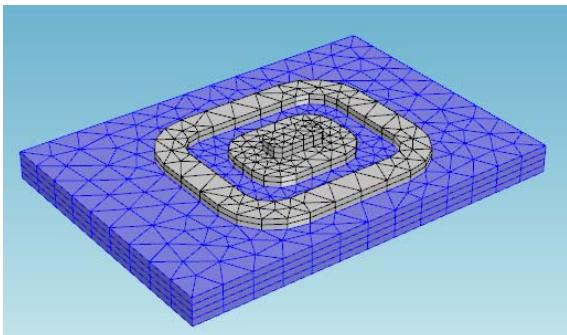
- II** Repeat the same swept operations for the first level domains but now for the second level.



- I2** Mesh the third level boundaries using the **Free Triangular** mesh operation.



- I3** Mesh the third level domain. Use the **Swept** mesh operation and enter 4 for the **Number of elements** in the corresponding **Distribution** attribute.



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.

If you have the:

- Acoustics Module, see [Vibrations of a Disk Backed by an Air-Filled Cylinder](#): Model Library path **Acoustics_Module/Verification_Models/coupled_vibrations_acsh**.
 - Batteries & Fuel Cells Module, see [Ohmic Losses and Temperature Distribution in a Passive PEM Fuel Cell](#): Model Library path **Batteries_and_Fuel_Cells_Module/PEMFC/passive_pem**.
 - Electrodeposition Module, see [Electrodeposition of an Inductor Coil](#): Model Library path **Electrodeposition_Module/Tutorial_Models/inductor_coil**.
 - Chemical Reaction Engineering Module, see [Steam Reformer](#): Model Library path **Chemical_Reaction_Engineering_Module/Heterogeneous_Catalysis/steam_reformer**.
-



Mesh Control Entities

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 and materials. An advantage is that the final mesh need not respect this curve exactly; it is used only to control element size.

Another situation where mesh control entities are useful is when you need precise control of mesh in certain regions of the geometry. In these regions you typically use a structured mesh with distribution nodes to control the mesh. In other regions of the geometry you may have to use free (unstructured) 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 7-7](#) 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** (A) and select **Virtual Operations>Mesh Control Edges** (X). Select the edges of the Bézier Polygon in the **Edges to include** selection.

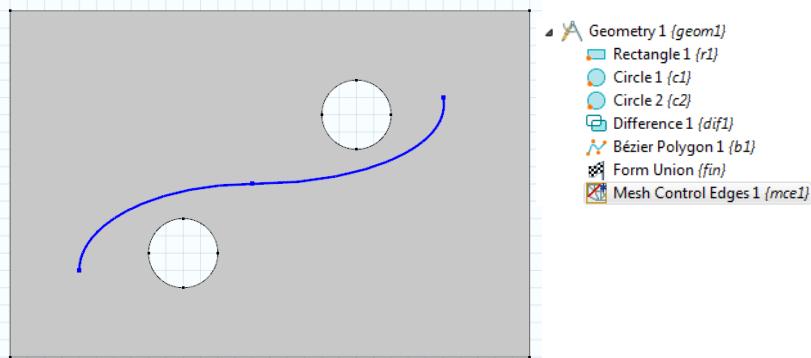


Figure 7-7: A geometry with a Bézier Polygon used to define mesh size inside the domain.

- 2 Click **Build Selected** (B). Note that the selected edges are removed.
- 3 Right-click **Mesh** (Globe) and select **Free Triangular** (X). Note that the edges removed in the previous step are now visible again.
- 4 Right-click **Free Triangular 1** and select **Size** (triangle).
- 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 7-8) and that the only trace of them is the fine mesh size inside the domain.

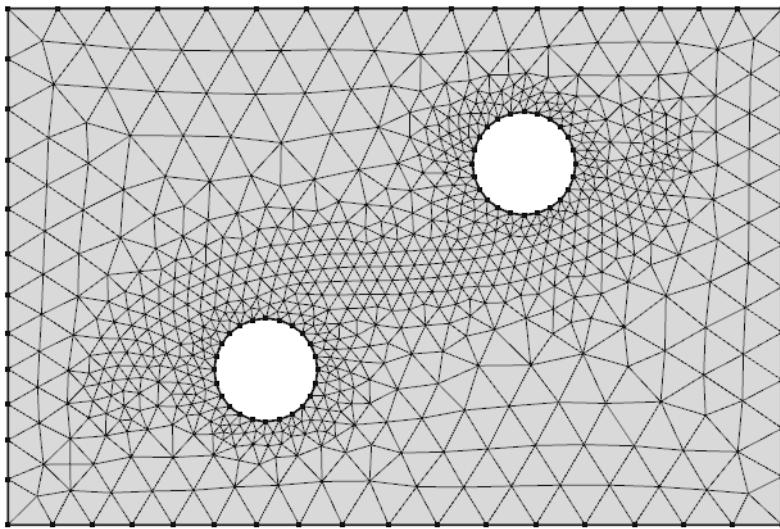
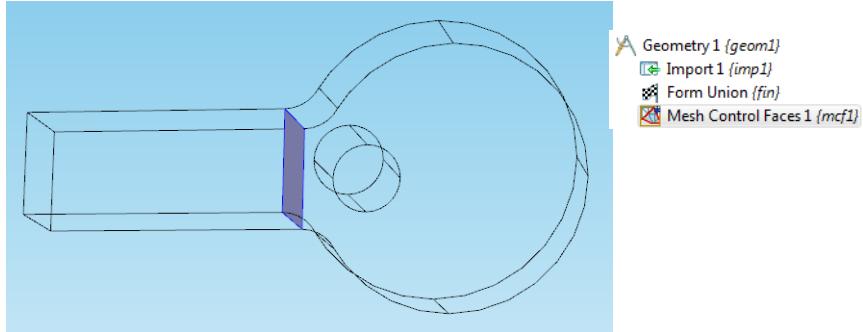


Figure 7-8: Fine mesh inside the domain.

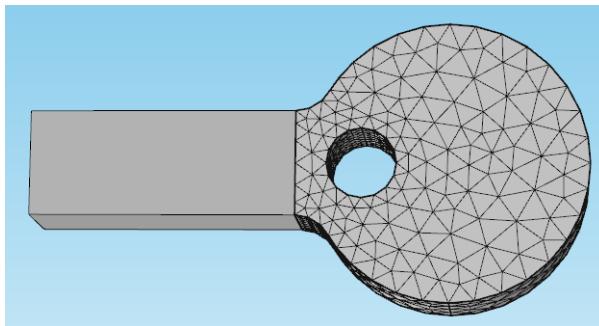
Using Structured and Unstructured Mesh with Boundary Layers

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** (A) and select **Virtual Operations>Mesh Control Faces** (M). Select the face separating the domains in the **Faces to include** selection.

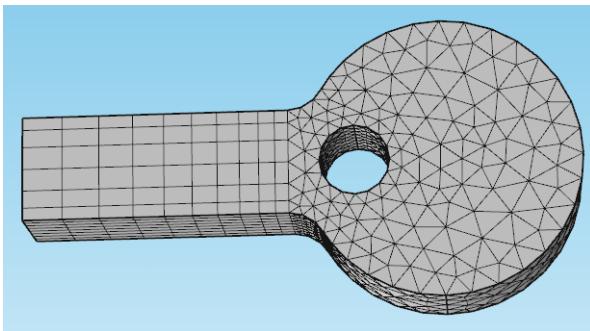


- 2 Click **Build Selected** (B). Note that the face is removed. There is now only one domain.
3 Right-click **Mesh** (M) and select **Free Tetrahedral** (T). Note that the face has reappeared, and that there are two domains.
4 Add the cylinder shaped domain with a hole to the selection list.
5 Click on the **Size** (S) node, and select the **Finer as Predefined** element size. Build the mesh (B).



- 6 Right-click **Mesh** (M) and select **Swept** (S).
7 Right-click **Swept 1** and select **Distribution** (D).

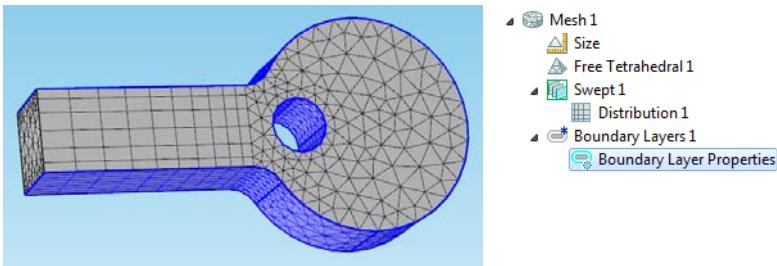
- 8** Select **Predefined distribution type** and enter 10 in the **Number of elements** field and 3 in the **Element ratio** field. Build the mesh ().



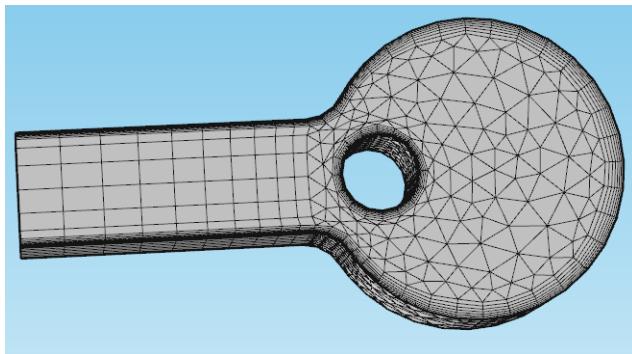
- 9** Right-click **Mesh** () and select **Boundary Layers** ().

- 10** Click the **Boundary Layer Properties** () node under **Boundary Layers 1**.

- II** Add (for example) the sides of the geometry to the selection list.



I2 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.



Importing and Exporting Meshes

About Mesh Export, Import, and Operations on Imported Meshes

It can be useful to import meshes already created by external software or, alternatively, to export a mesh generated by COMSOL Multiphysics into other software. Importing an externally generated mesh can be helpful when a mesh is already saved in a file and recreating the geometry would be difficult and time consuming.

The partitioning of the mesh into domains, boundaries, edges, and points is essential to set up the physics of the model. The available operations for imported meshes deliver some basic, but flexible, functionality. It is also possible to export a mesh for use in another software or for external manipulation of the mesh data.

Exporting a Mesh

You can export a mesh to a COMSOL Multiphysics binary file (`.mphbin`) or text file (`.mphtext`).



A 3D mesh can also be exported to an STL file.

To open the **Export Mesh** page, right-click the **Mesh** node and select **Export to File** () from the menu.

Select a file type among the available formats in the **File type** list and enter a filename including the path in the **Filename** field (or click **Browse** to specify the filename).

Click the **Export** button to export a mesh to the specified file. A confirmation message appears in the **Messages** window.



If you export to an STL file, COMSOL exports boundary elements only.

Importing Meshes

You can import a mesh from a COMSOL Multiphysics native file or from another meshing sequence. In 3D you can also import meshes from NASTRAN, STL, or VRML files. In 2D you can also import 2D meshes from NASTRAN (the third coordinate must then be the same for all mesh points).



Importing a mesh clears the geometry defined in the corresponding geometry sequence. It is not possible to use an imported mesh together with a user-defined geometry.

When a mesh is imported into COMSOL, the Import node automatically determines a partitioning of the mesh into domains, boundaries, edges, and points. If the automatically performed partitioning does not match the requirements, you can modify the face partitioning by manually adjusting the corresponding parameters.

To import additional meshes, add another Import node. Then COMSOL adds the elements and points of the newly imported mesh to the existing mesh.



Meshes from different Import nodes form an assembly.

USING SEVERAL MESHES

It is possible to use several meshing sequences of the imported type. Then the mesh of the first sequence (*master sequence*) defines a topology that the other sequences should match.

If you want to use the geometric multigrid solver, several meshing sequences must be added first.



[Import](#) and [Multigrid](#)

Operations on Imported Meshes

The following mesh import operation nodes make it possible to define the partitioning of an imported mesh into domains, boundaries, edges, and points, with respect to the physics settings of the model.

TABLE 7-2: OPERATIONS ON AN IMPORTED MESHES

NAME AND LINK	ICON	USE AND DESCRIPTION
Ball		To split geometric entities in an imported mesh by an element set defined by a ball.
Box		To split geometric entities in an imported mesh by an element set defined by a box.
Create Vertex		To create an additional vertex in an imported mesh.
Cylinder		To split geometric entities in an imported mesh by an element set defined by a cylinder.
Delete Entities		To delete geometric entities from an imported mesh.
Join Entities		To join adjacent geometric entities in an imported mesh.
Logical Expression		To split geometric entities in an imported mesh by specifying a logical expression.

Using Operations on an Imported Mesh

The following example shows how you can use the mesh import operations to control the partitioning of an imported mesh.

IMPORTED MESH

The following overview is based on using an imported mesh from the **feeder_clamp** model, found in the COMSOL Multiphysics Model Library and shown in [Figure 7-9](#).



Deformation of a Feeder Clamp: Model Library path **COMSOL Multiphysics/Structural_Mechanics/feeder_clamp**

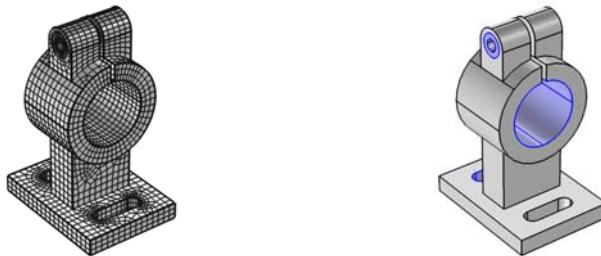


Figure 7-9: The Deformation of a Feeder clamp model showing an imported mesh, which is divided into 5 domains, 69 faces, 174 edges, and 114 points.

JOIN ENTITIES

To form a single domain, use a **Join Entities** () node, which operates on the domain level (that is, add **All domains** to the selection). As a result, you can obtain a mesh for the model with a single domain.

DELETE ENTITIES

To remove all edges, use a **Delete Entities** () node, which operates on the edge level (that is, add **All edges** to the selection). As a result, you can obtain a mesh for the model with no edges or points.

CYLINDER

To define a boundary that defines the contact between the feeder and the clamp, use a **Cylinder** () node which operates on the boundary level (that is, add **All boundaries**

to the selection, and use 10.001 as a cylinder radius, 0 and -20 for top and bottom, (15, 0, 35) as position, and **y-axis** as the axis type).



Figure 7-10: Using a Cylinder node to define the contact between the feeder and the clamp.

LOGICAL EXPRESSIONS

To define two boundaries that define screw channels, use a **Logical Expression** () node, which operates on the boundary level (that is, add **All boundaries** to the selection and use $(y+10)^2 + (z-55)^2 \leq 4$ as the expression).

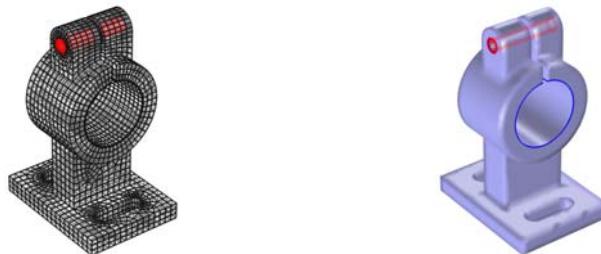


Figure 7-11: Using a Logical Expression to create boundaries on the two screw channels of the feeder clamp.

BALL

To create a boundary defining one of the washers used for the boundary loads of the model, use a **Ball** () node, which operates on the boundary level (use (5, -10, 55) as a ball center and 3.5 as a ball radius). The input boundary selection must be limited,

otherwise, the ball operation also splits one of the cylinder boundaries, which was created by the **Logical Expression** node.



Figure 7-12: Using a Ball node to define one of the washers of the feeder clamp.

By creating a duplicate of the **Ball** node and modifying the ball center (set x to 5) you can create a boundary for the second washer.

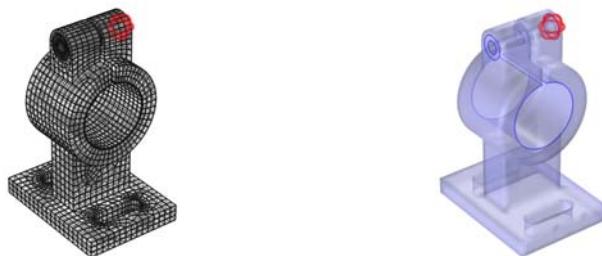


Figure 7-13: Using a Ball node to define a second washer.

BOX

To create the boundaries for the mounting holes, use a **Box** () node, which operates on the boundary level (use (0 - 30, -30 - 10, 0.1 - 4.9) as box limits and use the **Some vertex** condition).



Figure 7-14: Using a Box node to define the mounting holes on the feeder clamp.

CREATE VERTEX

Using the **Create Vertex** () node it is possible to add an additional vertex in a specified location (use (30, -30, 0) as vertex coordinates as in [Figure 7-15](#)).

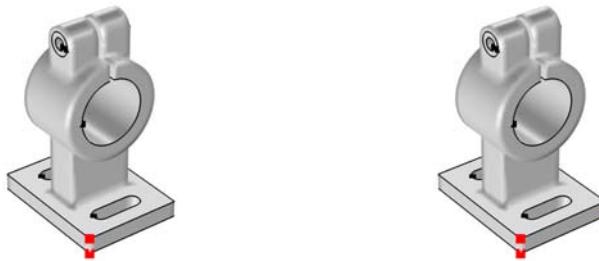


Figure 7-15: Using a Create Vertex node to add vertices at specific locations on the feeder clamp.

Ball

Use a **Ball** node () to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified ball.

To add a **Ball** node, right-click a 2D or 3D mesh node and select **Ball** (). Then use the following sections to specify the geometric entities to split, the properties of the ball, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified ball.
- Choose **Domain, Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

BALL CENTER

Specify the center of the ball in the **x**, **y**, and **z** (only in 3D) fields (SI unit: m).

BALL RADIUS

Specify the radius of the ball in the **Radius** field (SI unit: m). The default radius is 1.

CONDITION

Use the **Include element if ball contains** list to select the condition for which the element is enclosed in the specified ball. Choose **All vertices** to consider an element to be enclosed in the specified ball if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Box

Use a **Box** node () to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified box.

To add a **Box** node, right-click a 2D or 3D mesh node and select **Box** (). Then use the following sections to specify the geometric entities to split, the properties of the box, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified box.
- Choose **Domain, Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

BOX LIMITS

Specify the limits of the box in the **x minimum**, **x maximum**, **y minimum**, **y maximum**, **z minimum** (3D only), and **z maximum** (3D only) fields.

CONDITION

Use the **Include element if box contains** list to select the condition for which an element is enclosed in the specified box. Choose **All vertices** to consider an element to be enclosed in the specified box if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Create Vertex

Use a **Create Vertex** node () to create an additional vertex in the closest mesh point to a specified position of an imported mesh.

To add a **Create Vertex** node, right-click a 2D or 3D mesh node and select **Create Vertex** (). Then use the following section to specify the position of the new vertex:

CREATE VERTEX CLOSEST TO POINT

Use the **x**, **y**, and **z** (3D only) fields to specify the position of the vertex. The vertex appears in the mesh point closest to the specified position.

Cylinder

Use a **Cylinder** node () to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements enclosed in the specified cylinder.

To add a **Cylinder** node, right-click a 3D mesh node and select **Cylinder** (). Then use the following sections to specify the geometric entities to split, the properties of the cylinder, and the split condition:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified cylinder.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

SIZE AND SHAPE

Specify the radius of the cylinder in the field **r** and the positions of the upper and lower boundary circles in the **Top distance** and **Bottom distance** fields, respectively.

POSITION

Specify the position of cylinder in the **x**, **y**, and **z** fields.

AXIS

Use **Axis type** to set the direction of the cylinder axis. Choose **x-axis**, **y-axis**, or **z-axis** to let the cylinder axis coincide with one of the coordinate axes. It is also possible to customize the cylinder axis by choosing **Cartesian** or **Spherical** and using **x**, **y**, and **z**, or **theta** and **phi**, respectively.

CONDITION

Use the **Include element if cylinder contains** list to select the condition for an element to be enclosed in the specified cylinder. Choose **All vertices** to consider an element to be enclosed in the specified cylinder if all element vertices are enclosed, or choose **Some vertex** to consider it enclosed if at least one element vertex is enclosed.

Delete Entities

Use a **Delete Entities** node () to delete geometric entities from an imported mesh.

To add a **Delete Entities** node, right-click a 2D or 3D mesh node and select **Delete Entities** (). Then use the following sections to specify the geometric entities to delete:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to delete. You choose the geometric entity level from the **Geometric entity level** list: Choose **Domain**, **Boundary**, **Edge**, or **Point** to specify the domains, boundaries, edges, or vertices, respectively, that you want to delete. Use **All domains**, **All boundaries**, **All edges**, or **All points** to select all entities of the specified dimension.

ADJACENT ENTITIES

Select the **Delete adjacent lower dimensional entities** check box to also delete the adjacent entities of lower dimensions.

Finalize

The **Finalize** node () ends a meshing sequence of imported type. It performs an associativity update for geometric entity numbers. You cannot delete, disable, or move the **Finalize** node. The software automatically builds all nodes in a meshing sequence, including the **Finalize** node, if you select a node in Model Builder outside the meshing sequence.

Import

Use an **Import** node () to import a mesh from file or other meshing sequence. It is only possible to import a mesh if the geometry sequence is empty. If the sequence already contains a mesh, the imported mesh is added to the existing mesh, forming an assembly.

To import a mesh, right-click a **Mesh** node and select **Import** (). Then enter the properties for the import using the following section:

IMPORT

In the **Mesh source** list choose the type of data to import—**Any importable file**, **Meshing sequence**, and **COMSOL Multiphysics file** are always available. In addition, you can choose **STL/VRML file** in 3D and **NASTRAN file** in 2D and 3D.

For file import, specify the file name in the **Filename** field or click the **Browse** button. For import from another mesh in the model, select the meshing sequence from the **Source** list below. To import a mesh, click the **Import** button (). If you have changed some property, the software automatically re-imports the mesh when you click a build button.

Properties for NASTRAN Import

You can import 3D meshes (and planar 2D meshes) in the NASTRAN bulk data format, the most common format for exchanging 3D meshes among programs. This format supports hundreds of NASTRAN entries describing elements, loads, and materials, making it possible to define a complete finite element model. When you import a NASTRAN bulk data file into COMSOL Multiphysics, the software imports mesh and material information only.

To import mesh and material data from a NASTRAN file select **Mesh and materials** in the **Data to import** list. Select **Only mesh** to import the mesh only.



For information on the NASTRAN entries that COMSOL Multiphysics supports, see [Import](#) in the *COMSOL API Reference Manual (Meshing)*.

Generally, a NASTRAN bulk data file contains only solid elements. This means that COMSOL enriches the imported mesh data with boundary elements, edge elements, and vertex elements such that a valid mesh object is formed. Each element in the imported mesh object also gets a unique label.

To use element types in the file to determine the domain partitioning of the domain elements, select the **Split on element type** check box (deselected by default). To use material data in the file to determine the domain partitioning of the domain elements, select the **Split on material data** check box (selected by default). To ignore extended node points (that is, node points not in element vertices), select the **Ignore extended node points** check box (deselected by default).

Boundary Partitioning Properties

If the partitioning of the boundary elements in the mesh to import into boundaries (faces) is not complete you can use the **Boundary partitioning** list to control the partitioning:

- Select **Automatic** to let the software partition the boundary elements into boundaries (faces) automatically (the default setting).
- Select **Minimal** to make a minimal boundary partitioning. This is useful when you import a mesh from a measured geometry or a NASTRAN mesh with a predefined boundary partitioning. The automatic face partitioning is not desired then.
- Select **Manual** to manually control the partitioning using the following parameters that become available:

If the face partitioning of the boundary elements in the mesh to import is not complete, the boundary elements are automatically partitioned into faces so that you get edges where neighboring triangles' normals make a large angle. To control this algorithm, change **Boundary partitioning** to **Manual**. The value in the **Maximum angle within boundary** field then limits the angle between any two boundary elements in the same boundary (face). The angle between neighboring boundary elements in the same boundary (face) is kept less than the value in the **Maximum boundary neighbor angle** field.

If the **Detect planar boundaries** check box is selected, the mesh import detects (approximately) planar faces (boundaries). A planar face has an area larger than the total area of all faces times the value in the **Minimum relative area** field. The angle between neighboring boundary elements in the same planar face is kept less than the value in the **Maximum neighbor angle** field.

The following settings are only available for importing 3D meshes:

Removal of Small Boundaries

The mesh import removes small boundaries (faces) with an area that is less the mean area of all faces times the value in the **Removal of small boundaries** field.

Advanced parameters

If **Manual** is selected from the **Advanced parameters** list, some additional controls for detection of extruded faces and faces with constant curvature become visible. In an extruded face all boundary elements are (approximately) orthogonal to the extruded plane (work plane). The angle between such a triangle's normal and the extruded plane is kept less than the value in the **Maximum angle to extruded plane** field (in degrees). An extruded boundary (face) has an area larger than the total area of all faces times the value in the **Detect extruded boundaries** field. In a boundary (face) with constant curvature, the relative deviation of the curvature at neighboring triangles is at most the value (in degrees) in the **Maximum curvature deviation in boundary** field. A boundary (face) with constant curvature gets an area larger than the total area of all faces times the value in the **Detect constant curvature** field.

Join Entities

Use a **Join Entities** node () to join adjacent geometric entities in an imported mesh.

To add a **Join Entities** node, right-click a 2D or 3D mesh node and select **Join Entities** (). Then use the following sections to specify the geometric entities to join:

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to join. You choose the geometric entity level from the **Geometric entity level** list: Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to join. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

ADJACENT ENTITIES

Select the **Join adjacent lower dimensional entities** check box to also join the adjacent entities of lower dimensions (boundaries and edges for joined domains, for example).

Logical Expression

Use a **Logical Expression** node () to split geometric entities of an imported mesh by creating at least one new geometric entity for the elements that fulfill the specified logical expression.

To add a **Logical Expression** node, right-click a 2D or 3D mesh node (that has an imported mesh) and select **Logical Expression** (). Then use the sections below to specify the geometric entities to split, the expression, and the split condition.

GEOMETRIC ENTITY SELECTION

Define the geometric entities that you want to split. You choose the geometric entity level from the **Geometric entity level** list:

- Choose **Entire geometry** to split all geometric entities according to the specified cylinder.
- Choose **Domain**, **Boundary**, or **Edge** to specify the domains, boundaries, or edges, respectively, that you want to split. Use **All domains**, **All boundaries**, or **All edges** to select all entities of the specified dimension.

EXPRESSION

Enter a logical expression using **x**, **y**, or **z** (3D only), [Unary, Binary, and List Operators and Their Precedence Rules](#), and [Mathematical and Numerical Constants](#). For instance, the expression $(x*x+y*y)<1$ defines a ball split in 2D and an infinite cylinder split in 3D.



By default, the expression is set to 1 not inducing any split of geometric entities.

CONDITION

Use the **Include element if expression is fulfilled for** list to select the condition for which the logical expression is fulfilled for an element. Choose **All vertices** to make an element satisfy the expression if it is true for all element vertices, or choose **Some vertex** if it is true for at least one element vertex.

Backward Compatibility

If you open a model created in the 3.5a version of COMSOL Multiphysics a **Mesh Object** node representing the mesh appears in the meshing sequence to handle backward compatibility.

Mesh Object

The settings window for the **Mesh Object** node () contains the follow section:

MESH OBJECT

If you save the model as a .java file, COMSOL Multiphysics uses the filename specified in the **Filename** field to determine the path to a mesh file, containing the mesh, that appears together with the .java file. The software uses this mesh file when you run the resulting .java file. By default, the filename has the prefix \$FILENAME\$. If the filename starts with this prefix, COMSOL puts the mesh file in the same directory as the .java file. It is also possible to remove this prefix and specify the full path to the mesh file.

To create a new mesh for a geometry with a **Mesh Object** node in its meshing sequence you first need to delete the **Mesh Object** node.

Mesh Element Quality and Size

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 to high condition numbers for the Jacobians, which in turn can cause convergence issues.

COMSOL Multiphysics® includes built-in 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 built-in variables for mesh element quality (qual) and mesh element size (h) in a surface plot, for example:

- 1 Under **Results** () , right-click **Data Sets** () and select **Mesh** () to add a Mesh data set.
- 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 quality (qual)** from the predefined quantities (under **Mesh**). Then click the **Plot** button (.

Mesh Statistics

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. The window contains the following sections:

GEOMETRIC SCOPE

Define the geometric entities for which you want to display the statistics. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to view statistics for the entire mesh.
- Choose **Domain** to specify the domains for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains**.
- Choose **Boundary** to specify the boundaries for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the boundaries in the Graphics window or choose **All boundaries**.
- Choose **Edge** to specify the edges for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the edges in the Graphics window or choose **All edges**. This option is only available in 3D.
- Choose **Point** to specify the points for which the statistics is displayed. Choose **Manual** in the **Selection** list to select the points in the Graphics window or choose **All points**. This option is only available in 2D and 3D.

STATISTICS

In this section you find information on the status of the mesh: for example, if the mesh is empty, if the geometry is partially meshed, or if the geometry is completely meshed. You can select the element type for which you want to see statistics from the **Element type** list. The default is to display statistics for **All elements**.

ELEMENT QUALITY HISTOGRAM

This section displays a histogram plot of the mesh element quality for the specified element type and selection. The *x*-axis represents the element quality, and the *y*-axis represents the number of elements of similar quality. The absolute value of the mesh element quality is always between 0 and 1, where 0.0 represents a degenerated element and 1.0 represents the best possible element (see [Quality of Elements](#) in the *COMSOL*

API Reference Manual)

You can also create a histogram plot of the mesh element quality over the total area or volume of the elements by adding a Histogram plot (see [Histogram](#)) to a 1D Plot Group and using the **Mesh quality** (`qual1`) as the expression.

Meshing Operations

The following sections describe the meshing operation nodes and their settings windows. The following table lists the available meshing operations:

TABLE 7-3: OPERATION NODES

ICON	NAME	DESCRIPTION
	Boundary Layers	Create boundary layer meshes
	Convert	Convert elements in a mesh
	Copy Domain (2D, 3D), Copy Face (3D), and Copy Edge (2D, 3D)	Copy a mesh between domains, faces, or edges
	Edge	Create an edge mesh
	Free Quad	Create an unstructured quadrilateral mesh
	Free Tetrahedral	Create an unstructured tetrahedral mesh
	Free Triangular	Create an unstructured triangular mesh
	Mapped	Create a mapped mesh
	Reference	Reference to other meshing sequence
	Refine	Refine a mesh
	Swept	Create a swept mesh

Boundary Layers

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 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 a quadrilateral mesh.

Additional elements of arbitrary type can also be inserted into the layers if needed.

To create a boundary layer mesh, right-click a 2D or 3D mesh node and select **Boundary Layers** (). Then enter the properties for the boundary layer mesher using the following sections:

DOMAIN SELECTION

Specify the domains where you want a boundary layer mesh by first choosing an option from the **Geometric entity level** list:

- Choose **Entire geometry** to specify boundary layer mesh for the entire geometry.
- Choose **Domain** to specify the domains for which you want a boundary layer mesh. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains** to select all domains.

CORNER SETTINGS

The following options for handling boundary layers at sharp corners are available from the **Handling of sharp corners** list (in 2D) and the **Handling of sharp edges** list (3D):

- Select **Splitting** (the default) to introduce boundary layer splits at sharp corners. In the **Minimum angle for splitting** field you specify the minimum angle between adjacent boundary layer boundaries for a split to occur. Control the maximum angle of the elements in the split region by the **Maximum angle per split** parameter.
- Select **Trimming** to trim the boundary layer mesh at sharp corners. In the **Minimum angle for trimming** and in the **Maximum angle for trimming** fields you specify the minimum angle and maximum angle, respectively, between adjacent boundary layer boundaries for trimming to occur.
- Select **None** to not use any special treatment at sharp corners.

In the **Maximum layer decrement** field you can specify the maximum difference in number of boundary layers between neighboring points on boundary layer boundaries.

TRANSITION

Select the **Smooth transition to interior mesh** check box to smooth the transition in element size from the boundary layer mesh to the interior mesh. You can specify the

number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

When a **Boundary Layers** node is added, a **Boundary Layer Properties** node is automatically added as a subnode. Use this subnode to specify the boundary layers and the properties of the boundary layers. If you want to specify different boundary layer properties for more than one boundary selection, right-click the **Boundary Layers** node and add additional **Boundary Layer Properties** subnodes. However, adjacent boundaries must have the same number of boundary layers.



Boundary Layer Properties

Convert

Some geometries have domains that are well suited for swept meshes. If there are surrounding domains that cannot be swept, you can convert faces with quadrilateral mesh between these domains to faces with triangular mesh. This makes it possible to generate adjacent-free tetrahedral mesh. Pyramid elements are generated in the interface between the triangular mesh of the converted face and the hexahedral or prism mesh in the domain.

It is also possible to convert the entire mesh to tetrahedral mesh. This is useful because there are a few computations, such as the adaptive solver, that can only be used with a simplex mesh (that is, a mesh with only tetrahedral and triangular elements).

You can convert a mixed mesh, consisting of tetrahedral, pyramid, prism, and hexahedral elements, to a pure tetrahedral mesh. The mesh conversion splits elements into several tetrahedral elements. In 2D and on faces in 3D, you can convert a mesh with quadrilateral elements to a mesh with only triangular elements. In 3D, adjacent domain elements are also split to conform to the split face elements.

To convert a mesh, right-click a 2D or 3D mesh node and select **Convert** (convert) in the **More Operations** submenu. Then use the following sections to specify the parts of the mesh to convert and the method that the conversion uses to split the elements:

GEOMETRIC SCOPE (3D) / DOMAINS (2D)

First define the geometric entities where you want to convert the mesh elements. You choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to convert the mesh elements on all domains (and all boundaries in 3D).
- Choose **Domain** to specify the domains for which you want to convert mesh elements. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Boundary** to specify the boundaries for which you want to convert mesh elements. Choose **Manual** in the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries. This option is only available in 3D.

ELEMENT SPLIT METHOD

From the **Element split method** list, select **Insert diagonal edges** (the default setting) to split each quadrilateral element into two triangular elements and each hexahedral element into five tetrahedral element, or select **Insert center points** to split each quadrilateral element into four triangular elements and each hexahedral element into 28 tetrahedral elements. The conversion also affects quadrilateral elements on the boundaries of the specified domains in 3D.

Both element split methods split each prismatic element into three tetrahedral elements. When pyramid elements are involved in a split, also other splits can be performed.

Copy Domain (2D, 3D), Copy Face (3D), and Copy Edge (2D, 3D)

Add a **Copy Domain** node in 2D/3D (/) , a **Copy Face** node in 3D () , or a **Copy Edge** node in 2D/3D (/) to create identical meshes on domains, boundaries or edges.

It is possible to copy a mesh from one or several source domains, boundaries, or edges onto one or several destination domains, boundaries, or edges. One-to-one and many-to-one **Single Destination** and many-to-many **Array Copy** (array of one-to-one operations) are allowed copy types. The source (or their combination, if many-to-one is used) must be a connected set of exactly the same shape as the corresponding destination, up to a constant scaling factor. More precisely, the distance between any two geometry vertices on the destination is required to be the same, up to a constant scaling factor, as the distance between the corresponding geometry vertices on the

source. As an exception, **Copy Edge** in 3D can also be used for destination edges of different shapes.

To copy a mesh between domains, boundaries, or edges, right-click a mesh node and select **Copy Domain** (in 2D or 3D), **Copy Face** (3D only), or **Copy Edge** (in 2D or 3D) in the **More Operations** submenu. Then enter the properties for the copy meshing operation using the following sections:

SOURCE DOMAINS/BOUNDARIES/EDGES

Activate the selection list by selecting its **Activate Selection** button (), and select the domains/boundaries/edges to copy the mesh from in the **Graphics** window.



The source domains/boundaries/edges must be connected when **Single destination (many-to-one)** option is specified. In an assembly, an identity pair is not sufficient to connect boundaries across parts. Instead, consider forming a union of the parts or splitting the destination boundary (using imprints, for example) so that the mesh copy is a one-to-one copy operation using two or more **Copy Face** or **Copy Edge** nodes.

DESTINATION DOMAINS/DOMAINS/EDGES

Activate the selection list by selecting its **Activate Selection** button (), and select the domains/boundaries/edges to copy the mesh to in the **Graphics** window.

TYPE OF COPY

Select **Automatic** to let the software determine the proper copy method (this is the default), select **Single Destination (many-to-one)** to let the entire source mesh be copied onto each destination entity separately, and select **Array Copy (many-to-many)** to let each single source entity mesh be copied onto a corresponding single destination entity.



Array Copy can be used only if a bijective transformation of source to destination can be found (a transformation that sets 1-to-1 mapping between source and destination).

SWITCH SELECTIONS

Click the **Switch Source and Destination** button to switch source and destination selections. Edge or vertex map (available only for **Copy Domain** and **Copy Face** nodes) is also switched, if provided.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

ORIENTATION (ONLY AVAILABLE FOR COPY EDGE)

Select **Automatic orientation** to let the software determine the orientation of the source mesh on the destination automatically (this is the default), select **Same orientation** to let the source mesh be copied to the destination according to the direction of the edges, and select **Opposite orientation** to let the source mesh be copied to the destination in the opposite direction. Use the option **Show edge direction arrows** in the **View** node under the **Definitions** node to view the arrow direction.

In the case of several source edges the orientation refers to the direction between the destination edge and the source edge with the lowest edge index.

If you want to control the orientation of the source mesh on the destination when using the **Copy Domain** or **Copy Face** nodes you need to add an [Edge Map](#), a [One-Point Map](#), or a [Two-Point Map](#) as a local attribute. To add a local attribute node, select one of the options in the context menu of the **Copy Domain** or the **Copy Face** node.

Copying a face mesh in 3D is only possible if the destination face is not adjacent to any meshed domain. The copy node overwrites any existing mesh on the destination face.

Copying a mesh to a destination domain in 2D or face in 3D that is adjacent to a meshed domain (2D) or face (3D) is possible if the edges between these domains or faces have the same number of elements as the corresponding source edges. The mesh on the destination edges is kept and the copied domain or face elements are modified to fit with this edge mesh.

Copying a mesh to a destination domain in 3D that is adjacent to a meshed domain is also possible if each face between these domains has a mesh isomorphic to a mesh of the corresponding source face. The mesh on the destination face is kept and the copied domain elements are modified to fit with this face mesh.

The edges around the source and destination domains or faces are allowed to be partitioned differently, but only in such a way that several edges of the source domain or face map to one edge of the destination edge. Not the other way around.

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).

Edge

Add an **Edge** node () to mesh edges. You can control the number of elements and the distribution of elements in the edge mesh by using **Size** and **Distribution** nodes.

To create an edge mesh, right-click a mesh node and select **Edge** in the **More Operations** submenu. Then enter the properties using the following sections:

EDGES (3D) / BOUNDARIES (2D) / DOMAINS (1D)

To define the edges where you want to create a mesh, first choose the level of the geometric entities from the **Geometric entity level** list:

- Choose **Entire geometry** to specify an edge mesh for the entire geometry.
- Choose **Remaining** to specify an edge mesh for remaining, unmeshed edges.
- Choose **Edge** (3D), **Boundary** (2D), or **Domain** (1D) to specify the edges for which you want to create a mesh. Choose **Manual** in the **Selection** list to select the edges in the **Graphics** window or choose **All edges** (3D), **All boundaries** (2D), or **All domains** (1D) to select all edges.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

Free Quad

Add a **Free Quad** node () to create an unstructured quadrilateral mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using **Size** and **Distribution** nodes.

To create an unstructured quadrilateral mesh in 2D, right-click a mesh node and select **Free Quad**. To create an unstructured quadrilateral mesh in 3D, right-click a mesh node and select **Free Quad** in the **More Operations** submenu. Then enter the properties for the quadrilateral meshing operation using the following sections:

BOUNDARIES (3D) / DOMAINS (2D)

Define the boundaries (3D) or domains (2D) where you want to create an unstructured quad mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured quad mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create an unstructured quad mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create an unstructured quad mesh. Choose **Manual** in the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the *x*-scale, *y*-scale, and *z*-scale in 3D to positive real numbers. If any of the scale factors are not equal to one the software scales the geometry in the *x*, *y*, and *z* directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic, and they are useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

Free Tetrahedral

Add a **Free Tetrahedral** node () to create an unstructured tetrahedral mesh. You can control the number, size, and distribution of elements by using **Size** and **Distribution** subnodes.

To create an unstructured tetrahedral mesh, right-click a 3D mesh node and select **Free Tetrahedral**. Then enter the properties for the tetrahedral meshing operation using the following sections:

DOMAIN SELECTION

Define the domains where you want to create an unstructured tetrahedral mesh.

Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured tetrahedral mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create an unstructured tetrahedral mesh in the entire geometry.
- Choose **Domain** to specify the domains for which you want to create an unstructured tetrahedral mesh. Choose **Manual** in the **Selection** list to select the domains in the Graphics window or choose **All domains** to select all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the *x*-scale, *y*-scale, and *z*-scale to positive real numbers. If any of the scale factors are not equal to one the software scales the geometry in the *x*, *y*, and *z* directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic, and they are useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

Free Triangular

Add a **Free Triangular** node () to create an unstructured triangular mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using **Size** and **Distribution** nodes.

To create an unstructured triangular mesh in 2D, right-click a mesh node and select **Free Triangular**. To create an unstructured triangular mesh in 3D, right-click a mesh node and select **Free Triangular** in the **More Operations** submenu. Then enter the properties for the triangular meshing operation using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundaries (3D) or domains (2D) where you want to create an unstructured triangular mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify unstructured triangular mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to create an unstructured triangular mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create an unstructured triangular mesh. Choose **Manual** in the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

SCALE GEOMETRY

To scale the geometry during the meshing operation, change the x -scale, y -scale, and z -scale in 3D to positive real numbers. If any of the scale factors are not equal to one the software scales the geometry in the x , y , and z directions before meshing; after meshing, it restores the geometry and mesh to fit the original size. The scale factors make it possible to generate meshes that are anisotropic, and they are useful if the mesh generator creates many elements due to a thin geometry or if the mesh generation fails due to large aspect ratios in the geometry.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

TRIANGULATION

Here you can specify the triangulation method used when creating the triangular mesh. Select **Automatic** (default) to let the software use the best suited method, select **Delaunay** to use a method based on a Delaunay algorithm, or select **Advancing front** to use a method based on an advancing front algorithm.

Mapped

Add a **Mapped** node () to create a structured quadrilateral mesh on boundaries in 3D and domains in 2D. You can control the number, size, and distribution of elements by using **Size** (only the Maximum element size parameter is used) and **Distribution** subnodes.

For the mapping technique to work properly, the geometry must be fairly 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 vertices or isolated boundary segments.
- The shape of each domain must not differ too much from a rectangle.

For a geometry model that initially does not meet these criteria, it is often possible to modify it such that a mapped mesh can be successfully generated.

To create a mapped quadrilateral mesh for each domain, the mapped mesher maps a regular grid defined on a logical unit square onto each domain. The mapping method is based on transfinite interpolation. The settings in the **Size** and **Distribution** nodes used by a Mapped node determine the density of the logical meshes. For the mapping technique to work, the opposite sides of each logical unit square must be discretized by the same number of edge elements.

By default the relationship between the four sides of the logical unit square and the boundaries around a domain is based on a criterion related to the sharpest angle between boundaries. If you want to control this relationship you need to add an Edge Group node as a subnode; right-click the **Mapped** node and select **Edge Groups**.

To create a mapped quadrilateral mesh in 2D, right-click a mesh node and select **Mapped**. To create a mapped quadrilateral in 3D, right-click a mesh node and select **Mapped** in the **More Operations** submenu. Then enter the properties for the mapped meshing operation using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundaries (3D) or domains (2D) where you want to create a mapped quad mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify mapped quad mesh for remaining, unmeshed domains.

- Choose **Entire geometry** to create a mapped quad mesh in the entire geometry.
- Choose **Boundary** (3D) or **Domain** (2D) to specify the geometric entities for which you want to create a mesh. Choose **Manual** from the **Selection** list to select the boundaries or domains in the **Graphics** window or choose **All boundaries** (3D) or **All domains** (2D) to select all boundaries or all domains.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

ADVANCED SETTINGS

In 3D, you can choose between two different interpolation methods in the **Interpolation method** list. This specifies how the mapped meshing operation determines the positions of the interior mesh points. If you select **Transfinite in 2D** the positions of the interior mesh points are determined by transfinite interpolation in the 2D parameter space of the corresponding surface and if you select **Transfinite in 3D** transfinite interpolation is done in 3D to determine these positions. Select **Auto** to let the mapped meshing operation determine a suitable interpolation method automatically.

Select the **Adjust evenly distributed edge mesh** check box to allow the mapped mesher to automatically adjust the mesh on edges that are not already meshed and where no explicit distribution is applied.

Reference

Use a **Reference** node () to refer to another meshing sequence. Building a **Reference** node runs the operation nodes of the referenced sequence. If you have a **Scale** node preceding a **Reference** node, or as a subnode to a **Reference** node, you can create a finer or coarser version of the mesh generated by the referenced sequence.

To refer to another meshing sequence, right-click a **Mesh** node and select **Reference** in the **More Operations** submenu. Then use the following section to specify the sequence to reference:

REFERENCE

Select the meshing sequence to reference.

It is possible to *expand* a reference (that is, replacing the reference with a copy of the referred sequence). If the reference node has a **Scale** subnode, the attribute nodes in the expanded sequence are scaled accordingly. In some cases, such scaling of attributes cannot be done explicitly, and additional scale nodes are created instead.

To expand a reference, right-click a reference node and select **Expand** ().

Refine

Use this node to refine a mesh by splitting elements.

To refine a mesh, right-click a 2D or 3D **Mesh** node and select **Refine** () in the **More Operations** submenu. Then use the following sections to specify the parts of the mesh to refine and the method that the Refine node uses to refine the elements:

DOMAIN SELECTION

Define the domains where you want to refine the mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to refine the entire mesh.
- Choose **Domain** to specify the domains for which you want to refine the mesh.
Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

REFINE OPTIONS

Refinement Method

From the **Refinement method** list, select **Regular refinement** to use the regular refinement method or select **Split longest side** to use the refinement method that splits the longest side when refining the mesh. The regular refinement method divides each element into four triangular elements of the same shape in 2D or eight tetrahedral elements of the same shape in 3D. The longest refinement method bisects the longest edge of each element. For 2D geometries COMSOL Multiphysics defaults to the regular refinement method, and in 3D the refinement method that splits the longest side is the default. In 1D, COMSOL always uses regular refinement, where it divides each element into two elements of the same shape.

Number of Refinements

Enter the number of consecutive mesh refinements in the **Number of refinements** field (the default is one refinement).



It is only possible to refine domains meshed with simplex elements (that is, segments in 1D, triangles in 2D, and tetrahedra in 3D). To refine the mesh in other domains, you must convert the mesh into simplex elements using the [Convert](#) node.

REFINE ELEMENTS IN BOX

Check **Specify bounding box** to refine the mesh only within a box. If you refine the mesh only on certain domains, the mesh is refined only in the intersection between the box and the domains.

Specify the box either by entering the coordinates of the lower left corner and upper-right corner of the box or click **Draw box** to interactively specify the box (only available in 2D).

Swept

The **Swept** node () creates a swept mesh on a domain in 3D by sweeping the mesh from the source face along the domain to an opposite destination face. The source and destination may consist of several connected faces. To create a swept mesh, right-click a 3D mesh node and select **Swept**.

You can control the number, size, and distribution of elements using the **Size** and **Distribution** subnodes. The **Swept** node only reads properties from **Size** nodes defined on the entire geometry or on the domain level and **Distribution** nodes defined on the domain level.

About the Source, Destination, and Linking Faces

Each face about a domain in a Swept node is classified as either a source face, a destination face, or a linking face. You can specify the source faces and destination faces

manually, but in most cases this is not necessary because the swept meshes can identify source and destination faces from the geometry.

-
- 
- If the source faces are not meshed prior to the sweeping operation the Swept node automatically creates a quadrilateral (or triangular) mesh before sweeping it to the destination.
 - If the source faces contain a triangular mesh the resulting swept mesh consists of prism elements.
 - If the source faces contain a quadrilateral mesh the resulting swept mesh consists of hexahedral elements.
-

The default is to create a quadrilateral face mesh but, depending on the source faces, that is not always possible.

The Geometry Criteria

For the sweeping technique to work, the geometry must satisfy these criteria:

- Each domain must be bounded by one shell; that is, a domain must not contain holes except if they penetrate both the source and destination face.
- The source and destination for a domain must be opposite each other in the domain's topology.
- Each destination face must correspond to one or more source faces.
- Each source face must correspond to precisely one destination face or a subset of it.
- The cross section along the direction of the sweep for a domain must be topologically constant.

Coincident source and destination faces are allowed.



If any of the faces about a domain is meshed prior to the sweeping operation, the following must be fulfilled.

- If the source and destination faces are meshed, these meshes must match.
 - Structured quad meshes must be applied to the linking faces.
-

DOMAIN SELECTION

Specify the domains where you want a swept mesh. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Remaining** to specify swept mesh for remaining, unmeshed domains.
- Choose **Entire geometry** to specify swept mesh for the entire geometry.
- Choose **Domain** to specify the domains for which you want a swept mesh. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

SOURCE FACES

To specify the source faces directly, activate the **Source Faces** list by selecting its **Activate Selection** button (), and select the faces defining the source of the sweep operation in the **Graphics** window.

DESTINATION FACES

To specify the destination faces directly, activate the **Destination Faces** list by selecting its **Activate Selection** button (), and select the faces defining the destination of the sweep operation in the **Graphics** window.

SWEEP METHOD

Face Meshing Method

In the **Face meshing method** list you can specify how the unmeshed source faces, which are meshed automatically by the Swept node, are meshed:

- Select **Quadrilateral (Generate hexahedrons)** to generate a surface mesh with quadrilateral elements. This is the default meshing method, but it does not work for all surfaces.
- Select **Triangular (Generate prisms)** to generate a surface mesh with triangular elements.

Sweeping Path

Use **Swept path calculation** if you want to specify the shape of the sweep path. **Sweep following straight lines** means that all interior mesh points are located on straight lines between the corresponding source and destination points. **Sweep following circular arcs** means that all interior mesh points are located on circular arcs between the corresponding source and destination points. **Sweep using interpolation** means that the positions of the interior mesh points are determined by a general interpolation procedure. The default, **Automatic**, means that the sweeping algorithm automatically

tries to determine if the sweep path is straight or circular, otherwise, the general approach is used.

Destination Mesh

Use **Destination mesh generation** if you want to specify the method to be used for transferring the source mesh to the destination face. **Morph source to destination** means that the destination mesh is created from the source mesh by a morphing technique, and **Use rigid transformation** means that the destination mesh is created by a rigid transformation of the source mesh. The default, **Determine suitable method**, means that the algorithm automatically tries to determine a suitable method for creating the destination mesh.

CONTROL ENTITIES

Select the **Smooth across removed control entities** check box to smooth the transition in element size across removed control entities. You can specify the number of smoothing iterations in the **Number of iterations** field. In the **Maximum element depth to process** field you can specify the maximum element depth, from the boundary layer interface, for the mesh points to be smoothed.

LINKING FACES

You can choose between two different interpolation methods for the linking faces in the **Interpolation method for linking faces** list. This specifies how the mapped mesher, which is used by the swept mesher for the linking faces, determines the positions of the interior mesh points. For more information on the different option see [Mapped](#).

Mesh Attributes

The following sections describe the attribute nodes and their settings windows. The following table includes the available mesh attributes:

TABLE 7-4: ATTRIBUTE NODES

ICON	NAME	DESCRIPTION
	Boundary Layer Properties	Boundary layer properties
	Corner Refinement	Corner refinement properties
	Distribution	Mesh distribution properties
	Edge Groups	Edge groups used by Mapped nodes
	Edge Map	Edge map used by Copy Face and Copy Domain nodes
	One-Point Map	One-point map used by Copy Face and Copy Domain nodes
	Scale	Scale the element size of a mesh
	Size	Mesh size properties
	Two-Point Map	Two-point map used by Copy Face and Copy Domain nodes

Boundary Layer Properties

Add a **Boundary Layer Properties** node () to specify the location of the boundary layers and the properties, such as the number and thickness of the boundary layers.

To add a Boundary Layer Properties node as a subnode to a **Boundary Layers** node select **Boundary Layer Properties** in the right-click menu of the **Boundary Layers** node. Then enter the properties using the following sections:

BOUNDARIES

Define the boundaries where you want boundary layers. Choose **Manual** in the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.

BOUNDARY LAYER PROPERTIES

Use the **Number of boundary layers** field to specify the total number of boundary layers.

In the **Boundary layer stretching factor** field you specify the increase in thickness between two consecutive boundary layers as a scaling factor; for example, entering 1.3 means that the thickness increases by 30% from one layer to the next.

Use the **Thickness of first layer** list to specify the thickness of the first element layer—the layer adjacent to the corresponding boundary. If you select **Automatic** (the default setting), the thickness of the first layer is 1/20 of the local domain element height. Use the **Thickness adjustment factor** field to specify a scaling factor that multiplies this default size. Alternatively, select **Manual** from the **Thickness of first layer** list to specify the thickness of the first layer explicitly using the **Thickness** field.



The boundary layer meshing algorithm shrinks the boundary layers automatically if needed, for example, due to a narrow region. However, the stretching factor is always respected. In some cases the boundary layer meshing algorithm can choose to create fewer layers than specified. If this happens a warning is printed to the Log page of the **Progress** window.

Corner Refinement

Add a **Corner Refinement** node () to decrease the element size at sharp corners. The node considers a vertex in 2D or an edge in 3D to be a sharp corner if the angle between the adjacent selected boundaries, with respect to the selected domain, is greater than a specified angle.

It is possible to add Corner Refinement nodes both as global nodes and as local nodes. If there are several Corner Refinement nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last Corner Refinement node in the sequence.

To add a Corner Refinement node as a global node, right-click a mesh node and select **Corner Refinement**. To add a Corner Refinement node as a subnode to an operation node, right-click the operation node and select **Corner Refinement**. Then enter the properties using the following sections:

DOMAIN SELECTION

Specify the domains for which the node determines if the specified corners are sharp. Choose the level of the geometry from the **Geometric entity level** list:

- Choose **Entire geometry** to specify that the node should determine sharp corners with respect to all domains. The corner refinement also considers corners not adjacent to any domain.
- Choose **Domain** to specify the domains for which you want to determine sharp corners. Choose **Manual** in the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

BOUNDARY SELECTION

To specify the boundaries, select the **Activate Selection** button (), and select the boundaries in the **Graphics** window for which the corner refinement should determine the sharp corners.

ANGLE

Use the **Minimum angle between boundaries** field to specify the minimum angle between a pair of adjacent boundaries in the boundary selection for the refinement factor to apply at the vertex in 2D and edge(s) in 3D between the two boundaries. If a boundary pair is adjacent to one domain on each side (interior boundary) the corner refinement determines the angle(s) on the side(s) corresponding to the specified domain(s).

REFINEMENT

Use the **Element size scaling factor** field to specify a refinement factor (<1) that scales the element size for the vertices in 2D and edges in 3D corresponding to the sharp corners.

Distribution

Use the **Distribution** node () to specify the distribution of mesh elements along an edge, for example. It is possible to add **Distribution** nodes both as global nodes and as local nodes. If there are several **Distribution** nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last **Distribution** node in the sequence. Distribution properties always override properties defined by **Size** nodes sharing the same selections.

To add this node as a global node, right-click a mesh node and select **Distribution**. To add this node as a subnode to an operation node, right-click an operation node and select **Distribution**.

GEOMETRIC SCOPE (3D) / BOUNDARIES (2D) / DOMAIN SELECTION (1D)

Define the geometric entities where you want to specify a distribution. Choose the level of the geometry from the **Geometric entity level** list (only available in 3D):

- Choose **Domain** to specify the domains for the distribution. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Edge** to specify the edges for the distribution. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges.

Edge is the only option in 2D, and **Domain** is the only option in 1D.

DISTRIBUTION

There are three main distribution methods that you select from the **Distribution properties** list:

- Select **Explicit distribution** to use an explicit, user-defined element distribution. To define the distribution of mesh elements, enter a vector-valued expression of strictly increasing values starting with zero (using comma-separated numbers), specifying the relative arc length values of the mesh vertices along the edge or boundary.
- Select **Fixed number of elements** to use a fixed number of mesh elements, which you enter into the **Number of elements** field. This is the default option.
- Select **Predefined distribution type** to specify properties of a predefined distribution method that can be a geometric sequence (exponentially increasing or decreasing element size) or an arithmetic sequence (equal distance between elements); see below for details.

Predefined Distribution Type Settings

In the **Number of elements** field, enter the number of elements (the default is 5 elements). To specify the ratio in size between the last element and first element in the distribution, use the **Element ratio** field (the default value is 1.0; that is, the first and the last elements have the same size). From the **Distribution method** list, select **Arithmetic sequence** for a linear element distribution or select **Geometric sequence** for an exponential element distribution. Select the **Symmetric** check box to get a symmetric distribution, and select the **Reverse direction** check box to switch the element distribution to the opposite direction along the edge or boundary. If you have specified several edges in the selection the **Reverse direction** check box refers to the edge in the selection with lowest entity number (the *master edge* in the selection). For the other

edges, their direction (with respect to the distribution) is such that the rotation with respect to the master edge is minimized.



Meshing Operations

Edge Groups

Use an **Edge Groups** node () to specify the four groups of edges around a boundary (3D) or domain (2D) that is used to determine the **Mapped** mesh of the boundary/domain.

To add an Edge Groups node as a subnode to a **Mapped** node select **Edge Groups** in the right-click menu of the Mapped node. Then enter the properties using the following sections:

BOUNDARIES (3D) / DOMAIN SELECTION (2D)

Define the boundary/domain where you want to specify the edge groups. Choose **Manual** in the **Selection** list to select the boundary/domain in the **Graphics** window.

FIRST EDGE GROUP

Activate the **First Edge Group** list by selecting its **Activate Selection** button (), and select the edges for the first edge group in the **Graphics** window.

SECOND EDGE GROUP

Activate the **Second Edge Group** list by selecting its **Activate Selection** button (), and select the edges for the second edge group in the **Graphics** window.

THIRD EDGE GROUP

Activate the **Third Edge Group** list by selecting its **Activate Selection** button (), and select the edges for the third edge group in the **Graphics** window.

FOURTH EDGE GROUP

Activate the **Fourth Edge Group** list by selecting its **Activate Selection** button (), and select the edges for the fourth edge group in the **Graphics** window.

Edge Map

Use an **Edge Map** node () to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node. Using this node, the source mesh of the Copy Face or Copy Domain operation is transformed so that the source edge of the Edge Map node is mapped onto the destination edge of the Edge Map node with the specified orientation. To add an Edge Map subnode to a **Copy Face** or a **Copy Domain** node right-click the node and select **Edge Map**.

EDGES

Activate the **Source edge** list by selecting its **Activate Selection** button () , and select the edge that you want to define as source edge in the **Graphics** window.

Activate the **Destination edge** list by selecting its **Activate Selection** button () , and select the edge that you want to define as destination edge in the **Graphics** window.

ORIENTATION

Select **Automatic orientation** to let the software determine the orientation of the mesh of the source edge on the destination edge (this is the default), select **Same orientation** to let the mesh of the source edge be copied to the destination edge according to the directions of the edges, or select **Opposite orientation** to let the mesh of the source edge be copied to the destination edge in the opposite direction.



Copy Domain (2D, 3D), Copy Face (3D), and Copy Edge (2D, 3D)

One-Point Map

Use a **One-Point Map** node () to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node.

To add a One-Point Map node as a subnode to a **Copy Face** or a **Copy Domain** node right-click the node and select **One-Point Map** from the context menu. Then enter the properties using the following sections:

POINT SELECTION

Activate the **Point on source** list by selecting its **Activate Selection** button () , and select the point that you want to define as source point in the **Graphics** window.

Activate the **Point on destination** list by selecting its **Activate Selection** button (), and select the point that you want to define as destination point in the **Graphics** window.



Copy Domain (2D, 3D), Copy Face (3D), and Copy Edge (2D, 3D)

Scale

Use a **Scale** node () to scale the properties of the Size, Distribution, and Boundary Layer Properties nodes. It is possible to add Scale nodes both as global nodes and as local nodes to **Reference** nodes. A Scale node that exists as a global node affects the size of the mesh elements generated by the subsequent operation nodes. A Scale node that exists as a subnode to a Reference node affects the size of the mesh elements generated by the Reference node only.

If two or more Scale nodes exist on the same selection, the resulting scale factor on that selection is the product of the given scale factors.

To add a Scale node as a global node, right-click a **Mesh** node and select **Scale**. To add a Scale node as a subnode to a **Reference** node, right-click a **Reference** node and select **Scale**.

GEOMETRIC SCOPE

In this section you define the geometric entities where you want to specify a scale. Choose the level of the geometry from the **Geometric entity level** list.

- Choose **Entire geometry** to specify the scale for the entire geometry.
- Choose **Domain** to specify the domains for the scale specification. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.

- Choose **Boundary** to specify the boundaries for the scale specification. Choose **Manual** from the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.
- Choose **Edge** to specify the edges for the scale specification. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges. This option is only available in 3D.
- Choose **Point** to specify the points for the scale specification. Choose **Manual** from the **Selection** list to select the points in the **Graphics** window or choose **All points** to select all points. This option is only available in 2D and 3D.

SCALE

Specify the scale factor in the **Element size scale** field.

A scale factor less than 1 gives smaller (more) elements; a scale greater than 1 gives larger (fewer) elements.



It is not possible to use a coarser mesh size setting on a geometric entity adjacent to a higher dimensional entity with a finer mesh size setting; the finer setting on a geometric entity overrides the coarser setting on its boundary. Therefore a scale factor larger than 1 might have no effect if the dimensional level of the selection is less than the space dimension.

Size

Use a **Size** node () to specify the size of mesh elements. It is possible to add Size nodes both as global nodes and as local nodes. If there are several Size nodes in the sequence with a nonempty selection intersection, the mesher uses properties corresponding to the last Size node in the sequence.

A meshing sequence corresponding to a nonempty geometry contains a Size node at the first position in the sequence. This Size node, referred to as the *default Size node*, is defined for the entire geometry and cannot be removed. To override the mesh size settings defined by this Size node, add another Size node to the sequence.

To add a Size node as a global node, right-click a mesh node and select **Size**. To add a Size node as a subnode to an operation node, right-click an operation node and select **Size**.

GEOMETRIC SCOPE



This section is not available for the default Size node.

In this section you define the geometric entities where you want to specify a size. Choose the level of the geometry from the **Geometric entity level** list (only available in 3D):

- Choose **Entire geometry** to specify the size for the entire geometry.
- Choose **Domain** to specify the domains for the size specification. Choose **Manual** from the **Selection** list to select the domains in the **Graphics** window or choose **All domains** to select all domains.
- Choose **Boundary** to specify the boundaries for the size specification. Choose **Manual** from the **Selection** list to select the boundaries in the **Graphics** window or choose **All boundaries** to select all boundaries.
- Choose **Edge** to specify the edges for the size specification. Choose **Manual** from the **Selection** list to select the edges in the **Graphics** window or choose **All edges** to select all edges. This option is only available in 3D.
- Choose **Point** to specify the points for the size specification. Choose **Manual** from the **Selection** list to select the points in the **Graphics** window or choose **All points** to select all points. This option is only available in 2D and 3D.

ELEMENT SIZE

In the **Calibrate for** list you can select the physics for which the element size is calibrated.

If you select **Predefined** you can choose a predefined element size that automatically determines the parameters in the **Element Size Parameters** section. A predefined element size can generate an extremely fine, extra fine, finer, fine, normal, coarse, coarser, extra coarse, or extremely coarse mesh. Select **Custom** if you want to change the value for any parameters in the **Element Size Parameters** section. For a Size node that is not the default Size node, if you select **Custom**, a check box is automatically added by the node for each field in the **Element Size Parameters** section. You need to select the check box for a field to activate the corresponding parameter.

ELEMENT SIZE PARAMETERS

You can freely specify all element size parameters using numerical values or user-defined parameters. For a Size node that is not the default Size node, you need to select **Custom** in the **Element Size** section to be able to edit the parameters. The check box in front of each parameter (only available for Size nodes that are not the default Size node) determines if the parameter is active in the node. The following parameters control the mesh element size (the parameters where you define a size use the geometry's length unit):

- Use the value in the **Maximum element size** field to limit the allowed element size. For example, you may want to limit the maximum element size to a fraction of the wavelength to make sure that the wave propagation is fully resolved. By using a parametric sweep to vary the maximum element size you can solve the model using meshes with different mesh density to study how it affects the solution.
- The value in the **Minimum element size** field specifies the minimum allowed element size. You can use this value to, for example, prevent the generation of many elements around small curved parts of the geometry. This parameter is not available in 1D.
- The **Maximum element growth rate** determines the maximum 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.
- The value in the **Resolution of curvature** field determines 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. This parameter is not available in 1D.
- In the **Resolution of narrow regions** field you control the number of layers of elements that are created in narrow regions (approximately). The value must be a nonnegative scalar. A higher value gives a finer mesh in narrow regions. If the value of this parameter is less than one, the mesh generator might create elements that are anisotropic in size in narrow regions.



It is not possible to use a coarser mesh size setting on a geometric entity adjacent to a higher dimensional entity with a finer mesh size setting; the finer setting on a geometric entity overrides the coarser setting on its boundary. A warning is given when coarser settings are overridden.



Two-Point Map

Use a **Two-Point Map** node () to specify the orientation of the source mesh on the destination for a Copy Face or a Copy Domain node.

To add a Two-Point Map node as a subnode to a **Copy Face** or a **Copy Domain** node, right-click the node and select **Two-Point Map** from its context menu. Then enter the properties using the following sections:

SOURCE POINTS

Activate the **First point on source** list by selecting its **Activate Selection** button (), and select the point that you want to define as first source point in the **Graphics** window.

Activate the **Second point on source** list by selecting its **Activate Selection** button (), and select the point that you want to define as second source point in the **Graphics** window.

DESTINATION POINTS

Activate the **First point on destination** list by selecting its **Activate Selection** button (), and select the point that you want to define as first destination point in the **Graphics** window.

Activate the **Second point on destination** list by selecting its **Activate Selection** button (), and select the point that you want to define as second destination point in the **Graphics** window.



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.

VISUALIZING LINEARIZED ELEMENTS

The variable `linearizedelem` is 1 in elements that are linearized and 0 elsewhere. You can use this variable to identify mesh elements with linearized elements.

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 built-in `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.

8

Materials

This chapter includes information about how to work with materials in models and describes the material databases included with COMSOL Multiphysics® and the add-on modules.

Materials Overview

About Materials and Material Properties

MATERIALS

In COMSOL models, you can add one or more *materials*, which are named collections of *material properties*. Each such material is represented by a **Material** node (), typically with a name that describes the material, such as **Copper**, under **Model>Materials** in the **Model Builder**. Each material includes a number of physical properties with the values or functions (for temperature-dependent material properties, for example) that describe the material.

MATERIAL PROPERTIES AND PROPERTY GROUPS

The material properties are organized in material *property groups*, which appears as subnodes under the **Material** node in the **Model Builder**:

- The **Basic** property group contains common material properties that can generally be measured and are meaningful without any context.
- User-defined groups may contain a subset of the same quantities.
- Each predefined property group contains one or more material properties that are only meaningful together and in the context of a particular material model.
- The material property values are outputs of the material, which can be constant values or functions of *model inputs* (physical quantities like temperature and pressure) In principle, the physics user interfaces first ask a material which inputs it requires to compute its output properties, then asks the material to compute property values given values of the model inputs—for example, thermal conductivity (output) as function of temperature (input).
- Each property group can also define a set of local properties and functions that can be used together with model inputs in output property expressions. This makes it possible to, for example, create generic materials for certain classes of some type of material and use the local properties to parameterize the material.

About the Material Databases

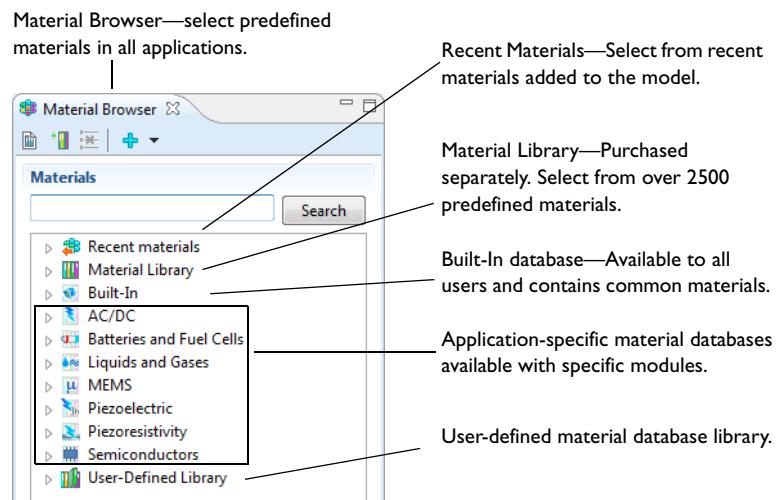


Figure 8-1: The Material Browser and the material databases.

All COMSOL Multiphysics modules have predefined material data available in collections of materials—material databases—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 collections of 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. You can also create custom materials and material libraries by researching and entering material properties.

The **Material Browser** window provides access to all material databases (including the Material Library). The following material databases are available in the Material Browser (some require additional module licenses):

- **Recent Materials:** From the **Recent Materials** folder (blue icon), 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** (purple icon), contains data for over 2500 materials and 20,000 property functions.
- **Built-In.** Included with COMSOL Multiphysics, the **Built-In** database (grey icon) contains common solid materials with electrical, structural, and thermal properties.

- *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 () 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-type and n-type silicon materials.
- *Semiconductors*. Included in the Semiconductor Module, the Semiconductors database contains silicon, gallium arsenide, and germanium materials for use with this module.
- *User-Defined Library*. The **User-Defined Library** folder () is where user-defined material databases (libraries) are created. When you have created a new database, it also displays in the **Material Browser**.



The material databases shipped with COMSOL Multiphysics are read-only. This includes the Material Library and any materials shipped with the optional modules.



- [Module-Specific Material Databases](#)
- [Adding a New User-Defined Library](#)

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 nodes. When the first material is added to a model, COMSOL Multiphysics automatically assigns that material to all domains in the geometry (or all boundaries or edges if the model only contains surfaces or edges). 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.

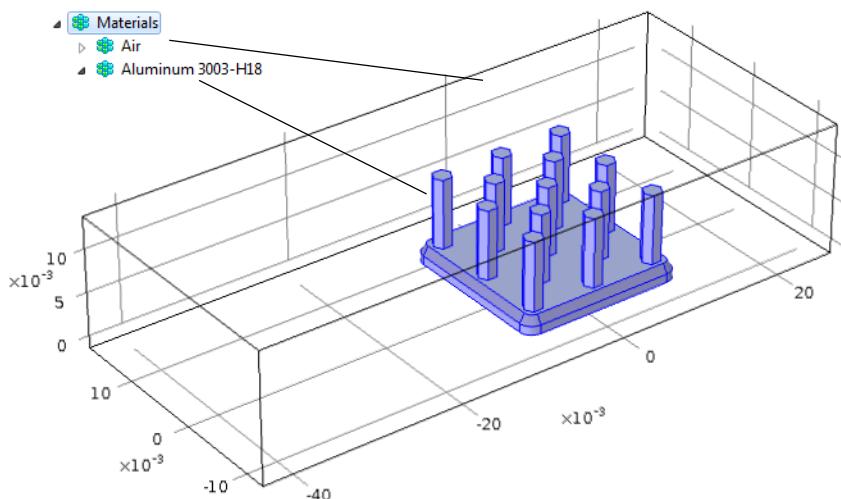


Figure 8-2: Assigning materials to a heat sink model. Air is assigned as the material to the box surrounding the heat sink, and aluminum to the heat sink itself.

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 8-2). The **Conjugate Heat Transfer** user interface, selected during model set-up, uses a **Fluid** model to simulate non-isothermal 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 I** settings use the material properties associated to the **Aluminum 3003-H18** materials node, and the **Fluid I** settings define the flow using the **Air** material properties. The other nodes under **Conjugate Heat Transfer** define the initial and boundary conditions.

All physics properties automatically use the correct material properties from the **Material** nodes 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.



The Material Settings Window

There are also some physics nodes where you can explicitly select a material from which material properties are retrieved (for example, the **Fluid Properties** node's **Settings** window for two-phase flow modeling). The default setting is then typically to use the **Domain material** on each domain (that is, the materials defined on the same domains as the physics that uses the material data). 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 are then 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, for example.



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. The numbers 1, 2, and 3 denote the first, second, and third direction, respectively, in the active coordinate system. In the general case, you can define a 3-by-3 tensor, for example, k_{ij} in the order $k_{11}, k_{21}, k_{31}, k_{12}, k_{22}, k_{32}, k_{13}, k_{23}$, and k_{33} . 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. Functions can be plotted directly from the function nodes' settings window by first specifying suitable ranges for the input arguments.

- Many physics user interfaces also define variables for the material properties that they use. For example, `solid.rho` is the density in the Solid Mechanics user interface and is equal to the density in a material when it is used in the domains where the Solid Mechanics user interface is active. If you define the density in the Solid Mechanics user 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 and Adding Materials

- 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**.



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).



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** settings window opens.



For detailed instructions, see [About the Output Material Properties](#).

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.



You can also modify and extend existing materials that you load from any of the material libraries. When added to the model, the material is a copy of the properties and the for the material from the library, and you can modify that material's properties in the same way as a user-defined material.

Adding New 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** settings window opens.
- 3 Add a selection to the **Material** node (the domains, for example, that consist of this material).
- 4 Add material properties as required using the settings in the **Material** node's settings window. See [The Material Settings Window](#).
- 5 Right-click the **Material** node to rename it, for example, using the name of the material it represents.

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.

Adding a New User-Defined Library

When you first open the **Material Browser**, an empty **User-Defined Library** is available for you to start creating your own library of materials (see [Figure 8-1](#)). These steps describe how to copy this existing library and rename it to one suitable for your purposes.

- 1** In the **Material Browser**, click the **New Material Library** button () on the toolbar.
- 2** In the **New 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 on Windows might be similar to
`C:\Users\Your_Name\.comsol\v43b\materials.`
 - on Linux, the file path is typically `~/.comsol/v43b/material`.
 - On the Mac, it is typically `<home folder>/Library/Preferences/COMSOL/v43b/material` (if missing, click the Finder’s **Go** menu and hold down the Option key to show the **Library** folder). 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. The new name must include underscores (_) between words, for example **My_Metals.mph**.
- 4** Click **Save**. The empty database, with a new name, is added to the Material Browser.

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** from the context menu.
- 3** Right-click to **Rename Selected** material (as required) that has been added to your library. See [Figure 8-3](#).

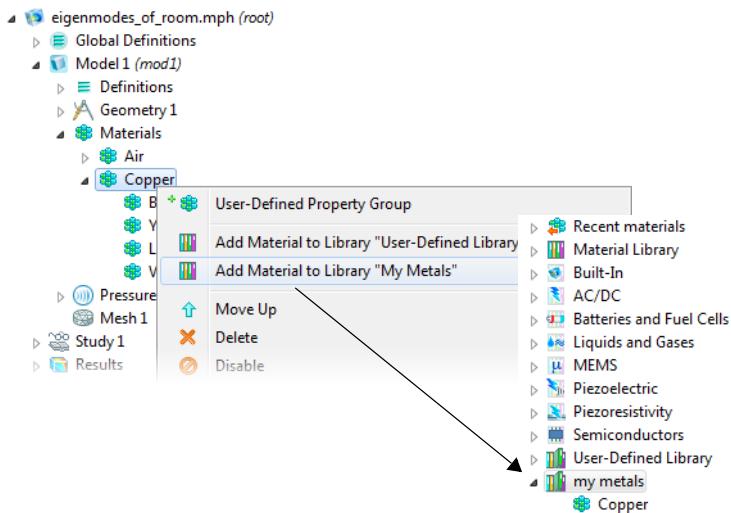


Figure 8-3: Adding a predefined material to a new user-defined library.

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 [Adding a New User-Defined Library](#) and then import the file to the Material Browser.

Material Nodes and Material Browser

The Materials Branch

Use the nodes under **Materials** () to add predefined or user-defined materials, to specify specific material properties using model inputs, functions, values, and expressions as needed, or to 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 using the materials' node labels from the model tree according to the settings under **View>Model Builder Node Label**.

The **Selection** column lists the geometric entities selected for the material (the domains, boundaries, or edges where the material is defined).

ERRORS RELATING TO THE MATERIAL NODES

If a material property in a physics user 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. The Material nodes in also indicate when there is a material error. For example, if some property is deleted but needed in a part of the geometry, then the icon indicates where the error is located (see [Figure 8-4](#)).

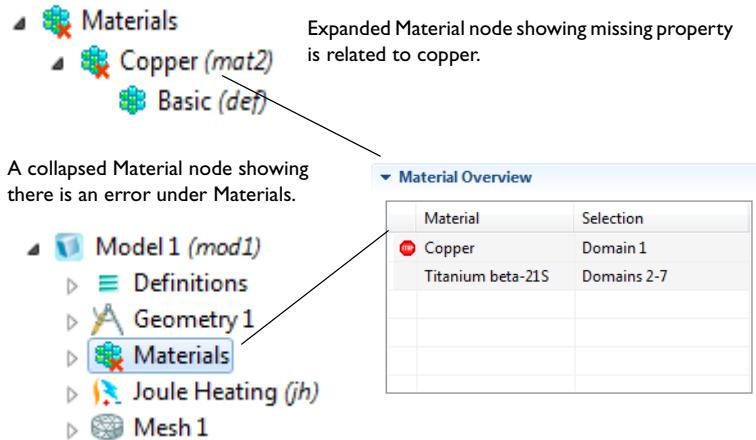


Figure 8-4: An example of a Material node error.

The Material Browser Window

The **Material Browser** window () contains a number of databases with a broad collection of elastic, solid mechanics, electromagnetic, fluid, chemical, thermal, piezoelectric, and piezoresistive properties of materials. Use the Material Browser to find predefined materials and add them to the Model Builder, or create a custom material library.

MATERIALS

In this section you can browse all the available material databases or search for specific materials. There is also a **Recent Materials** folder where you find the most recently used materials. **Search** a specific material by name (or, for the Material Library product, by UNS number or DIN number). You can also browse for a specific material. When the material is located, right-click to **Add Material to Model**.

INFORMATION

When browsing the material databases, in particular the **Material Library**, some materials include additional information—UNS number, DIN number, and composition—that appears in this section.

MATERIAL PROPERTIES

While browsing the databases, predefined material properties for the selected material are listed in a table under **Property** together with their **Expression**, **Unit**, and the **Property group** to which the material property belongs. If **Property group** is empty, the material property is a **Basic** property.

FUNCTIONS

For 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. The functions are used, for example, to define temperature-dependent material properties.

INPUTS

For some materials, predefined function inputs are listed under **Input** in a table together with their **Unit** and **Property groups**. Inputs appear for material properties defined using functions that require the input. Typical inputs are the temperature and the pressure, for temperature- and pressure-dependent material properties.

The Material Settings Window

The **Material** settings window () summarizes the predefined or user-defined material properties for a material. This is where you can add or change material properties to fit your model and assign the material to all types of geometric entities: domains (most common), boundaries, edges (3D models only), or points.

After adding a material, click the material node (for example, **Material 1** or **Copper**) in the **Model Builder**. The **Material** settings window opens.

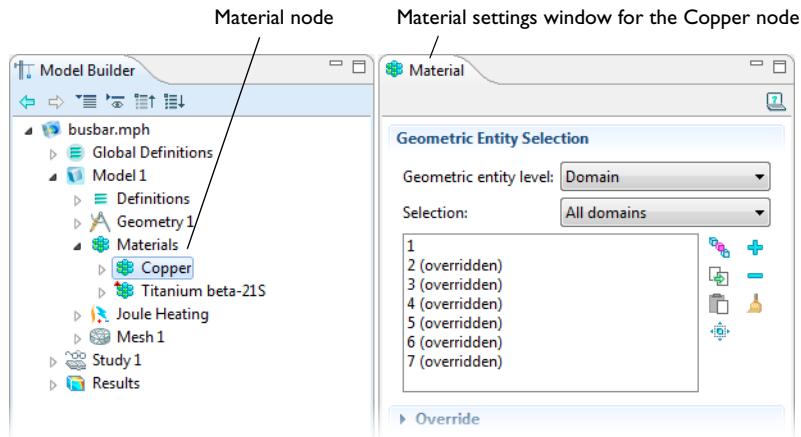


Figure 8-5: Click the Copper node to open the Material settings window for the node.

GEOMETRIC ENTITY SELECTION

- 4 Assign the material to some or all entities on a specific **Geometric entity level**—**Domain**, **Boundary**, **Edge** (3D only), or **Point**—on the geometry in the **Graphics** window (the geometry in the model).

 By default, the first material in the model is active in all domains (or all boundaries or edges if the model only contains surfaces or edges). By assigning other materials to some or all domains, the first material is overridden and remains active only in domains where no other material, added below it in the **Materials** branch, is active.

 If the model contains features on different geometric entity levels, such as solid mechanics in domains coupled to beams on edges, and the features 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.

OVERRIDE

This section shows if the material, in some or all parts of the geometry where it is active, is overridden by another material added underneath it in the Materials branch, or if it overrides another material above it.

The **Overridden by** list shows the names of the materials that override this material. The **Selection** list in the **Geometric Entity List** section displays **(overridden)** for the geometric entities where this material is overridden.

The **Overrides** list shows the names of the materials that this material overrides.

MATERIAL PROPERTIES

You can add material properties to the model if they are not already included. To do so, browse the available material property categories (**Basic Properties**, **Acoustics**, and so on), and select a material property or a collection of material properties in one of the property groups or material models that appear under the main level of material property categories. Right-click the material property or property group and select **Add to Material**, or click the **Add to Material** button (+) to add the material property or group of properties to the material.

 Review the properties listed in the **Material Contents** table before adding new material properties.

For example, under **Acoustics>Viscous Model** select **Bulk viscosity (muB)** and right-click to **Add to Material** or click the **Add** button (+). If you add a material model like the **Viscous Model** with more than one property, all its material properties are added to the **Material Contents**. In this example, a **Viscous model** node is added to the **Model Builder**, and its associated properties are added to the **Material Contents** table.

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** settings windows.

- When material properties are added from the **Basic** node's or a user-defined group node's **Property Group** settings window, they are listed under **Output Properties and Model Inputs** in that settings window.
- When material properties are added from the **Material** settings window, the available material properties are listed under **Material Properties** and are added to the list under **Material Contents** with the property group listed. The list under **Material Contents** also contains material properties added from a subnode with a **Property Group** settings window.

From the **Domain type** list, choose **Solid** or **Non-solid** (for gases and liquids). Some features use this information to choose the correct frame for the computation.



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.

MATERIAL CONTENTS

This section lists all the material properties that are defined for the material or required by the physics in the model. The table lists the **Property**, **Name**, **Value**, and **Unit** for the material property as well as the **Property group** that the material property belongs to. The **Property group** corresponds to the subnodes in the **Model Builder** with the same name. If required, edit the values or expression for the property's **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 feature in the model but is undefined. When you enter a value in the **Value** column, the material property is added to its 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 in the model.

APPEARANCE

The settings in this section make it possible to control or change the default appearance of a material in the Graphics window when working in the materials or physics parts of the model tree.



In 3D models, the material is rendered including color and texture when **Scene Light** is active. In 2D models and in 3D models when **Scene Light** is turned off, only a change of color is visible.

The **Family** list provides quick settings approximating the appearance of a number of common materials—**Air**, **Aluminum**, **Brick**, **Concrete**, **Copper**, **Gold**, **Iron**, **Lead**, **Magnesium**, **Plastic**, **Steel**, **Titanium**, and **Water**. Select **Custom** to make further adjustments of the specific settings for colors, texture, reflectance, and so on. The default custom settings are inherited from the material selected last from the **Family** list.



The texture and reflectance properties only take effect 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.

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** gives 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. The diffuse color 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. Ambient color 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.



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 that 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 between 0–1 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 because 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 determines the size of the specular highlight. 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 it can also represent many other materials.

- Reflectance at normal incidence is the amount of incoming light (0–1) from the normal direction (of the surface) that is reflected.
- Surface roughness is a value that describes microreflectance on the surface. Higher values create a rougher look of the surface, with fewer highlights. A value from 0–1 is suitable.

Property Groups

The **Property Group** settings window is where output properties and the model inputs are added, local properties are defined, and expressions for material properties are entered in a specific property group such as **Basic**. The property groups are subnodes to a material node. The **Property Group** settings window is displayed when you click the property group node (for example, **Basic**) under the material node (typically with the material's name—**Aluminum**, for example) in the **Model Builder**.

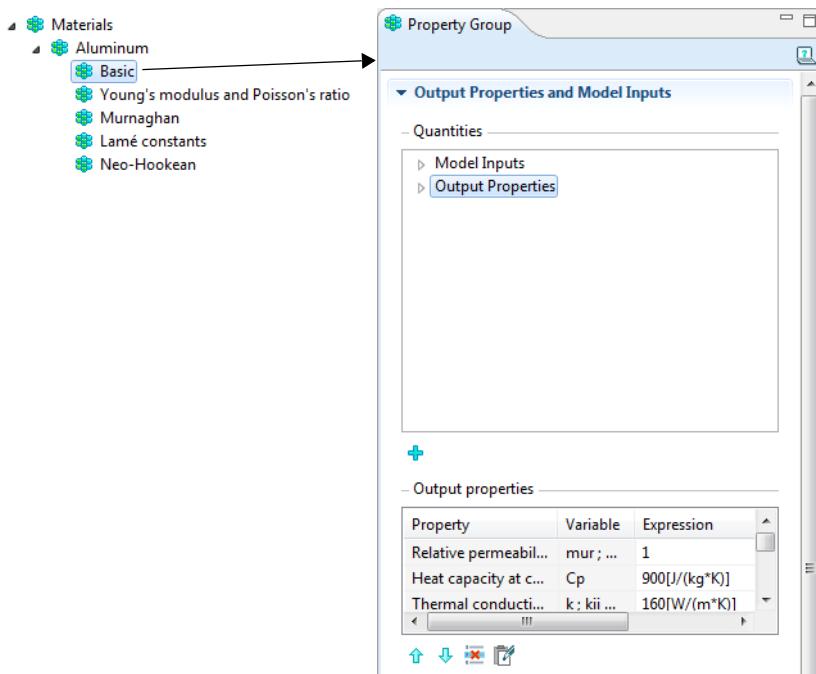


Figure 8-6: An example of a Basic Property Group settings window.

OUTPUT PROPERTIES AND MODEL INPUTS

The predefined material properties in the property group appears in the **Output properties** table. Under **Quantities** you can add additional material properties to the **Output properties** list or add model inputs to the **Model inputs** list.



Output Properties under **Quantities** is only available from the **Basic** material properties and with user-defined property groups.

The model inputs are physical quantities such as temperature that are used as inputs in the expressions that define the output properties (for example, to describe a temperature-dependent physical quantity). For example, adding **Temperature** as a model input with the variable name **T** makes it possible to use an expression for the heat capacity at constant pressure C_p such as $300[\text{J} / (\text{kg}\cdot\text{K})] \cdot T[1/\text{K}]$ 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** list's **Expression** column; edit directly in the table or by clicking the **Edit** button (), which opens a dialog box for easier specification of orthotropic and anisotropic material properties (tensors): Select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** when entering the data in the material property's 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 for a 3-by-3 tensor, for example, k_{ij} in the order $k_{11}, k_{21}, k_{31}, k_{12}, k_{22}, k_{32}, k_{13}, k_{23}$, and k_{33} . 1, 2, and 3 represent the first, second, and third direction in the active coordinate system.

Use the **Move Up** (), **Move Down** (), and **Delete** () buttons to organize the tables as required.

LOCAL PROPERTIES

Here you can enter a user-defined **Property** and its corresponding **Expression** and organize the table as required. These local properties are useful for parameterizing functions that describe material properties, if they contain other inputs than those that are model inputs (such as temperature and pressure). For example, a local property can be a reference value at a certain temperature. Use the **Move Up** (), **Move Down** (), and **Delete** () buttons to organize the tables as required.



You can use local properties to parameterize a material (for example, to create a generic “template” material for a particular symmetry class of anisotropic materials). You can then adjust the local property values for each instance of the material.

About Automatic Adding of Property Groups to a Material

Material property groups are automatically added to the material node in the **Model Builder**. You can also add additional predefined property groups or create a **User-Defined Property Group** by right-clicking the Material node. The available properties are collected in property groups according to the physical context.

Each property group has a **Property Group** settings window. When a **Model Builder** node is clicked (for example, **Basic**), the **Property Group** settings window displays specific information about that property group. The physical properties for all property groups are summarized in a **Material Contents** table on the **Material** settings window.

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.

In the **Basic>Property Group** window, you can add **Output Properties** under **Quantities** subsection. You can also add **Model Inputs** to, for example, create a temperature-dependent material property.

About the Output Material Properties



Some of these material groups are only used by physics interfaces in the add-on modules and detailed information is in the applicable documentation.

This section describes all available property groups and the material properties that they contain. These material properties can be added to models from two settings windows: the **Material** window and its subnodes' **Property Group** windows.

The Basic group contains over 25 basic properties for use with all materials.



Materials

BASIC MATERIAL PROPERTIES

These common material properties belong to the **Basic** property group.

- When this information is accessed from the **Basic>Property Group** window, it is listed under **Quantities>Output Properties** and **Variable** is listed in the table.
- When this information is accessed from the **Material** window, it is listed under **Material Properties>Basic Properties** and **Name** is listed in the table under **Material Contents**.

TABLE 8-1: BASIC MATERIAL PROPERTIES

PROPERTY	NAME/VARIABLE	SI UNIT
Absorption coefficient	kappaR	l/m
Characteristic acoustic impedance	Z	Pa·s/m

TABLE 8-I: BASIC MATERIAL PROPERTIES

PROPERTY	NAME/VARIABLE	SI UNIT
Coefficient of thermal expansion	alpha	1/K
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	epsilonor	1
Resistivity	res	Ω·m
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



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.

ACOUSTICS

Under Acoustics you find two acoustic material models with their associated material properties: a **Thermoacoustics Model** and a **Viscous Model**.

The **Thermoacoustics Model** property group adds thermal and viscous properties for a thermoacoustics material model.

These material property groups (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Acoustics Module.

TABLE 8-2: ACOUSTICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
THERMOACOUSTICS MODEL		
Bulk viscosity	μ_B	$\text{Pa}\cdot\text{s}$
Density	ρ	kg/m^3
Dynamic viscosity	μ	$\text{Pa}\cdot\text{s}$
Heat capacity at constant pressure	C_p	$\text{J}/(\text{kg}\cdot\text{K})$
Thermal conductivity	k	$\text{W}/(\text{m}\cdot\text{K})$
VISCOUS MODEL		
Bulk viscosity	μ_B	$\text{Pa}\cdot\text{s}$

ELECTROCHEMISTRY

These material property groups for electrochemistry (including their associated physical properties) can be added to models from the **Material** window. These property

groups require the Batteries & Fuel Cells Module, Corrosion Module, or Electrodeposition Module.

TABLE 8-3: ELECTROCHEMISTRY MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELECTRODE POTENTIAL		
Equilibrium potential	Eeq	V
Temperature derivative of equilibrium potential	dEeqdT	V/K
Reference concentration	cEeqref	mol/m ³
ELECTROLYTE CONDUCTIVITY		
Electrolyte conductivity	sigmal	S/m
LINEARIZED RESISTIVITY		
	This material node defines the electric resistivity (and conductivity) as a linear function of temperature.	
Reference resistivity	rho0	Ωm
Resistivity temperature coefficient	alpha	I/K
Reference temperature	Tref	K
SPECIES PROPERTIES		
Transport number	transNum	I

ELECTROMAGNETIC MODELS

These material property groups for various electromagnetic material models (including their associated physical properties) can be added to models from the **Material** window. These properties require the AC/DC Module or RF Module.

TABLE 8-4: ELECTROMAGNETIC MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
BH CURVE	This material node requires the AC/DC Module.	
Local Properties	normH	-
Magnetic flux density norm	normB	T
DIELECTRIC LOSSES		
	This material node requires the AC/DC Module.	
Relative permittivity (imaginary part)	epsilonBis	1
Relative permittivity (real part)	epsilonPrim	1
Dielectric loss factor	eta_epsilon	

TABLE 8-4: ELECTROMAGNETIC MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
HB CURVE	This material node requires the AC/DC Module.	
Local Parameters	normB	-
Magnetic field norm	normH	A/m
LINEARIZED RESISTIVITY	This material node defines the electric resistivity (and conductivity) as a linear function of temperature. and requires the AC/DC Module.	
Reference resistivity	rho0	Ωm
Resistivity temperature coefficient	alpha	1/K
Reference temperature	Tref	K
LOSS TANGENT	This material node assumes zero conductivity and requires the RF Module.	
Loss tangent	delta	-
Relative permittivity (real part)	epsilonPrim	1
MAGNETIC LOSSES	This material node requires the RF Module.	
Relative permeability (real part)	murPrim	-
Relative permeability (imaginary part)	murBis	-
REFRACTIVE INDEX	This material node assumes a relative permeability of unity and zero conductivity and requires the RF Module.	
Refractive index	n	1
Refractive index, imaginary part	ki	-

GAS MODELS

This material property group for an ideal gas (including its associated physical properties) can be added to models from the **Material** page.

TABLE 8-5: 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)

PIEZOELECTRIC MODELS

These material property groups for piezoelectric materials (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Acoustics Module, MEMS Module, or Structural Mechanics Module.

TABLE 8-6: 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	epsilonnrS	I
Loss factor for elasticity matrix	cE	I
Loss factor for coupling matrix	e	I
Loss factor for electrical permittivity	epsilonT	I
STRAIN-CHARGE FORM		
Compliance matrix sE	sE	I/Pa
Coupling matrix	dET	C/N
Relative permittivity	epsilonnrT	I
Loss factor for compliance matrix	sE	I
Loss factor for coupling matrix	d	I
Loss factor for electrical permittivity	epsilonT	I

PIEZORESISTIVE MODELS

These material property groups for piezoresistive materials (including their associated physical properties) can be added to models from the **Material** window.

TABLE 8-7: GAS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELASTORESISTANCE FORM	The Elastoresistance Form node adds the elastoresistive coupling matrix property. Requires the MEMS Module.	
PIEZORESISTANCE FORM	The Piezoresistance Form node adds the piezoresistive coupling matrix property. Requires the MEMS Module.	

SOLID MECHANICS

These material property groups for material models in solid mechanics (including their associated physical properties) can be added to models from the **Material** window.

TABLE 8-8: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
LINEAR ELASTIC MATERIAL		
ANISOTROPIC		
Elasticity matrix	D	Pa
Loss factor for elasticity matrix D	eta_D	I
ANISOTROPIC, VOIGT NOTATION		
Elasticity matrix, Voigt notation	DVo	Pa
Loss factor for elasticity matrix D, Voigt notation	eta_DVo	I
BULK MODULUS AND SHEAR MODULUS		
Bulk modulus	K	N/m ²
Shear modulus	G	N/m ²
LAMÉ PARAMETERS		
Lamé parameter λ	lambLame	N/m ²
Lamé parameter μ	muLame	N/m ²
ORTHOTROPIC		
Young's modulus	Evector	Pa
Poisson's ratio	nuvector	I
Shear modulus	Gvector	N/m ²
Loss factor for orthotropic Young's modulus	eta_Evector	I
Loss factor for orthotropic shear modulus	eta_Gvector	I
ORTHOTROPIC, VOIGT NOTATION		
Shear modulus, Voigt notation	GvectorVo	N/m ²
Loss factor for orthotropic shear modulus, Voigt notation	eta_GvectorVo	I
PRESSURE-WAVE AND SHEAR-WAVE SPEEDS		
Pressure-wave speed	cp	m/s
Shear-wave speed	cs	m/s
YOUNG'S MODULUS AND POISSON'S RATIO		
Young's modulus	E	Pa
Poisson's ratio	nu	I

TABLE 8-8: SOLID MECHANICS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
YOUNG'S MODULUS AND SHEAR MODULUS		
Young's modulus	E	Pa
Shear modulus	G	N/m ²
LINEAR VISCOELASTIC MATERIAL		
Long-term shear modulus	Gv	N/m ²
Bulk modulus	K	N/m ²
POROELASTIC MATERIAL		
Biot-Willis coefficient	alphaB	I
Porosity	epsilon	I
Permeability	kappa	m ²

	<ul style="list-style-type: none"> The <i>Geomechanics Module User's Guide</i> and Table 8-11 The <i>Nonlinear Structural Materials Module User's Guide</i> and Table 8-9 The <i>Fatigue Module User's Guide</i> and Table 8-10

SOLID MECHANICS: NONLINEAR STRUCTURAL MATERIALS MODULE

These material property groups for material models in solid mechanics using the Nonlinear Structural Materials Module (including their associated physical properties) can be added to models from the **Material** window.

TABLE 8-9: HYPERELASTIC AND ELASTOPLASTIC MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELASTOPLASTIC MATERIAL		
Initial yield stress	sigmags	Pa
Isotropic tangent modulus	Et	Pa
Kinematic tangent modulus	Ek	Pa
Hill's coefficients	Hillcoefficients	(m ² ·s ⁴)/kg ²
Initial tensile and shear yield stresses	ys	N/m ²
HYPERELASTIC MATERIALS		
ARRUDA-BOYCE		
Macroscopic shear modulus	mu0	N/m ²
Number of segments	Nseg	I

TABLE 8-9: HYPERELASTIC AND ELASTOPLASTIC MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
BLATZ-KO		
Model parameters	phiBK	I
Model parameters	betaBK	I
Shear modulus	muBK	Pa
GAO		
Model parameters	aG	Pa
Model parameters	nG	I
GENT		
Macroscopic shear modulus	muG	Pa
Model parameters	jmG	I
MOONEY-RIVLIN		
Model parameters	C01, C02, C03, C10, C11, C12, C20, C21, C30	Pa
MURNAGHAN	The Murnaghan node adds five model parameters. The model is based on strain invariants and is typically used in acoustoelasticity.	
Murnaghan third-order elastic moduli	I	Pa
Murnaghan third-order elastic moduli	m	Pa
Murnaghan third-order elastic moduli	n	Pa
Lamé parameter λ	lambLame	Pa
Lamé parameter μ	muLame	Pa
VARGA		
Model parameters	c1VA	Pa
Model parameters	c2VA	Pa
YEOH		
Model parameters	c1YE	Pa
Model parameters	c2YE	Pa
Model parameters	c3YE	Pa

SOLID MECHANICS: FATIGUE MODULE

These material property groups for material models in solid mechanics using the Fatigue Module (including their associated physical properties) can be added to models from the **Material** window.

TABLE 8-10: ELASTOPLASTIC AND FATIGUE BEHAVIOR MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
ELASTOPLASTIC MATERIAL>RAMBERG-OSGOOD		
Cyclic hardening coefficient	K_ROcyclic	Pa
Cyclic hardening coefficient	n_ROcyclic	l
FATIGUE BEHAVIOR>STRAIN BASED		
COFFIN-MANSON		
Fatigue ductility coefficient	epsilonf_CM	l
Fatigue ductility exponent	c_CM	l
Shear fatigue ductility coefficient	gammaf_CM	l
Shear fatigue ductility exponent	cgamma_CM	l
FATEMI-SOCIE		
Normal stress sensitivity coefficient	k_FS	l
WANG-BROWN		
Normal stress sensitivity coefficient	S_WB	l
FATIGUE BEHAVIOR>STRESS BASED		
BASQUIN		
Fatigue strength coefficient	sigmaf_Basquin	Pa
Fatigue strength exponent	b_Basquin	l
Shear fatigue strength coefficient	tauf_Basquin	Pa
Shear fatigue strength exponent	bgamma_Basquin	l
FINDLEY		
Normal stress sensitivity coefficient	k_Findley	l
Limit factor	f_Findley	Pa
MATAKE		
Normal stress sensitivity coefficient	k_Matake	l
Limit factor	f_Matake	Pa

TABLE 8-10: ELASTOPLASTIC AND FATIGUE BEHAVIOR MATERIAL PROPERTIES

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
NORMAL STRESS		
Limit factor	f_NormalStress	Pa

SOLID MECHANICS>GEOMECHANICS MATERIAL MODEL

These material property groups for material models in solid mechanics (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Geomechanics Module.

TABLE 8-11: GEOMECHANICS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
CAM-CLAY MATERIAL MODEL		
Swelling index	kappaSwelling	
Compression index	lambdaComp	
Initial void ratio	e0	
Cam-Clay M parameter	M	
DRUCKER-PRAGER		
Drucker-Prager alpha coefficient	alphaDrucker	
Drucker-Prager k coefficient	kDrucker	Pa
HOEK BROWN		
Hoek-Brown m parameter	mHB	
Hoek-Brown s parameter	sHB	
Geological strength index	GSI	
Disturbance factor	Dfactor	
Intact rock parameter	miHB	
LADE-DUNCAN		
Lade-Duncan k coefficient	kLade	
MATSUOKA-NAKAI		
Matsuoka-Nakai mu coefficient	muMatsuoka	
MOHR-COULOMB		
Cohesion	cohesion	Pa
Angle of internal friction	internalphi	rad
OTTOSEN		
Ottosen a parameter	aOttosen	
Ottosen b parameter	bOttosen	
Size factor	k1Ottosen	

TABLE 8-11: GEOMECHANICS MODELS MATERIALS

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	SI UNIT
Shape factor	k2Ottosen	I
YIELD STRESS PARAMETERS		
Uniaxial tensile strength	sigmaut	Pa
Uniaxial compressive strength	sigmauc	Pa
Biaxial compressive strength	sigmabc	Pa

SEMICONDUCTORS

These material property groups for all the material models in semiconductors (including their associated physical properties) can be added to models from the **Material** window. These property groups require the Semiconductor Module.



The Property Group, Variable Names, and SI Unit columns are applicable to all materials in the Semiconductor Module. However, the Values and References columns listed in [Table 8-12](#) are specifically for Silicon.

TABLE 8-12: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	VALUE FOR SILICON	SI UNIT	REFERENCE FOR SILICON
BASIC				
Relative permittivity	epsilon0r	11.7		1
Thermal conductivity	k	131 W/(m·K)		1
Density	rho	2329 kg/m ³		1
Heat capacity at constant pressure	Cp	700 J/(kg·K)		1
SEMICONDUCTOR MATERIAL				
Band gap	Eg0	1.12 V (valid at 300 K)	V	1
Effective density of states, conduction band	Nc	$2.8 \times 10^{19} \text{ } \text{cm}^{-3}$ $\times (T/300 \text{ K})^{3/2}$	cm^{-3}	1
Effective density of states, valence band	Nv	$1.04 \times 10^{19} \text{ } \text{cm}^{-3}$ $\times (T/300 \text{ K})^{3/2}$	cm^{-3}	1
Electron affinity	chi0	4.05 V	V	1
Electron mobility	mun	1450 cm ² /(V·s)	$\text{m}^2/(\text{V}\cdot\text{s})$	1

TABLE 8-12: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	VALUE FOR SILICON	SI UNIT	REFERENCE FOR SILICON
Hole mobility	μ_{up}	$500 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	1
GENERATION-RECOMBINATION>AUGER RECOMBINATION				
Auger recombination factor, electrons	C_n	$2.8 \times 10^{-31} \text{ cm}^6/\text{s}$ (valid at 300 K)	m^6/s	2
Auger recombination factor, holes	C_p	$9.9 \times 10^{-32} \text{ cm}^6/\text{s}$ (valid at 300 K)	m^6/s	2
GENERATION-RECOMBINATION>DIRECT RECOMBINATION				
Direct recombination factor	C	$0 \text{ m}^3/\text{s}$	m^3/s	N/A
GENERATION-RECOMBINATION>SHOCKLEY-READ-HALL RECOMBINATION				
Electron lifetime, SRH	τ_{aun}	$10 \mu\text{s}$	s	3
Hole lifetime, SRH	τ_{aup}	$10 \mu\text{s}$	s	3
MOBILITY MODELS>ARORA MOBILITY MODEL				
Electron mobility reference	$\mu_{\text{un0_ref_arora}}$	$1252 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Hole mobility reference	$\mu_{\text{up0_ref_arora}}$	$407 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Electron mobility reference minimum	$\mu_{\text{un_min_ref_arora}}$	$88 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Hole mobility reference minimum	$\mu_{\text{up_min_ref_arora}}$	$53.4 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Electron reference impurity concentration	$N_{\text{n0_ref_arora}}$	$1.26 \times 10^{17} \text{ l}/\text{cm}^3$	l/m^3	4
Hole reference impurity concentration	$N_{\text{p0_ref_arora}}$	$2.35 \times 10^{17} \text{ l}/\text{cm}^3$	l/m^3	4
Alpha coefficient	$\alpha_{\text{0_arora}}$	0.88		4
Mobility reference minimum exponent	$\beta_{\text{1_arora}}$	-0.57		4
Mobility reference exponent	$\beta_{\text{2_arora}}$	-2.33		4

TABLE 8-12: SEMICONDUCTOR MATERIAL PROPERTIES (ALL MATERIALS) AND VALUES AND REFERENCES FOR SILICON

PROPERTY GROUP AND PROPERTY	NAME/VARIABLE	VALUE FOR SILICON	SI UNIT	REFERENCE FOR SILICON
Impurity concentration reference exponent	beta3_arora	2.4	I	4
Alpha coefficient exponent	beta4_arora	-0.146	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Reference temperature	Tref_arora	300 K	K	4
MOBILITY MODELS>FLETCHER MOBILITY MODEL				
Fletcher mobility coefficient 1	F1_fl	$1.04 \times 10^{21} \text{ I}/(\text{cm}\cdot\text{V}\cdot\text{s})$	$\text{I}/(\text{cm}\cdot\text{V}\cdot\text{s})$	5
Fletcher mobility coefficient 2	F2_fl	$7.45 \times 10^{13} \text{ I}/\text{cm}^2$	I/m^2	5
Reference temperature	Tref_fl	300 K	K	5
MOBILITY MODELS>POWER LAW MOBILITY MODEL				
Electron mobility reference	mun0_pl	$1448 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Hole mobility reference	mup0_pl	$473 \text{ cm}^2/(\text{V}\cdot\text{s})$	$\text{m}^2/(\text{V}\cdot\text{s})$	4
Electron exponent	alphan_pl	2.33	I	4
Hole exponent	alphap_pl	2.23	I	4
Reference temperature	Tref_pl	300 K	K	4

Model Inputs

The following model inputs (which are scalar or vector-field physical quantities that appear as inputs in, for example, a temperature-dependent material property) can be added to models from the **Property Group** window (vector fields have three components enclosed by curly braces).

TABLE 8-13: MODEL INPUTS

MODEL INPUT	NAME/VARIABLE
Absolute Pressure	pA
Concentration	c

TABLE 8-13: MODEL INPUTS

MODEL INPUT	NAME/VARIABLE
Current Density	{J1, J2, J3}
Electric Field	{E1, E2, E3}
Frequency	freq
Magnetic Field	{H1, H2, H3}
Magnetic Flux Density	{B1, B2, B3}
Number Density	nd
Strain Reference Temperature	Tempref
Stress Tensor	{F1, F2, F3}
Temperature	T
Velocity Field	{u1, u2, u3}

Using Functions in Materials

Functions are useful for describing material properties as, for example, functions of temperature or pressure.

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 add an analytic function of one or more input arguments.
 - Select **Interpolation** to add an interpolation function that can interpolate from structured data (defined on a grid) or unstructured data (defined on a generic point cloud).
 - Select **Piecewise** to add a piecewise function that is useful if a material property has different definitions on different intervals. The intervals must not overlap, and there cannot be any holes between intervals.

	<ul style="list-style-type: none">• Example of Defining an Analytic Function• Analytic• Interpolation• Piecewise
	Once a function is created, you can use it for any property in the same property group.

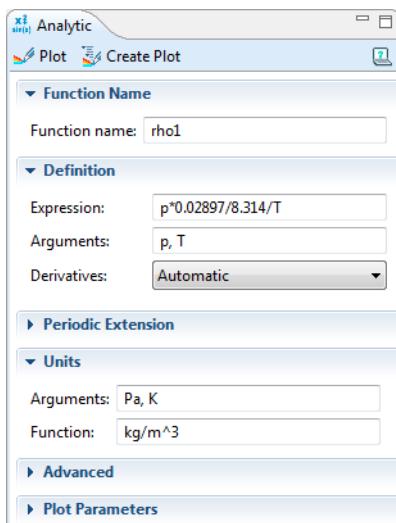
Example of Defining an Analytic Function

Assume that you want to define the density ρ_1 for a material as a function of pressure and temperature: $\rho_1 = \rho_1(p, T)$. You can name the function `rho1(p,T)` and use the expression `p*0.02897/8.314/T` to define the function.

- 1 Open the **Material Browser**.
- 2 Add a new material to the model (or use an existing material where density is not defined, or redefine the current expression for the density).
- 3 Add a **Density** property to the material.
 - a In the **Model Builder**, click the material node.
 - b On the **Material** page, click to expand the **Material Properties** section. Under **Basic Properties**, right-click **Density** and **Add to Material**.
A **Density** property is added to the **Basic** property group.
- 4 In the **Model Builder**, under the material node, right-click **Basic** and select **Functions>Analytic**. This adds an **Analytic** subnode () under **Basic**.
- 5 On the **Analytic** settings window, enter `rho1` in **Function Name**.
- 6 Under **Definition**:
 - a In the **Expression** field enter `p*0.02897/8.314/T`.
 - b In the **Arguments** column, enter `p,T`.

7 Under Units:

- a In the **Arguments** field, enter Pa, K as the units for the pressure and the temperature, respectively.
- b In the **Function** field: enter kg/m³ as the unit for the function's output (density). The function rho1 can now be used to define the density in your material.



8 Click the material node. On the Material settings window, under Material Contents, enter rho1(p,T) in the Value column (in the Density row).

Material Contents				
Property	Name	Value	Unit	Property group
Density	rho	<code>rho1(p,T)</code>	kg/...	Basic

Click the **Basic** node to notice that the **Density** analytic function is defined on the **Property Group** settings window under **Output properties**. See [Figure 8-7](#).

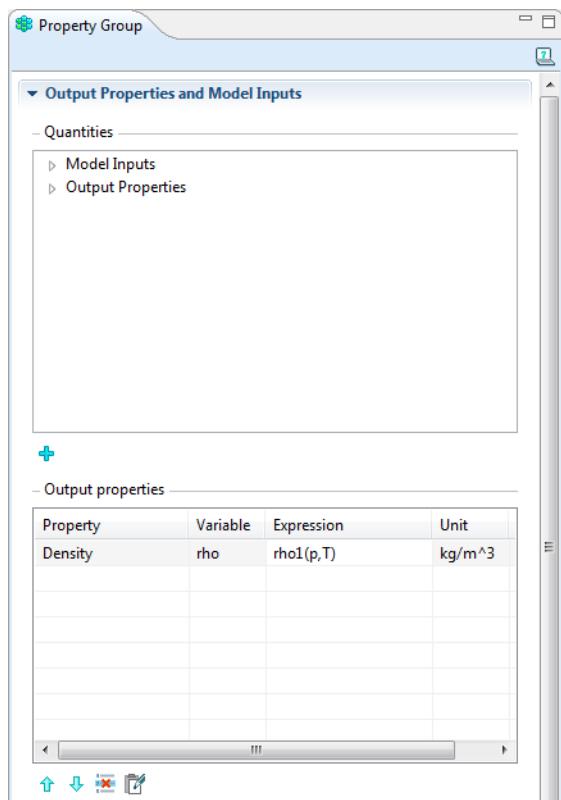


Figure 8-7: A density property is defined using an analytic function.

Module-Specific Material Databases

-
- 
 - See [About the Material Databases](#) for information about which modules have these material databases.
 - For more information about customizing the material's appearance in the Graphics window, see [The Material Settings Window](#).
-

AC/DC Material Database

The electromagnetic material properties that can be stored in the material databases are:

- Electrical conductivity and resistivity
- Relative permittivity
- Relative permeability
- Nonlinear BH-curves
- Refractive index

The database contains electromagnetic and other material properties for these materials:

PREDEFINED MATERIALS

Copper

Soft Iron (without losses)

Soft Iron (with losses)

Quartz

Graphite

Graphite felt

Silicon Carbide

Some properties depend on the magnetic flux density, location, or temperature. The database contains, depending on the material and in addition to the more common material properties, the following properties:

PREDEFINED PROPERTIES
Remnant flux density
Reference temperature
Temperature coefficient
Nonlinear BH-curves
Resistivity at reference temperature

Liquids and Gases Material Database

The Liquids and Gases materials database contains thermal and fluid dynamic properties for a set of common liquids and gases. All properties are given as functions of temperature and at atmospheric pressure, except the density, which for gases is also a function of the local pressure. The database also contains surface and interface tensions for a selected set of liquid/gas and liquid/liquid systems. All functions are based on data collected from scientific publications.

TABLE 8-14: LIQUIDS AND GASES MATERIALS

GROUP	MATERIAL	GROUP	MATERIAL
Gases	Air	Liquids	Engine oil
References 1, 2, 7, and 8	Nitrogen	References 2, 3, 4, 5, 6, 7, 9, and 10	Ethanol
	Oxygen		Diethyl ether
	Carbon dioxide		Ethylene glycol
	Hydrogen		Gasoline
	Helium		Glycerol
	Steam		Heptane
	Propane		Mercury
	Ethanol vapor		Toluene
	Diethyl ether vapor		Transformer oil
	Freon 12 vapor		Water
	SiF4		

MEMS Material Database

The MEMS material database contains 33 materials commonly used in MEMS applications. The materials are divided into the following groups: metals, semiconductors, insulators, and polymers.

The basic structure of this library comes from the book *Microsensors, MEMS, and Smart Devices* (Ref. 1). The material properties come from two primary sources: the *CRC Handbook of Chemistry and Physics* (Ref. 2) and *MacMillan's Chemical and Physical Data* (Ref. 3). Some of the mechanical properties in the library are instead more MEMS-specific values from *The MEMS Handbook* (Ref. 4), and most of the semiconductor properties are values from Ref. 5. Ref. 6 provides a valuable resource for cross-checking the insulation material properties.

The table below lists the materials and the corresponding groups:

TABLE 8-15: MEMS MATERIALS

MATERIAL	GROUP
Aluminium (Al)	Metals
Silver (Ag)	Metals
Gold (Au)	Metals
Chrome (Cr)	Metals
Copper (Cu)	Metals
Indium (In)	Metals
Titanium (Ti)	Metals
Iron (Fe)	Metals
Nickel (Ni)	Metals
Lead (Pb)	Metals
Palladium (Pd)	Metals
Platine (Pt)	Metals
Antimon (Sb)	Metals
Tungsten (W)	Metals
C [100]	Semiconductors
GaAs	Semiconductors
Ge	Semiconductors
InSb	Semiconductors
Si(c)	Semiconductors

TABLE 8-15: MEMS MATERIALS

MATERIAL	GROUP
Poly-Si	Semiconductors
Silicon (single-crystal)	Semiconductors
Al ₂ O ₃	Insulators
SiC (6H)	Insulators
Si ₃ N ₄	Insulators
SiO ₂	Insulators
ZnO	Insulators
Borosilicate	Insulators
Nylon	Polymers
PMMA	Polymers
Polymide	Polymers
Polyethylene	Polymers
PTFE	Polymers
PVC	Polymers

Piezoelectric Materials Database

The Piezoelectric materials database included with this module contains the following materials:

MATERIAL
Barium Sodium Niobate
Barium Titanate
Barium Titanate (poled)
Lithium Niobate
Lithium Tantalate
Lead Zirconate Titanate (PZT-2)
Lead Zirconate Titanate (PZT-4)
Lead Zirconate Titanate (PZT-4D)
Lead Zirconate Titanate (PZT-5A)
Lead Zirconate Titanate (PZT-5H)
Lead Zirconate Titanate (PZT-5J)
Lead Zirconate Titanate (PZT-7A)

MATERIAL
Lead Zirconate Titanate (PZT-8)
Quartz
Rochelle Salt
Bismuth Germanate
Cadmium Sulfide
Gallium Arsenide
Tellurium Dioxide
Zinc Oxide
Zinc Sulfide
Ammonium Dihydrogen Phosphate
Aluminum Nitride

All materials define the following material properties needed for piezoelectric modeling:

MATERIAL PROPERTY	DESCRIPTION
c_E	Elasticity matrix
e	Coupling matrix, stress-charge
ϵ_{rS}	Relative permittivity, stress-charge
s_E	Compliance matrix
d	Coupling matrix, strain-charge
ϵ_{rT}	Relative permittivity, strain-charge
ρ	Density

Piezoresistivity Materials Database

The Piezoresistivity materials database is included with the MEMS Module and contains the following materials:

MATERIAL
p-Silicon (single-crystal, lightly doped)
n-Silicon (single-crystal, lightly doped)
p-Silicon (polycrystalline, lightly doped)
n-Silicon (polycrystalline, lightly doped)

All materials define the following material properties needed for modeling the piezoresistance effect:

MATERIAL PROPERTY	DESCRIPTION
D	Elasticity matrix
D_{Vo}	Elasticity matrix, Voigt notation
ρ	Density
ϵ_r	Relative permittivity
σ	Electrical conductivity
Π	Piezoresistive coupling matrix
m_l	Elastoresistive coupling matrix

Both the electrical conductivity and the piezoresistive or elastoresistive coupling matrix are strong functions of the material dopant density. The material models include appropriate functions, although the piezoresistive and elastoresistive matrices scale only with the conductivity, which is appropriate only at lower dopant densities (below approximately 10^{16} cm^{-3}). The low doping level piezoresistance and elastoresistance values are based on those given in [Ref. 1](#). The conductivity is computed from an empirical functional fit to experimental data given in equation 8 of [Ref. 2](#). Data on the piezoresistance properties of Silicon at higher doping levels is available in [Ref. 3](#) and [Ref. 4](#). Because this data does not include all components of the coupling matrix, it is not included in the material models.

The dopant density must be entered for the material as a model input in the piezoresistive or conductive material node. It can be entered as a constant value or as an expression (for example, a spatially varying function could be used).

Batteries and Fuel Cells Materials Database

The Batteries and Fuel Cells Materials database is included with the Batteries & Fuel Cells Module and contains the materials listed in [Table 8-16](#). The material property groups (including all associated properties) are listed in [Table 8-3](#).

TABLE 8-16: BATTERIES & FUEL CELLS MODULE MATERIALS DATABASE

MATERIAL	INTENDED USE
Sulfuric Acid (Lead-Acid Battery)	Electrolyte
Pb Electrode (Negative, Lead-Acid Battery)	Equilibrium potential
PbO ₂ Electrode (Positive, Lead-Acid Battery)	Equilibrium potential
LixC ₆ Electrode (Negative, Li-ion Battery)	Equilibrium potential

TABLE 8-16: BATTERIES & FUEL CELLS MODULE MATERIALS DATABASE

MATERIAL	INTENDED USE
LixMn2O4 Electrode (Positive, Li-ion Battery)	Equilibrium potential
LixCoO2 Electrode (Positive, Li-ion Battery)	Equilibrium potential
HxLiN5 Electrode (Negative, NiMH Battery)	Equilibrium potential
NiOHO-Hx Electrode (Positive discharge, NiMH Battery)	Equilibrium potential
NiOHO-Hx Electrode (Positive charge, NiMH Battery)	Equilibrium potential
1:2 EC:DMC/LiPF6 (Li-ion Battery)	Electrolyte
2:1 EC:DMC/LiPF6 (Li-ion Battery)	Electrolyte
1:1 EC:DEC/LiPF6 (Li-ion Battery)	Electrolyte

Semiconductor Materials Database

The Semiconductor Materials database is included with the Semiconductor Module and contains the materials listed in [Table 8-17](#). The material property groups (including all associated properties) are listed in [Table 8-12](#).

The Silicon material contains parameters for all the material property groups, while the other materials contain only the material parameters in the Semiconductor and Basic material groups.

[Table 8-12](#) also gives the references used for the silicon material properties. The material properties for materials other than Silicon are obtained from Appendix 6 of Ref. 1.

TABLE 8-17: SEMICONDUCTOR MODULE MATERIALS DATABASE

MATERIAL
Silicon
Gallium Arsenide
Germanium

References for the Material Databases

REFERENCES FOR THE LIQUIDS AND GASES MATERIAL DATABASE

1. *ASHRAE Handbook of Fundamentals*, American Society of Heating, Refrigerating and Air Conditioning Engineers, 1993.
2. E. R. G. Eckert and M. Drake, Jr., *Analysis of Heat and Mass Transfer*, Hemisphere Publishing, 1987.

3. H. Kashiwagi, T. Hashimoto, Y. Tanaka, H. Kubota, and T. Makita, "Thermal Conductivity and Density of Toluene in the Temperature Range 273–373K at Pressures up to 250 MPa," *Int. J. Thermophys.*, vol. 3, no. 3, pp. 201–215, 1982.
4. C. A. Nieto de Castro, S.F.Y. Li, A. Nagashima, R.D. Trengove, and W.A. Wakeham, "Standard Reference Data for the Thermal Conductivity of Liquids," *J. Phys. Chem. Ref. Data*, vol. 15, no. 3, pp. 1073–1086, 1986.
5. B.E. Poling, J.M. Prausnitz, and J.P. O'Connell, *The Properties of Gases and Liquids*, 5th ed., McGraw-Hill, 2001.
6. C.F. Spencer and B.A. Adler, "A Critical Review of Equations for Predicting Saturated Liquid Density," *J. Chem. Eng. Data*, vol. 23, no. 1, pp. 82–88, 1978.
7. N.B. Vargnaftik, *Tables of Thermophysical Properties of Liquids and Gases*, 2nd ed., Hemisphere Publishing, 1975.
8. R.C. Weast (editor), *CRC Handbook of Chemistry and Physics*, 69th ed., CRC Press, 1988.
9. M. Zabransky and V. Ruzicka, Jr., "Heat Capacity of Liquid n-Heptane Converted to the International Temperature Scale of 1990," *Phys. Chem. Ref. Data*, vol. 23, no. 1, pp. 55–61, 1994.
10. M. Zabransky, V. Ruzicka, Jr., and E.S. Domalski, "Heat Capacity of Liquids: Critical Review and Recommended Values. Supplement I," *J. Phys. Chem. Ref. Data*, vol. 30, no. 5, pp. 1199–1397, 2002.

REFERENCES FOR THE MEMS MATERIALS DATABASE

1. J.W. Gardner, V.K. Varadan, and O.O. Awadelkarim, *Microsensors, MEMS, and Smart Devices*, John Wiley & Sons, 2001.
2. D.R. Lide (editor), *CRC Handbook of Chemistry and Physics*, 84th edition, CRC Press, 2003.
3. A.M. James and M.P. Lord, *MacMillan's Chemical and Physical Data*, MacMillan's Press, 1992.
4. M. Gad-el-Hak (editor), *The MEMS Handbook*, CRC Press, 2002.
5. *New Semiconductor Materials. Characteristics and Properties*, [www.ioffe.ru/
SVA/NSM](http://www.ioffe.ru/SVA/NSM), 2003.
6. *Ceramics WebBook*, www.ceramics.nist.gov/srd/scd/scdquery.htm, 2003.

REFERENCES FOR THE PIEZORESISTIVITY MATERIALS DATABASE

1. C. S. Smith, “Piezoresistance Effect in Germanium and Silicon,” *Physical Review*, vol. 94, no. 1, pp. 42–49, 1957.
2. C. Jacoboni, C. Canali, G. Ottaviani, and A. Alberigi Quaranta, “A Review of Some Charge Transport Properties of Silicon,” *Solid-State Electronics*, vol. 20, pp. 77–89, 1977.
3. O. N. Tufte and E. L. Stelzer, “Piezoresistance Properties of Heavily Doped n-Type Silicon,” *Physical Review*, vol. 133, no. 6A, pp. A1705–A1716, 1964.
4. O. N. Tufte and E. L. Stelzer, “Piezoresistive Properties of Silicon Diffused Layers,” *Journal of Applied Physics*, vol. 34, no. 2, pp. 313–318, 1963.

REFERENCES FOR THE SEMICONDUCTOR MATERIALS DATABASE

1. M. Shur, *Physics of Semiconductor Devices*, Prentice Hall, 1990.
2. S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Springer Verlag, 1984.
3. Approximate values are given, based loosely on Ref. 2. In practice these values should depend on the local doping, as discussed in Ref. 2. Use the user defined option to implement this functionality.
4. N. D. Arora, J. R. Hauser, and D. J. Roulston, “Electron and Hole Mobilities in Silicon as a Function of Concentration and Temperature,” *IEEE Transactions on Electron Devices*, vol. 29, no. 2, pp. 292–295, 1982.
5. J. M. Dorkel and Ph. Leturcq, “Carrier Mobilities in Silicon Semi-empirically Related to Temperature, Doping and Injection Level,” *Solid-State Electronics*, vol. 24, no. 9, pp. 821–825, 1981.

Overview of the Physics

This chapter gives an overview of the available physics user interfaces as well as some general guidelines for effective modeling.

The Physics User Interfaces

This section provides an overview of the physics user interfaces included in COMSOL Multiphysics®.

-
- Advanced Physics Sections
 - The Physics 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 Nodes and Toolbar Buttons
-

Introduction to the Physics User 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 user interfaces* that consist of nodes and settings that set up the equations and variables for specific areas of physics. An extensive set of physics-dependent variables makes it easy to visualize and evaluate the important physical quantities using conventional terminology and notation.



Suites of physics user interfaces that are optimized for specific disciplines together with specialized model libraries are available in a group of optional products. See [The COMSOL Modules and Interfacing Options](#) for a list.

A complement to the user interfaces for physics, special user interfaces for equation-based modeling simplify the setup of PDEs for modeling that does not explicitly refer to any particular application field. In addition, other user interfaces supplement the physics with special functionality such as the Sensitivity and Moving Mesh user interfaces.

Physics Groups in the Model Wizard

The **Add Physics** page in the **Model Wizard** contains the main groups of physics and mathematics user interfaces as in [Table 9-1](#) (some items only display if a license includes the add-on modules).

TABLE 9-1: PHYSICS GROUPS IN THE MODEL WIZARD

ICON	BRANCH NAME	DESCRIPTION OF PHYSICS IN THE BRANCH
	Recently Used	Contains the most recently used user interfaces for easy access.
	AC/DC	Low-frequency electromagnetics such as electrostatics and electric currents.
	Acoustics	Acoustics.
	Chemical Species Transport	Chemical species transport by, for example, convection and diffusion, solving for the species concentrations.
	Electrochemistry	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	Fluid flow such as laminar single-phase flow and, with add-on modules, multiphase flow and turbulent flow.
	Heat Transfer	Heat transfer in solids and fluids and thermal multiphysics applications such as Joule heating.
	Plasma	Plasma modeling. This branch is only available if a license includes the Plasma Module.
	Radio Frequency	High-frequency electromagnetic field simulations solving the full Maxwell equations. This branch is only available if a license includes the RF Module.
	Structural Mechanics	Structural mechanics, studying displacements and stresses in solids, for example.
	Mathematics	Mathematics user 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 Guide

The table lists the physics in COMSOL Multiphysics and their availability for 1D, 1D axisymmetric, 2D, 2D axisymmetric, and 3D geometries.

PHYSICS USER INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
AC/DC				
Electric Currents		ec	all dimensions	stationary
Electrostatics		es	all dimensions	stationary; time dependent
Magnetic Fields		mf	2D, 2D axisymmetric	stationary; frequency domain
Acoustics				
Pressure Acoustics				
Pressure Acoustics, Frequency Domain		acpr	all dimensions	eigenfrequency; frequency domain
Chemical Species Transport				
Transport of Diluted Species		chds	all dimensions	stationary; time dependent
Fluid 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

PHYSICS USER INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
 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
Curvilinear Coordinates		cc	all dimensions	stationary; eigenvalue
 PDE Interfaces				
Coefficient Form PDE		c	all dimensions	stationary; eigenvalue; time dependent
General Form PDE		g	all dimensions	stationary; eigenvalue; time dependent
Wave Form PDE		wahw	all dimensions	time dependent
Weak Form PDE		w	all dimensions	stationary; eigenvalue; time dependent
 Lower Dimensions				
Coefficient Form Boundary PDE		cb	all dimensions	stationary; eigenvalue; time dependent
Coefficient Form Edge PDE		ce	3D	stationary; eigenvalue; time dependent
Coefficient Form Point PDE		cp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent

PHYSICS USER INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
General Form Boundary PDE		gb	all dimensions	stationary; eigenvalue; time dependent
General Form Edge PDE		ge	3D	stationary; eigenvalue; time dependent
General Form Point PDE		gp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent
Weak Form Boundary PDE		wb	all dimensions	stationary; eigenvalue; time dependent
Weak Form Edge PDE		we	3D	stationary; eigenvalue; time dependent
Weak Form Point PDE		wp	3D, 2D, 2D axisymmetric	stationary; eigenvalue; time dependent

ODE and DAE Interfaces

Global ODEs and DAEs		ge	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
Domain ODEs and DAEs		dode	all dimensions	stationary; time dependent; eigenvalue
Events		ev	all dimensions	time dependent
Boundary ODEs and DAEs		bode	all dimensions	stationary; time dependent; eigenvalue
Edge ODEs and DAEs		eode	3D	stationary; time dependent; eigenvalue
Point ODEs and DAEs		pode	3D, 2D, 2D axisymmetric	stationary; time dependent; eigenvalue

Optimization and Sensitivity

Optimization Requires the Optimization Module		opt	all dimensions	stationary; eigenfrequency; time dependent; frequency domain; eigenvalue
--	--	-----	----------------	--

PHYSICS USER INTERFACE	ICON	TAG	SPACE DIMENSION	AVAILABLE PRESET STUDY TYPE
Sensitivity		sens	all dimensions	stationary; eigenfrequency; frequency domain; eigenvalue; time dependent (available with the Optimization Module)

Classical PDEs

Laplace Equation		lpeq	all dimensions	stationary
Poisson's Equation		poeq	all dimensions	stationary
Wave Equation		waeq	all dimensions	time dependent
Helmholtz Equation		hzeq	all dimensions	stationary
Heat Equation		hteq	all dimensions	stationary; time dependent
Convection-Diffusion Equation		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 Physics

When creating a model in COMSOL Multiphysics, you can select a single physics user interface that describes one type of physics or select several physics user interfaces for multiphysics modeling and coupled-field analyses.

MODELING USING A SINGLE PHYSICS

Most physics user interfaces contain stationary, eigenvalue, and time-dependent (dynamic) study types. As already mentioned, these physics provide features and windows where you can create models using material properties, boundary conditions, sources, initial conditions, and so on. Each physics user interfaces comes with a template that automatically supplies the appropriate underlying PDEs.

If you cannot find a physics user interface that matches a given problem, try one of the user interfaces for PDEs, which makes it possible to define a custom model in general mathematical terms. Indeed, COMSOL can model virtually any scientific phenomena or engineering problems that originate from the laws of science.

MULTIPHYSICS MODELING USING MULTIPLE PHYSICS USER INTERFACES

When modeling real-world systems, you often need to include the interaction between different kinds of physics: *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 into a single model using the multiphysics capabilities of COMSOL. For the example just mentioned, you can use the predefined Joule Heating user interface, which is a combination of the Electric Currents and Heat Transfer user 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 user interface that combines two or more coupled physics for common multiphysics applications.

You can also combine physics user 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 user interfaces suitable for the phenomena of interest. If you find them among the available physics user interfaces, use them; if not, add one or more user interface for equation-based modeling.

When coupling multiple physics in a multiphysics model (without using a predefined multiphysics user interface), the couplings can occur in domains and on boundaries. COMSOL automatically identifies potential *model inputs* for quickly forming couplings between physics. For example, a velocity field from fluid flow is a model input for the convective heat transport in heat transfer. In that case, the model input automatically transfers the velocity field from the fluid to the heat transfer part.

Show More Physics Options

There are several general options available for the physics user interfaces and for individual nodes. This section is a short overview of these options, and includes links to additional information when available.



The links to the features described in the *COMSOL Multiphysics Reference Manual* (or any external guide) do not work in the PDF, only from the on line help.



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 options for the physics interfaces and other parts of the model tree, click the **Show** button (▶) on the **Model Builder** and then select the applicable option.

After clicking the **Show** button (▶), additional sections get displayed on the settings window when a node is clicked and additional nodes 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 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 nodes in the **Model Builder**.

Availability of each node, and whether it is described for a particular node, is based on the individual 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
Show More Options and Expand Sections	Advanced Physics Sections The Model Wizard and Model Builder
Discretization	Show Discretization Discretization (Node)
Discretization—Splitting of complex variables	Compile Equations
Consistent and Inconsistent Stabilization	Show Stabilization Numerical Stabilization
Constraint Settings	Weak Constraints and Constraint Settings
Override and Contribution	Physics Exclusive and Contributing Node Types

OTHER COMMON SETTINGS

At the main level, some of the common settings found (in addition to the Show  options) are the Interface Identifier, Domain, Boundary, or Edge Selection, and Dependent Variables.

At the nodes' level, some of the common settings found (in addition to the Show  options) are Domain, Boundary, Edge, or Point Selection, Material Type, Coordinate System Selection, and Model Inputs. Other sections are common based on application area and are not included here.

SECTION	CROSS REFERENCE
Coordinate System Selection	Coordinate Systems
Domain, Boundary, Edge, and Point Selection	About Geometric Entities About Selecting Geometric Entities
Interface Identifier	Predefined Physics Variables Variable Naming Convention and Scope Viewing Node Names, Identifiers, Types, and Tags
Material Type	Materials

SECTION	CROSS REFERENCE
Model Inputs	About Materials and Material Properties Selecting Physics Adding Multiphysics Couplings
Pair Selection	Identity and Contact Pairs Continuity on Interior Boundaries

Equations for the Physics

Physics Nodes—Equation Section

For each physics node there is an **Equation** section always available on the settings window. This has options to display mathematical equations applicable to the node.



Equation View



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 lists depend on the study types and other physics-specific factors. See [Figure 9-2](#) for an example comparing the equations that display for a **Stationary** or **Time Dependent** study for a **Heat Transfer in Solids** user 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.



Study Step Types

Node Contributions Display a Dotted Line Under Part of the Equation

For all physics nodes (excluding the main physics user interface node level), the equation that displays includes a dotted line underneath where the node's contribution is made to the equation. See [Figure 9-1](#) for an example where a section of the heat transfer equation is underlined, indicating where the **Heat Transfer in Solids** node contributes to it.

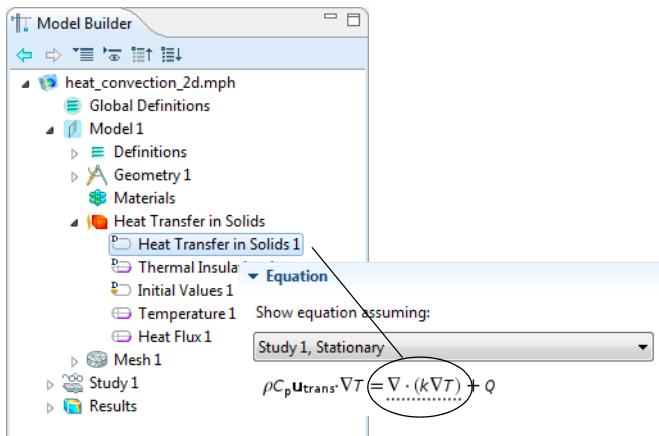


Figure 9-1: The Heat Transfer in Solids contribution to the equation for a 3D model.

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 physics.

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).

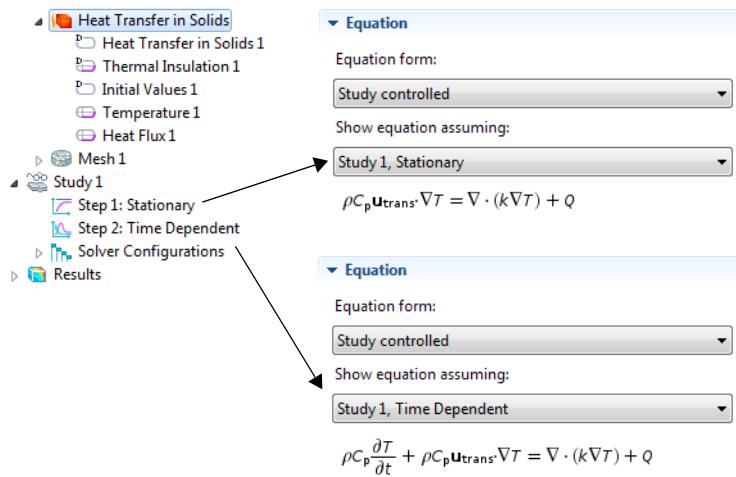


Figure 9-2: An example of the Equation section on a Heat Transfer interface. Selecting the study type updates the equation accordingly.

Physics Nodes Advanced Settings

Various settings are available by clicking the **Show** button () and selecting from the list of sections to display. See [Advanced Physics Sections](#) for more information.

Creating Multiphysics Models

This section describes different ways of creating multiphysics models and to connect different physics in a model.

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’s 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.



See [Marangoni Convection](#) for an example of combining the Laminar Flow and Heat Transfer in Fluids interfaces (Model Library path [COMSOL_Multiphysics/Multiphysics/marangoni_convective](#)).

Using Predefined Multiphysics

COMSOL Multiphysics and the add-on modules provide a number of predefined multiphysics user interfaces. They are included in the **Model Wizard**’s list of physics. The multiphysics user interfaces include the functionality of two or more physics with additional nodes and functionality 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.

The predefined multiphysics user interfaces include all functionality in the add-on modules if your license includes them; otherwise they include the corresponding but more limited physics user 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 user 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 in a model that you start using one of the predefined multiphysics user interfaces. If you want to add additional

physics in the **Model Builder**, right-click the model node, choose **Add Physics** and then use the **Model Wizard** that opens.



For links to more information about the add-on modules and the multiphysics interfaces available go to www.comsol.com/products/multiphysics.

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 user interface:

- 5 Right-click the model node in the **Model Builder** and choose **Add Physics**.
- 6 Add one or more new physics user interface 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 nodes in the physics user interfaces, including couplings between the physics.

Building a Multiphysics Model Directly

Another approach is to include multiple physics when you start creating the model:

- 1 On the **Add Physics** page, select a physics user interface for the model.
- 2 Click the **Add Selected** button () underneath the list of available physics. The selected physics then appears in the list under **Selected Physics**.

- 3 If needed, you can change the names of the dependent variables. For some physics user interfaces you can also change the number of dependent variables.
- 4 Continue selecting physics 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 to add a study and start modeling.

Adding Multiphysics Couplings

Model inputs can 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. COMSOL Multiphysics connects the model input to an existing field (dependent variable) within the physics user interface (but not to available fields in other physics).

With more than one physics in the model, coupling of the fields is easy: all applicable fields that can serve as inputs in another physics automatically appears in the other physics user interface's settings window's **Model Inputs** section. For example, with a Heat Transfer in Fluids (ht) user interface and a Laminar Flow (spf) user interface, you can select **Velocity field (spf/fpl)**, which the **Fluid Properties I {fpl}** 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/fluidI)**, which the **Heat Transfer in Fluids I {fluidI}** 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 itself defines the field because it is then automatically connected to that field. For example, with a Heat Transfer in Fluids (ht) interface the **Temperature** list is grayed out in the **Heat Transfer in Fluids I {fluidI}** node. This automatic connection selects the **Temperature (ht/fluidI)** field because it is defined by the Heat Transfer in Fluids. 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 User Interfaces

To delete a physics user interface from a model:

- 1 Select the main node for the physics 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**.

The AC/DC User Interfaces

This chapter explains the physics user 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.

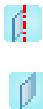
The Electromagnetics User Interfaces

For simulating electromagnetic fields, COMSOL Multiphysics® has three physics interfaces.

With the first two, you can perform static simulations to solve for electric properties:

- Electrostatics
- Electric Currents
- Magnetic Fields

These interfaces are available in all space dimensions.



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.



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

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:

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \cdot \mathbf{D} = \rho$$

$$\nabla \cdot \mathbf{B} = 0$$

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

$$\begin{aligned}\mathbf{D} &= \epsilon_0 \mathbf{E} + \mathbf{P} \\ \mathbf{B} &= \mu_0(\mathbf{H} + \mathbf{M}) \\ \mathbf{J} &= \sigma \mathbf{E}\end{aligned}\tag{10-1}$$

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 a vacuum is given as c_0 and the permittivity of a vacuum is derived from the relation:

$$\epsilon_0 = \frac{1}{c_0^2 \mu_0} = 8.854 \cdot 10^{-12} \text{ F/m} \approx \frac{1}{36\pi} \cdot 10^{-9} \text{ F/m}$$

The electromagnetic constants ϵ_0 , μ_0 , and c_0 are available in COMSOL Multiphysics as predefined physical constants.

The *electric polarization vector* \mathbf{P} describes how the material is polarized when an electric field \mathbf{E} is present. It can be interpreted as the volume density of *electric dipole* moments. \mathbf{P} is generally a function of \mathbf{E} . Some materials can have a nonzero \mathbf{P} also when there is no electric field present.

The *magnetization vector* \mathbf{M} similarly describes how the material is magnetized when a magnetic field \mathbf{H} is present. It can be interpreted as the volume density of *magnetic dipole* moments. \mathbf{M} is generally a function of \mathbf{H} . Permanent magnets, for instance, have a nonzero \mathbf{M} also when there is no magnetic field present.

For linear materials, the polarization is directly proportional to the electric field, $\mathbf{P} = \epsilon_0 \chi_e \mathbf{E}$, 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:

$$\begin{aligned}\mathbf{D} &= \epsilon_0(1 + \chi_e) \mathbf{E} = \epsilon_0 \epsilon_r \mathbf{E} = \epsilon \mathbf{E} \\ \mathbf{B} &= \mu_0(1 + \chi_m) \mathbf{H} = \mu_0 \mu_r \mathbf{H} = \mu \mathbf{H}\end{aligned}$$

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, respectively.

GENERALIZED CONSTITUTIVE RELATIONS



The [Charge Conservation](#) node describes the macroscopic properties of the medium (relating the electric displacement \mathbf{D} with the electric field \mathbf{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 $\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E} + \mathbf{D}_r$ 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 \mathbf{B} and \mathbf{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:

$$\begin{aligned}\mathbf{B} &= \nabla \times \mathbf{A} \\ \mathbf{E} &= -\nabla V - \frac{\partial \mathbf{A}}{\partial t}\end{aligned}$$

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

$$\mathbf{n}_2 \times (\mathbf{E}_1 - \mathbf{E}_2) = \mathbf{0}$$

$$\mathbf{n}_2 \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

$$\mathbf{n}_2 \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s$$

$$\mathbf{n}_2 \cdot (\mathbf{B}_1 - \mathbf{B}_2) = 0$$

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 two. 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:

$$-\mathbf{n}_2 \times \mathbf{E}_2 = \mathbf{0}$$

$$-\mathbf{n}_2 \times \mathbf{H}_2 = \mathbf{J}_s$$

$$-\mathbf{n}_2 \cdot \mathbf{D}_2 = \rho_s$$

$$-\mathbf{n}_2 \cdot \mathbf{B}_2 = 0$$

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 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](#).



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 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*

$$\nabla \cdot (\epsilon \mathbf{E}) = \rho$$

By combining these, one can deduce the following differential equation for the space charge density in a homogeneous medium

$$\frac{\partial \rho}{\partial t} + \frac{\sigma}{\epsilon} \rho = 0$$

This equation has the solution

$$\rho(t) = \rho_0 e^{-t/\tau}$$

where

$$\tau = \frac{\epsilon}{\sigma}$$

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 10-1](#) below,

TABLE 10-1: SUITABLE PHYSICS INTERFACE AND STUDY TYPE FOR DIFFERENT TIME-SCALE REGIMES.

CASE	INTERFACE	STUDY TYPE
$\tau \gg t$	Electrostatics	Stationary
$\tau \ll t$	Electric Currents	Stationary
$\tau \sim t$	Electric Currents	Time Dependent or Frequency Domain (in AC/DC Module or MEMS Module)

FIRST CASE: $\tau > > t$

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 $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$ between \mathbf{D} and \mathbf{E} , you can rewrite Gauss' law as a variant of Poisson's equation

$$-\nabla \cdot (\epsilon_0 \nabla V - \mathbf{P}) = \rho$$

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.



Such separate charge transport formulations can be found in the Plasma Module and the Chemical Reaction Engineering Module.

SECOND CASE: $\tau \ll t$

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

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}^e$$

where \mathbf{J}^e is an externally generated current density. The static form of the equation of continuity then reads

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma \nabla V - \mathbf{J}^e) = 0$$

To handle current sources the equation can be generalized to

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

This equation is used in the static study type for the Electric Currents interface.

Theory of Electrostatics

The [Electrostatics User 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](#). Also discussed in this section is [The Electrostatics Interface in Time-Dependent or Frequency-Domain Studies](#).

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 $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$ between the electric displacement \mathbf{D} and the electric field \mathbf{E} , it is possible to represent Gauss' law as the following equation:

$$-\nabla \cdot (\epsilon_0 \nabla V - \mathbf{P}) = \rho$$

In this equation, the physical constant, ϵ_0 (SI unit: F/m) is the permittivity of vacuum, \mathbf{P} (SI unit: C/m²) is the electric polarization vector, and ρ (SI unit: C/m³) 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:

$$-\nabla \cdot d(\epsilon_0 \nabla V - \mathbf{P}) = \rho$$

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.

The Electrostatics Interface in Time-Dependent or Frequency-Domain Studies

The [Electrostatics User Interface](#) can be solved also in a dynamic study (Time-Dependent or Frequency-Domain). The equation system solved, however, is always the one presented in the previous section for the stationary case, in which no transient electromagnetic effects are taken into account. The difference is that the sources of the problem (charge densities, electric potential) are assumed to be time-varying (in a Time-Dependent study) or time-harmonic (in a Frequency-Domain study). The support for dynamic studies simplifies the coupling of the Electrostatics interface with other physics. Using the interface in a dynamic study is a valid approximation only if the time-scale (or the frequency) of the study is so slow that transient electromagnetic effects can be neglected; for example, in acoustic or structural problems.

The Electrostatics interface also supports the small-signal analysis study sequence, that can be used when on a static bias charge or voltage is superposed a time-harmonic perturbation.

Theory of Electric Currents

The [Electric Currents User 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](#)

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:

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e$$

where σ is the electrical conductivity (SI unit: S/m), and \mathbf{J}_e is an externally generated current density (SI unit: A/m²). The static form of the equation of continuity then states:

$$\nabla \cdot \mathbf{J} = -\nabla \cdot (\sigma \nabla V - \mathbf{J}_e) = 0$$

To handle current sources, you can generalize the equation to:

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

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:

$$-\nabla \cdot d(\sigma \nabla V - \mathbf{J}_e) = dQ_j \quad (10-2)$$

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 $\partial\mathbf{D}/\partial t = 0$.

Maxwell's Equations

This implies that it is possible to rewrite Maxwell's equations in the following manner:

$$\nabla \times \mathbf{H} = \mathbf{J} = \sigma(\mathbf{E} + \mathbf{v} \times \mathbf{B}) + \mathbf{J}_e$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \cdot \mathbf{D} = \rho$$

$$\nabla \cdot \mathbf{J} = 0$$

Here \mathbf{J}_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,

$$\mathbf{B} = \nabla \times \mathbf{A}$$

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t}$$

and the constitutive relation $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$, Ampère's law can be rewritten as

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times (\mu_0^{-1} \nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) + \sigma \nabla V = \mathbf{J}_e \quad (10-3)$$

The equation of continuity, which is obtained by taking the divergence of the above equation, adds the following equation:

$$\nabla \cdot \left(-\sigma \frac{\partial \mathbf{A}}{\partial t} + \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) - \sigma \nabla V + \mathbf{J}_e \right) = 0 \quad (10-4)$$

Equation 10-3 and Equation 10-4 form a system of equations for the two potentials \mathbf{A} and V .

Gauge Transformations

The electric and magnetic potentials are not uniquely defined from the electric and magnetic fields through

$$\begin{aligned}\mathbf{E} &= -\frac{\partial \mathbf{A}}{\partial t} - \nabla V \\ \mathbf{B} &= \nabla \times \mathbf{A}\end{aligned}$$

Introducing two new potentials

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathbf{A} + \nabla \Psi \\ \tilde{V} &= V - \frac{\partial \Psi}{\partial t}\end{aligned}$$

gives the same electric and magnetic fields:

$$\begin{aligned}\mathbf{E} &= -\frac{\partial \mathbf{A}}{\partial t} - \nabla V = -\frac{\partial (\tilde{\mathbf{A}} - \nabla \Psi)}{\partial t} - \nabla \left(\tilde{V} + \frac{\partial \Psi}{\partial t} \right) = -\frac{\partial \tilde{\mathbf{A}}}{\partial t} - \nabla \tilde{V} \\ \mathbf{B} &= \nabla \times \mathbf{A} = \nabla \times (\tilde{\mathbf{A}} - \nabla \Psi) = \nabla \times \tilde{\mathbf{A}}\end{aligned}$$

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 \mathbf{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 \mathbf{A} . In the static limit, \mathbf{A} and V are not coupled via the gauge selection and thus any gauge can be chosen for \mathbf{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:

$$\nabla \cdot \left(-\sigma \frac{\partial \mathbf{A}}{\partial t} + \mathbf{J}_e \right) = 0$$

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):

$$\nabla \times \mathbf{H} = \mathbf{J} = \sigma(\mathbf{E} + \mathbf{v} \times \mathbf{B}) + j\omega \mathbf{D} + \mathbf{J}^e$$

In the transient case the inclusion of this term leads 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:

$$\begin{aligned} -\nabla \cdot ((j\omega\sigma - \omega^2 \epsilon_0) \mathbf{A} - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) + (\sigma + j\omega\epsilon_0) \nabla V - (\mathbf{J}^e + j\omega \mathbf{P})) &= 0 \\ (j\omega\sigma - \omega^2 \epsilon_0) \mathbf{A} + \nabla \times (\mu_0^{-1} \nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) + (\sigma + j\omega\epsilon_0) \nabla V &= \mathbf{J}^e + j\omega \mathbf{P} \end{aligned}$$

The constitutive relation $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$ 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:

$$\tilde{\mathbf{A}} = \mathbf{A} - \frac{j}{\omega} \nabla V \quad \tilde{V} = 0$$

When \tilde{V} vanishes from the equations, only the second one is needed,

$$(j\omega\sigma - \omega^2 \epsilon_0) \tilde{\mathbf{A}} + \nabla \times (\mu_0^{-1} \nabla \times \tilde{\mathbf{A}} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \tilde{\mathbf{A}}) = \mathbf{J}^e + j\omega \mathbf{P}$$

Working with $\tilde{\mathbf{A}}$ is often the best option when it is possible to specify all source currents as external currents \mathbf{J}^e or as surface currents on boundaries.

Theory of Magnetic Fields

Simulation of magnetic fields is of interest when studying magnets, motors, transformers, and conductors carrying static or alternating currents.

The [Magnetic Fields User 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.



For a deeper theoretical background to the magnetic vector potential used, 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

$$\mathbf{J} = \sigma \mathbf{v} \times \mathbf{B} + \mathbf{J}^e$$

where \mathbf{J}^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

$$\nabla \times (\mu_0^{-1} \nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}^e$$

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 $\mathbf{D} = \epsilon_0 \mathbf{E}$ to rewrite Ampère's law as

$$(j\omega\sigma - \omega^2\epsilon_0)\mathbf{A} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}^e$$

Transient Equation

The transient equation this physics interface solves is Ampère's law, here illustrated with the constitutive relation $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$.

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times (\mu_0^{-1}\nabla \times \mathbf{A} - \mathbf{M}) - \sigma \mathbf{v} \times (\nabla \times \mathbf{A}) = \mathbf{J}_e$$

The Electrostatics User Interface

The **Electrostatics (es)** user 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 node, 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** (the default boundary condition), and **Initial Values**. Right-click the **Electrostatics** node to add other nodes that implement, for example, boundary conditions and space charges.

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 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

 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](#).



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 shares 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.

-
- 
- [Show More Physics Options](#)
 - Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface
 - Theory of Electrostatics
-



Electric Sensor: Model Library path **COMSOL_Multiphysics/**
Electromagnetics/electric_sensor

Domain, Boundary, Edge, Point, and Pair Nodes for the Electrostatics Interface

The [Electrostatics User Interface](#) has these domain, boundary, edge, point, and pair nodes 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

$$\mathbf{n} \cdot [(\epsilon_0 \nabla V - \mathbf{P})_1 - (\epsilon_0 \nabla V - \mathbf{P})_2] = -\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = 0$$

AVAILABLE

These nodes are available for this interface and listed in alphabetical order. Also see [Table 10-2](#) 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
- 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 10-2](#) lists the interior and exterior boundaries available with this interface. It also includes edge, point, and pair availability.

TABLE 10-2: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTROSTATICS INTERFACE

NODE	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
Ground	x	x	edges, points, and pairs

TABLE 10-2: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTROSTATICS INTERFACE

NODE	INTERIOR	EXTERIOR	ALSO AVAILABLE FOR
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

	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. There are also Line Charge (on Axis) and Point Charge (on Axis) available.
	<ul style="list-style-type: none"> Continuity on Interior Boundaries Identity and Contact Pairs
	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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define the electric potential and the equation based on Gauss' law that describes the potential field or select **All domains** as required.

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** (the default), 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 **D** with the electric field **E**) and the applicable material properties, such as the relative permittivity.

Select:

- **Relative permittivity** (the default) to use the constitutive relation $\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E}$. Then the default is to take the **Relative permittivity** ϵ_r (dimensionless) values **From material**. If **User defined** is selected, select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix. The default is 1.
- **Polarization** to use the constitutive relation $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$. Then enter the components based on space dimension for the **Polarization** vector **P** (SI unit: C/m²). The defaults are 0 C/m².
- **Remanent electric displacement** to use constitutive relation $\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E} + \mathbf{D}_r$, where \mathbf{D}_r is the remanent displacement (the displacement when no electric field is present). Then the default is to take the **Relative permittivity** ϵ_r (dimensionless) 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** **D_r** (SI unit: C/m²). The defaults are 0 C/m².

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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.

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.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

SPACE CHARGE DENSITY

Enter a value or expression for the **Space charge density** ρ_q (SI unit: C/m³). The default is 0 C/m³.

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Ground

The **Ground** node is the default boundary condition and implements ground (zero potential) 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** nodes 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 define.



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 this node 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**.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent**

variables to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

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** nodes 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 define.



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 this node 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). The default is 0 V.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Surface Charge Density

The **Surface Charge Density** node provides the following surface-charge boundary condition for exterior boundaries (left) and interior boundaries (right):

$$-\mathbf{n} \cdot \mathbf{D} = \rho_s, \quad \mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s$$

Specify the surface charge density ρ_s at an outer boundary or at an interior boundary between two nonconducting media. Also right-click to add **Harmonic Perturbation**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid**, or **From material**.

PAIR SELECTION

If this node 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/m²).



Harmonic Perturbation—Exclusive and Contributing Nodes

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 define.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

PAIR SELECTION

If this node 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** $\mathbf{n} \cdot \mathbf{J}_i$ (SI unit: A/m²) and the **Normal electron current density** $\mathbf{n} \cdot \mathbf{J}_e$ (SI unit: A/m²). The defaults are 0 A/m² for both.

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 define.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

PAIR SELECTION

If this node 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: C/m²). The defaults are 0 C/m².

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.



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 define.

PERIODIC CONDITION

Select a **Type of periodicity**—**Continuity** (the default) or **Antiperiodicity**.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.



- [Periodic Condition and Destination Selection](#)
- [Periodic Boundary Conditions](#)

Thin Low Permittivity Gap

Use the **Thin Low Permittivity Gap** node

$$\mathbf{n} \cdot \mathbf{D}_1 = \frac{\epsilon_0 \epsilon_r}{d_s} (V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{D}_2 = \frac{\epsilon_0 \epsilon_r}{d_s} (V_2 - V_1)$$

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 define.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

PAIR SELECTION

If this node 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 (dimensionless) values **From material**. Select **User defined** to enter a different value or expression. Enter a **Surface thickness** d_s (SI unit: m). The default is 5 mm.

Line Charge



For 3D models, use the **Line Charge** node to specify line charges along the edges of a geometry. Also right-click to add **Harmonic Perturbation**.

EDGE SELECTION

From the **Selection** list, choose the edges to define.



Beware that constraining the potential on edges usually yields a current outflow that is mesh dependent.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

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 and the default is 0 C/m.

-
- 
 - [Line Charge \(on Axis\)](#)
 - [Line Charge \(Out-of-Plane\)](#)
 - [Harmonic Perturbation—Exclusive and Contributing Nodes](#)
-

Line Charge (on Axis)



For 2D axisymmetric models, use the **Line Charge (on Axis)** node to specify line charges along the symmetry axis. Also right-click to add **Harmonic Perturbation**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

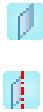
Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

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 and the default is 0 C/m.

-
- 
 - [Line Charge](#)
 - [Line Charge \(Out-of-Plane\)](#)
 - [Harmonic Perturbation—Exclusive and Contributing Nodes](#)
-

Line Charge (Out-of-Plane)



For 2D and 2D axisymmetric models, points are selected and this is the same as a line out-of-plane. Also right-click to add **Harmonic Perturbation**.

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 define.



Beware that constraining the potential on points usually yields a current outflow that is mesh dependent.

MATERIAL TYPE

Select a **Material type**—**Solid**, **Non-solid** (the default), or **From material**.

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 and the default is 0 C/m.

-
- [Line Charge](#)
 - [Line Charge \(on Axis\)](#)
 - [Harmonic Perturbation—Exclusive and Contributing Nodes](#)

Point Charge



The **Point Charge** node adds a point source to 3D models. The point charge represents an electric displacement field flowing out of the point. Also right-click to add **Harmonic Perturbation**.

POINT SELECTION

From the **Selection** list, choose the points to define.



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. The default is 0 C.



- [Point Charge \(on Axis\)](#)
- [Line Charge \(Out-of-Plane\)](#)
- [Harmonic Perturbation—Exclusive and Contributing Nodes](#)

Point Charge (on Axis)



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. Also right-click to add **Harmonic Perturbation**.

POINT SELECTION

From the **Selection** list, choose the points to define.



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. The default is 0 C.

-
- Point Charge
 - Line Charge (Out-of-Plane)
 - Harmonic Perturbation—Exclusive and Contributing Nodes
-

Change Cross-Section

 This node 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** node 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.

PAIR SELECTION

If this node 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: m²). 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.

 For 2D models, see [Change Thickness \(Out-of-Plane\)](#).

Change Thickness (Out-of-Plane)



This node is available for 2D models. This setting overrides the global **Thickness** setting made in any interface that uses this node.

Use the **Change Thickness (Out-of-Plane)** node 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.

PAIR SELECTION

If this node 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.



For 1D models, see [Change Cross-Section](#).

The Electric Currents User Interface

The **Electric Currents (ec)** user 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 node, 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 nodes that implement, for example, boundary conditions and current 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 **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)

	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).
---	--



Enter a default value for the **Out-of-plane thickness** d (SI unit: m) (see [Equation 10-1](#)). The default value of 1 m 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 **Quintic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

-
- 
- [Show More Physics Options](#)
 - [Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface](#)
 - [Theory of Electric Currents](#)
-



Pacemaker Electrode: Model Library path **COMSOL_Multiphysics/Electromagnetics/pacemaker_electrode**

Domain, Boundary, Edge, Point, and Pair Nodes for the Electric Currents Interface

The **Electric Currents** User Interface has these domain, boundary, edge, point, and pair nodes available.

ABOUT THE BOUNDARY CONDITIONS

The exterior and interior boundary conditions listed in [Table 10-2](#) 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 NODES

These nodes are available for this interface and listed in alphabetical order. Also see [Table 10-2](#) 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](#)
- [Initial Values](#)
- [Line Current Source](#)
- [Line Current Source \(on Axis\)](#)
- [Normal Current Density](#)
- [Point Current Source](#)
- [Sector Symmetry](#)

These nodes are described for the Electrostatics interface:

- [Change Cross-Section](#)
- [Change Thickness \(Out-of-Plane\)](#)
- [Electric Potential](#)
- [Ground](#)
- [Periodic Condition](#)



You can right-click many of the nodes to add the Harmonic Perturbation feature. For more information see [Harmonic Perturbation—Exclusive and Contributing Nodes](#).

	<ul style="list-style-type: none"> Continuity on Interior Boundaries Identity and Contact Pairs
 	<p>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.</p>
	<p>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.</p>

Table 10-2 lists the interior and exterior boundaries available with this interface. It also includes edge, point, and pair availability.

TABLE 10-3: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS (INCLUDING EDGE, POINT, AND PAIR AVAILABILITY) FOR THE ELECTRIC CURRENTS INTERFACE

NODE	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
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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define the electric potential and the continuity equation that describes the potential field or select **All domains** as required.

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** or **Linearized resistivity**.

User Defined

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 for the electrical conductivity σ in the field or matrix. The default is 0 S/m.



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.

Linearized Resistivity

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 resistivity** ρ_0 (SI unit: $\Omega\text{-m}$), **Reference temperature** T_{ref} (SI unit: K), and **Resistivity temperature coefficient** α (SI unit: $1/\text{K}$) are taken **From material**, which means that the values are taken from the domain (or boundary) material. T is the current temperature, which can be a value that is specified as a model input or the temperature from a heat transfer interface. The definition of the temperature field appears in the **Model Inputs** section.



Only certain material models support the **Linearized resistivity**. See [AC/DC Material Database](#).

To specify other values for any of these properties, select **User defined** from the list and then enter a value or expression for each. The default values are:

- 1 $\Omega\text{-m}$ for the Reference resistivity
- 273.15 K for the Reference temperature, and
- 0 $1/\text{K}$ for the Resistivity temperature coefficient

ELECTRIC FIELD

See **Electric Field** as described for the [Charge Conservation](#) node for the Electrostatics interface.

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** nodes.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the

interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains.

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.

External Current Density

The **External Current Density** node adds an externally generated current density \mathbf{J}_e , which appears in Ohm's law

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e$$

and in the equation that the interface defines. Also right-click to add **Harmonic Perturbation**.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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.

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/m²). The defaults are 0 A/m².

Current Source

The **Current Source** node adds a distributed current source Q_j in the equation that the interface defines. Use this node with caution as it may violate the current conservation law that is inherent in Maxwell-Ampère's law. Also right-click to add **Harmonic Perturbation**.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

CURRENT SOURCE

Enter a value or expression for the **Current source** Q_j (SI unit: A/m³). The default is 0 A/m³.

Electric Insulation

The **Electric Insulation** node, which is the default boundary condition, 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.

Electric insulation as the default boundary condition is not applicable to interior boundaries.



To add electric insulation to an interior boundary, add an **Electric Insulation** node in addition to the one that represents the default boundary condition.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node 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

The **Boundary Current Source** node adds a current source Q_j on the boundary.

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = Q_j$$

It is applicable to interior boundaries that represent either a source or a sink of current. Also right-click to add **Harmonic Perturbation**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

PAIR SELECTION

If this node 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: A/m²). The default is 0 A/m².

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. Also right-click to add **Harmonic Perturbation**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to apply a current flow as the boundary condition using the normal current density.

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.

NORMAL CURRENT DENSITY

Select a **Type**—**Inward current density** (the default) or **Current density**.

- If **Inward current density** is selected, enter a value or expression for the **Normal current density** J_n (SI unit: A/m²). Use a positive value for an inward current flow or a negative value for an outward current flow. The default is 0 A/m².
- If **Current density** is selected, enter values or expressions for the components of the **Current density** \mathbf{J}_0 (SI unit: A/m²). The defaults are 0 A/m².

Distributed Impedance

The **Distributed Impedance** node adds a distributed impedance boundary condition to a model. Also right-click to add a **Harmonic Perturbation** subnode.



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:

$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = \frac{\sigma_s}{d_s}(V - V_{\text{ref}})$$
$$\mathbf{n} \cdot (\mathbf{J}_1 - \mathbf{J}_2) = \frac{1}{\rho_s}(V - V_{\text{ref}})$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

DISTRIBUTED IMPEDANCE

Enter the reference potential V_{ref} (SI unit: V). The default is 0 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 (dimensionless). The defaults take values **From material**. Select **User defined** to enter different values or expressions. The default electrical conductivity is $1 \cdot 10^{-2}$ S/m and the default relative permittivity is 1.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: $\Omega \cdot \text{m}^2$) and the **Surface capacitance** C_s (SI unit: F/ m^2). The default surface impedance is $1 \cdot 10^{-8} \Omega \cdot \text{m}^2$ and the default surface capacitance is 0 F/ m^2 .

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 as a **Pair Contact Impedance** node.

$$\mathbf{n} \cdot \mathbf{J}_1 = \frac{\sigma}{d_s}(V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{J}_2 = \frac{\sigma}{d_s}(V_2 - V_1)$$

$$\mathbf{n} \cdot \mathbf{J}_1 = \frac{1}{\rho_s}(V_1 - V_2)$$

$$\mathbf{n} \cdot \mathbf{J}_2 = \frac{1}{\rho_s}(V_2 - V_1)$$

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.



These parameters work the same as with [Distributed Impedance](#).

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

Select a **Material type**—**Solid** (the default), **Non-solid**, or **From material**.

PAIR SELECTION

If this node 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

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 (dimensionless). The defaults take values **From material**. Select **User defined** to enter different values or expressions. The default electrical conductivity is $1 \cdot 10^{-2}$ S/m and the default relative permittivity is 1.
- If **Surface impedance** is selected, enter values or expressions for the **Surface resistance** ρ_s (SI unit: $\Omega \cdot m^2$) and the **Surface capacitance** C_s (SI unit: F/m²). The default surface impedance is $1 \cdot 10^{-8} \Omega \cdot m^2$ and the default surface capacitance is 0 F/m².



Thin-Film Resistance: Model Library path **COMSOL_Multiphysics/**
Electromagnetics/thin_film_resistance

Sector Symmetry

Select **Sector Symmetry** at interfaces between rotating objects where sector symmetry is used. It is only available for pairs. A default [Electric Insulation](#) node is added. Right-click to add additional **Fallback Features**.



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

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** (the default) or **Antiperiodicity**.

Based on space dimension, enter values or expressions in the table for the **Axis of rotation \mathbf{a}_{rot}** .

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.

To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.



- Continuity on Interior Boundaries
 - Identity and Contact Pairs
-

Line Current Source



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 define.



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. The default is 0 A/m.



[Line Current Source \(on Axis\)](#) for 2D axisymmetric models

Line Current Source (on Axis)



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 define.

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.



Line Current Source

Point Current Source



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. Also right-click to add **Harmonic Perturbation**.

POINT SELECTION

From the **Selection** list, choose the points to define.



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.



- [Line Current Source](#) for 2D models
- [Point Current Source \(on Axis\)](#) for 2D axisymmetric models

Point Current Source (on Axis)



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 define.



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.



- [Point Current Source](#) for 3D models
- [Line Current Source](#) for 2D models

The Magnetic Fields User Interface

The **Magnetic Fields (mf)** user 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 node is **Ampère's Law**, 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 physics user 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 nodes that implement, for example, boundary conditions and external currents.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics user 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 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

This section allows the specification of a background magnetic vector potential (that generates a background magnetic flux density). The only option to **Solve for** is **Full field**.

COMPONENTS

The current vector has the same direction as the magnetic vector potential. 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. Therefore, the only **Components** option is **Out-of-plane vector potential**.

THICKNESS

For 2D models, enter a value or expression for the global **Out-of-plane thickness** d (SI unit: m). The default 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**, or **Cubic**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

- [Show More Physics Options](#)
- [Domain, Boundary, Point, and Pair Nodes for the Magnetic Fields Interface](#)
- [Theory of Magnetic Fields](#)



[Quadrupole Lens](#): Model Library path **COMSOL_Multiphysics/Electromagnetics/quadrupole**

Domain, Boundary, Point, and Pair Nodes for the Magnetic Fields Interface

The [Magnetic Fields User Interface](#) has these domain, boundary, point, and pair nodes available, which are listed in alphabetical order.



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.

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 \mathbf{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.

[Table 10-2](#) lists the interior and exterior boundaries available with this interface.

TABLE 10-4: INTERIOR AND EXTERIOR BOUNDARY CONDITIONS FOR THE MAGNETIC FIELDS INTERFACE

NODE	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

Available Nodes

These nodes are available for this interface and listed in alphabetical order. Also see [Table 10-2](#) 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 \(Lorentz Term\)](#)



For 2D axisymmetric models, COMSOL takes the axial symmetry boundaries (at $r = 0$) into account and adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.



Infinite Element Domains and Perfectly Matched Layers

Ampère's Law

The **Ampère's Law** node 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

For a default node, the setting inherits the selection from the parent node and cannot be edited; that is, the selection is automatically selected and is the same as for the physics user interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose the domains to define the magnetic vector potential and the equation based on Ampère's law that defines the potential or select **All domains** as required.

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

This section is described for the [Current Conservation](#) feature.

ELECTRIC FIELD

The default **Relative permittivity** ϵ_r (dimensionless) 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 describes 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.



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 a **Constitutive relation**—**Relative permeability** (the default), **HB curve**, **Magnetic losses**, **Remanent flux density**, or **Magnetization**.

Relative Permeability

Select **Relative permeability** μ_r (dimensionless) to use the constitutive relation $\mathbf{B} = \mu_0 \mu_r \mathbf{H}$. If **User defined** is selected, choose **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and enter values or expressions in the field or matrix.

HB Curve

Select **HB curve** $|\mathbf{H}|$ (SI unit: A/m) to use a curve that relates magnetic flux density \mathbf{B} and the magnetic field \mathbf{H} as $|\mathbf{H}| = f(|\mathbf{B}|)$.

Magnetic Losses

Select **Magnetic losses** μ' and μ'' (dimensionless) 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.

Remanent Flux Density

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 (dimensionless) 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 **Remanent flux density** \mathbf{B}_r .

Magnetization

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.

Initial Values

The **Initial Values** node 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.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter values or expressions for the **Magnetic vector potential A** (SI unit: Wb/m). The defaults are 0 Wb/m.

External Current Density

The **External Current Density** node 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. Also right-click to add **Harmonic Perturbation**.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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.

EXTERNAL CURRENT DENSITY

Enter a value or expression for each component of the **External current density \mathbf{J}_e** (SI unit: A/m²). The defaults are 0 A/m².



[Harmonic Perturbation—Exclusive and Contributing Nodes](#)

Velocity (Lorentz Term)

This node is only valid in 2D and 2D axisymmetry when only solving for the out-of-plane component of the magnetic vector potential.



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.

The **Velocity (Lorentz term)** node adds velocity \mathbf{v} . The external current is equal to $\sigma\mathbf{v} \times \mathbf{B}$.

An operational definition of when it can be used is that the moving domain should only contain an induced magnetic source (magnetization plus 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 or Hall generators/thrusters, for example).



If you are not sure how to proceed, contact COMSOL Technical Support:
<http://www.comsol.com/support>.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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)

User defined is selected by default. Enter the components for the **Velocity** vector \mathbf{v} (SI unit: m/s) or, if present, select any velocity field defined in the model.



For example, using the velocity field is useful when coupling to the velocity field of a fluid for a magnetohydrodynamic model.

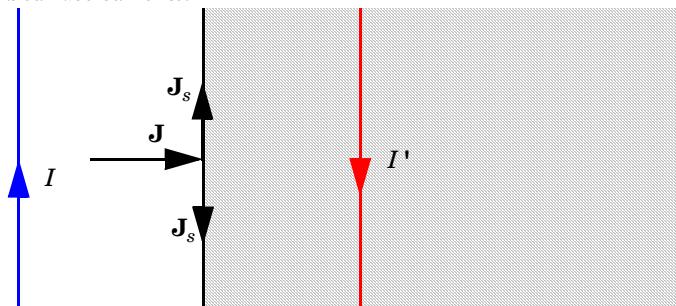
Magnetic Insulation

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 $\mathbf{n} \times \mathbf{A} = 0$.



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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, that is, **All boundaries** is automatically selected and this applies to all the *external* boundaries. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.



Weak constraints perform poorly when applied on vector elements. They should be used when the magnetic vector potential is discretized with Lagrange elements, for example when solving for out-of-plane component in a two-dimensional model.

Magnetic Field

The **Magnetic Field** node adds a boundary condition for specifying the tangential component of the magnetic field at the boundary:

$$\mathbf{n} \times \mathbf{H} = \mathbf{n} \times \mathbf{H}_0$$

Also right-click to add **Harmonic Perturbation** node.



Harmonic Perturbation—Exclusive and Contributing Nodes

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node 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) vector coordinates. The defaults are 0 A/m.

Surface Current

The **Surface Current** node adds a boundary condition for a surface current density \mathbf{J}_s :

$$\begin{aligned}-\mathbf{n} \times \mathbf{H} &= \mathbf{J}_s \\ \mathbf{n} \times (\mathbf{H}_1 - \mathbf{H}_2) &= \mathbf{J}_s\end{aligned}$$

Also right-click to add **Harmonic Perturbation**.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

MATERIAL TYPE

Select a **Material type**—**Non-solid** (the default), **Solid**, or **From material**.

PAIR SELECTION

If this node 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}_{s0} (SI unit A/m) coordinates. The defaults are 0 A/m.



Harmonic Perturbation—Exclusive and Contributing Nodes

Magnetic Potential

The **Magnetic Potential** node 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 define.

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 this node 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

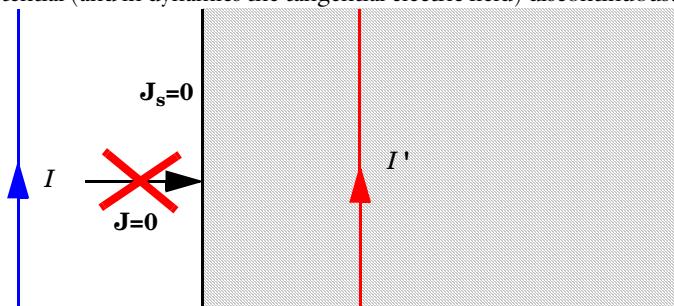
Enter a value or expression for the **Magnetic vector potential \mathbf{A}_0** (SI unit: Wb/m) coordinates.

CONSTRAINT SETTINGS

These settings are the same as for [Magnetic Insulation](#).

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 define.

PAIR SELECTION

If this node 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)** node, selected from the **Points** menu, to specify a line current out of the modeling plane. In axially symmetric geometries this is the rotational (azimuthal) direction, in 2D geometries this is the z-direction.

POINT SELECTION

From the **Selection** list, choose the points to define.

LINE CURRENT (OUT-OF-PLANE)

Enter a value or expression for the **Out-of-plane current** I_0 (SI unit: A). The default is 0 A.



Pressure Acoustics

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.

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 also includes a brief introduction to acoustics, gives some examples of standard acoustics problems, and provides a short introduction to the mathematical formulation of the governing equations.

In this section:

- [Acoustics Explained](#)
- [Examples of Standard Acoustics Problems](#)
- [Mathematical Models for Acoustic Analysis](#)



[Overview of the Physics](#) and [Building a COMSOL Model](#)

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 the atmospheric pressure (about 100,000 pascals), which varies slowly. Associated with a sound pressure wave is a flow of energy—the intensity. 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 propagating at the speed of sound.

The propagation of sound in solids happens through small-amplitude elastic oscillations of its shape. These elastic waves are transmitted to surrounding fluids as ordinary sound waves. The elastic sound waves in the solid are the counter part to the pressure waves or compressible waves propagating in the fluid.

Examples of Standard Acoustics Problems

Depending on the basic dependent variable used to model the acoustic field, the acoustical interfaces can be divided into the following main categories.

- Pressure acoustics—The dependent variable is the acoustic pressure p .
- Acoustic-solid interaction—The dependent variables are the pressure p and the displacement field \mathbf{u} in the solid. This type of problem requires the addition of the Acoustics Module.
- Poroelastic waves—The dependent variables are the pressure p inside the saturating fluid and the total displacement \mathbf{u} of the porous matrix. This type of problem requires the addition of the Acoustics Module.
- Aeroacoustics—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. This type of problem requires the addition of the Acoustics Module.
- Thermoacoustics—The dependent variables are the acoustic pressure p , the particle-velocity field \mathbf{v} , and the acoustic temperature variation T . This is a detailed acoustic model solving the full set of linearized equations for a compressible flow: Navier-Stokes (momentum conservation), continuity (mass conservation), and energy conservation equations. This type of problem requires the addition of the Acoustics Module.

These standard problems or scenarios occur frequently when analyzing acoustics:

THE RADIATION PROBLEM

A vibrating structure (a speaker, for example) radiates sound into the surrounding space. A radiation boundary condition or a PML (perfectly matched layer) is necessary to model the unbounded open domain.

THE SCATTERING PROBLEM

An incident wave impinges on a body and creates a scattered wave. A radiation boundary condition or a PML is necessary. This could be a sonar application in underwater acoustics or an analysis of the scattered sound field around a human head.

THE SOUND FIELD IN AN INTERIOR SPACE

The acoustic waves stay in a finite volume so no radiation condition is necessary. This could be the sound inside a room or a car interior. A more advanced example is the sound inside a transducer like a microphone; in this case, the acoustic field should be solved with the Thermoacoustics interface. Analysis using thermoacoustics requires the Acoustics Module.

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. A typical transmission problem is that of modeling the behavior of mufflers.

AEROACOUSTICS PROBLEMS

The sound (noise) field is influenced by a background flow. This could be the propagating sound from a jet engine. Analysis of these types of problems require the addition of the Acoustics Module.

POROELASTIC WAVES PROBLEM

If the acoustic waves are propagating inside the saturating fluid of porous material the detailed coupling between the fluid pressure and the solid displacement need to be taken into account. In cases where only the fluid pressure is of interest the porous material may be modeled using an equivalent fluid model. Analysis of this type of problem requires the addition of the Acoustics Module.

TRANSDUCER PROBLEMS

Transducers transformation of one form of energy to another type (electrical, mechanical, or acoustical). This type of problem is common in acoustics and is a true

multiphysics problem involving electric, structural, and acoustic interfaces. Typical problems of this type involve modeling loudspeakers, microphones, and piezo transducers. Analysis of these types of problems require the addition of the Acoustics Module.

Mathematical Models for Acoustic Analysis

Standard acoustic problems involve solving for the small acoustic pressure variations p' on top of the stationary background pressure p_0 . Mathematically this represents a linearization (small parameter expansion) around the stationary quiescent values.

The governing equations for a compressible lossless (no thermal conduction and no viscosity) fluid flow problem are the momentum equation (Euler's equation) and the continuity equation. These are given by:

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\frac{1}{\rho} \nabla p \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0\end{aligned}$$

where ρ is the total density, p is the total pressure, and \mathbf{u} is the velocity field. In classical pressure acoustics all thermodynamic processes are assumed reversible and adiabatic, known as an isentropic process. The small parameter expansion is performed on a stationary fluid of density ρ_0 (SI unit: kg/m³) and at pressure p_0 (SI unit: Pa) such that:

$$\begin{aligned}p &= p_0 + p' & p' \ll p_0 \\ \rho &= \rho_0 + \rho' & \text{with} & \rho' \ll \rho_0 \\ \mathbf{u} &= \mathbf{0} + \mathbf{u}'\end{aligned}$$

where the primed variables represent the small acoustic variations. Inserting these into the governing equations and only retaining terms linear in the primed variables yields:

$$\begin{aligned}\frac{\partial \mathbf{u}'}{\partial t} &= -\frac{1}{\rho_0} \nabla p' \\ \frac{\partial \rho'}{\partial t} + \rho_0(\nabla \cdot \mathbf{u}') &= 0\end{aligned}$$

One of the dependent variables, the density, is removed by expressing it in terms of the pressure using a Taylor expansion (linearization):

$$\rho' = \frac{\partial \rho_0}{\partial p} \Big|_s p' = \frac{1}{c_s^2} p'$$

where c_s is recognized as the (isentropic) speed of sound (SI unit: m/s) at constant entropy s . The subscripts s and 0 are dropped in the following. Finally, rearranging the equations (divergence of momentum equation inserted into the continuity equation) and dropping the primes yields the wave equation for sound waves in a lossless medium:

$$\frac{1}{\rho c^2} \frac{\partial^2 p}{\partial t^2} + \nabla \cdot \left(-\frac{1}{\rho} (\nabla p - \mathbf{q}_d) \right) = Q_m \quad (11-1)$$

The speed of sound is related to the compressibility of the fluid where the waves are propagating. The combination ρc^2 is called the *bulk modulus*, commonly denoted K (SI unit: N/m²). The equation is further extended with two optional source terms: the *dipole source* \mathbf{q}_d (SI unit: N/m³) and the *monopole source* Q_m (SI unit: 1/s²).

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 $\omega = 2\pi f$ (SI unit: rad/s) is the angular frequency and f (SI unit: Hz) is 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:

$$\nabla \cdot \left(-\frac{1}{\rho} (\nabla p - \mathbf{q}_d) \right) - \frac{\omega^2}{\rho c^2} p = Q_m \quad (11-2)$$

With the two source terms removed, this equation can also be treated as an eigenvalue PDE to solve for eigenmodes and eigenfrequencies.

Typical boundary conditions for the wave equation and the Helmholtz equation are:

- Sound-hard boundaries (walls)
- Sound-soft boundaries
- Impedance boundary conditions
- Radiation boundary conditions

In lossy media, an additional term of first order in the time derivative needs to be introduced to model attenuation of the sound waves:

$$\frac{1}{\rho c^2} \frac{\partial^2 p}{\partial t^2} - d_a \frac{\partial p}{\partial t} + \nabla \cdot \left(-\frac{1}{\rho} (\nabla p - \mathbf{q}_d) \right) = Q_m$$

where d_a is the damping coefficient. 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 Pressure Acoustics, Frequency Domain

The Pressure Acoustics, Frequency Domain User 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 physics user 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:

$$\nabla \cdot \left(-\frac{1}{\rho_c} (\nabla p - \mathbf{q}_d) \right) - \frac{\omega^2 p}{\rho_c c_c^2} = Q_m \quad (11-3)$$

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 11-3 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 11-3](#) gives

$$\nabla \cdot \left(-\frac{1}{\rho_c} (\nabla p - \mathbf{q}_d) \right) - \frac{1}{\rho_c} \left(\frac{\omega^2}{c_c^2} - k_z^2 \right) p = Q_m \quad (11-4)$$

The *out-of-plane wave number*, k_z , can be set on the Pressure Acoustics page. By default its value is 0. 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,

$$p(r, \phi, z) = p(r, z)e^{-im\phi} \quad (11-5)$$

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 settings window.

As a result of [Equation 11-5](#), the equation to solve for the acoustic pressure in 2D axisymmetric geometries becomes

$$\frac{\partial}{\partial r} \left[-\frac{r}{\rho_c} \left(\frac{\partial p}{\partial r} - q_r \right) \right] + r \frac{\partial}{\partial z} \left[-\frac{1}{\rho_c} \left(\frac{\partial p}{\partial z} - q_z \right) \right] - \left[\left(\frac{\omega}{c_c} \right)^2 - \left(\frac{m}{r} \right)^2 \right] \frac{rp}{\rho_c} = rQ_m$$

1D AXISYMMETRY

In 1D axisymmetric geometries,

$$p(r, \phi, z) = p(r)e^{-i(k_z z + m\phi)}$$

leading to the radial equation

$$\frac{\partial}{\partial r} \left[-\frac{r}{\rho_c} \left(\frac{\partial p}{\partial r} - q_r \right) \right] - \left[\left(\frac{\omega}{c_c} \right)^2 - \left(\frac{m}{r} \right)^2 - k_z^2 \right] \frac{rp}{\rho_c} = r Q_m$$

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 :

$$\frac{d}{dx} \left(-\frac{1}{\rho_c} \left(\frac{dp}{dx} - q_d \right) \right) - \frac{\omega^2}{\rho_c c_c^2} p = Q_m$$

Eigenfrequency Study

In the eigenfrequency formulation the source terms are absent, the eigenmodes and eigenfrequencies are solved for:

$$\nabla \cdot \left(-\frac{1}{\rho_c} \nabla p \right) + \frac{\lambda^2 p}{\rho_c c^2} = 0 \quad (11-6)$$

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 11-6](#) 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 User Interface

The **Pressure Acoustics, Frequency Domain (acpr)** user 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 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.

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 a 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 and SI units for the most important physical quantities in the Pressure Acoustics, Frequency Domain interface:

TABLE II-I: 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}_d	newton/meter ³	N/m ³
Monopole source	Q_m	1/second ²	1/s ²
Speed of sound	c	meter/second	m/s
Acoustic impedance	Z	pascal-second/meter	Pa·s/m
Normal acceleration	a_n	meter/second ²	m/s ²

TABLE II-1: PRESSURE ACOUSTICS, FREQUENCY DOMAIN INTERFACE PHYSICAL QUANTITIES

QUANTITY	SYMBOL	SI UNIT	ABBREVIATION
Source location	r_0	meter	m
Wave direction	\mathbf{n}_k	(dimensionless)	

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 add other features that implement, for example, boundary conditions.



Physics Nodes—Equation Section

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics user 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.

EQUATION

Expand the **Equation** section to see the equations solved for with the **Equation form** specified. The default selection is **Equation form** is set to **Study controlled**. The available studies are selected under **Show equations assuming**.



When the **Equation form** is set to **Study controlled**, the scaling and non-reflecting boundary settings are optimized for the numerical performance of the different solvers.



To display the **Pressure Acoustics Equation Settings** section for 2D and 1D models, click to expand the **Equation** section, then select **Frequency domain** as the **Equation form** and enter the settings as described in [Scaling Factor and Non-reflecting Boundary Condition Approximation](#).

PRESSURE ACOUSTICS EQUATION SETTINGS



For 1D axisymmetric models, the **Circumferential wave number m** (dimensionless) default is 0 and the **Out-of-plane wave number k_z** (SI unit: rad/m) default is 0 rad/m. Enter different values or expressions as required.



For 2D axisymmetric models, the **Circumferential wave number m** (dimensionless) default is 0. Enter a different value or expression as required.



For 2D models, the **Out-of-plane wave number k_z** (SI unit: rad/m) default is 0 rad/m. Enter a different value or expression as required.

Scaling Factor and Non-reflecting Boundary Condition Approximation

For all model dimensions, and if required, click to expand the **Equation** section, then select **Frequency domain** as the **Equation form** and enter the settings as described below.

The default **Scaling factor** Δ is $1/\omega^2$ and **Non-reflecting boundary condition approximation** is **Second order**. These values correspond to the equations for a Frequency Domain study when the equations are study controlled.

To get the equations corresponding to an Eigenfrequency study, change the **Scaling factor** Δ to 1 and the **Non-reflecting boundary conditions approximation** to **First order**.

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 L_p based on the root mean square (rms) pressure p_{rms} , such that

$$L_p = 20 \log\left(\frac{p_{\text{rms}}}{p_{\text{ref}}}\right) \quad \text{with} \quad p_{\text{rms}} = \sqrt{\frac{1}{2} p p^*}$$

where p_{ref} is the reference pressure and the star (*) represents the complex conjugate. This is 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 $20 \mu\text{Pa}$ ($20 \cdot 10^{-6} \text{ Pa}$).
- **Use reference pressure for water** to use a reference pressure of $1 \mu\text{Pa}$ ($1 \cdot 10^{-6} \text{ Pa}$).
- **User-defined reference pressure** to enter a reference pressure p_{ref} , SPL (SI unit: Pa). The default value is the same as for air, $20 \mu\text{Pa}$.

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.

DISCRETIZATION

To display this section, click the **Show** button () and select **Discretization**. Select **Quadratic** (the default), **Linear**, **Cubic**, **Quartic**, or **Quintic** for the **Pressure**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex** (the default).

-
- 
- Show More Physics Options
 - Domain, Boundary, Edge, Point, and Pair Nodes for the Pressure Acoustics, Frequency Domain Interface
 - Theory for Pressure Acoustics, Frequency Domain
-



Eigenmodes of a Room: Model Library path **COMSOL_Multiphysics/Acoustics/eigenmodes_of_room**

Domain, Boundary, Edge, Point, and Pair Nodes for the Pressure Acoustics, Frequency Domain Interface

The Pressure Acoustics, Frequency Domain User Interface has these domain, boundary, edge, point, and pair nodes available and listed in alphabetical order.

- Continuity
- Cylindrical Wave Radiation
- Dipole Source
- Destination Selection
- Impedance
- Incident Pressure Field
- Interior Sound Hard Boundary (Wall)
- Initial Values
- Monopole Source
- Normal Acceleration
- Periodic Condition
- Plane Wave Radiation
- Pressure Acoustics Model
- Pressure
- Sound Hard Boundary (Wall)
- Sound Soft Boundary
- Spherical Wave Radiation
- Symmetry

	Continuity in the total pressure is the default condition on interior boundaries.
 	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.
	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** node 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to compute the acoustic pressure field and the equation that defines it, or select **All domains** as required.

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: kg/m³) 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** node to add a the domain source term Q_m to the governing equation. A monopole source added to a domain has a uniform strength in all directions. In advanced models this source term can, for example, be used to represent a domain heat source causing pressure variations.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

MONOPOLE SOURCE

Enter a **Monopole source** Q_m (SI unit: $1/s^2$). The default is $0\ 1/s^2$.

Dipole Source

Use the **Dipole Source** node to add the domain source term \mathbf{q}_d to the governing equation. This source will is typically stronger in two opposite directions. In advanced models this term may, for example, be used to represent a uniform constant background flow convecting the sound field.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

DIPOLE SOURCE

Enter coordinates for the **Dipole source** \mathbf{q}_d (SI unit: N/m^3). These are the individual components of the dipole source vector. The defaults are $0\ N/m^3$.

Initial Values

The **Initial Values** node 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the initial values for the **Pressure** p (SI unit: Pa) and the **Pressure, first time derivative**, $\partial p / \partial t$ (SI unit: Pa/s). The defaults are 0 Pa and 0 Pa/s, respectively.

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:

$$-\mathbf{n} \cdot \left(-\frac{1}{\rho_0} (\nabla p - \mathbf{q}_d) \right) = 0$$

For zero dipole source and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary:

$$\frac{\partial p}{\partial \mathbf{n}} = 0$$

Sound-hard boundaries are available for all study types. Note that this condition is identical to the [Symmetry](#) condition.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

Normal Acceleration

The **Normal Acceleration** adds an inward normal acceleration a_n :

$$-\mathbf{n} \cdot \left(-\frac{1}{\rho_0} (\nabla p - \mathbf{q}_d) \right) = a_n$$

Alternatively, specify the acceleration \mathbf{a}_0 of the boundary. The part in the normal direction is used to define the boundary condition:

$$\mathbf{n} \cdot \left(-\frac{1}{\rho_0} (\nabla p - \mathbf{q}_d) \right) = \mathbf{n} \cdot \mathbf{a}_0$$

This feature represents an external source term. It can also be used to manually couple acoustics with a structural analysis for modeling acoustic-structure interaction.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

NORMAL ACCELERATION

Select a **Type**—**Inward Acceleration** (the default) or **Acceleration**.

- If **Inward Acceleration** is chosen, enter the value of the **Inward acceleration a_n** (SI unit: m/s²). The default is 0 m/s². Use a positive value for inward acceleration or a negative value for outward acceleration.
- If **Acceleration** is chosen, enter values for the components of the **Acceleration \mathbf{a}_0** (SI unit: m/s²). The defaults are 0 m/s².

Sound Soft Boundary

The **Sound Soft Boundary** adds a boundary condition for a sound soft boundary, where the acoustic pressure vanishes: $p = 0$. It 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. If the node is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PAIR SELECTION

If this node 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**.

To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

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 node 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 define. If the node 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). The default is 0 Pa.

PAIR SELECTION

If this node 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

These are the same settings as for [Sound Soft Boundary](#).

Impedance

The **Impedance** node adds an impedance boundary condition, which is a generalization of the sound-hard and sound-soft boundary conditions:

$$-\mathbf{n} \cdot \left(-\frac{1}{\rho_c} (\nabla p_t - \mathbf{q}_d) \right) = -\frac{i \omega p_t}{Z_i}$$

Here Z_i is the acoustic input impedance of the external domain and it has the unit of a specific acoustic impedance. From a physical point of view, the acoustic input impedance is the ratio between the local pressure and local 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.



In the two opposite limits $Z_i \rightarrow \infty$ and $Z_i \rightarrow 0$, this boundary condition is identical to the [Sound Hard Boundary \(Wall\)](#) condition and the [Sound Soft Boundary](#) condition, respectively.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

IMPEDANCE

Enter the value of the **Impedance** Z_i (SI unit: Pa·s/m). The default value is set to the specific impedance of air $1.2 \text{ kg/m}^3 \cdot 343 \text{ m/s}$.

Symmetry

The **Symmetry** node adds a boundary condition where there is symmetry in the pressure. Use this condition to reduce the size of a model by cutting it in half where there are symmetries. In pressure acoustics this boundary condition is mathematically identical to the [Sound Hard Boundary \(Wall\)](#) condition.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. If the node is selected from the **Pairs** submenu, this list cannot be edited and it shows the boundaries in the selected pairs.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

Plane Wave Radiation

The **Plane Wave Radiation** node adds a radiation boundary condition for a plane wave. If required, right-click the main node to add an [Incident Pressure Field](#) to model an incoming wave. This radiation condition allow an outgoing plane wave to leave the modeling domain with minimal reflections, when the angle of incidence is near to normal.

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. When using the radiation condition on an open far-field boundary it is recommended to construct the boundary such that the incidence angle is near to normal, this of course requires a priori knowledge of the problem and the solution.

An estimate of the reflection coefficient R_s , for the spurious waves reflecting off the plane wave radiation boundary, is, for incident plane waves at angle θ , given by the expression:

$$R_s = \left| \frac{\cos\theta - 1}{\cos\theta + 1} \right|^N$$



where N is the order of the boundary condition (here 1 or 2). So at normal incidence ($\theta = 0$) there are no spurious reflections, while, for example, at an incidence angle of 30° for $N = 2$ (plane wave radiation in the frequency domain) the amplitude of the spurious reflected wave is 0.5 % of the incident.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.



Acoustics of a Muffler: Model Library path **COMSOL_Multiphysics/Acoustics/automotive_muffler**

Spherical Wave Radiation

The **Spherical Wave Radiation** node adds a radiation boundary condition for a spherical wave, for which you define the source location. If required, right-click the main node to add an **Incident Pressure Field** to model an incoming wave. This radiation condition allows an outgoing spherical wave to leave the modeling domain with minimal reflections. The geometry of the modeling domain should be adapted to have the outgoing spherical waves coincide with the boundary, this is in order to minimize spurious reflections.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Spherical Wave Radiation

Enter coordinates for the **Source location** r_0 (SI unit: m). The defaults are 0 m.

Cylindrical Wave Radiation

The **Cylindrical Wave Radiation** node adds a radiation boundary condition for a cylindrical wave, for which you define the source location and the source axis direction. If required, right-click the main node to add an [Incident Pressure Field](#) to model an incoming wave. This radiation condition allows an outgoing cylindrical wave to leave the modeling domain with minimal reflections. The geometry of the modeling domain should be adapted to have the outgoing cylindrical waves coincide with the boundary, this is in order to minimize spurious reflections.

Boundary Selection

From the **Selection** list, choose the boundaries to define.

Cylindrical Wave Radiation

Enter coordinates for the **Source location** r_0 (SI unit: m) (the defaults are 0 m) and the **Source axis** direction r_{axis} (dimensionless) (the defaults are 0).

Incident Pressure Field

The **Incident Pressure Field** node is a subnode to all nonreflecting boundary conditions (plane, cylindrical, or spherical wave radiation). Right-click the [Plane Wave Radiation](#), [Spherical Wave Radiation](#), or [Cylindrical Wave Radiation](#) nodes to add this subnode. If the incident pressure field p_i is a predefined plane wave, it is of the type:

$$p_i = p_0 e^{-i(\mathbf{k} \cdot \mathbf{r})} = p_0 e^{-ik_{\text{eq}} \left(\frac{\mathbf{r} \cdot \mathbf{e}_k}{\|\mathbf{e}_k\|} \right)}$$

where p_0 is the wave amplitude, \mathbf{k} is the wave vector (with amplitude $k_{\text{eq}} = |\mathbf{k}|$ 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. Then enter a **Pressure amplitude** p_0 (SI unit: Pa) (the default is 0 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 Pa.

Periodic Condition

The **Periodic Condition** node adds a periodic boundary condition that can be used to reduce the model size by using symmetries and periodicities in the geometry and physics being modeled.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. The software automatically identifies the boundaries as either source boundaries or destination boundaries.



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) or **Antiperiodicity**.

CONSTRAINT SETTINGS

These are the same settings as for [Sound Soft Boundary](#).



- [Periodic Condition](#) and [Destination Selection](#)
- [Periodic Boundary Conditions](#)

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:

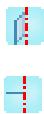
$$-\mathbf{n} \cdot \left(\frac{1}{\rho_c} (\nabla p - \mathbf{q}_d) \right)_1 = 0 \quad -\mathbf{n} \cdot \left(\left(\frac{1}{\rho_c} \right) (\nabla p - \mathbf{q}_d) \right)_2 = 0$$

where the subscripts 1 and 2 represent the two sides of the boundary. For zero dipole charge and constant fluid density, this means that the normal derivative of the pressure is zero at the boundary. On an interior sound hard boundary the pressure is not continuous but is treated as a so-called slit feature.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Axial Symmetry



The **Axial Symmetry** node is a default node added for all 2D and 1D axisymmetric models. The boundary condition is active on all boundaries on the symmetry axis.

BOUNDARY SELECTION



The boundaries section shows on which boundaries the node 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.



- Continuity on Interior Boundaries
- Identity and Contact 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):

$$-\mathbf{n} \cdot \left[-\left(\frac{1}{\rho_c} (\nabla p_t - \mathbf{q}_d) \right)_1 - \left(\frac{1}{\rho_c} (\nabla p_t - \mathbf{q}_d) \right)_2 \right] = 0$$

BOUNDARY SELECTION



This list cannot be edited. It shows the boundaries in the selected pairs.

PAIR SELECTION

When this node 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

These are the same settings as for [Sound Soft Boundary](#).

Chemical Species Transport

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.

Theory for Transport of Diluted Species

The Transport of Diluted Species user 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 user 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.

Mass Balance Equation

The default node attributed to the Transport of Diluted Species user interface assumes chemical species transport through diffusion and convection and implements the mass balance equation:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot (D \nabla c) + R \quad (12-1)$$

Equation 12-1 includes these quantities:

- c is the concentration of the species (SI unit: mol/m³)
- D denotes the diffusion coefficient (SI unit: m²/s)
- R is a reaction rate expression for the species (SI unit: mol/(m³.s))
- \mathbf{u} is the velocity vector (SI unit: m/s)

The first term on the left-hand side of Equation 12-1 corresponds to the accumulation (or indeed consumption) of the species.

The second term accounts for the convective transport due to a velocity field \mathbf{u} . This field can be expressed analytically or be obtained from coupling this physics 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 the velocity vector component variable names from a fluid-flow user interface, can be entered into the appropriate field. The velocity fields from existing fluid-flow user interfaces are available directly as predefined fields (model inputs) for multiphysics couplings.

On the right-hand side of the mass balance equation ([Equation 12-1](#)), 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, and any equation that relates to another variable, such as temperature, can be entered there. The node has a matrix that you can use to describe the diffusion coefficient if it is vectorized or is a tensor. Anisotropic diffusion can therefore be simulated.

Finally, the second term on the right-hand side of [Equation 12-1](#) 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.

Convective Term Formulation

The default node attributed to [The Transport of Diluted Species User Interface](#) assumes chemical species transport through diffusion and convection and implements the mass balance equation in [Equation 12-1](#).

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:

$$\text{non-conservative: } \frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot (D \nabla c) + R \quad (12-2)$$

$$\text{conservative: } \frac{\partial c}{\partial t} + \nabla \cdot (c \mathbf{u}) = \nabla \cdot (D \nabla c) + R \quad (12-3)$$

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: $\text{mol}/(\text{m}^3 \cdot \text{s})$) 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.



To switch between the two formulations, click the **Show** button () and select **Advanced Physics Options**.

Solving a Diffusion Equation Only

Remove the convection term from [Equation 12-2](#) and [Equation 12-3](#) by clearing the Convection check box in the Transport Mechanisms section for [The Transport of Diluted Species User Interface](#). The equation then becomes

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c) + R$$

Crosswind Diffusion

Transport of diluted species applications can often result in models with very high cell Péclèt number—that is, systems where convection or migration dominates over diffusion. Streamline diffusion and crosswind diffusion are of paramount importance to obtain physically reasonable results. The Transport of Diluted Species user interface provides two crosswind diffusion options using different formulations. Observe that crosswind diffusion makes the equation system nonlinear even if the transport equation is linear.

DO CARMO AND GALEÃO

This is the formulation described in [Numerical Stabilization](#). The method reduces over- and undershoot to a minimum, also for anisotropic meshes.

In some cases, the resulting nonlinear equation system can be difficult to converge. This can happen when the cell Péclèt number is very high and the model contains many thin layers such as contact discontinuities. You then have three options:

- Refine the mesh, specially in regions with thin layers.
- Use a nonlinear solver with constant damping factor less than one.
- Switch to the Codina crosswind formulation.

CODINA

The Codina formulation is described in Ref. 1. It adds diffusion strictly in the direction orthogonal to the streamline direction. Compared to the do Carmo and Galeão formulation, the Codina formulation adds less diffusion but does a less good job at reducing overshoot and undershoot. It also works less well for anisotropic meshes. The advantage is that the resulting nonlinear system is easier to converge and that under-resolved gradients are less smeared out.

Reference

1. R. Codina, “A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 110, pp. 325–342, 1993.

The Transport of Diluted Species User 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 user interface to model transport of a diluted species in chemical systems by convection and diffusion.

In the Transport of Diluted Species user 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.



The optional Chemical Reaction Engineering Module has an extension of this physics user interface for modeling multicomponent convection, diffusion, and migration (electrokinetic flow).

The **Transport of Diluted Species (chds)** user 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 physics user 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 physics user 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 physics user 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 nodes with appropriate boundary conditions and rate expression terms, for example.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first physics user 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.



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

Diffusion is always included. By default, the **Convection** check box is selected under **Additional transport mechanisms**. Click to clear the check box to model using the diffusion equation only. The [Dynamic Transport Feature Node](#) name changes according to the selection made.



If you have the Batteries & Fuel Cells Module, Electrodeposition Module, Corrosion Module, Microfluidics Module, or the Chemical Reaction Engineering Module, you can also choose to model using the **Migration in electric field** option.

DEPENDENT VARIABLES



The dependent variable is **Concentration** c . The species are dependent variables, and their names must be unique with respect to all other dependent variables in the model.

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, a weak term that reduces spurious oscillations is added to the transport equation. The resulting system will be non-linear. There are two options for **Crosswind diffusion type**:
 - **Do Carmo and Galeão**—the default option. This type of crosswind diffusion reduces undershoot and overshoot to a minimum but can in rare cases give equations systems that are difficult to fully converge.
 - **Codina**. This option is less diffusive compared to the Do Carmo and Galeão option but can result in more undershoot and overshoot. It is also less effective for anisotropic meshes. The Codina option activates a text field for the **Lower gradient limit g_{lim}** (SI unit: mol/m⁴). It defaults to 0.1[mol/m³]/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.

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 form** (the default) or **Conservative form**. The conservative formulation should be used for a compressible flow.

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** (the default) or **Complex**.

-
- 

- Convective Term Formulation
 - Show More Physics Options
 - Domain, Boundary, and Pair Nodes for the Transport of Diluted Species User Interface
 - Theory for Transport of Diluted Species
-

-
- | | |
|---|---|
|  | <ul style="list-style-type: none">• 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 |
|---|---|
-

Domain, Boundary, and Pair Nodes for the Transport of Diluted Species User Interface

The [Transport of Diluted Species User Interface](#) has these domain, boundary, and pair nodes available and listed in alphabetical order.

- | | |
|---|---|
| <ul style="list-style-type: none">• Concentration• Dynamic Transport Feature Node• Flux• Flux Discontinuity• Inflow• Initial Values• No Flux (the default boundary condition) | <ul style="list-style-type: none">• Outflow• Periodic Condition• Reactions• Symmetry• Thin Diffusion Barrier• Thin Impermeable Barrier |
|---|---|

 	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.
--	---

- | | |
|---|--|
|  | <ul style="list-style-type: none">• Continuity on Interior Boundaries• Identity and Contact Pairs |
|---|--|
-



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).



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](#) 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define material properties and other parameters that govern the transport equations, or select **All domains** as required.



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 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 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

Select an option from the **Bulk material** list. The default is **None**.

Under **Diffusion coefficient** select the appropriate scalar or tensor type to describe the diffusion transport—**Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic**—then enter the values in the corresponding field for D_c (SI unit: m^2/s).

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.



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 . The default value is 0 mol/m³.

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.



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/(m³·s)), for the species to be solved for. Enter a value or expression in the field.

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

NO FLUX

Select **Apply for all species** (the default) 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.

Concentration

The **Concentration** 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 define.

PAIR SELECTION

If this node 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**.

To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

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/(m²·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/m³) 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 define.

INWARD FLUX

Specify the flux of each species individually. Select the check box for the **Species** to specify the **Inward flux** $N_{0,c}$ (SI unit: mol/(m²·s)), 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.



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** node, except the concentrations of all species must be specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

CONCENTRATION

For the concentration of each species $c_{0,c}$ (SI unit: mol/m³) enter a value or expression.

CONSTRAINT SETTINGS

These settings are the same as for the **Concentration** node.

Outflow

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:

$$\mathbf{n} \cdot (-D\nabla c) = 0$$



In models where only diffusion occurs, this boundary condition is not available.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

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** node, but applies to all species and cannot be applied to individual species.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Flux Discontinuity

The **Flux Discontinuity** node represents a discontinuity in the mass flux across an interior boundary:

$$-\mathbf{n} \cdot (\mathbf{N}_d - \mathbf{N}_u) = N_0 \quad \mathbf{N} = (c\mathbf{u} - D\nabla c)$$

where the value N_0 specifies the jump in flux evaluated the boundary.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

PAIR SELECTION

If this node 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.



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** $N_{0,c}$, SI unit: mol/($\text{m}^2 \cdot \text{s}$) 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 node 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 right-click and add the **Destination Selection** subnode. With this subnode the boundaries that constitute the source and destination surfaces can be manually specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.



- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

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 define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect. A **No Flux** node is added by default to the **Thin Diffusion Barrier** pair.

THIN DIFFUSION BARRIER

Enter a **Layer thickness** d_s (SI unit: m). The default is 0.005 m (5 mm).

Enter a **Diffusion coefficient** $D_{s,c}$ (SI unit: mol/(m²·s)). The default is 0 mol/(m²·s).

Thin Impermeable Barrier

Use the **Thin Impermeable Barrier** feature node to model a thin mass transfer barrier. The feature is available on interior boundaries and introduces a discontinuity in the concentration across the boundary. On each side of the boundary a no flux condition is prescribed for the mass transport. The **Thin Impermeable Barrier** boundary feature can be used to avoid meshing thin structures.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Fluid Flow

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.

Theory of Laminar Flow

The Single-Phase Flow Laminar Flow theory is described in this section.



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 user interfaces are based on the Navier-Stokes equations, which in their most general form read

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (13-1)$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \boldsymbol{\tau}] + \mathbf{F} \quad (13-2)$$

$$\rho C_p \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = -(\nabla \cdot \mathbf{q}) + \boldsymbol{\tau} : \mathbf{S} - \frac{T}{\rho} \frac{\partial \rho}{\partial T} \Big|_p \left(\frac{\partial p}{\partial t} + (\mathbf{u} \cdot \nabla) p \right) + Q \quad (13-3)$$

where

- ρ is the density (SI unit: kg/m³)
- \mathbf{u} is the velocity vector (SI unit: m/s)
- p is pressure (SI unit: Pa)
- $\boldsymbol{\tau}$ is the viscous stress tensor (SI unit: Pa)
- \mathbf{F} is the volume force vector (SI unit: N/m³)
- 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/m²)
- Q contains the heat sources (SI unit: W/m³)
- \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

$$\mathbf{a}:\mathbf{b} = \sum_n \sum_m a_{nm} b_{nm} \quad (13-4)$$

This is sometimes referred to as the double dot product.

[Equation 13-1](#) is the continuity equation and represents the conservation of mass.

[Equation 13-2](#) is a vector equation and represents the conservation of momentum.

[Equation 13-3](#) 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 13-1](#) through [Equation 13-3](#), 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:

$$\tau = 2\mu\mathbf{S} - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I} \quad (13-5)$$

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.



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 13-5](#).

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.



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 13-3](#) is decoupled from [Equation 13-1](#) and [Equation 13-2](#).



If you also have the Heat Transfer Module, the temperature equation is described in the section the Heat Equation found in the Heat Transfer Theory chapter.

2D AXISYMMETRIC FORMULATIONS

A 2D axisymmetric formulation of [Equation 13-1](#) and [Equation 13-2](#) only require $\partial/\partial\phi$ to be zero. That is, there must be no gradients in the azimuthal direction. A common additional assumption is however that $u_\phi=0$. In that case, the ϕ -equation can be removed from [Equation 13-2](#). The resulting equation system will be both easier to converge and computationally less expensive compared to retaining the ϕ -equation. The default 2D axisymmetric formulation of Equations [Equation 13-1](#) and [Equation 13-2](#) therefore assumes that

$$\begin{aligned}\partial/\partial\phi &= 0 \\ u_\phi &= 0\end{aligned}$$



If you have the CFD Module the **Swirl Flow** property is available, which reduces the above assumptions to $\partial/\partial\phi = 0$ and reintroduces the ϕ -equation to [Equation 13-2](#). Also see the *CFD Module User's Guide* for more information.

Compressible Flow

The Navier-Stokes equations solved by default in all single-phase flow interfaces are the compressible formulation of the continuity:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (13-6)$$

and the momentum equations:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nabla \cdot \left(\mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3} \mathbf{u} (\nabla \cdot \mathbf{u}) \mathbf{I} \right) + \mathbf{F} \quad (13-7)$$

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

$$Ma = \frac{|\mathbf{u}|}{a}$$

where a is the speed of sound. A flow is formally incompressible when $Ma = 0$. 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 13-1](#) through [Equation 13-3](#), 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 physics user interfaces uses the same boundary conditions as the incompressible user interfaces, which implies that the compressible user 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. One 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. Another 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 $\text{Ma} = 0.3$. For this reason the compressible formulation is referred to as *compressible flow ($\text{Ma} < 0.3$)* in COMSOL Multiphysics®.



The CFD Module includes physics user 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 13-6](#) reduces to

$$\rho \nabla \cdot \mathbf{u} = 0 \quad (13-8)$$

and [Equation 13-7](#) becomes

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \mathbf{F} \quad (13-9)$$

The Reynolds Number

Fundamental to the analysis of fluid flow is the Reynolds number:

$$\text{Re} = \frac{\rho U L}{\mu}$$

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 $\text{Re}^c = \rho|\mathbf{u}|h/(2\mu)$ using the element length h for L and the magnitude of the velocity vector \mathbf{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 Fluid Flow

The momentum equations ([Equation 13-7](#) and [Equation 13-9](#)) 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](#).



- Numerical Stabilization
 - Iterative
-

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 of HM type.

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.

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.



Crosswind diffusion also enables the iterative solvers to use inexpensive presmothers such as SSOR. If crosswind diffusion is deactivated, 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 13-9](#) reduces to:

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F} \quad (13-10)$$

Solving [Equation 13-10](#) 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 13-10](#):

$$\rho \frac{\mathbf{u} - \text{nojac}(\mathbf{u})}{\Delta t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot [-p \mathbf{I} + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \mathbf{F}$$

where $\tilde{\Delta t}$ is a *pseudo time step*. Since $\mathbf{u} - \text{nojac}(\mathbf{u})$ 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 $\tilde{\Delta t}$.

Pseudo time stepping is not active per default. However, the pseudo time step $\tilde{\Delta t}$ can be chosen individually for each element based on the local CFL number:

$$\tilde{\Delta t} = \text{CFL}_{\text{loc}} \frac{h}{|\mathbf{u}|}$$

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 CFL_{loc} is set to the built-in variable `CFLCMP`. The automatic setting will then suggest a PID regulator as default solver. The PID regulator will start with a small CFL number and increase CFL_{loc} as the solution comes closer to convergence.



For details about the CFL regulator, see [About Pseudo Time Stepping](#).

The default manual expression is

$$\begin{aligned} & 1.3^{\min(\text{niterCMP}, 9)} + \\ & \text{if}(\text{niterCMP} > 20, 9 \cdot 1.3^{\min(\text{niterCMP} - 20, 9)}, 0) + \\ & \text{if}(\text{niterCMP} > 40, 90 \cdot 1.3^{\min(\text{niterCMP} - 40, 9)}, 0) \end{aligned} \quad (13-11)$$

The variable `niterCMP` is the nonlinear iteration number. It is equal to one for the first nonlinear iteration. CFL_{loc} starts at 1.3 and increases with 30% each iteration until it reaches $1.3^9 \approx 10.6$. It remains there until iteration 20 where it starts to increase until it reaches approximately 106. A final increase after iteration 40 then takes it to 1060. [Equation 13-11](#) can for some advanced flows increase CFL_{loc} too slow or too quickly. CFL_{loc} can then be tuned to the specific application.

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. 10, for example). COMSOL uses incremental pressure-correction schemes.

 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}_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:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u}_c^n \cdot \nabla \mathbf{u}^{n+1} = -\nabla p^n + \nabla \cdot \left(\mu (\nabla \mathbf{u}^{n+1} + (\nabla \mathbf{u}^{n+1})^T) - \frac{2}{3} \mu (\nabla \cdot \mathbf{u}^{n+1}) \mathbf{I} \right) + \mathbf{F}$$

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}_c values. At the first order it is discretized it reads:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}_c^n}{\text{timestep}}$$

- 2 Solve Poisson's equation to adjust the pressure:

$$\text{timestep} \Delta(p^{n+1} - p^n) = \frac{\partial p}{\partial t} - \nabla \cdot \rho \mathbf{u}^{n+1} \quad (13-12)$$

- 3 Update the corrected velocity:

$$\mathbf{u}_c^{n+1} = \mathbf{u}^n + \frac{\text{timestep}}{\rho} \nabla(p^{n+1} - p^n)$$



For incompressible flows, the $\frac{2}{3}\mu(\nabla \cdot \mathbf{u}^{n+1})\mathbf{I}$ term in [Equation 13-11](#) and $\frac{\partial p}{\partial t}$ term in [Equation 13-12](#) 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

$$\mathbf{F} = (\rho_0 + \Delta\rho)\mathbf{g} \quad (13-13)$$

where \mathbf{g} is the gravity vector. Enter an expression for [Equation 13-13](#) in the Volume force fields in the Volume Force feature; however, further simplifications are often possible. Because \mathbf{g} can be written in terms of a potential, Φ , it is possible to write [Equation 13-13](#) 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 13-8](#) and [Equation 13-9](#) in terms of the hydrodynamic pressure $P = p + \rho_0\Phi$:

$$\rho\nabla \cdot \mathbf{u} = 0 \quad (13-14)$$

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + (\rho_0 \mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) + \mathbf{g}\Delta\rho \quad (13-15)$$

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 13-14](#) and [Equation 13-15](#) form has its limitations. The main assumption is that the density fluctuations must be small; that is, $\Delta\rho/\rho_0 \ll 1$. 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. 8](#).



See [Volume Force](#) for the node settings.



See [Wall](#) for the node settings. Note that some modules have additional theory sections describing options available with that module.

SLIP

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:

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad (-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = \mathbf{0}$$

The no penetration term will take precedence over the Neumann part of the condition and the above expression is therefore equivalent to

$$\begin{aligned}\mathbf{u} \cdot \mathbf{n} &= 0, \quad \mathbf{K} - (\mathbf{K} \cdot \mathbf{n})\mathbf{n} = \mathbf{0} \\ \mathbf{K} &= \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)\mathbf{n}\end{aligned}$$

expressing that there is no flow across the boundary and no viscous stress in the tangential direction.

If you have the CFD Module or Heat Transfer Module, the boundary conditions for k and ε are



$$\nabla k \cdot \mathbf{n} = 0 \quad \nabla \varepsilon \cdot \mathbf{n} = 0$$

If you have the Microfluidics Module, the boundary condition for v is

$$\tilde{\nabla v} \cdot \mathbf{n} = 0$$

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.



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.



For 2D and 2D axisymmetric models, the velocity is given as a scalar U_w and the condition prescribes



$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{u} \cdot \mathbf{t} = U_w$$

where $\mathbf{t} = (n_y - n_x)$ for 2D and $\mathbf{t} = (n_z - n_r)$ for axial symmetry.

For 3D models, the velocity is set equal to a given vector \mathbf{u}_w projected onto the boundary plane:

$$\mathbf{u} = \frac{\mathbf{u}_w - (\mathbf{n} \cdot \mathbf{u}_w)\mathbf{n}}{\|\mathbf{u}_w - (\mathbf{n} \cdot \mathbf{u}_w)\mathbf{n}\|} \|\mathbf{u}_w\|$$

The normalization makes \mathbf{u} have the same magnitude as \mathbf{u}_w even if \mathbf{u}_w is not exactly parallel to the wall.

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:

$$-p + \mu \frac{\partial u_n}{\partial n} = F_n \quad (13-16)$$

where $\partial u_n / \partial n$ is the normal derivative of the normal velocity field component.

Equation 13-16 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.



[Inlet](#) and [Outlet](#) for the node settings. Note that some modules have additional theory sections describing options available with that module.

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:

$$\left(\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I} \right) \mathbf{n} = \mathbf{0}, \quad p = p_0$$

$$\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \mathbf{n} = \mathbf{0}, \quad p = p_0$$

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 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.



Inlet and Outlet for the node settings. Note that some modules have additional theory sections describing options available with that module.

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

$$p = 2\mu \frac{\partial u_n}{\partial n} + f_0 \quad (13-17)$$

If $\partial u_n / \partial n$ is small, Equation 13-17 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.



Inlet, Outlet, Open Boundary, and Boundary Stress for the individual node settings. Note that some modules have additional theory sections describing options available with that module.

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 Multiphysics® 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. 11](#).

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. 12](#).

Particle Tracing in Fluid Flow



The Particle Tracing Module is available to assist with these types of modeling problems.



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.

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

$$m \frac{d^2 \mathbf{x}}{dt^2} = \mathbf{F}(t, \mathbf{x}, \frac{d\mathbf{x}}{dt})$$

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. 9). That expression is valid for spherical particles for a wide range of particle Reynolds numbers. The Reynolds particle number is defined as

$$\text{Re}_p = \frac{|\mathbf{u} - \mathbf{u}_p| 2r\rho}{\mu}$$

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

$$\mathbf{F} = \pi r^2 \rho |\mathbf{u} - \mathbf{u}_p| (\mathbf{u} - \mathbf{u}_p) [1.84 \text{Re}_p^{-0.31} + 0.293 \text{Re}_p^{0.06}]^{3.45}$$

References for the Single-Phase Flow, Laminar Flow User 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 β 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. D.J. Tritton, *Physical Fluid Dynamics*, 2nd ed., Oxford University Press, 1988.
9. J.M. Coulson and J.F. Richardson, “Particle Technology and Separation Processes,” *Chemical Engineering, Volume 2*, Butterworth-Heinemann, 2002.
10. 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.
11. B. Rivière, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*, SIAM, 2008.
12. 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.

Single-Phase Flow, Laminar Flow User Interface



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.



- [The Laminar Flow User Interface](#)
- [Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow](#)
- [Theory of Laminar Flow](#)

The Laminar Flow User Interface

The **Laminar Flow (spf)** user 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 node is **Fluid Properties**, which adds the Navier-Stokes equations and provides an interface for defining the fluid material and its properties.

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 nodes that implement, for example, boundary conditions and volume forces.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

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

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

These dependent variables (fields) are defined for this interface—**Velocity field \mathbf{u}** (SI unit: m/s) and its components, 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.

CONSISTENT STABILIZATION

To display this section, click the **Show** button () and select **Stabilization**.

The consistent stabilization methods are applicable to the **Navier-Stokes equations**—**Streamline diffusion** and **Crosswind diffusion**. These check boxes are selected by default. If required, click to clear one or both of the **Streamline diffusion** and **Crosswind diffusion** check boxes.

INCONSISTENT STABILIZATION

To display this section, click the **Show** button () and select **Stabilization**. By default, the **Isotropic diffusion** check box is not selected for the **Navier-Stokes equations**. Click to select as required.

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** sets the local CFL number (from the Courant–Friedrichs–Lewy condition) to the built-in variable **CFLCMP** which in turns trigger a PID regulator for the CFL number. If **Manual** is selected, enter a **Local CFL number** CFL_{loc} (dimensionless).

-
- 
 - The Projection Method for the Navier-Stokes Equations
 - Pseudo Time Stepping for Laminar Flow Models and About Pseudo Time Stepping
-

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 interface. 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.
- **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** (the default) or **Complex** for each of the variables in the table.

-
- 
 - Show More Physics Options
 - Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow
 - Theory of Laminar Flow
-

-
- 
- [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**
-

Domain, Boundary, Pair, and Point Nodes for Single-Phase Flow

The following nodes are for all interfaces found under the **Fluid Flow>Single-Phase Flow** branch () in the **Model Wizard**. Other interfaces also share many of these domain, boundary, pair, and point nodes (listed in alphabetical order):

- [Boundary Stress](#)
- [Flow Continuity](#)
- [Fluid Properties](#)
- [Initial Values](#)
- [Inlet](#)
- [Open Boundary](#)
- [Outlet](#)
- [Periodic Flow Condition](#)
- [Pressure Point Constraint](#)
- [Symmetry](#)
- [Volume Force](#)
- [Wall](#)



For 2D axisymmetric models, COMSOL Multiphysics® takes the axial symmetry boundaries (at $r = 0$) into account and adds an **Axial Symmetry** node to the model that is valid on the axial symmetry boundaries only.



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.



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.

Fluid Properties

The **Fluid Properties** node 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.

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

The default **Density** ρ (SI unit: kg/m³) uses the value **From material**. Select **User defined** to enter a different value or expression.

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** node specifies the volume force \mathbf{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 nodes are added to the same domain, then the sum of all contributions are added to the momentum equations.

VOLUME FORCE

Enter the components of the **Volume force \mathbf{F}** (SI unit: N/m³). The defaults for all components are 0 N/m³.



The Boussinesq Approximation

Initial Values

The **Initial Values** node 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

COORDINATE SYSTEM SELECTION

The **Global coordinate system** is selected by default. The **Coordinate system** list contains any additional coordinate systems that the model includes.

INITIAL VALUES

Enter values or expressions for the initial value of the **Velocity field \mathbf{u}** (SI unit: m/s) and the **Pressure p** (SI unit: Pa). The default values are 0 m/s and 0 Pa, respectively.

Wall

The **Wall** node includes a set of boundary conditions describing the fluid flow condition at a wall.

- [No Slip](#) (the default)
- [Slip](#)
- [Sliding Wall](#)

- Moving Wall
- Leaking Wall

-
- | | |
|---|--|
|  | <ul style="list-style-type: none">• Slip• Sliding Wall• Moving Mesh User Interface |
|---|--|
-

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

BOUNDARY CONDITION

Select a **Boundary condition** for the wall.

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.

	For 3D models, enter the components of the Velocity of the sliding wall \mathbf{u}_w (SI unit: m/s). If the velocity vector entered is not in the plane of the wall, COMSOL Multiphysics® projects it onto the tangential direction. Its magnitude is adjusted to be the same as the magnitude of the vector entered.
---	--



For 2D models, 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).

Moving Wall

If the wall moves, so must the fluid. Hence, this boundary condition prescribes $\mathbf{u} = \mathbf{u}_w$. Enter the components of the **Velocity of moving wall** \mathbf{u}_w (SI unit: m/s).



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 $\mathbf{u} = \mathbf{u}_l$. Enter the components of the **Fluid velocity** \mathbf{u}_l (SI unit: m/s).

CONSTRAINT SETTINGS

To display this section, click the **Show** button () and select **Advanced Physics Options**.

For the **No Slip**, **Moving Wall**, and **Leaking Wall** boundary conditions, select an option from the **Apply reaction terms on:** list—**All physics (symmetric)** or **Individual dependent variables**. The other types of wall boundary conditions with constraints use **Individual dependent variables** constraints only.

Select the **Use weak constraints** check box (not available for the **Sliding Wall** condition) 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.



In most cases the inlet boundary conditions are available, 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.



- Theory for the Pressure, No Viscous Stress Boundary Condition
- Theory for the Normal Stress Boundary Condition

BOUNDARY CONDITION

Select a **Boundary condition** for the inlet—**Velocity** (the default), **Pressure**, **No Viscous Stress**, or **Normal Stress**.



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

The **Velocity** boundary condition is available for the **Inlet** and **Outlet** boundary nodes.

- Select **Normal inflow velocity** (the default) to specify a normal inflow velocity magnitude $\mathbf{u} = -\mathbf{n}U_0$ where \mathbf{n} is the boundary normal pointing out of the domain. Enter the velocity magnitude U_0 (SI unit: m/s). The default is 0 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.

PRESSURE, NO VISCOSUS STRESS

The **Pressure, no viscous stress** boundary condition is available for the [Inlet](#) and [Outlet](#) boundary nodes. It specifies vanishing viscous stress along with a Dirichlet condition on the pressure. Enter the **Pressure** p_0 (SI unit: Pa) at the boundary. The default is 0 Pa.



Depending on the pressure field in the rest of the domain, an inlet boundary with this condition can become an outlet boundary.

NORMAL STRESS

The **Normal stress** boundary condition is available for the [Inlet](#), [Outlet](#), [Open Boundary](#), and [Boundary Stress](#) nodes. Enter the magnitude of **Normal stress** f_0 (SI unit: N/m²). This implicitly specifies that $p \approx f_0$. The default is 0 N/m².

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.

When [Velocity](#) or [Pressure, No Viscous Stress](#) are selected as the **Boundary condition**, and to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required.

Outlet

The **Outlet** node 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.



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.

BOUNDARY CONDITION

Select a **Boundary condition** for the outlet—[Pressure, No Viscous Stress](#) (the default), [Velocity, No Viscous Stress](#), [Pressure](#), or [Normal Stress](#).



The **Pressure, no viscous stress, Velocity, and Normal stress** boundary conditions are described for the **Inlet** node.

Pressure

The **Pressure** 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 $\text{Re}^c = \rho|\mathbf{u}|h/(2\mu) \gg 1$ (h is the local mesh element size). It does however work well in most other situations as well.

No Viscous Stress

The **No Viscous Stress** condition specifies vanishing viscous stress on the outlet. This condition does not provide sufficient information to fully specify the flow at the outlet and must at least be combined with pressure constraints on adjacent points.

If **No viscous stress** is selected, 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.



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.

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.

When **Velocity**, **Pressure**, **No Viscous Stress**, or **Pressure** are selected as the **Boundary condition**, and to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required.

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 Dirichlet condition and a Neumann condition:

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{0}, \quad \left(-p\mathbf{I} + \left(\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I} \right) \right) \mathbf{n} = \mathbf{0}$$

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{0}, \quad (-p\mathbf{I} + \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T))\mathbf{n} = \mathbf{0}$$

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:

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= \mathbf{0}, & \mathbf{K} - (\mathbf{K} \cdot \mathbf{n})\mathbf{n} &= \mathbf{0} \\ \mathbf{K} &= \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\mathbf{n} \end{aligned}$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.



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** node that implements this condition on the axial symmetry boundaries only.

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.

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.

BOUNDARY CONDITIONS

Select a **Boundary condition** for the open boundaries—**Normal Stress** (the default) or **No Viscous Stress**. These options are described for the **Inlet** and **Outlet** nodes, respectively.

Boundary Stress

The **Boundary Stress** node adds a boundary condition that represents a very general class of conditions also known as traction boundary conditions.

BOUNDARY CONDITION

Select a **Boundary condition** for the boundary stress—**General stress** (the default), **Normal Stress** (described for the Inlet node), or **Normal stress, normal flow**.

General Stress

When **General stress** is selected, enter the components for the **Stress F** (SI unit: N/m²). 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

$$p = 2\mu \frac{\partial u_n}{\partial n} - \mathbf{n} \cdot \mathbf{F} \quad (13-18)$$

If $\partial u_n / \partial n$ is small, [Equation 13-18](#) 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/m²).

In addition to the stress condition set in the **Normal Stress** condition, the **Normal stress, normal flow** condition also prescribes that there must be no tangential velocities on the boundary:

$$\left(-p\mathbf{I} + \left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathbf{I}\right)\right)\mathbf{n} = -f_0\mathbf{n}, \quad \mathbf{t} \cdot \mathbf{u} = 0$$

$$(-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T))\mathbf{n} = -f_0\mathbf{n}, \quad \mathbf{t} \cdot \mathbf{u} = 0$$

using the compressible and the incompressible formulation respectively.

This boundary condition also implicitly sets a constraint on the pressure that for 2D flows is

$$p = 2\mu \frac{\partial u_n}{\partial n} + f_0 \quad (13-19)$$

If $\partial u_n / \partial n$ is small, [Equation 13-19](#) states that $p \approx f_0$.

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.

If **Normal Stress, Normal Flow** is selected as the **Boundary condition**, then to **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms 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.



If the boundaries are curved, it is recommended to only include two boundaries.



No input is required when **Compressible flow (Ma<0.3)** is selected as the **Compressibility** under the **Physical Model** section for the interface. 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.

PRESSURE DIFFERENCE

When **Incompressible flow** is selected as the **Compressibility** under the **Physical Model** section for the interface, this section is available.

Enter a value or expression for the pressure difference, $p_{\text{src}} - p_{\text{dst}}$ (SI unit: Pa). This pressure difference can, for example, drive the flow in a fully developed channel flow. The default is 0 Pa.



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 the **Use weak constraints** check box to use weak constraints and create dependent variables for the corresponding Lagrange multipliers.



[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 and it 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 subnodes.



- Continuity on Interior Boundaries
 - Identity and Contact Pairs
-

Pressure Point Constraint

The **Pressure Point Constraint** node 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.

P R E S S U R E C O N S T R A I N T

Enter a point constraint for the **Pressure** p_0 (SI unit: Pa). The default is 0 Pa.

C O N S T R A I N T S E T T I N G S

To display this section, click the **Show** button () and select **Advanced Physics Options**. To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**. Or select **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Heat Transfer Modeling

This chapter describes the different types of Heat Transfer user 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**.

Heat Transfer Theory

The Heat Transfer User Interface theory is described in this section. This section reviews the theory about the heat transfer equations in COMSOL Multiphysics® and heat transfer in general. For more detailed discussions of the fundamentals of heat transfer, see [Ref. 1](#) and [Ref. 3](#).

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:

$$\rho C_p \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = -(\nabla \cdot \mathbf{q}) + \boldsymbol{\tau} : \mathbf{S} - \frac{T}{\rho} \frac{\partial p}{\partial T} \Big|_p \left(\frac{\partial p}{\partial t} + (\mathbf{u} \cdot \nabla) p \right) + Q \quad (14-1)$$

where

- ρ is the density (SI unit: kg/m³)
- 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/m²)
- p is pressure (SI unit: Pa)
- $\boldsymbol{\tau}$ 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/m³)

For a detailed discussion of the fundamentals of heat transfer, see [Ref. 1](#).



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 14-1](#), 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, \mathbf{q} , is proportional to the temperature gradient:

$$q_i = -k \frac{\partial T}{\partial x_i} \quad (14-2)$$

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

$$q_i = -\sum_j k_{ij} \frac{\partial T}{\partial x_j}$$



Fourier's law expect that the thermal conductivity tensor is symmetric. Non symmetric tensor leads to unphysical results.

The second term on the right of [Equation 14-1](#) 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 14-2](#) into [Equation 14-1](#), 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 in Fluids physics 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 on 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 that are similar but not identical.

This section briefly describes the theory for the variables for [Total Energy Flux](#) and [Total Heat 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)

$$\rho \mathbf{u}(H_0 + \Psi) - k \nabla T + \tau \cdot \mathbf{u} + q_r \quad (14-3)$$

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 14-3](#) τ is the viscous stress tensor and q_r is the radiative heat flux. Ψ in [Equation 14-3](#) 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

$$\rho \mathbf{u} \left(H + \frac{1}{2}(\mathbf{u} \cdot \mathbf{u}) \right) - k \nabla T + \tau \cdot \mathbf{u} + q_r \quad (14-4)$$

For a simple compressible fluid, the enthalpy, H , has the form ([Ref. 5](#))

$$H = H_{\text{ref}} + \int_{T_{\text{ref}}}^T C_p dT + \int_{p_{\text{ref}}}^p \frac{1}{\rho} \left(1 + \frac{T}{\rho} \left(\frac{\partial \rho}{\partial T} \right) \right) dp \quad (14-5)$$

where p is the absolute pressure. The reference enthalpy, H_{ref} , is the enthalpy at reference temperature, T_{ref} , and reference pressure, p_{ref} . 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: $H_{\text{ref}} = C_{p,\text{ref}} T_{\text{ref}}$
- Gas/liquid: $H_{\text{ref}} = (C_{p,\text{ref}}/\gamma_{\text{ref}})T_{\text{ref}} + p_{\text{ref}}/\rho_{\text{ref}}$

where the subscript “ref” indicates that the property is evaluated at the reference state.

The two integrals in [Equation 14-5](#) are sometimes referred to as the *sensible enthalpy* ([Ref. 7](#)). These are evaluated 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.



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 depends on the pressure, that dependency must be prescribed either via model input or by using the variable pA , which is the variable for the absolute pressure.

TOTAL HEAT FLUX

The total heat flux vector is defined as ([Ref. 6](#)):

$$\rho \mathbf{u} U - k \nabla T + q_r \quad (14-6)$$

where U is the internal energy. It is related to the enthalpy via

$$H = U + \frac{P}{\rho} \quad (14-7)$$

What is the difference between [Equation 14-4](#) and [Equation 14-7](#)? 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 is a pressure drop along the channel that drives the flow. Since there is no viscous heating and the walls are isolated, [Equation 14-5](#) gives that $H_{\text{in}} > H_{\text{out}}$. Since everything else is constant, [Equation 14-4](#) shows that the energy flux into the channel is higher than the energy flux out of the channel. On the other hand $U_{\text{in}} = U_{\text{out}}$, 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 $H_{\text{in}} = H_{\text{out}}$ (first law of thermodynamics) and $U_{\text{in}} < U_{\text{out}}$ (since work has been converted to heat).

Heat Flux and Heat Source Variables

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 14-1: HEAT FLUX VARIABLES

VARIABLE	NAME	GEOMETRIC ENTITY LEVEL
<code>tflux</code>	Total Heat Flux	Domains, boundaries
<code>dflux</code>	Conductive Heat Flux	Domains, boundaries
<code>trlflux</code>	Translational Heat Flux	Domains, boundaries
<code>teflux</code>	Total Energy Flux	Domains, boundaries
<code>ntflux</code>	Total Normal Heat Flux	Boundaries
<code>ndflux</code>	Normal Conductive Heat Flux	Boundaries
<code>naflux</code>	Normal Convective Heat Flux	Boundaries
<code>ntrflux</code>	Normal Translational Heat Flux	Boundaries
<code>nteflux</code>	Normal Total Energy Flux	Boundaries
<code>Qtot</code>	Domain Heat Sources	Domains
<code>Qbtot</code>	Boundary Heat Sources	Boundaries
<code>Q1</code>	Line heat source (Line and Point Heat Sources)	Edges
<code>Qp</code>	Point heat source (Line and Point Heat Sources)	Points

DOMAIN HEAT FLUXES

On domains the heat fluxes are vector quantities. Their definition can vary depending on the active physics nodes and selected properties.

Total Heat Flux

On domains the total heat flux, `tflux`, corresponds to the conductive and convective heat flux.

For solid domains, for example heat transfer in solids and biological tissue domains, the total heat flux is defined by:

$$\text{tflux} = \text{trlflux} + \text{dflux}$$

For fluid domains (for example, heat transfer in fluids), the total heat flux is defined by:

$$\text{tflux} = \text{aflux} + \text{dflux}$$

Conductive Heat Flux

The conductive heat flux variable, dflux , is evaluated using the temperature gradient and the effective thermal conductivity:

$$\text{dflux} = -k_{\text{eff}} \nabla T$$

In the general case k_{eff} is the thermal conductivity, k .

For heat transfer in porous media, $k_{\text{eff}} = k_{\text{eq}}$, where k_{eq} is the equivalent conductivity defined in the Heat Transfer in Porous Media feature.



The Heat Transfer in Porous Media feature requires one of the following products: Batteries & Fuel Cells Module, CFD Module, Chemical Reaction Engineering Module, Corrosion Module, Electrochemistry Module, Electrodeposition Module, Heat Transfer Module, or Subsurface Flow Module.

Translational Heat Flux

Similar to convective heat flux but defined for solid domains with translation. The variable name is trflux .

Total Energy Flux

The total energy flux, teflux , is defined when viscous heating is enabled:

$$\text{teflux} = \rho \mathbf{u} H_0 + \text{dflux} + \tau \cdot \mathbf{u}$$

where the total enthalpy, H_0 , is defined as

$$H_0 = H + \frac{\mathbf{u} \cdot \mathbf{u}}{2}$$

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:

$$\text{ntflux} = \text{mean}(\text{tflux}) \cdot \mathbf{n}$$

Normal Conductive Heat Flux

The variable `ndflux` is defined by:

$$\text{ndflux} = \text{mean(dflux)} \cdot \mathbf{n}$$

Normal Convective Heat Flux

The variable `naflux` is defined by:

$$\text{naflux} = \text{mean(aflux)} \cdot \mathbf{n}$$

Normal Translational Heat Flux

The variable `ntrflux` is defined by:

$$\text{ntrflux} = \text{mean(trlflux)} \cdot \mathbf{n}$$

Normal Total Energy Flux

The variable `nteflux` is defined by:

$$\text{nteflux} = \text{mean(teflux)} \cdot \mathbf{n}$$

DOMAIN HEAT SOURCES

The sum of the domain heat sources added by different physics features are available in one variable, Q_{tot} (SI unit: W/m³). This variable $Qtot$ is the sum of:

- Q which is the heat source added by [Heat Source](#)(described for the Heat Transfer interface and [Electromagnetic Heat Source](#) (described for the Joule Heating interface) feature.

BOUNDARY HEAT SOURCES

The sum of the boundary heat sources added by different boundary conditions is available in one variable, $Q_{b,\text{tot}}$ (SI unit: W/m²). This variable $Qbtot$ is the sum of:

- Q_b which is the boundary heat source added by the [Boundary Heat Source](#) boundary condition.
- Q_{sh} which is the boundary heat source added by the [Boundary Electromagnetic Heat Source](#) boundary condition (described for the Joule Heating interface).

LINE AND POINT HEAT SOURCES

The sum of the line heat sources is available in a variable called `Q1` (SI unit: W/m).

The sum of the point heat sources is available in a variable called Q_p (SI unit: W).



In 2D axisymmetric models, the SI unit for the variable Q_p is W/m.

About the Boundary Conditions for the Heat Transfer User Interfaces

TEMPERATURE AND HEAT FLUX BOUNDARY CONDITIONS

The heat equation accepts two basic types of boundary conditions: specified temperature and specified heat flux. The specified temperature is of a constraint type and prescribes the temperature at a boundary:

$$T = T_0 \quad \text{on } \partial\Omega$$

while the latter specifies the inward heat flux

$$-\mathbf{n} \cdot \mathbf{q} = q_0 \quad \text{on } \partial\Omega$$

where

- \mathbf{q} is the conductive heat flux vector (SI unit: W/m²) where $\mathbf{q} = -k\nabla T$.
- \mathbf{n} is the normal vector of the boundary.
- q_0 is inward heat flux (SI unit: W/m²), 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 $q_0 = h \cdot (T_{\text{inf}} - T)$, 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.



Convective heat flux requires either the Heat Transfer Module or the CFD Module.

OVERRIDING MECHANISM FOR HEAT TRANSFER BOUNDARY CONDITIONS



This section includes information for features that may require additional modules.

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 14-2](#) 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 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, Thermal Contact

TABLE 14-2: 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 14-2](#) to determine if *A* is overridden by *B* or not:

- Locate the line that corresponds to the *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*.



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



Surface-to-Surface radiation requires the Heat Transfer Module.

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



Convective Heat Flux requires either the Heat Transfer Module or the CFD Module.

Consider a boundary where **Convective Heat Flux** is applied. Then a **Symmetry** boundary condition is applied on the same boundary afterward.

- **Convective Heat Flux** 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 Heat Flux** is overridden by **Symmetry**.



In Example 2 above, if **Symmetry** followed by **Convective Heat Flux** 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

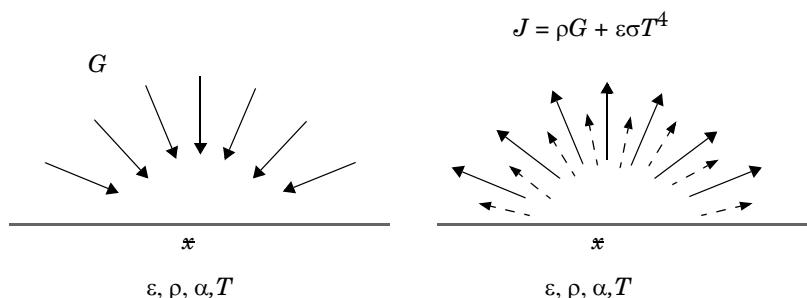


Figure 14-1: Arriving irradiation (left), leaving radiosity (right).

Consider Figure 14-1. 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:

$$J = \rho G + \varepsilon \sigma T^4 \quad (14-8)$$

The net inward radiative heat flux, q , is then given the difference between the irradiation and the radiosity:

$$q = G - J \quad (14-9)$$

Using [Equation 14-8](#) and [Equation 14-9](#) 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 :

$$q = (1 - \rho)G - \varepsilon\sigma T^4 \quad (14-10)$$

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:

$$\alpha = \varepsilon = 1 - \rho \quad (14-11)$$

Thus, for ideal gray bodies, q is given by:

$$q = \varepsilon(G - \sigma T^4) \quad (14-12)$$

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 14-12](#) holds for both radiation types, but the irradiation term, G , is different for each of them. The Heat Transfer interface supports radiation.



Surface-to-surface radiation requires the Heat Transfer Module.

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

$$G = \sigma T_{\text{amb}}^4 \quad (14-13)$$

Inserting [Equation 14-13](#) into [Equation 14-12](#) results in the net inward heat flux for surface-to-ambient radiation

$$q = \varepsilon \sigma (T_{\text{amb}}^4 - T^4) \quad (14-14)$$

For boundaries where a surface-to-ambient radiation is specified, COMSOL Multiphysics adds this term to the right-hand side of [Equation 14-14](#).

Consistent and Inconsistent Stabilization Methods for the Heat Transfer User Interfaces

The different versions of the Heat Transfer interface have this advanced option to set the stabilization method parameters. This section provides information pertaining to these options.

To display this section, click the **Show** button () and select **Stabilization**.

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

Streamline diffusion introduces artificial diffusion in the streamline direction. This is often enough to obtain a smooth numerical solution provided that the exact solution of the heat equation does not contain any discontinuities. At sharp gradients, however, undershoots and overshoots can occur in the numerical solution. Crosswind diffusion addresses these spurious oscillations by adding diffusion orthogonal to the streamline direction—that is, in the crosswind direction.

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.



- [Stabilization Techniques](#)
- [Show Stabilization](#)

References for the Heat Transfer User Interfaces

1. F.P. Incropera, D.P. DeWitt, T.L. Bergman and A.S. Lavine, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, Sixth edition, 2006.
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.
8. W. Wagner, and H-J Kretzschmar, *International Steam Tables*, 2nd ed., Springer, 2008.

The Heat Transfer User Interface

The Heat Transfer user interfaces model heat transfer by conduction and convection. Surface-to-ambient radiation effects around edges and boundaries can also be included. 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 .

The Heat Transfer user interfaces include the equations, boundary conditions, and sources for modeling conductive and convective heat transfer and solving for the temperature.

After selecting a version of the physics user interface in the **Model Wizard**, default nodes are added under the main node. For example:

- If **Heat Transfer in Solids** () is selected, a **Heat Transfer in Solids (ht)** node is added with a default **Heat Transfer in Solids** model as a subnode.
- If **Heat Transfer in Fluids** () is selected, a **Heat Transfer in Fluids (ht)** node is added with a default **Heat Transfer in Fluids** model as a subnode.

The benefit of the different versions of the Heat Transfer user interfaces, with **ht** as the common default identifier, is that it is easy to add the default settings when selecting the interface from the Model Wizard. At any time, right-click the parent node to add a **Heat Transfer in Fluids** or **Heat Transfer in Solids** node—the functionality is always available.



The **Joule Heating (jh)** () multiphysics interface is also available with the basic COMSOL Multiphysics license. See [The Joule Heating User Interface](#) for information.

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 used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern

`<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `ht`.

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

This section does not require any input.

CONSISTENT 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.

INCONSISTENT STABILIZATION

To display this section, click the **Show** button () and select **Stabilization**. The **Isotopic diffusion** check box is not selected by default.

DISCRETIZATION

To display this section, click the **Show** button () and select **Discretization**.

- Select a **Temperature—Quadratic** (the default), **Linear**, **Cubic**, **Quartic**, or **Quintic**.
- The **Compute boundary fluxes** check box is selected by default.
- The **Apply smoothing to boundary fluxes** check box is selected by default.
- In the table, specify the **Value type when using splitting of complex variables**—**Real** (the default) or **Complex**.

DEPENDENT VARIABLES

The Heat Transfer user interfaces have 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

Add both a **Heat Transfer (ht)** and **Moving Mesh (ale)** interface (found under the **Mathematics>Deformed Mesh** branch in the Model Wizard) then click the **Show** button () and select **Advanced Physics Options** to display this section.

When the model contains moving mesh, the **Enable conversions between material and spatial frame** check box is selected by default.

This option has no effect when the model does not contain a moving frame since the material and spatial frames are identical in this case. With moving mesh, and when this option is active, the heat transfer features automatically account for deformation effects on heat transfer properties. In particular the effects for volume changes on the density are considered. Rotation effects on thermal conductivity of an anisotropic material and, more generally, deformation effects on arbitrary thermal conductivity, are also covered. When the **Enable conversions between material and spatial frame** check box is not selected, the feature inputs (for example, **Heat Source**, **Heat Flux**, **Boundary Heat Source**, and **Line Heat Source**) are not converted and all are defined on the **Spatial** frame.

-
- 
 - [About Frames in Heat Transfer](#)
 - [Show More Physics Options](#)
 - [Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer User Interfaces](#)
 - [Consistent and Inconsistent Stabilization Methods for the Heat Transfer User Interfaces](#)
 - [Heat Transfer Theory](#)
 - [Show Stabilization](#)
-

Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer User Interfaces

The **Heat Transfer User Interface** has these domain, boundary, edge, point, and pair nodes and subnodes available (listed in alphabetical order).



To locate and search all the documentation, in COMSOL Multiphysics, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

- [Boundary Heat Source](#)
- [Continuity](#)
- [Heat Flux](#)
- [Heat Source](#)
- [Heat Transfer in Fluids](#)
- [Heat Transfer in Solids](#)
- [Initial Values](#)
- [Line Heat Source](#)
- [Outflow](#)
- [Periodic Heat Condition](#)
- [Point Heat Source](#)
- [Surface-to-Ambient Radiation](#)
- [Symmetry](#)
- [Temperature](#)
- [Thermal Insulation](#) (the default boundary condition)
- [Thin Thermally Resistive Layer](#)
- [Translational Motion](#)



If you also have the Heat Transfer Module, there are several other feature nodes available and described in the *Heat Transfer Module User's Guide*.



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.

Heat Transfer in Solids

The **Heat Transfer in Solids** node uses the heat equation version in [Equation 14-15](#) as the mathematical model for heat transfer in solids:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \quad (14-15)$$

For a steady-state problem the temperature does not change with time and the first term disappears. The equation includes the following material properties: the density ρ , the heat capacity C_p , and the thermal conductivity k (a scalar or a tensor if the thermal conductivity is anisotropic), and a heat source (or sink) Q —one or more heat sources can be added separately.

When parts of the model are moving in the material frame, right-click the **Heat Transfer in Solids** node to add a **Translational Motion** node to take this into account.

	<p>If you have the CFD Module, also right-click to add a Pressure Work node.</p> <p>If you have the Heat Transfer Module, also right-click to add Pressure Work or Opaque nodes. The Opaque subnode is automatically added to the entire selection when Surface-to-surface radiation is activated. The selection can be edited.</p>
---	--

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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, SOLID

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, representing an anisotropic thermal conductivity) are available as `ht.kxx`, `ht.kyy`, and so on (using the default 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} .

 Fourier's law expect that the thermal conductivity tensor is symmetric. A non symmetric tensor can lead to unphysical results.

THERMODYNAMICS; SOLID

The default **Density** ρ (SI unit: kg/m³) 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.

Thermal Diffusivity

In addition, the thermal diffusivity α is defined as $k/(\rho C_p)$ (SI unit: m²/s) is also a predefined quantity. The thermal diffusivity can be interpreted as a measure of thermal inertia (heat propagates slowly where the thermal diffusivity is low, for example). The components of a thermal diffusivity α in the case that it is a tensor (α_{xx} , α_{yy} , and so on, representing an anisotropic thermal diffusivity) are available as `ht.alphaTdx`, `ht.alphaTdy`, and so on (using the default interface identifier `ht`). The single scalar mean thermal diffusivity `ht.alphaTdMean` is the mean value of the diagonal elements

α_{xx} , α_{yy} , and α_{zz} . The denominator ρC_p is the effective volumetric heat capacity, and is also available as a predefined quantity, `ht.C_eff`.

-
- 
 - [Axisymmetric Transient Heat Transfer](#): Model Library path `COMSOL_Multiphysics/Heat_Transfer/heat_transient_axi`
 - [2D Heat Transfer Benchmark with Convective Cooling](#): Model Library path `COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d`
-

Translational Motion

Right-click the [Heat Transfer in Solids](#) node to add the **Translational Motion** node, which provides movement by translation to model heat transfer in solids. It adds the following contribution to the right-hand side of [Equation 14-15](#), defined in the parent node:

$$-\rho C_p \mathbf{u} \cdot \nabla T$$

The contribution describes the effect of a moving coordinate system that is required to model, for example, a moving heat source.



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 define.



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** $\mathbf{u}_{\text{trans}}$ (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:

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + Q \quad (14-16)$$

For a steady-state problem the temperature does not change with time and the first term disappears. This equation includes the following material properties, fields, and sources:

- **Density** ρ (SI unit: kg/m³)
- **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K))—describes the amount of heat energy required to produce a unit temperature change in a unit mass.
- **Thermal conductivity** k (SI unit: W/(m·K))—a scalar or a tensor if the thermal conductivity is anisotropic.
- **Velocity field** \mathbf{u} (SI unit: m/s)—either 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.
- The **Ratio of specific heats** γ (dimensionless)—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.



Heat Transfer by Free Convection: Model Library path
[COMSOL_Multiphysics/Multiphysics/free_convection](#)

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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

This section controls both the variable as well as any property value (reference pressures) used when solving for pressure. There are usually two ways to calculate 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 option 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 default **Absolute pressure** p_A (SI unit: Pa) is **User defined** and is 1 atm (101,325 Pa). When additional physics interfaces are added to the model, the pressure variables solved can also be selected from the list. For example, if a fluid-flow interface is added you can select **Pressure (spf/fp)** from the list.

When a **Pressure** variable is selected, the **Reference pressure** check box is selected by default and the default value of p_{ref} is 1 [atm] (1 atmosphere).



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 work with physics interfaces that it is being coupled to. In such models, check the coupling between any interfaces using the same variable.

Velocity Field

The default **Velocity field u** (SI unit: m/s) is **User defined**. When **User defined** is selected, enter values or expressions for the components based on space dimension. The defaults are 0 m/s. Or select an existing velocity field in the model (for example, **Velocity field (spf/fpl)** from a Laminar Flow 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 (except boundary coordinate systems). The coordinate system is used for interpreting directions of orthotropic and anisotropic thermal conductivity.

HEAT CONDUCTION, FLUID

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 **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, FLUID

The only available **Fluid type** is **Gas/Liquid**.

The default **Density** ρ (SI unit: kg/m³), **Heat capacity at constant pressure** C_p (SI unit: J/(kg·K)), and **Ratio of specific heats** γ (dimensionless) for a general gas or liquid use values **From material**. Select **User defined** to enter other values or expressions.

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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).

Heat Source

The **Heat Source** describes heat generation within the domain. You 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 a linear heat source, or as a total heat source (power).

DOMAIN SELECTION

From the **Selection** list, choose the domains to add the heat source to.

HEAT SOURCE

Click the **General source** (the default), **Linear source**, or **Total power** button.

- If **General source** is selected, enter a value for the distributed heat source Q (SI unit: W/m³) when the default option, **User defined**, is selected. The default is 0 W/m³(that is, no heat source). See also [Additional General Source Options](#).

- If **Linear source** ($Q = q_s \cdot T$) is selected, enter the **Production/absorption coefficient** q_s (SI unit: W/(m³·K)). The default is 0 W/(m³·K).
 - If **Total power** is selected, enter the total heat source, P_{tot} , (SI unit: W). The default is 0 W. In this case $Q = P_{\text{tot}}/V$, where V is the total volume of the selected domains.
-



In 3D and 2D axial symmetry, $V = \int 1$.



In 2D and 1D axial symmetry:



$$V = dz \int 1$$



where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available to define dz .

In 1D:



$$V = A_c \int 1$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available to define A_c .



The advantage of writing the source in this second 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.

Additional General Source Options

For the general heat source Q , there are predefined heat sources available (in addition to a **User defined** heat source) when simulating heat transfer together with electrical or

electromagnetic physics user interfaces. Such sources represent, for example, ohmic heating and induction heating.



The following options are also available from the **General source** list above but require additional interfaces and/or licenses as indicated.

- With the addition of an Electric Currents physics interface, the **Total power dissipation density (ec/cucn1)** heat source is available from the General source list.
- With the addition of any version of the Electromagnetic Waves user interface (which requires the RF Module), the **Total power dissipation density (emw/weel)** and **Electromagnetic power loss density (emw/weel)** heat sources are available from the General source list.
- With the addition of a Magnetic Fields user interface (a 3D model requires the AC/DC Module), the **Electromagnetic heating (mfi/all)** heat source is available from the General source list.
- With the addition of a Magnetic and Electric Fields user interface (which requires the AC/DC Module), the **Electromagnetic heating (mef/alcl)** heat source is available from the General source list.
- For the Heat Transfer in Porous Media user interface, with the addition of physics interfaces from the Batteries & Fuel Cells Module, Corrosion Module, or Electrodeposition Module, heat sources from the electrochemical current distribution interfaces are available.

FRAME SELECTION

To display this section, add both a **Heat Transfer (ht)** and a **Moving Mesh (ale)** user interface (found under the **Mathematics>Deformed Mesh** branch in the Model Wizard). Then click the **Show** button () and select **Advanced Physics Options**.

When the model contains a moving mesh, the **Enable conversions between material and spatial frame** check box is selected by default on the Heat Transfer interface, which in turn enables this section. Use **Frame Selection** to select the frame where the input variables are defined. If **Spatial** is selected, the variables take their values from the edit

fields. If **Material** (the default) is selected, a conversion from the material to the spatial frame is applied to the edit field values.

-
- 
- [About Frames in Heat Transfer](#)
 - [The Heat Transfer User Interface](#)
 - [Stabilization Techniques](#)
-

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

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 define.

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. To **Apply reaction terms** on all dependent variables, select **All physics (symmetric)**. Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required.
- Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.
- Select the **Discontinuous Galerkin constraints** button when **Classic constraints** do not work satisfactorily.

 The **Discontinuous Galerkin constraints** 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.

 [Show More Physics Options](#)

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



In most cases, the **Outflow** node does not 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 a [Thermal Insulation](#) condition, and it means that there is no heat flux across the boundary.

BOUNDARY SELECTION



In most cases, the 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 define.

PAIR SELECTION

If this node 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 to select the **General inward heat flux** (the default), **Inward heat flux**, or **Total heat flux** button.

General Inward Heat Flux

If **General inward heat flux** q_0 (SI unit: W/m²) 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. The default is 0 W/m².

Inward Heat Flux

If **Inward heat flux** is selected, enter the **Heat transfer coefficient** h (SI unit: W/(m².K)). The default is 0 W/(m².K). 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 $q_0 = h(T_{\text{ext}} - T)$.



For a thorough introduction about how to calculate heat transfer coefficients, see Incropera and DeWitt in [Ref. 1](#).

Total Heat Flux

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. The default is 0 W. In this case $q_0 = q_{\text{tot}}/A$, where A is the total area of the selected boundaries.



In 3D and 2D axial symmetry, $A = \int 1$.



In 2D and 1D axial symmetry:



$$A = dz \int 1$$



where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available to define dz .

In 1D:


$$A = A_c \int l$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available to define A_c .

FRAME SELECTION

The settings are the same for the [Heat Source](#) node and described under the **Frame Selection** section.

- 
- [About Frames in Heat Transfer](#)
 - [The Heat Transfer User Interface](#)
-

Surface-to-Ambient Radiation

Use the **Surface-to-Ambient Radiation** 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 define.

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.

SURFACE-TO-AMBIENT RADIATION

The default **Surface emissivity** ε (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** to add a periodic heat condition to boundaries. Right-click to add a **Destination Selection** node.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.



- Periodic Condition and Destination Selection
- Periodic Boundary Conditions

Boundary Heat Source

The **Boundary Heat Source** models a heat source (or heat sink) that is embedded in the boundary. When selected as a **Pair Boundary Heat Source**, it also prescribes that the temperature field is continuous across the pair.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

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** (the default) or **Total boundary power** button.

- If **General source** is selected, enter a value for the boundary heat source Q_b (SI unit: W/m²) when the default option, **User defined**, is selected. A positive Q_b is heating and a negative Q_b is cooling. The default is 0 W/m². For the general boundary heat source Q_b , there are predefined heat sources available when simulating heat transfer

together with electrical or electromagnetic physics user interfaces. Such sources represent, for example, ohmic heating and induction heating.

- If **Total boundary power** is selected, enter the total power (total heat source) $P_{b,tot}$ (SI unit: W). The default is 0 W. In this case $Q_b = P_{b,tot}/A$, where A is the total area of the selected boundaries.
-



In 3D and 2D axial symmetry, $A = \int 1$.



In 2D and 1D axial symmetry:



$$A = dz \int 1$$



where dz is the out-of-plane thickness. If the out-of-plane property is not active, a text field is available to define dz .

In 1D:


$$A = A_c \int 1$$

where A_c is the cross-sectional area. If the out-of-plane property is not active, a text field is available to define A_c .

FRAME SELECTION

The settings are the same for the [Heat Source](#) node and described under the **Frame Selection** section.



- [About Frames in Heat Transfer](#)
 - [The Heat Transfer User Interface](#)
-

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 this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.



- Continuity on Interior Boundaries
 - Identity and Contact Pairs
-

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. It can be added to pairs by selecting **Pair Thin Thermally Resistive Layer** from the **Pairs** menu. The resistive material can also be defined through the **Thermal Resistance**:

$$R_s = \frac{d_s}{k_s}$$

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 the u and d subscripts refer to the upside and the downside of the slit, respectively.



When using the **Pair Thin Thermally Resistive Layer** node, then the u and d subscripts refer to the upside and the downside of the pair, respectively, instead of the slit.



Like any pair feature, the **Pair Thin Thermally Resistive Layer** condition contributes with any other pair feature. However, do not use two conditions on the same pair. In order to model a thin resistive layer made of several materials, use the **Multiple layers** option, which is available with the Heat Transfer Module.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

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.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

THIN THERMALLY RESISTIVE LAYER

Select **Layer properties** (the default) or **Thermal resistance** from the **Specify** list.

- If **Layer properties** is selected, enter a value or expression for the **Layer thickness d_s** (SI unit: m). The default is 0.005 m. The default **Thermal conductivity k_s** (SI unit: W/(m·K)) is taken **From material**. Select **User defined** to enter another value or expression. The default is 0.01 W/(m·K).
- If **Thermal resistance** is selected, enter a value or expression for the **Thermal resistance R_s** (SI unit: $s^3 \cdot K/kg$). The default is $s^3 \cdot K/kg$.

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 node from the **Edges** submenu.



The **Line Heat Source** node is only available in 3D. This is because in 2D it is a boundary and in 1D it is a domain.

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 Multiphysics 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 define.

LINE HEAT SOURCE

Click the **General source** (the default) or **Total line power** button.

- When **General source** is selected, enter a value for the distributed heat source, Q_l (SI unit: W/m) in unit power per unit length. Positive Q_l is heating while a negative Q_l is cooling. The default is 0 W/m.
- If **Total line power** is selected, enter the total power (total heat source) $P_{l,tot}$ (SI unit: W). The default is 0 W.

FRAME SELECTION

The settings are the same for the **Heat Source** node and described under the **Frame Selection** section.



- [About Frames in Heat Transfer](#)
- [The Heat Transfer User Interface](#)

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 node from the **Points** menu.



The **Point Heat Source** is available in 3D and 2D. In 1D it is not available because points are boundaries there (possibly interior boundaries).



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 Multiphysics returns a finite value, but that value must be interpreted in a weak sense.

POINT SELECTION

From the **Selection** list, choose the points to define.

POINT HEAT SOURCE

Enter the **Point heat source** Q_p (SI unit: W) in unit power. Positive Q_p is heating while a negative Q_p is cooling. The default is 0 W.

The Joule Heating User Interface

The **Joule Heating (jh)** user interface () , found under the **Heat Transfer>Electromagnetic Heating** branch () in the **Model Wizard**, combines all functionality from the Electric Currents and Heat Transfer user interfaces for modeling *Joule heating (resistive heating or ohmic heating)*. This multiphysics user 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: $1/(\rho_0(1 + \alpha(T - T_{\text{ref}})))$, 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.

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 nodes that implement boundary conditions and sources, for example.



Except where described in this section, the Joule Heating multiphysics user interface shares its functionality with [The Electric Currents User Interface](#) and [The Heat Transfer User Interface](#).

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique.

Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

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 include Joule heating. 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, which are described in the *Heat Transfer Module User's Guide*.

- To include heat transfer out of the plane, select the **Out-of-plane heat transfer** check box (2D models only).
- To include surface-to-surface radiation as part of the heat transfer and to add a **Radiation Settings** section, select the **Surface-to-surface radiation** check box.
- To include radiation in participating media as part of the heat transfer and to add a **Participating media Settings** section, select the **Radiation in participating media** check box.
- To enable the **Biological Tissue** node, select the **Heat Transfer in biological tissue** check box.

In addition, 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.

RADIATION SETTINGS



This section is available when the **Surface-to-surface radiation** check box is selected. See **Radiation Settings** as described for [The Heat Transfer User Interface](#).

PARTICIPATING MEDIA SETTINGS



This section is available when the **Radiation in participating media** check box is selected. See **Participating Media Settings** as described for [The Heat Transfer User Interface](#).

OUT-OF-PLANE THICKNESS



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: Ω). The default value is 50 Ω .

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.

ADVANCED

Click the **Show** button () and select **Advanced Physics Options** to display this section. The **Enable conversions between material and spatial frame** check box is selected by default.

DISCRETIZATION

To display this section, click the **Show** button () and select **Discretization**.

Select **Linear**, **Quadratic**, **Cubic**, or **Quartic** for the **Temperature**, **Surface radiosity**, and **Electric potential**. Specify the **Value type when using splitting of complex variables**—**Real** or **Complex**.



- Show More Physics Options
- Domain, Boundary, Edge, Point, and Pair Nodes for Joule Heating



Joule Heating of a Microactuator—Distributed Parameter Version:
Model Library path **COMSOL_Multiphysics/Tutorial_Models/**
thermal_actuator_jh_distributed

Domain, Boundary, Edge, Point, and Pair Nodes for Joule Heating

Because **The Joule Heating User Interface** is a multiphysics user interface, most nodes are shared with, and described for, other physics user interfaces. Below are links to the domain, boundary, edge, pair, and point nodes, listed in alphabetical order.



To locate and search all the documentation, in COMSOL Multiphysics, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

These features are described in this section:

- Boundary Electromagnetic Heat Source

- Electromagnetic Heat Source
- Initial Values
- Joule Heating Model

These nodes are described for the Heat Transfer interface:

- 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



If you also have the Heat Transfer Module, there are several other feature nodes available and described in the *Heat Transfer Module User's Guide*.

These nodes are described for the Electric Currents, Electrostatics, or another electromagnetics user 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



If you also have the AC/DC Module, the **Electrical Contact** node (and other specialized features) is available and described in the *AC/DC Module User's Guide*.

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:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \quad (14-17)$$

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. Also right-click to add a [Translational Motion](#) subnode.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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:

$$1/(\rho_0(1 + \alpha(T - T_{\text{ref}})))$$

where T is the dependent variable for temperature. By default, the **Reference temperature** T_{ref} (SI unit: K), **Reference resistivity** ρ_0 (SI unit: $\Omega \cdot \text{m}$), and **Resistivity temperature coefficient** α (SI unit: $1/\text{K}$) 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 displays under the list)—**Relative permittivity**, **Polarization**, or **Remanent electric displacement**. Select:

- Select **Relative permittivity** to use the constitutive relation $\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E}$ (the default). Then specify ϵ_r (dimensionless). The default is to take its value **From material**, or select **User defined** to specify a different value or expression.
- Select **Polarization** to use the constitutive relation $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$. Then specify \mathbf{P} , the polarization vector (SI unit: C/m^2). Enter its components based on space dimension in the fields in the **Polarization** table.
- Select **Remanent electric displacement** to use the constitutive relation $\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E} + \mathbf{D}_r$, where \mathbf{D}_r (SI unit: C/m^2) is the displacement when no electric field is present. The default **Relative permittivity** ϵ_r uses values **From material** or select **User defined** to specify a different value or expression. Enter the **Remanent displacement** \mathbf{D}_r components based on space dimension in the table.

HEAT CONDUCTION

The default **Thermal conductivity** k (SI unit: $\text{W}/(\text{m}\cdot\text{K})$) 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 **Density** ρ (SI unit: kg/m³) 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.

Electromagnetic Heat Source

The **Electromagnetic Heat Source** is added as a default node. This node represents the source term Q (SI unit: W/m³) in the heat equation implemented by the Joule Heating Model node; see [Equation 14-17](#). The resistive heating (ohmic heating) due to the electric current is

$$Q = \mathbf{J} \cdot \mathbf{E}$$

where \mathbf{J} is the current density (SI unit: A/m²), and \mathbf{E} is the electric field strength (SI unit: V/m).

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter values or expressions for the initial values of the **Temperature** T (SI unit: K) and the **Electric potential** V (SI unit: V). The default value for the temperature is room temperature, 293.15 K (20° C) and for the electric potential it is 0 V.

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/m²) in the heat transfer part of the model. It is a default node.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

About Frames in Heat Transfer

This section discusses heat transfer analysis with moving frames, when spatial and material frames do not coincide. When the **Enable conversions between material and spatial frame** check box is selected, all heat transfer physics account for deformation effects on heat transfer properties.

The entire physics (equations and variables) are defined on the spatial frame. When a moving mesh is detected, the user inputs for certain features are defined on the material and are converted so that all the corresponding variables contain the value on the spatial frame.

-
- 
- [The Heat Transfer User Interface](#)
 - [Domain, Boundary, Edge, Point, and Pair Nodes for the Heat Transfer User Interfaces](#)
 - [Heat Transfer Theory](#)
-

Frame Physics Feature Nodes and Definitions

This subsection contains the list of all heat transfer nodes and the corresponding definition frame.



Some of the physics require additional licenses, for example, a Heat Transfer Module or a CFD Module.

The following explains the different values listed in the *definition frame* column in [Table 14-3](#), [Table 14-4](#), and [Table 14-5](#):

Material: •The inputs are entered by the user and defined on the material frame.
Because the heat transfer variables and equations are defined on the spatial frame, the inputs are internally converted to the spatial frame.

Spatial: •The inputs are entered by the user are defined on the spatial frame. No conversion is done.

Material/(Spatial): •For these physics nodes, select from a menu to decide if the inputs are defined on the material or spatial frame. The default definition frame is the material frame.

(Material)/Spatial: •For these physics nodes, select from a menu to decide if the inputs are defined on the material or spatial frame. The default definition frame is the spatial frame.

N/A: There is no definition frame for this physics node.

Domain Nodes

TABLE I4-3: DOMAIN PHYSICS NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Heat Transfer in Solids	Material
Translational Motion	Material
Heat Transfer in Fluids	Spatial
Biological Tissue	Material
Heat Transfer with Phase Change	Spatial
Heat Transfer in Porous Media	Material (Solid part) Spatial (Fluid part)
Thermal Dispersion	Spatial
Immobile Fluids	Spatial
Geothermal Heating	Material
Infinite Elements	Spatial
Pressure Work	Spatial
Viscous Heating	Spatial
Heat Source	Material//(Spatial)
Bioheat	Material
Opaque	N/A
Out-of-Plane Convective Heat Flux	Spatial
Out-of-Plane Radiation	Spatial
Out-of-Plane Heat Flux	Spatial
Radiation in Participating Media	Spatial
Change Thickness	Spatial
Initial Values	Spatial

Boundary Nodes

TABLE 14-4: BOUNDARY PHYSICS NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Temperature	Spatial
Thermal Insulation	N/A
Outflow	N/A
Symmetry	N/A
Heat Flux	(Material)/Spatial
Inflow Heat Flux	Spatial
Open Boundary	Spatial
Thin Thermally Resistive Layer	Material
Thermal Contact	Material
Surface-to-Ambient Radiation	Spatial
Surface-to-Surface Radiation	Spatial
Prescribed Radiosity	Spatial
Reradiating Surface	Spatial
Radiation Group	N/A
Periodic Heat Condition	Spatial
Boundary Heat Source	Material//(Spatial)
Heat Continuity	Spatial
Pair Thin Thermally Resistive Layer	Material
Pair Thermal Contact	Material
Pair Boundary Heat Source	Material//(Spatial)
Convective Heat Flux	Spatial
Highly Conductive Layer	Material
Layer Heat Source	Material
Opaque Surface	Spatial
Continuity on Interior Boundary	Spatial

Edge and Point Nodes

TABLE 14-5: EDGE AND POINT NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Line Heat Source	Material//(Spatial)
Point Heat Source	Material

TABLE 14-5: EDGE AND POINT NODES FOR FRAMES

NODE NAME	DEFINITION FRAME
Edge Heat Flux	(Material)/Spatial
Point Heat Flux	Spatial
Temperature	Spatial
Point Temperature	Spatial
Edge Surface-to-Ambient	Spatial
Point Surface-to-Ambient	Spatial

TABLE 14-6: HEAT TRANSFER IN THIN SHELLS NODES

NODE NAME	DEFINITION FRAME
Thin Conductive Layer	Material
Heat Source	Material/(Spatial)
Change Thickness	Spatial
Initial Values	Spatial
Out-of-Plane Convective Heat Flux	Spatial
Out-of-Plane Radiation	Spatial
Out-of-Plane Heat Flux	Spatial
Heat Flux	Spatial/(Material)
Surface-to-Ambient Radiation	Spatial
Temperature	Spatial
Change Effective Thickness	Spatial
Edge Heat Source	Material/(Spatial)

Conversion Between Material and Spatial Frames

This subsection explains how the user inputs are converted. The conversion depends on the dimension of the variables (scalars, vectors, or tensors) and on their density order.

DENSITY, HEAT SOURCE, HEAT FLUX

Scalar density variables do not have the same value in the material and in the spatial frame.

In heat transfer physics, the following variables are relative scalars of weight one (also called scalar densities): the mass density ρ , the heat source Q , the heat flux q_0 , the heat transfer coefficient h , and the production/absorption coefficient q_s .

When a feature has its *definition frame* on the spatial frame, no transformation is done because the user input is defined on the spatial frame. For example, if $\rho=500[\text{kg}/\text{m}^3]$ is defined in the [Heat Transfer in Fluids](#) (*definition frame* = spatial frame) the variable `ht.rho` is equal to $500[\text{kg}/\text{m}^3]$ (on the spatial frame).

When a feature has its *definition frame* on the material frame, the user input is defined on the material frame so it has to be multiplied by `spatial.detInvF` to get the corresponding value on the spatial frame. For example, if $\rho=500[\text{kg}/\text{m}^3]$ is defined in the [Heat Transfer in Solids](#) (*definition frame* = material frame) the variable `ht.rho` is equal to `spatial.detInvF*500[\text{kg}/\text{m}^3]` (on the spatial frame). As a consequence, to evaluate or integrate the mass density on the material frame, the value of `ht.rho` has to be multiplied by `spatial.detF`.

`spatial.detF` has different definitions based on the dimension of the geometric entity where it is evaluated. On domains it corresponds to the local volume change from the material to the spatial frame while it corresponds to local surface or length change on boundaries and edges. `spatial.detInvF` is the inverse of `spatial.detF`.

VELOCITY VECTOR

The relationship between $\mathbf{u}_{(x,y,z)}$ and $\mathbf{u}_{(X,Y,Z)}$ is

$$\mathbf{u}_{(x,y,z)} = \mathbf{F}^T \mathbf{u}_{(X,Y,Z)}$$

where F is the coordinate transform matrix from the material to the spatial frame:

$$F = \begin{bmatrix} x_X & y_X & z_X \\ x_Y & y_Y & z_Y \\ x_Z & y_Z & z_Z \end{bmatrix}$$

with x_X corresponding to the derivative of x with respect to X .

THERMAL CONDUCTIVITY

Thermal conductivity is a tensor density. The relationship between the value on the spatial frame and the material frame is

$$\mathbf{k}_{(x,y,z)} = \frac{1}{\det(\mathbf{F})} \mathbf{F}^T \mathbf{k}_{(X,Y,Z)} \mathbf{F}$$

where $\mathbf{k}_{(x,y,z)}$ is the thermal conductivity tensor in the spatial frame and $\mathbf{k}_{(X,Y,Z)}$ is the thermal conductivity tensor in the material frame. F is the coordinate transform matrix from the material frame to the spatial frame defined in the paragraph above.

THERMAL CONDUCTIVITY OF HIGHLY CONDUCTIVE LAYER

The same transformations are applied to thermal conductivity but with different transformation matrices. The transformation matrix uses tangential derivatives and is defined as

$$F_{\text{tang}} = \begin{bmatrix} xT_X & yT_X & zT_X \\ xT_Y & yT_Y & zT_Y \\ xT_Z & yT_Z & zT_Z \end{bmatrix}$$

where xT_X corresponds to the tangential derivative x with respect to X , and so on.

AXISYMMETRIC GEOMETRIES

In 1D axisymmetric and 2D axisymmetric models an additional conversion is done between the material frame and the spatial frame. The density variables (density, heat source, heat flux, and so forth) are multiplied by

$$\frac{R}{r}$$

which corresponds to the ratio of the material first cylindrical coordinate over the spatial one.

For example, if you enter a heat source $Q = 500[\text{W/m}^3]$ in the material frame in axisymmetric cases, the conversion leads to:

$$Q = 500[\text{W/m}^3] * \frac{R}{r} \text{spatial.detInvF}$$

Structural Mechanics

This chapter explains how to use the Solid Mechanics user 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.

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).

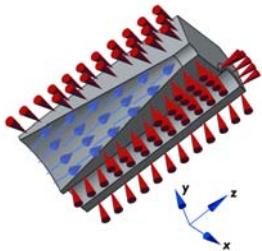


Figure 15-1: Loads and constraints applied to a 3D solid using the Solid Mechanics interface.

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.

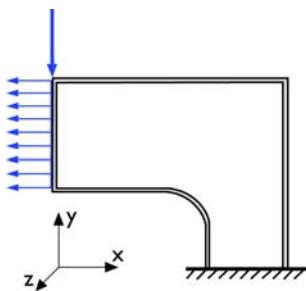


Figure 15-2: Plane stress models plates where the loads are only in the plane; it does not include any out-of-plane stress components.

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.

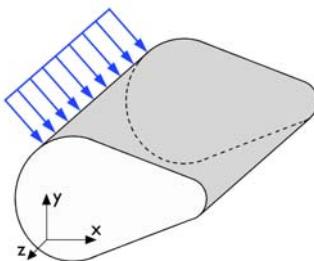


Figure 15-3: A geometry suitable for plane strain analysis.

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 $\varphi = 0$, $r \geq 0$. Therefore the geometry is drawn only in the half plane $r \geq 0$ and recover the original 3D solid by rotating the 2D geometry about the z -axis.

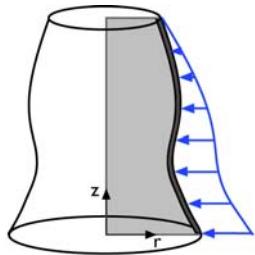


Figure 15-4: Rotating a 2D geometry to recover a 3D solid.

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 15-1](#). These symbols are available with the applicable structural mechanics feature nodes.

TABLE 15-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY NODE	NOTES
	Added Mass ¹	Added Mass	
	Antisymmetry ¹	Antisymmetry	
	Body Load ¹	Body Load	

TABLE 15-I: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY NODE	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 node.
	Damping ¹	Spring Foundation	Can be displayed together with the Spring symbol, depending on the values given in the node.
	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 node.
	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 node.
	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 node.
	Prescribed Acceleration	Prescribed Acceleration	
	Prescribed Displacement	Prescribed Displacement	
	Prescribed Velocity ¹	Prescribed Velocity	
	Rigid Connector ¹	Rigid Connector	

TABLE 15-1: STRUCTURAL MECHANICS BOUNDARY CONDITION PHYSICS SYMBOLS

SYMBOL	SYMBOL NAME	DISPLAYED BY NODE	NOTES
	Roller	Roller	
	Spring ¹	Spring Foundation Thin Elastic Layer	Can be displayed together with the Damping symbol, depending on the values given in the node.
	Symmetry	Symmetry	
	Thin-Film Damping ²	Thin-Film Damping	

¹ Requires the Structural Mechanics Module
² Requires the MEMS Module

About Coordinate Systems and Physics Symbols

Physics symbols connected to a node 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 node.

 For cases when physics symbol display is dependent on values actually given in the node, it may be necessary to move to another node before the display is actually updated on the screen.

Displaying Physics Symbols in the Graphics Window—An Example

- I 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**.



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 15-1](#) 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 15-5](#).

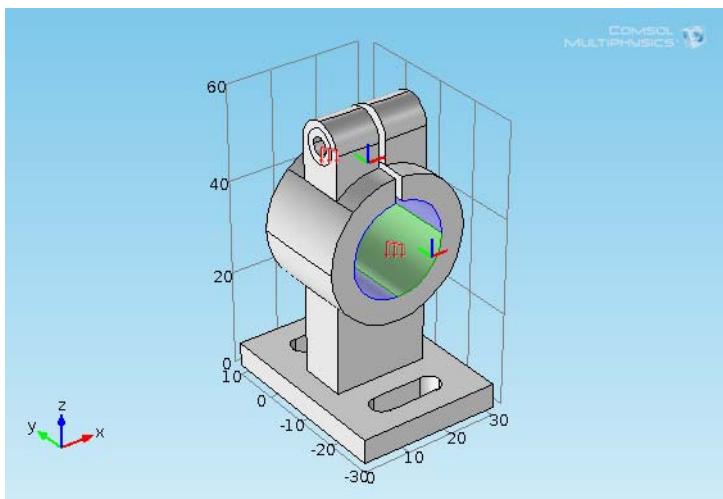


Figure 15-5: Example of the Boundary Load physics symbols as displayed in the COMSOL Multiphysics Model Library model [Deformation of a Feeder Clamp](#).

- 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 15-6.

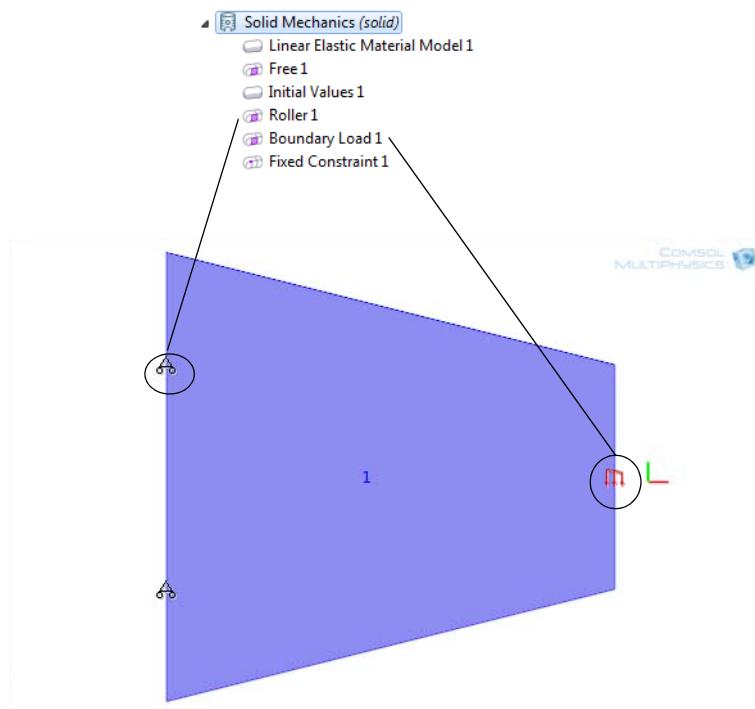


Figure 15-6: Example of Roller and Boundary Load physics symbols as displayed in the COMSOL Multiphysics Model Library model 'Tapered Membrane End Load'.

The Solid Mechanics User Interface

The **Solid Mechanics (solid)** user interface (Solid icon), found under the **Structural Mechanics** branch (Structural Mechanics icon) in the **Model Wizard**, has the equations and functionality 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 physics user 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 material models, boundary conditions, and loads.

INTERFACE IDENTIFIER

The interface identifier is a text string that can be used to reference the respective physics user interface if appropriate. Such situations could occur when coupling this interface to another physics user interface, or when trying to identify and use variables defined by this physics user 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 user 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

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



For 2D models, 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.

Use a [Change Thickness](#) node to change thickness in parts of the geometry if necessary.

STRUCTURAL TRANSIENT BEHAVIOR

From the **Structural transient behavior** list, select **Include inertial terms** (the default) or **Quasi-static**. Use **Quasi-static** to treat the elastic behavior as quasi-static (with no mass effects; that is, no second-order time derivatives). Selecting this option will give a more efficient solution for problems where the variation in time is slow when compared to the natural frequencies of the system. The default solver for the time stepping is changed from Generalized alpha to BDF when **Quasi-static** is selected.

REFERENCE POINT FOR MOMENT COMPUTATION

Enter the coordinates for the **Reference point for moment computation** \mathbf{x}_{ref} (SI unit: m; COMSOL 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** \mathbf{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 user interface to a Shell directly attached to the boundaries of the solid domain, or to another Solid Mechanics user 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 **Quintic**. Specify the **Value type when using splitting of complex variables—Real** or **Complex** (the default). The **Frame type** in the Solid Mechanics interface is always **Material**.

-
- 
 - Show More Physics Options
 - Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics
 - About the Body, Boundary, Edge, and Point Loads
 - Theory of Solid Mechanics
-

- 
 - **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**
-

Domain, Boundary, Edge, Point, and Pair Nodes for Solid Mechanics

The **Solid Mechanics User Interface** has these domain, boundary, edge, point, and pair nodes listed in alphabetical order. The list also includes subnodes.

- Body Load
- Boundary Load
- Change Thickness
- Damping
- Edge Load
- Fixed Constraint
- Free
- Initial Values
- Linear Elastic Material
- Periodic Condition
- Point Load
- Prescribed Displacement
- Roller



If there are subsequent boundary conditions specified on the same geometrical entity, the last one takes precedence.

	For information about the Perfectly Matched Layers feature, see Infinite Element Domains and Perfectly Matched Layers .
	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.
	<ul style="list-style-type: none">• Harmonic Perturbation—Exclusive and Contributing Nodes• Continuity on Interior Boundaries• Identity and Contact Pairs
	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** node 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains to define a linear elastic solid and compute the displacements, stresses, and strains, or select **All domains** as required.

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.

Density

The default **Density** ρ (SI unit: kg/m³) uses values **From material**. If **User defined** is selected, enter another value or expression.

Specification of Elastic Properties for Isotropic Materials

For an **Isotropic Solid model**, from the **Specify** list select a pair of elastic properties for an isotropic material—**Young's modulus and Poisson's ratio**, **Young's modulus and shear modulus**, **Bulk modulus and shear modulus**, **Lamé parameters**, or **Pressure-wave and shear-wave speeds**. For each pair of properties, select from the applicable list to use the value **From material** or enter a **User defined** value or expression.



Each of these pairs define the elastic properties and it is possible to convert from one set of properties to another (see Table 15-2).

The individual property parameters are:

- **Young's modulus** (*elastic modulus*) E (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 default is 0 Pa.
- **Poisson's ratio** ν (dimensionless), which defines the normal strain in the perpendicular direction, generated from a normal strain in the other direction and follows the equation $\varepsilon_{\perp} = -\nu\varepsilon_{\parallel}$. The default is 0.
- **Shear modulus** G (SI unit: N/m²), which is a measure of the solid's resistance to shear deformations. The default is 0 N/m².
- **Bulk modulus** K (SI unit: N/m²), which is a measure of the solid's resistance to volume changes. The default is 0 N/m².

- **Lamé parameter** λ (SI unit: N/m²) and **Lamé parameter** μ (SI unit: N/m²). The defaults are 0 N/m².
- **Pressure-wave speed** (longitudinal wave speed) c_p (SI unit: m/s). The default is 0 m/s.
- **Shear-wave speed** (transverse wave speed) c_s (SI unit: m/s). The default is 0 m/s.



This is the wave speed for a solid continuum. In plane stress, for example, the actual speed with which a longitudinal wave travels is lower than the value given.

Change Thickness



The **Change Thickness** node is available in 2D.

Use the **Change Thickness** node 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 to 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** subnode, which is used in time-dependent, eigenfrequency, and frequency domain studies to model undamped or damped problems. The node adds Rayleigh damping by default.

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

For a default node, the setting inherits the selection from the parent node. Or select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

DAMPING SETTINGS

Rayleigh damping is the only available **Damping type**.

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 this damping model, the damping parameter ξ is expressed in terms of the mass m and the stiffness k as

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

That is, Rayleigh damping is proportional to a linear combination of the stiffness and mass; there is no direct physical interpretation of the *mass damping parameter* α_{dM} and the stiffness damping parameter β_{dM} .

Initial Values

The **Initial Values** node adds initial values for the displacement field and structural velocity field that can serve as an initial condition for a transient simulation or as an initial guess for a nonlinear analysis. Right-click to add additional **Initial Values** nodes.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the

interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter values or expressions for the initial values of the **Displacement field \mathbf{u}** (SI unit: m) (the displacement components u , v , and w in 3D) (the default is 0 m), and the **Structural velocity field $\partial\mathbf{u}/\partial t$** (SI unit: m/s) (the default is 0 m/s).

About the Body, Boundary, Edge, and Point Loads

Add force loads acting on all levels of the geometry to [The Solid Mechanics User 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 modeling gravity or centrifugal loads, for example.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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

Select a **Load type—Load defined as force per unit volume** (the default) or **Total force**.

For 2D models, **Load defined as force per unit area** is also an option.

Then enter values or expressions for the components in the matrix based on the selection and the space dimension:

- **Body load \mathbf{F}_V** (SI unit: N/m³)

- **Total force \mathbf{F}_{tot}** (SI unit: N). For total force, COMSOL divides the total force by the volume of the domains where the load is active.
- *For 2D models:* **Load \mathbf{F}_A** (SI unit: N/m²). The body load as force per unit volume is then the value of F divided by the thickness.

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.

PAIR SELECTION

If this node 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.

FORCE

Select a **Load type**—**Load defined as force per unit area** (the default), **Pressure**, or **Total force**. For 2D models, **Load defined as force per unit length** is also an option.



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.

Then enter values or expressions for the components in the matrix based on the selection and the space dimension:

- **Load \mathbf{F}_A** (SI unit: N/m²). The body load as force per unit volume is then the value of F divided by the thickness.
- *For 2D models:* **Load \mathbf{F}_L** (SI unit: N/m).

- **Total force \mathbf{F}_{tot}** (SI unit: N). For total force, 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.

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

Select a **Load type—Load defined as force per unit area** (the default) or **Total force**. Then enter values or expressions for the components in the matrix based on the selection:

- **Load \mathbf{F}_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}_{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.

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}_p** (SI unit: N).

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 selected from the **More** submenu.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose, the geometric entity (domains, boundaries, edges, or points) to define.

PAIR SELECTION

If this node 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**.

To **Apply reaction terms on** all dependent variables, select **All physics (symmetric)**.

Otherwise, select **Current physics (internally symmetric)** or **Individual dependent variables** to restrict the reaction terms as required. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.



Boundary Conditions

Prescribed Displacement

The **Prescribed Displacement** node adds a condition where the displacements are prescribed in one or more directions to the geometric entity (domain, boundary, edge, or point).

If a displacement is prescribed in one direction, this leaves the solid free to deform in the other directions. Also define more general displacements as a linear combination of the displacements in each direction.

-
- 
- 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 define.

PAIR SELECTION

If this node 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** (the default) or a **General notation**.

Standard Notation

To define the displacements individually, click the **Standard notation** button.

Select one or all of the **Prescribed in x direction**, **Prescribed in y direction**, and for 3D models, **Prescribed in z direction** check boxes. Then enter a value or expression for u_0 , v_0 , and for 3D models, w_0 (SI unit: m). 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 u_0 and w_0 (SI unit: m).

General Notation

Click the **General notation** to specify the displacements using a general notation that includes any linear combination of displacement components. For example, for 2D models, use the relationship

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

For **H matrix** H (dimensionless) select **Isotropic**, **Diagonal**, **Symmetric**, or **Anisotropic** and then enter values as required in the field or matrix. Enter values or expressions for the **R vector** R (SI unit: m)

For example, to achieve the condition $u = v$, use the settings

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

which force the domain to move only diagonally in the xy -plane.

CONSTRAINT SETTINGS

See [Fixed Constraint](#) for these settings.

Free

The **Free** node is the default boundary condition. It means that there are no constraints and no loads acting on the boundary.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.

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. See [Fixed Constraint](#) for all the settings.

CONSTRAINT SETTINGS

See [Fixed Constraint](#) for these settings.

Periodic Condition

The **Periodic Condition** node adds a periodic boundary condition. This periodicity makes $u_i(x_0) = u_i(x_1)$ for a displacement u_i . You can control the direction that the periodic condition applies to. 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. 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 right-click the **Periodic Condition** node to add 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**.

CONSTRAINT SETTINGS

See [Fixed Constraint](#) for these settings.

-
-
- [Periodic Condition](#) and [Destination Selection](#)
 - [Periodic Boundary Conditions](#)
-

Theory of Solid Mechanics

The Solid Mechanics User 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 \mathbf{X} (bold font is used to denote coordinate vectors), while its spatial coordinates change with time and applied forces such that it follows a path

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \quad (15-1)$$

in space. Because the material coordinates are constant, the current spatial position is uniquely determined by the displacement vector \mathbf{u} , pointing from the reference position to the current position. The global Cartesian components of this displacement

vector in the spatial frame, by default called u , v , and w , are the primary dependent variables in the Solid Mechanics interface.

By default, the Solid Mechanics interface uses the calculated displacement and [Equation 15-1](#) to define the difference between spatial coordinates \mathbf{x} and material coordinates \mathbf{X} . This means the material coordinates relate to the original geometry, while the spatial coordinates are solution dependent.

Material coordinate variables X , Y , and Z must be used in coordinate-dependent expressions that refer to positions in the original geometry, for example, for material properties that are supposed to follow the material during deformation. On the other hand, quantities that have a coordinate dependence in physical space, for example, a spatially varying electromagnetic field acting as a force on the solid, must be described using spatial coordinate variables x , y , and z .

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.

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. 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*.

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:

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

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

About Linear Elastic Materials

The total strain tensor is written in terms of the displacement gradient

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

or in components as

$$\varepsilon_{mn} = \frac{1}{2} \left(\frac{\partial u_m}{\partial x_n} + \frac{\partial u_n}{\partial x_m} \right) \quad (15-2)$$

The Duhamel-Hooke's law relates the stress tensor to the strain tensor and temperature:

$$s = s_0 + C : (\varepsilon - \varepsilon_0 - \alpha\theta)$$

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, $\theta = T - T_{\text{ref}}$, and α is the thermal expansion tensor.

The elastic energy is

$$W_s = \frac{1}{2}(\varepsilon - \varepsilon_0 - \alpha\theta) : C : (\varepsilon - \varepsilon_0 - \alpha\theta) \quad (15-3)$$

or using the tensor components:

$$W_s = \sum_{i,j,m,n} \frac{1}{2} C^{ijmn} (\varepsilon_{ij} - \varepsilon_{ij}^0 - \alpha_{ij}\theta)(\varepsilon_{mn} - \varepsilon_{mn}^0 - \alpha_{mn}\theta)$$

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

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

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 15-2](#).

TABLE 15-2: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E, v)$	$D(K, G)$	$D(\lambda, \mu)$
Young's modulus	E		$\frac{9KG}{3K+G}$	$\mu \frac{3\lambda + 2\mu}{\lambda + \mu}$
Poisson's ratio	v		$\frac{1}{2} \left(1 - \frac{3G}{3K+G} \right)$	$\frac{\lambda}{2(\lambda + \mu)}$
Bulk modulus	K	$\frac{E}{3(1-2v)}$		$\lambda + \frac{2\mu}{3}$
Shear modulus	G	$\frac{E}{2(1+v)}$		μ
Lamé parameter λ	λ	$\frac{Ev}{(1+v)(1-2v)}$	$K - \frac{2G}{3}$	
Lamé parameter μ	μ	$\frac{E}{2(1+v)}$	G	

TABLE 15-2: EXPRESSIONS FOR THE ELASTIC MODULI.

DESCRIPTION	VARIABLE	$D(E, v)$	$D(K, G)$	$D(\lambda, \mu)$
Pressure-wave speed	c_p		$\sqrt{\frac{K+4G/3}{\rho}}$	
Shear-wave speed	c_s		$\sqrt{G/\rho}$	

According to Table 15-2, 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 :

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

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:

$$\begin{aligned}\varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_{xy} &= \frac{\gamma_{xy}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\ \varepsilon_y &= \frac{\partial v}{\partial y} & \varepsilon_{yz} &= \frac{\gamma_{yz}}{2} = \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \varepsilon_z &= \frac{\partial w}{\partial z} & \varepsilon_{xz} &= \frac{\gamma_{xz}}{2} = \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right).\end{aligned}\quad (15-4)$$

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

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

Stress-Strain Relationship

The symmetric stress tensor σ describes stress in a material:

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

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}).

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 15-2](#) for these three strain components. Thus, three components ε_{i3} are treated as extra degrees of freedom. For isotropy and orthotropy, 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 15-2](#).



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

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

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

For any tensor, the physical components are defined as

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

where no summation is done over repeated indices.

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

$$s^{ij} = s_0^{ij} + C^{ijkl} (\varepsilon_{kl} - \alpha_{kl} \theta - \varepsilon_{0kl})$$

where $\theta = T - T_{\text{ref}}$.

For an isotropic material:

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

where λ and μ are the first and second Lamé elastic parameters.

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).

	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).
	For 2D axisymmetric models, the boundary loads apply on edges (boundaries).
	For 3D solids, the boundary loads apply on faces (boundaries).

Table 15-3 shows how to define distributed loads on different geometric entity levels; the entries show the SI unit in parentheses.

TABLE 15-3: 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.



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](#).

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

$$\int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt}) dv \\ + \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}_{\text{test}}^t \cdot \mathbf{F}_p)$$

Setting up Equations for Different Studies

The Solid Mechanics interface supports stationary (static), eigenfrequency, time-dependent (transient), 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:

$$\begin{aligned}\delta W = & \int_V (-\varepsilon_{\text{test}} : s + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V) dv \\ & + \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}_{\text{test}}^t \cdot \mathbf{F}_p)\end{aligned}$$

TIME-DEPENDENT STUDIES

$$\begin{aligned}\int_V & (-\varepsilon_{\text{test}} : (s + \beta_{dM} s_t) + \mathbf{u}_{\text{test}} \cdot \mathbf{F}_V - \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_{tt} - \alpha_{dM} \rho \mathbf{u}_{\text{test}} \cdot \mathbf{u}_t) dv \\ & + \int_S (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_S) ds + \int_L (\mathbf{u}_{\text{test}} \cdot \mathbf{F}_L) dl + \sum_p (\mathbf{U}_{\text{test}}^t \cdot \mathbf{F}_p)\end{aligned}\quad (15-5)$$

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

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

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

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

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

where

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

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

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

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

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:

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

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

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 15-2](#). The contribution from the beta-damping that shown in [Equation 15-5](#) corresponds to the case of small strains. In case of geometric nonlinearity, it becomes

$$\int_V (-\beta_d M \nabla \mathbf{u}_{\text{test}} : P_t) dv$$

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

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

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 15-7 shows.

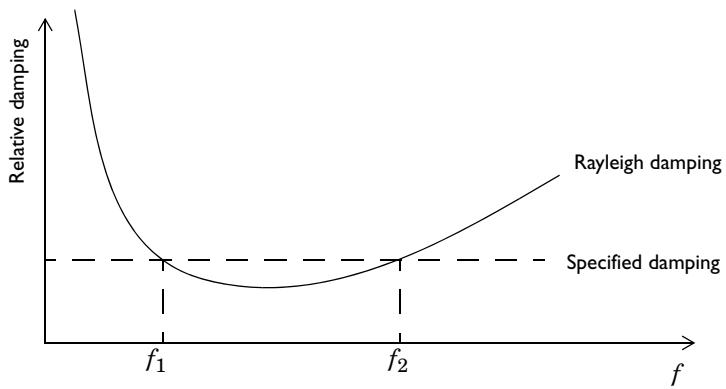


Figure 15-7: An example of Rayleigh damping.

Equation-Based Modeling

This chapter describes the use of the user interfaces for mathematics and equation-based modeling, found under the **Mathematics** branch (Δu) in the **Model Wizard**. With those user interfaces you can solve various types of PDEs using different formulations. You can also solve ODEs and other global equations as well as create curvilinear coordinates.

The Mathematics User Interfaces

The mathematics user interfaces are a collection of tools for equation-based modeling and for performing special tasks, rather than for modeling specific physics. These user interfaces support several PDE formulations as well as general ways to add ODEs, algebraic equations, other global (space-independent) equations, and curvilinear coordinates.



For a list of the available user interfaces found under the Model Wizard's **Mathematics** branch (Δu) including the identifier names, see [Physics Guide](#).

PDE USER INTERFACES

These user interfaces are for entering PDEs in different forms:

- *Coefficient form* for PDEs conforming to the template explained in [The Coefficient Form PDE User Interfaces](#).
- *General form* for conservation laws and PDEs resulting from nonlinear material models. See [The General Form PDE User Interfaces](#).
- *Weak form* using the weak formulation of the PDE for maximum flexibility. See [The Weak Form PDE User Interfaces](#).
- [The Wave Form PDE](#) solves PDEs with first-order derivatives in time and space using optimized algorithms with respect to speed and memory consumption.

Except for the Wave Form PDE, the PDE user interfaces are available in domains, boundaries, edges, and at points. The user interfaces for the different equation forms are identical except for the default node on the top geometric entity level. Also see [Modeling with PDEs](#).

CLASSICAL PDES

These specialized instances of the coefficient form PDE provide user interfaces for a number of [The Classical PDE User Interfaces](#). Also see [Compact and Standard Notations for Classical PDEs](#).

ODE AND DAE USER INTERFACES

Use these user 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 [Modeling with ODEs and DAEs](#) as well as [The ODE and DAE User Interfaces](#).

EVENTS USER INTERFACE

The [Events User Interface](#) is a specialized user interface used to create solver events. An event can be explicit or implicit, the difference being that for explicit events you 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.

WALL DISTANCE USER INTERFACE

The [Wall Distance User Interface](#), which computes the distance to the boundary (wall), is primarily intended for use in connection to turbulence modeling for fluid flow.

CURVILINEAR COORDINATES USER INTERFACE

[Curvilinear Coordinates](#) provides the possibility to create curvilinear coordinates using a diffusion method, an elasticity method, a flow method, or user-defined coordinate directions.

Modeling with PDEs

The physics user interfaces in COMSOL Multiphysics and add on Modules use *partial differential equations*, PDEs, as a mathematical model of physical reality. You can access these PDEs in a number of different ways:

- [The PDE User Interfaces](#) allow you to specify whole or part of your problem completely using PDEs. This approach may, for example, be suitable for modeling unusual equations from various fields of physics, or for learning mathematical modeling.
- Auxiliary equation-based nodes are available in all physics user interfaces. These let you add extra equation contributions and constraints to the predefined mathematical model.
- Equation View nodes display the PDEs (on weak form) and constraints underlying the physics interfaces, and allow you to modify them.

This section describes the theory behind the PDE user interfaces, but also contains information useful for understanding and modifying the mathematical models implemented in the physics interfaces.

-
- 
 - [The PDE User Interfaces](#)
 - [The Wave Form PDE](#)
 - [About Auxiliary Equation-Based Nodes](#)
-

About Equation Forms

Partial differential equations may be entered into COMSOL Multiphysics on four different formats:

- [The General Form PDE](#)
- [The Coefficient Form PDE](#)
- [The Weak Form PDE](#)
- [The Wave Form PDE](#)

Which one to choose is mostly a matter of convenience: certain equations are quicker and easier to specify in one particular form. Internally, equations written on general or coefficient form are converted to weak form, which is therefore the most fundamental

one. In particular, the weak form is closely linked to the theory behind the *finite element method*, FEM. In a similar way, the wave form is linked to a particular *discontinuous Galerkin* version of FEM, particularly suited for solving wave-propagation problems.

Notational Conventions

The PDE user interfaces use a slightly different notation, compared to other physics user interfaces and the associated documentation. The difference lies in the definition of the symbol ∇ , pronounced *nabla* or *del*. For the physics interface equation sections and nodes (see [Advanced Physics Sections](#)), the ∇ symbol applied to a scalar or vector variable denotes the following coordinate system-independent *gradient*, *divergence*, and *curl* operations:

$$\begin{aligned}\nabla u &= \text{grad}(u) \\ \nabla \cdot \mathbf{u} &= \text{div}(\mathbf{u}) \\ \nabla \times \mathbf{u} &= \text{curl}(\mathbf{u})\end{aligned}$$

In the PDE interfaces, *nabla/del* is interpreted as the vector of partial derivatives:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right)$$

The spatial coordinates are denoted x_1, \dots, 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

$$\text{div}(\mathbf{u}) = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} = \frac{1}{r} \frac{\partial}{\partial r} (r u_r) + \frac{\partial u_z}{\partial z}$$

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 must compensate for the missing terms related to the curvature of the coordinate system. In particular, note that you must typically multiply your entire equation as well as boundary conditions, by a volume factor—in an axisymmetric geometry, for example, with a factor r —in order to recast it into one of COMSOL Multiphysics’ equation forms.

The following related examples follow the same principle:

- The symbol Δ is the Laplace operator

$$\frac{\partial^2}{\partial x_1^2} + \dots + \frac{\partial^2}{\partial x_n^2}$$

- $\nabla \cdot (c \nabla u)$ means

$$\frac{\partial}{\partial x_1} \left(c \frac{\partial u}{\partial x_1} \right) + \dots + \frac{\partial}{\partial x_n} \left(c \frac{\partial u}{\partial x_n} \right)$$

- $\beta \cdot \nabla u$ means

$$\beta_1 \frac{\partial u}{\partial x_1} + \dots + \beta_n \frac{\partial u}{\partial x_n}$$

where β_1, \dots, β_n are the components of the vector β .



The axisymmetric versions of physics user interfaces take the cylindrical system into account, and no compensation is therefore needed.



- [The PDE User Interfaces](#)
- [The Wave Form PDE](#)

PDE User Interface Variables

The following list shows symbolic expressions for quantities appearing in the definition of PDEs and corresponding variable names which can be used in PDE coefficients and are available for results evaluation and visualization

EXPRESSION	NAME	DESCRIPTION
u_i	ui	The solution variable (dependent variable)
$\frac{\partial u_i}{\partial x_j}$	$uixj$	The derivative of the solution variable u_i with respect to the spatial coordinate x_j , for example, uy
$(\nabla_T u_i)_j$	$uitxj$	The x_j component of the gradient of u_i projected onto a boundary or edge, for example, uTy
$\frac{\partial^2 u_i}{\partial x_j \partial x_k}$	$uixjxk$	The second derivative of the solution variable u_i with respect to the spatial coordinates x_j and x_k , for example, uxx, uxy
$\frac{\partial u_i}{\partial t}$	uit	The derivative of the solution variable u_i with respect to time
$\frac{\partial^2 u_i}{\partial t^2}$	$uitt$	The second derivative of the solution variable u_i with respect to time
$\frac{\partial^2 u_i}{\partial x_j \partial t}$	$uixjt$	The mixed derivative of the solution variable u_i with respect to time and the spatial coordinate x_j

The General Form PDE

The general form PDE format is closely related to the conservation laws which govern many areas of physics. Assuming that you are working with a single dependent variable u , the general form reads:

$$\left\{ \begin{array}{ll} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{\Gamma} = f & \text{in } \Omega \\ -\mathbf{n} \cdot \boldsymbol{\Gamma} = g - qu + h^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ u = r & \text{on } \partial\Omega_d \end{array} \right. \quad (16-1)$$

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 line (equation) of [Equation 16-1](#) is the PDE, which must be satisfied in Ω . The second, third and fourth equations are the boundary conditions, which must hold on $\partial\Omega$. The second equation is a generalization of a *Neumann* boundary condition. The third equation is a general *constraint*, of which the *Dirichlet* boundary condition on the fourth line is a special case.

The terms $\boldsymbol{\Gamma}, f, g, q, R$ and r are user-defined coefficients. They can be functions of the spatial coordinates, the solution u , and the space derivatives of u (see [PDE User Interface Variables](#)), as well as of other predefined and user-defined variables. The coefficients f, g, q, R and r are scalar, whereas $\boldsymbol{\Gamma}$ is the *flux vector*.

In practical applications, $\boldsymbol{\Gamma}$ typically represents the flux of a conserved quantity such as heat, charge, mass or momentum. This flux is usually related in some empirical way, via a material law, to the gradient of the dependent variable. Therefore, $\boldsymbol{\Gamma}$ is usually a vector whose components are functions of derivatives of the dependent variable. The flux vector may also contain terms which are proportional to a velocity field when there is convective transport of the conserved quantity present. The structure of [Equation 16-1](#) implies that the normal component of $\boldsymbol{\Gamma}$ will be continuous across any surface in the interior of the domain, Ω .

BOUNDARY CONDITIONS FOR THE GENERAL FORM PDE

In finite element terminology, the boundary condition on the second line of [Equation 16-2](#), corresponding to a Neumann boundary condition, is called a *natural boundary condition*, 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 Γ at the boundary.

Constraints and 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 general constraint on line 3 of [Equation 16-2](#) specifies that an arbitrary expression is equal to zero on the boundary, $R = 0$. The Dirichlet condition on line 4 of the same equation is a special case directly specifying the value of the dependent variable at the boundary, $u = r$. This makes the constraint a more general boundary condition.

The term $-h^T \mu$ in the generalized Neumann condition is a reaction term enforcing the constraint $R = 0$. When reaction terms are applied symmetrically on all dependent variables,

$$h = \frac{dR}{du}$$

but also other definitions are possible. The variable μ is a Lagrange multiplier, which is eliminated by the solvers when using standard constraints, and therefore does not normally appear explicitly in equations.



For details about the time-dependent and eigenvalue formulations, see [Solving Time-Dependent Problems](#) and [Solving Eigenvalue Problems](#).



- [The General Form PDE User Interfaces](#)
- [Domain, Boundary, Pair, Edge, and Point Conditions for PDEs](#)

The Coefficient Form PDE

Many PDEs originating from physics and other fields may be cast into a generic form containing derivatives up to second order in both time and space, but no mixed derivatives. In COMSOL multiphysics, you can define a PDE of this type by specifying coefficients for the derivatives of different orders. This results in a *coefficient form* PDE, which for one dependent variable u reads:

$$\left\{ \begin{array}{ll} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f & \text{in } \Omega \\ \mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) = g - q u + h^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ u = r & \text{on } \partial\Omega_d \end{array} \right. \quad (16-2)$$

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 line (equation) of [Equation 16-2](#) is the PDE, which must be satisfied in Ω . The second and third equations are the boundary conditions, which must hold on $\partial\Omega$. The second equation is a generalization of a *Neumann* boundary condition. The third equation is a general constraint, with a *Dirichlet* boundary condition as a special case. For more information about the boundary conditions, see [The General Form PDE](#).

To define a PDE on coefficient form in one of the PDE interfaces, you specify the coefficients c , α , γ , β , and a and the boundary terms f , g , R and r . They can all be functions of the spatial coordinates as well as of dependent variables and other predefined or user-defined variables and parameters. A PDE is guaranteed to be *linear* when the coefficients vary only with the spatial coordinates (or are constants). A PDE is *nonlinear* if the c , α , β , a , h or q coefficients depend on u or its derivatives (for example, the components of ∇u), or if γ , f , g , R or r are nonlinear in u .

For a single dependent variable u , all the coefficients in the above equation are scalars except α , β , and γ , which are vectors with n components. The coefficient c may be given alternatively as a scalar or an n -by- n matrix to model anisotropic materials. When the coefficient form is used for modeling a system of equations, the coefficients are extended with additional vector and matrix dimensions referring to the dependent variable index. See further [Multiple Dependent Variables—Equation Systems](#).

COEFFICIENT FORM VERSUS GENERAL FORM

Comparing [Equation 16-2](#) to [Equation 16-1](#) it is easy to see that the coefficient form is just a special case of the general form. In fact, applying the following substitutions

in the general form, [Equation 16-1](#), turns it into the coefficient form:

$$\begin{aligned}\Gamma &= -c \nabla u - \alpha u + \gamma \\ F &= f - \beta \nabla u - \alpha u\end{aligned}\quad (16-3)$$

This duality lets you choose the representation in which it is easiest to implement a particular PDE. There is no difference in performance.

INTERPRETING PDE COEFFICIENTS

The COMSOL Multiphysics PDE formulations can model a variety of problems, but this guide as well as the user interface uses descriptive names for the coefficients 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.
- α is the absorption coefficient.
- γ is the conservative flux source term $\frac{\partial u}{\partial t}^2$ $\frac{\partial u}{\partial t}$ Damping/ $\frac{\partial u}{\partial t}$ Convection
- f is the source term.

For the Neumann boundary condition of the coefficient form

$$\mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) = g - qu + h^T \boldsymbol{\mu}$$

- g is the boundary source term.
- q is the boundary absorption coefficient.

 There are many interesting PDE problems to which these interpretations do not apply. For example, a time-harmonic PDE such as the Helmholtz equation represents a time-dependent phenomenon transformed into the frequency domain, making the α coefficient a mass rather than absorption term.

COMPACT AND STANDARD NOTATIONS FOR CLASSICAL PDES

Many classical PDEs are instances of the coefficient form PDE. The classical PDEs have their own interfaces which are found under the **Mathematics>Classical PDEs** branch ( 2) in the **Model Wizard**. **Table 16-1** shows the available classical PDEs using two notations: the compact notation of vector analysis (used in this documentation) and an expanded component notation.

TABLE 16-1: 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 $-\Delta u - u = 1$.
---	--

-
- Multiple Dependent Variables—Equation Systems
 - The Coefficient Form PDE User Interfaces
 - The PDE User Interfaces
 - The Classical PDE User Interfaces
 - Domain, Boundary, Pair, Edge, and Point Conditions for PDEs
 - Modeling Anisotropic Materials
-

Multiple Dependent Variables—Equation Systems

All PDE interfaces and equation forms support multiple dependent variables. In the case of several dependent variables u_1, u_2, \dots, u_N , a general form system of equations takes the following form:

$$\left\{ \begin{array}{ll} e_a^{lk} \frac{\partial^2 u_k}{\partial t^2} + d_a^{lk} \frac{\partial u_k}{\partial t} + \nabla \cdot \boldsymbol{\Gamma}_l = F_l & \text{in } \Omega \\ -\mathbf{n} \cdot \boldsymbol{\Gamma}_l = G_l + h_{ml} \mu_m & \text{on } \partial\Omega \\ 0 = R_m & \text{on } \partial\Omega_c \\ u_n = r_n & \text{on } \partial\Omega_d \end{array} \right. \quad (16-4)$$

The equation index l and k ranges from 1 to N , while the general constraint index m ranges from 1 to M_c and the Dirichlet condition index n ranges from 1 to M_d . The total number of constraints is therefore $M = M_c + M_d$. This discussion uses the summation convention. F_l, G_l, R_m and r_n are scalars, whereas $\boldsymbol{\Gamma}_l$ is a spatial vector. The mass and damping coefficients e_a and d_a are N-by-N matrices, while the constraint force Jacobian h is an M-by-N matrix. Note that in this case there are several Lagrange multipliers: $\mu_1, \mu_2, \dots, \mu_M$.

For a more compact form, let \mathbf{u} be a vector with components u_k , let $\boldsymbol{\Gamma}$ be a matrix with components Γ_{lj} , and so on. Then the system of equations takes on the same form as given in [Equation 16-1](#) for a single dependent variable.

It is also possible to write the system entirely on component form, where Γ_{lj} are components of the vector $\boldsymbol{\Gamma}_l$, and n_j components of the normal vector \mathbf{n} . Then the system of equations becomes:

$$\left\{ \begin{array}{ll} e_a^{lk} \frac{\partial^2 u_k}{\partial t^2} + d_a^{lk} \frac{\partial u_k}{\partial t} + \frac{\partial}{\partial x_j} (\Gamma_{lj}) = F_l & \text{in } \Omega \\ -n_j \Gamma_{lj} = G_l + h_{ml} u_m & \text{on } \partial\Omega \\ 0 = R_m & \text{on } \partial\Omega_c \\ u_n = r_n & \text{on } \partial\Omega_d \end{array} \right.$$

-
- 🔍
 - The General Form PDE
 - The General Form PDE User Interfaces
 - Boundary Condition Types
-

THE COEFFICIENT FORM EQUATION SYSTEM

The coefficient form of an equation system with N dependent variables u_1, u_2, \dots, u_N can be easily obtained from the general form PDE shown in [Equation](#) using the substitutions:

$$\left\{ \begin{array}{l} \Gamma_{lj} = -c^{l k j i} \frac{\partial u_k}{\partial x_i} - \alpha^{l k j} u_k + \gamma^{l j} \\ F_l = f_l - \beta^{l k i} \frac{\partial u_k}{\partial x_i} - a^{l k} u_k \end{array} \right.$$

Where index k and l run over dependent variables from 1 to N, while index i and j run over space dimensions from 1 to K. This means that for the case of a system of equations with N dependent variables in K space dimensions, the coefficients have the following sizes:

- e_a is an N-by-N matrix
- d_a is an N-by-N matrix
- c is an N-by-N-by-K-by-K four-dimensional array
- α is an N-by-N-by-K three-dimensional array
- β is an N-by-N-by-K three-dimensional array
- a is an N-by-N matrix

- f is an N-vector
- g is an N-vector
- q is an N-by-N matrix

-
- 
 - The Coefficient Form PDE
 - The Coefficient Form PDE User Interfaces
 - Boundary Condition Types
-

Solving Time-Dependent Problems

The general form equation shown in [Equation 16-1](#) as well as the coefficient form equation in [Equation 16-2](#) contain time-derivative terms of the same form. These terms only take effect for Time Dependent, Eigenvalue and Eigenfrequency study steps, and derived versions of these. When solving a Stationary, Frequency Domain or similar study step, all time derivatives are assumed to be zero. In this case, the value of the e_a and d_a coefficients does not matter.



To activate the d_a and e_a coefficients and convert the model into a time-dependent model, select a Time Dependent study.

When solving a Time Dependent study step, the *mass coefficient*, e_a , becomes important. The name *mass coefficient*, or *mass matrix* in case of a system of equations, stems from the fact that in many physics applications, e_a contains the mass density. The d_a coefficient in such equations usually represents damping of wave-like phenomena. However, if $e_a = 0$, then d_a is often called the mass coefficient instead. The default settings are $e_a = 0$ and $d_a = 1$, representing a parabolic time-dependent PDE such as the heat equation. Using $e_a = 1$ and $d_a = 0$ represents an undamped wave equation.



When solving a Time Dependent study step, the time variable is called t and may be used anywhere in equation coefficients. For other study steps, t is undefined. If you want to solve a model which depends explicitly on time using a Stationary study, you must first define a model parameter called t and give it a suitable value.

If, for a system of equations, the e_a matrix is nonzero and singular, or if $e_a = 0$ and d_a is singular, the system becomes a *differential-algebraic equation* (DAE) systems. The COMSOL solvers for time-dependent problems handle DAEs.



Time-Dependent Solver

USING MIXED SPACE-TIME DERIVATIVES

The coefficient forms in equation [Equation 16-2](#) only contains coefficients for pure space and time derivatives up to second order. The only directly available time-derivative coefficients are therefore e_a and d_a , using the subscript a because they are similar to the a coefficient in the absorption term except that they multiply $\partial^2 u / \partial t^2$ and $\partial u / \partial t$ instead of u . In analogy, it is possible to define coefficients e_c , \mathbf{e}_α , \mathbf{e}_β and d_c , \mathbf{d}_α , \mathbf{d}_β for mixed space-time derivatives, such that the equation becomes instead

$$e_a \frac{\partial^2 u}{\partial t^2} + \nabla \cdot \left(-e_c \nabla \frac{\partial^2 u}{\partial t^2} - \mathbf{e}_\alpha \frac{\partial^2 u}{\partial t^2} \right) + \mathbf{e}_\beta \cdot \nabla \frac{\partial^2 u}{\partial t^2} + \\ d_a \frac{\partial u}{\partial t} + \nabla \cdot \left(-d_c \nabla \frac{\partial u}{\partial t} - \mathbf{d}_\alpha \frac{\partial u}{\partial t} \right) + \mathbf{d}_\beta \cdot \nabla \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u + \dots) = \dots$$

These mixed coefficients are not directly available in the general or coefficient form PDE models. Instead, enter them in the existing γ and f terms:



In 1D, add `-d_c*uxt-d_a1*ut` to the γ term, and add `-d_be*uxt` to the f term, and similarly for second-order derivatives.



In 2D, add `-d_c*uxt-d_a11*ut` to the first γ component, and add `-d_c*uyt-d_a12*ut` to the second γ component. Add `-d_be1*uxt-d_be2*uyt` to the f term, and similarly for second-order derivatives.

USING TIME DERIVATIVES IN BOUNDARY CONDITIONS

To specify a flux or source boundary condition containing time-derivative terms as in

$$\mathbf{n} \cdot (c \nabla u + \dots) = g - e_q \frac{\partial^2 u}{\partial t^2} - d_q \frac{\partial u}{\partial t} - qu + h^T \boldsymbol{\mu},$$

simply 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 example, a [Global Equations](#) settings window.

 Constraints and Dirichlet boundary conditions must not contain time derivatives like `ut` and `utt` in the R and r coefficients unless they are enforced weakly, using weak constraints. See [Boundary Conditions](#).

Solving Eigenvalue Problems

When solving a PDE using an Eigenvalue study step, COMSOL Multiphysics assumes that all dependent variables vary with time as $u(t) = \hat{u}e^{-\lambda t}$, where \hat{u} is a complex amplitude field. Therefore the time derivatives in [Equation 16-1](#) and [Equation 16-2](#) are interpreted as

$$\begin{aligned}\frac{\partial u}{\partial t} &= -\lambda \hat{u} \\ \frac{\partial^2 u}{\partial t^2} &= \lambda^2 \hat{u}\end{aligned}$$

which for example leads to the general form eigenvalue PDE

$$\left\{ \begin{array}{ll} \lambda^2 \hat{u} - \lambda \hat{u} + \nabla \cdot \boldsymbol{\Gamma} = f & \text{in } \Omega \\ -\mathbf{n} \cdot \boldsymbol{\Gamma} = g - q \hat{u} + h^T \boldsymbol{\mu} & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ \hat{u} = r & \text{on } \partial\Omega_d \end{array} \right.$$

The eigenvalue solver further ignores any source or flux terms which are independent of the dependent variables.

As an alternative to defining eigenvalue PDEs using the time-derivative coefficients e_a and d_a , you may write the eigenvalue explicitly in the equations using the variable name `lambda`. For example, instead of specifying $e_a = 1$ you can set `a = lambda^2` with exactly the same result. In many cases, this formulation is preferable, in particular when

the eigenvalue problem does not arise from a time derivative in a time-harmonic assumption.



After solving an eigenvalue problem, the eigenvalue is always available for postprocessing under the variable name `lambda`, independently of whether the problem has been specified using the e_a and d_a coefficients or using the variable `lambda`.



Eigenfrequency studies are exactly analogous to Eigenvalue studies except that they in addition define the variable `freq` using the definition $\text{freq} = i\lambda/(2\pi)$. The variable name `freq` may be used in equations and postprocessing in the same way as `lambda`.

The eigenvalue solvers solve eigenvalue problems which are at most quadratic polynomials in the eigenvalue `lambda` exactly in one step. Therefore, damped eigenvalue solutions are easily found when both e_a and d_a are nonzero. Using the variable `lambda`, more complicated eigenvalue problems can be specified. Such problems must be solved using an iterative procedure.

Each time you run the eigenvalue solver, the PDE is expanded in a Taylor series in `lambda` around the eigenvalue linearization point λ_0 . Only the linear and quadratic terms are retained, while higher order terms are dropped. Running the solver repeatedly, updating the eigenvalue linearization point to the last eigenvalue found, will usually converge to an eigenvalue solving the full nonlinear eigenvalue problem.



[Eigenvalue Solver](#) and [Eigenvalue](#).

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. For more information about the test operator and other operators, see [Operators, Functions, and Constants](#).

All physics interfaces are implemented as weak form equations which you may study and modify in the Equation View nodes. COMSOL Multiphysics also converts all equation-based models specified in [The Coefficient Form PDE User Interfaces](#) and [The General Form PDE User Interfaces](#) to the weak form before solving.

In addition, it is possible in COMSOL Multiphysics to add extra weak form contributions and auxiliary variables to any physics interface in the model.

-
- 
 - [Physics Nodes—Equation Section and Equation View](#)
 - [About Auxiliary Equation-Based Nodes](#)
 - [The Weak Form PDE User Interfaces and Weak Form PDE](#)
-

Introduction to the Weak Form

The general form and coefficient form PDEs in equations [Equation 16-1](#) and [Equation 16-2](#) specify PDEs on *strong form*, in the sense that they in principle require the PDE to be satisfied at every point. And for this to be possible, all terms must be sufficiently continuous for derivatives and well-defined pointwise values to exist. In many cases, the natural phenomena a PDE intends to model are in fact best described as discontinuous, and may also contain source terms which are only defined as a total over a small region, without a well-defined pointwise value.

In these situations a weak equation turns out to be a better model of physics than can be provided by the more commonly used strong form PDEs. In addition, the weak form is particularly suitable for discretization and numerical solution using the finite element method. One reason for this is precisely the lower continuity requirement on

the solution, which only needs to be sufficiently smooth on each mesh element separately.

EXAMPLE: CONVERSION FROM GENERAL FORM TO WEAK FORM

As an example, consider the general form presented in [Equation 16-1](#), in particular the stationary form of the domain equation:

$$\nabla \cdot \Gamma = f$$

Assuming a single dependent variable u , introduce a corresponding arbitrary test function v . Multiply the equation by this test function and integrate over the domain:

$$\int_{\Omega} v \nabla \cdot \Gamma dV = \int_{\Omega} v f dV \quad (16-5)$$

Requiring this integral equation to hold is clearly a weaker statement than the original equation, in particular if [Equation 16-5](#) is required to hold only for all test functions v from a limited class of functions. In the finite element method, the test functions v (and also solution u) are usually limited to the set of piecewise polynomials of a given order on each mesh element.

This polynomial can also be written as a sum of individual *shape functions*. Therefore, the original strong form PDE is transformed into a weak form equation which must only be satisfied in a local integral sense over each shape function. When you increase the number of shape functions by refining the mesh or increasing the polynomial order, you simultaneously decrease the space of solutions u which can possibly satisfy [Equation 16-5](#). Therefore, well-posed and consistent finite element formulations converge towards the single solution u which satisfies the original strong form PDE.

To further simplify solution of [Equation 16-5](#), the left-hand side integral can be integrated by parts, using Gauss law:

$$-\int_{\Omega} \nabla v \cdot \Gamma dV + \int_{\partial\Omega} v \mathbf{n} \cdot \Gamma dA = \int_{\Omega} v f dV \quad (16-6)$$

This has two main advantages. First of all, it reduces the maximum order of spatial derivatives. If Γ is a function of the gradient of u , for example $\Gamma = -c\nabla u - au + \gamma$ as in the coefficient form PDE, the transformed weak equation now contains only first-order derivatives compared to second-order derivatives in the original strong form PDE. Secondly, it makes it clear what the *natural boundary condition* is for this equation. The second integral on the left-hand side disappears if the normal component of Γ

vanishes on the boundary. Alternately, if the value of the normal component is known, for example such that

$$-\mathbf{n} \cdot \boldsymbol{\Gamma} = g - qu + h^T \mu \quad (16-7)$$

on $\delta\Omega$, this value can be inserted as a boundary condition into the weak form equation, which then becomes

$$-\int_{\Omega} \nabla v \cdot \boldsymbol{\Gamma} dV = \int_{\Omega} vf dV + \int_{\delta\Omega} vG dA \quad (16-8)$$

This final weak formulation of the standard general form PDE therefore also explains why the Neumann boundary condition on the second line of [Equation 16-1](#) looks the way it does.

The Weak Form PDE

The weak form does not define any coefficients and does not even separate the different equations in a system of equations. When specifying a PDE and its boundary conditions on weak form, you specify contributions to a generic weak equation which for a 3D model reads:

$$0 = \sum_{i=1}^{N_3} \int_{\Omega_i} W_3^i dV_i + \sum_{j=1}^{N_2} \int_{\delta\Omega_j} W_2^j dA_j + \sum_{k=1}^{N_1} \int_{\delta^2\Omega_k} W_1^k dL_k + \sum_{m=1}^{N_0} \sum_{\delta^3\Omega_m} W_0^m \quad (16-9)$$

This weak equation has

- N_3 domain contributions W_3^i , each integrated over domain selection Ω_i
- N_2 boundary contributions W_2^j , each integrated over boundary selection $\delta\Omega_j$
- N_1 edge contributions W_1^k , each integrated over edge selection $\delta^2\Omega_k$
- N_0 point contributions W_0^m , each summed over point selection $\delta^3\Omega_m$

Note that all contributions are summed into the same integral equation without any particular order. Therefore, whether you write two equations in separate contributions or sum them into one single contribution normally does not matter. There is, however, a small caveat: the index on dV_i , dA_j and dL_k indicate that each contribution may be integrated in a different way. The integration may be performed with respect to either material or spatial coordinates, and in addition using different numerical quadrature

orders. But while working inside a single PDE interface, all contributions are integrated in the same way.

USING THE TEST OPERATOR

When specifying a weak contribution, you may use all variables normally available for evaluation in equation contributions and during postprocessing. This includes independent variables (coordinates), dependent variables and their derivatives, and other predefined and user-defined variables, parameters and constants. In addition, you must use the `test` operator to distinguish between test functions and the solution.



The `test` operator must always occur linearly in each weak form contribution. Contributions or terms without any `test` operator are ignored, while terms nonlinear in the `test` operator are considered an error.

In many cases, it is sufficient to let the `test` operator act directly on the dependent variables and their derivatives. For example, the weak form of the heat equation in two dimensions is `q*test(u) - k*(ux*test(ux)+uy*test(uy))`, where `q` is a heat source and `k` is the thermal conductivity. In other cases, it is more convenient to insert an expression or user-defined variable into the test function. One example of this is geometrically nonlinear solid mechanics, where the weak form can be written as a sum of terms `S_i*test(E_i)` where `S_i` is a stress measure and `E_i` its conjugate strain measure.

When a nonlinear expression, like for example a Green-Lagrange strain, is inserted into the `test` operator, the operator acts as a linear differential operator. This means that the argument expression is effectively first differentiated with respect to each dependent variable it contains and the results multiplied by the test function of the corresponding variable. Therefore, the heat equation may alternately be implemented as `q*test(u) - 0.5*k*test(ux^2+uy^2)`. Using the chain rule on the second term with the `test` operator as differential operator, returns the standard weak form of the equation given above.



For more information about the `test` operator and other operators, see [Operators, Functions, and Constants](#).

Specifying and Interpreting Boundary Conditions

The formulation of the boundary conditions in general form ([Equation 16-1](#)) and coefficient form ([Equation 16-2](#)) imposes both Dirichlet and Neumann conditions at the same time:

$$\left\{ \begin{array}{ll} -\mathbf{n} \cdot \boldsymbol{\Gamma} = g - qu + h^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega_c \\ u = r & \text{on } \partial\Omega_d \end{array} \right.$$

where $\boldsymbol{\Gamma}$ is the *flux vector* ($\boldsymbol{\Gamma} = -c\nabla u - au + \gamma$ for a coefficient form equation) and $\partial\Omega_c$ and $\partial\Omega_d$ are parts of the overall boundary, $\partial\Omega$, where general constraints and Dirichlet conditions have been specified. Combining conditions of different types on the same boundary is possible because of a new dependent variable μ , which is defined only on the boundary. This unknown variable μ is called a *Lagrange multiplier*, and usually has a physical interpretation. For example, 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 *constraint force Jacobian*. It decides how the Lagrange multipliers enforcing the constraint are scaled and distributed over the equations. The default settings in a [Constraint](#) node uses

$$h^T = -\left(\frac{dR}{du}\right)^T$$

while a [Dirichlet Boundary Condition](#) node by default corresponds to $h^T = 1$. For example:

- The Dirichlet condition is $u = r$ and the default constraint settings imply $h^T = 1$. The Neumann condition becomes:

$$-\mathbf{n} \cdot \boldsymbol{\Gamma} = g - qu + \mu$$

The Lagrange multiplier, μ , adjusts so as to satisfy the requested Dirichlet condition. Specifying a nonzero g changes the value of the Lagrange multiplier on the same boundary but does not affect the actual solution u . Therefore, this equation can usually be ignored, leaving effectively a pure Dirichlet condition.

- When no constraint is applied on a boundary, the value of R is zero, or equivalently, the Dirichlet condition reads $0 = 0$. Therefore h^T is zero and the Neumann condition is:

$$-\mathbf{n} \cdot \boldsymbol{\Gamma} = g - qu$$

This is the generalized Neumann condition without a Lagrange multiplier.



- [Boundary Condition Types](#)
- [The PDE User Interfaces](#)

EXAMPLE: SYSTEM OF TWO VARIABLES IN THE GENERAL FORM

The following example demonstrates a number of possible boundary condition combinations for a stationary system with two dependent variables u_1 and u_2 and two constraints when reaction terms are applied symmetrically on all physics. This is the default, and most useful, implementation. Written in general form:

$$\begin{cases} \nabla \cdot \boldsymbol{\Gamma}_1 = F_1 & \text{in } \Omega \\ \nabla \cdot \boldsymbol{\Gamma}_2 = F_2 & \text{in } \Omega \end{cases}$$

with the default Neumann boundary conditions

$$\begin{cases} -\mathbf{n} \cdot \boldsymbol{\Gamma}_1 = G_1 + \frac{\partial R_1}{\partial u_1} \mu_1 + \frac{\partial R_2}{\partial u_1} \mu_2 & \text{on } \partial\Omega \\ -\mathbf{n} \cdot \boldsymbol{\Gamma}_2 = G_2 + \frac{\partial R_1}{\partial u_2} \mu_1 + \frac{\partial R_2}{\partial u_2} \mu_2 & \text{on } \partial\Omega \end{cases}$$

writing out the symmetric application of reaction terms on all dependent variables, and the Dirichlet boundary conditions:

$$\begin{cases} 0 = R_1 & \text{on } \partial\Omega \\ 0 = R_2 & \text{on } \partial\Omega \end{cases}$$

The same set of boundary conditions are accessible in all PDE interfaces. To illustrate the flexibility of the [Constraint](#) boundary condition $R = 0$, consider these cases:

Case 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:

$$\begin{cases} -\mathbf{n} \cdot \Gamma_1 = G_1 & \text{on } \partial\Omega \\ -\mathbf{n} \cdot \Gamma_2 = G_2 & \text{on } \partial\Omega \end{cases}$$

Case 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 $u_2 = r_2$. Using default settings for the constraint reaction terms,

$$h = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and 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 last 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.

Case 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 default Neumann conditions including reaction terms 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.

Case 4: The same as Case 3 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: $u_1 = r_1$. By default, 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, the PDEs and the Dirichlet conditions are identical, while the Neumann conditions are different. In fact, both the Dirichlet and the Neumann conditions are now applied on u_1 , and nothing is specified for u_2 .



This example shows that when mixing Dirichlet and Neumann conditions on Coefficient Form PDEs and General Form PDEs, 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.

Case 5: Finally, let $R_1 = u_2 - u_1$ and $R_2 = 0$. Also, assume that u_1 and u_2 exist on two adjacent domains rather than on the same domain. The normal vectors as seen from the two sides are then $\mathbf{n}_1 = -\mathbf{n}_2 = \mathbf{n}$. 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 using the default symmetric reaction terms are:

$$\begin{cases} -\mathbf{n}_1 \cdot \Gamma_1 = G_1 - \mu_1 & \text{on } \partial\Omega \\ -\mathbf{n}_2 \cdot \Gamma_2 = G_2 + \mu_1 & \text{on } \partial\Omega \end{cases}$$

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, the Neumann conditions cannot be ignored. Instead, adding the two conditions, it becomes apparent that the solution and flux on the boundary must fulfill:

$$\begin{cases} 0 = u_2 - u_1 & \text{on } \partial\Omega \\ -\mathbf{n}_1 \cdot \Gamma_1 - \mathbf{n}_2 \cdot \Gamma_2 = G_1 + G_2 & \text{on } \partial\Omega \end{cases}$$

In particular, if $G_1 = G_2 = 0$, the last condition simplifies to:

$$-\mathbf{n} \cdot (\Gamma_1 - \Gamma_2) = 0 \text{ on } \partial\Omega$$

This means that both the variables u_1 and u_2 and the corresponding fluxes are equal at the boundary. In fact, if u_1 and u_2 represent the same quantity, this is the same continuity condition that holds implicitly at every mesh element boundary in the model, where nothing else has been specified.



In all of these examples, the values of the Lagrange multipliers do not matter. However, they often have a physical significance. In structural mechanics, the term $h^T \mu$ in the Neumann condition is the reaction force necessary to satisfy the kinematic constraints described by the Dirichlet conditions.

Symmetric and Nonsymmetric Constraints

Constraints formulated through the coefficient R in [The Coefficient Form PDE User Interfaces](#) and [The General Form PDE User Interfaces](#) by default give rise to *globally symmetric bidirectional* constraints. This happens when the constraint settings specify that reaction terms are to be applied symmetrically on all physics.

A bidirectional symmetric 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 in this case

$$\mathbf{n} \cdot (c\nabla u + au - \gamma) = g - qu - \left(\frac{\partial R}{\partial u}\right)^T \boldsymbol{\mu}$$

and for the general form, the flux condition is

$$-\mathbf{n} \cdot \Gamma = g - qu - \left(\frac{\partial R}{\partial u}\right)^T \boldsymbol{\mu}$$

The last term on the right-hand side in both expressions is the globally symmetric constraint reaction term, or *generalized constraint force*. Thus, with symmetric 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 symmetric constraints. As an example, consider 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 if the default reaction term definition is used. This is because the globally symmetric constraint force is not zero:

$$-\mathbf{n} \cdot \Gamma_2 = G_2 + \frac{\partial R_1}{\partial u_2} \boldsymbol{\mu}_1 + \frac{\partial R_2}{\partial u_2} \boldsymbol{\mu}_2 = G_2 + \frac{\partial r_1}{\partial u_2} \boldsymbol{\mu}_1 \neq G_2$$

To remedy this limitation with bidirectional constraints, the **Constraint Settings** section allows you to **Apply reaction terms** on either dependent variables from **Current physics (internally symmetric)** or **Individual dependent variables**. Both options imply a unidirectional and possibly nonsymmetric constraint in the sense that some dependent variables are considered as constants for the purpose of enforcing the constraint.



To display the **Constraint Settings** section in Constraint nodes, click the **Show** button () on the **Model Builder** tool bar and select **Advanced Physics Options**.

When constraint reaction terms are applied only on the current physics, flux conditions in other interfaces are left untouched by the constraint. If reaction terms are applied

only to individual variables, this leaves flux conditions untouched on all but the specific variables. For the above example, both settings have the same desired effect if u_1 and u_2 belong to different interfaces. If these belong to the same interface, applying reaction terms to **Current physics (internally symmetric)** has the same effect as the default application to **All physics (symmetric)**.

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. Symmetric 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 default bidirectional symmetric constraint attempts to apply the test function on the time derivative of u , which is not supported. The solution is to apply the reaction terms on **Individual dependent variables**. Note that the 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 both in a pointwise sense and in a weak sense.



Turbulent fluid flow requires the CFD Module or Heat Transfer Module.



For descriptions about how to use even more general pointwise and weak nonsymmetric constraints, see [Pointwise Constraint](#) and [Weak Constraint](#), respectively. Also see [Boundary Condition Types](#).

The PDE User Interfaces

COMSOL Multiphysics® includes different PDE user interfaces for equation-based modeling, distinguished by the equation formulation used for entering the equations—Coefficient Form, General Form, and Weak Form. The user interfaces are identical except for the default node added to the top geometric entity level where the user interface is active. You can still use a General Form to specify equations in a Coefficient Form PDE user interface, or add Weak Form contributions to a General Form PDE user interface.



The Wave Form PDE is also available and described in another section.



- [Modeling with PDEs](#)
- [Notational Conventions](#)

Creating a Model Using a PDE User 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 interfaces 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 (➔) on the **Model Wizard** window toolbar.
- 5 Optionally, choose a **Study Type**.
- 6 Click the **Finish** button (✅).

SPECIFYING A SYSTEM OF EQUATIONS

COMSOL Multiphysics allows the creation of equations with more than one dependent variable. To do this, on the **Add Physics** page under **Dependent variables**, enter the **Number of dependent variables** in the field. COMSOL then automatically assigns variable names, typically u_1 , u_2 , u_3 , and so on. You can also edit the default variable name (as long as it is valid and unique) in the **Dependent variables** table. Several scalar PDEs can also be coupled using a multiphysics approach



For any form of PDE interface you add to a model, additional equation nodes can be added in Coefficient Form, General Form, or Weak Form.

MODELING WITH PDES ON BOUNDARIES, EDGES, AND POINTS

The **Coefficient Form PDE**, **General Form PDE**, and **Weak Form PDE** are also available on boundaries, edges, and at points in the geometry.

Extra weak equations can be added by adding auxiliary dependent variables to a **Weak Contribution (PDEs)** node. Use such weak form equations as a way to handle thin layers; COMSOL then solves the problem by modeling rather than meshing. This approach reduces the solution time.



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.



- Modeling with PDEs
- Notational Conventions
- The PDE User Interfaces

Discretization Section Shape Function Types and Element Orders

The PDE and weak form interfaces have different shape functions available with the associated element order (the order of the shape functions). Select the **Shape function type** and the **Element order** as, in most cases, **Constant**, **Linear**, **Quadratic**, **Cubic**, **Quartic**, or **Quintic** (order 0–5, respectively). [Table 16-2](#) is an overview of the available shape function types and the element orders supported.



Not all shape functions are available for all space dimensions and types of equations, and not all shape functions support all orders.

TABLE 16-2: 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)

The Coefficient Form PDE User Interfaces

The **Coefficient Form PDE (c)** interface (Δu) found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, covers many well-known PDEs.

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 **Coefficient Form PDE** node to add other nodes that implement other boundary conditions, for example.

	The Coefficient Form PDE interface is also available in other forms from the PDE interfaces>Lower Dimensions submenu— Coefficient Form Boundary PDE , Coefficient Form Edge PDE , and Coefficient Form Point PDE . Also see Modeling with PDEs on Boundaries, Edges, and Points .
	The Coefficient Form PDE discusses the formulation and settings pertaining to the coefficient form, as well as the general PDE terminology used in COMSOL Multiphysics.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `c` (in domains), `cb` (on boundaries), `ce` (on edges), or `cp` (at points).

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

The default setting is to include **All domains**, **All boundaries**, **All edges**, or **All points** in the model. To choose specific geometric entities, select **Manual** from the **Selection** list.

UNITS

By default, the PDE interfaces are dimensionless, but units can be defined 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. Select the units from a list of physical quantities or enter the unit directly.

From the list, select the **Dependent variable quantity** that defines the unit for the dependent variable u . The default is **Dimensionless [1]**. Select **None** to enter a unit (for example, K, m/s, or mol/m³) in the **Unit** field.

Select the **Source term quantity** that defines the unit for the source term f (the unit for the right—and left—side of the PDE). **None** is the default quantity, and m⁻² is the default **Unit**, which is a consistent with a dimensionless dependent variable. Enter another unit (for example, W/m³ or A/m³) in the **Unit** field as required.



For the **Classical PDE>Heat Equation** interface, the **Dependent variable quantity** defaults to **Temperature (K)** and the **Source term quantity** defaults to **Heat source (W/m³)**.

DEPENDENT VARIABLES

Enter the **Number of dependent variables** (the default is 1) and set the field and dependent variable names. The default **Field name** and **Dependent variables** 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) 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 a **Shape function type** (finite element types)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

The **Compute boundary fluxes** check box is selected by default. Click to clear the check box as required. The default **Value type when using splitting of complex variables** is **Complex**. Or select **Real**.

- 
- Modeling with PDEs
 - Discretization Section Shape Function Types and Element Orders
 - Computing Accurate Fluxes
 - Compile Equations
-



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**.

The General Form PDE User Interfaces

The **General Form PDE (g)** interface (Δu), found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, is a flexible way to specify PDEs in a general form.

When this user 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 General Form PDE interface is also available in other forms from the **PDE interfaces>Lower Dimensions** submenu—**General Form Boundary PDE**, **General Form Edge PDE**, and **General Form Point PDE**. Also see [Modeling with PDEs on Boundaries, Edges, and Points](#).



The **General Form PDE** discusses the formulation and settings pertaining to the general form.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique.

Only letters, numbers and underscores (_) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is **g** (in domains), **gb** (on boundaries), **ge** (on edges), or **gp** (at points).



See [The Coefficient Form PDE User Interfaces](#) remaining settings.



- [General Form PDE](#)
- [Domain, Boundary, Pair, Edge, and Point Conditions for PDEs](#)
- [Modeling with PDEs](#)

The Weak Form PDE User Interfaces

The **Weak Form PDE (w)** user interface, found under the **Mathematics>PDE Interfaces** branch (**Δu**) in the **Model Wizard**, are identical to [The Coefficient Form PDE User Interfaces](#) and [The General Form PDE User Interfaces](#) except for the default node on the top geometric entity level being a **Weak Form PDE** node.



The Weak Form PDE interface is also available in other forms from the **PDE interfaces>Lower Dimensions** submenu—**Weak Form Boundary PDE**, **Weak Form Edge PDE**, and **Weak Form Point PDE**. Also see [Modeling with PDEs on Boundaries, Edges, and Points](#).

In all interfaces, weak expressions can be added, which COMSOL Multiphysics adds to the overall equation. Adding one of these interfaces creates a **PDE** node (**fdu**) for PDE modeling using a weak formulation. The same type of Weak Form PDE node may also be added 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 **Weak Form 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 used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `w` (in domains), `wb` (on boundaries), `we` (on edges), or `wp` (at points).



See [The Coefficient Form PDE User Interfaces](#) for the rest of the settings.



- [Weak Form PDE](#)
- [Domain, Boundary, Pair, Edge, and Point Conditions for PDEs](#)
- [Modeling with PDEs](#)

The Classical PDE User Interfaces

Many classical PDEs are instances of Coefficient Form PDEs. The classical PDEs have their own interfaces found under the **Mathematics>Classical PDEs** branch ([▼2](#)) in the **Model Wizard**. Classical PDEs can also be added to all of the forms of PDE interfaces as domain nodes.

The following Classical PDE user interfaces and nodes are available. All of these have the same settings as [The Coefficient Form PDE User Interfaces](#):

- [Laplace Equation](#)
- [Poisson's Equation](#)
- [Wave Equation](#)
- [Helmholtz Equation](#)

- **Heat Equation**
 - **Convection-Diffusion Equation**
-



- Compact and Standard Notations for Classical PDEs
 - Domain, Boundary, Pair, Edge, and Point Conditions for PDEs
-

Domain, Boundary, Pair, Edge, and Point Conditions for PDEs

The PDE interfaces have the following domain, boundary, pair, edge, and point conditions described in this section and listed in alphabetical order. Some nodes are selected from the **Classical PDEs** submenu:

- Coefficient Form PDE
- Constraint
- Convection-Diffusion Equation
- Dirichlet Boundary Condition
- Flux/Source
- General Form PDE
- Heat Equation
- Helmholtz Equation
- Initial Values
- Laplace's Equation
- Periodic Condition
- Poisson's Equation
- Source, Edge Source, and Point Source
- Wave Equation
- Weak Form PDE
- Zero Flux

There are also auxiliary equation-based nodes found under the **More**, **Edges**, and **Points** submenus. To display these submenus in the context menu, click the **Show**

button () on the **Model Builder** toolbar and select **Advanced Physics Options**. Then choose from the following (listed in alphabetical order):

- Auxiliary Dependent Variable
- Discretization (Node)
- Pointwise Constraint
- Weak Constraint
- Weak Contribution (PDEs)
- Weak Contributions on Mesh Boundaries



There is generally a **More** submenu for the domain level as well as one for the boundary level on a physics context menu. See [Physics Node Context Menu Layout](#) for an example.



- Modeling with PDEs
 - The PDE User Interfaces and [Classical PDE Domain Nodes](#)
-

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, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

INITIAL VALUES

Enter a value or expression for the **Initial value for u** , u (dimensionless) and the **Initial time derivative of u** $\frac{\partial u}{\partial t}$ (SI unit: 1/s). The defaults are 0 for both dependent variables.

Coefficient Form PDE

The **Coefficient Form PDE** node is the default equation for [The Coefficient Form PDE User Interfaces](#), and is available for the other forms from the context menu. Specify the

coefficients for a coefficient form PDE (see [The Coefficient Form PDE](#) and [Equation 16-2](#))

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

DIFFUSION COEFFICIENT

Enter a value or expression for the diffusion coefficient c . 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.

-
- 
- [Interpreting PDE Coefficients](#)
 - [Specifying and Interpreting Boundary Conditions](#)
 - [The PDE User Interfaces](#)
 - [Modeling with PDEs](#)
-

General Form PDE

The **General Form PDE** node is the default equation for [The General Form PDE User Interfaces](#), and it is available for the other forms from the context menu. Specify the coefficients for a general form PDE (see [The General Form PDE](#) and [Equation 16-1](#)).



Except for Conservative Flux described in this section, see [Coefficient Form PDE](#) for the rest of the settings.

CONSERVATIVE FLUX

Enter 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.

-
- 
- [The PDE User Interfaces](#)
 - [Interpreting PDE Coefficients](#)
 - [Modeling with PDEs](#)
-

Weak Form PDE

The **Weak Form PDE** node is the default node on the top geometric entity level in [The Weak Form PDE User Interfaces](#) and may also be added to [The Coefficient Form PDE User Interfaces](#) and [The General Form PDE User Interfaces](#). It contains one weak

form expression for each dependent variable in the interface (see [Equation 16-9](#)).

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

WEAK EXPRESSIONS

Enter the weak expressions that COMSOL Multiphysics (together with any other weak expressions on the same domain) sets equal to 0 in the **weak** field. For example, in a 2D model with one dependent variable, the default expression on the domain level is $-test(ux)*ux-test(uy)*uy+1[m^{-2}]*test(u)$. This is the weak formulation of Poisson's equation with the right-hand side $f = 1$. On other geometric entity levels, the default weak expression is 0.



- [The PDE User Interfaces](#)
 - [Modeling with PDEs](#)
-

Source, Edge Source, and Point Source

You can add additional source term nodes on different geometry levels: **Source** on domains, **Edge Source** on edges (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.

SOURCE TERM

Enter a value or expression for the source term f . The default is 0.

Classical PDE Domain Nodes

The nodes available from the **Classical PDEs** submenu can be added to any PDE interface at the domain level. The same node is also available as its own interface from the **Mathematics>Classical PDEs** branch ( 2) in the **Model Wizard**.



See **Coefficient Form PDE** for all the settings and **Compact and Standard Notations for Classical PDEs** for the equations that the Classical PDE user interface solve.

The available interfaces and domain nodes are:

LAPLACE'S EQUATION

The **Laplace Equation** is a classic PDE of elliptic type that can describe the behavior of some kind of potential or the steady-state heat equation.

POISSON'S EQUATION

The **Poisson's Equation** is a classical PDE of elliptic type that can describe, for example, electrostatics with a space charge density.

HELMHOLTZ EQUATION

The **Helmholtz Equation** is a classical PDE of elliptic type that can represent, for example, a time-independent form of the wave equation.

WAVE EQUATION

The **Wave Equation** is a classic PDE of hyperbolic type. It is a second-order PDE that describes waves, such as sound waves, light waves, and water waves.

HEAT EQUATION

The **Heat Equation** is a classical PDE of parabolic type that describes time-dependent heat transfer by diffusion or other diffusion processes.

CONVECTION-DIFFUSION EQUATION

The **Convection-Diffusion Equation** is a classical PDE that describes time-dependent transport by convection and diffusion.

Dirichlet Boundary Condition

The **Dirichlet Boundary Condition** specifies a value of u on the boundary of the domain: $u = r$. By default, it is a unidirectional condition, applying reaction terms on u but not on any variables appearing in r .

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the boundaries, edges, or points to define or select **All boundaries**, **All edges**, or **All points** from the list.

DIRICHLET BOUNDARY CONDITION

The Dirichlet boundary condition for each dependent variable (for example, u_2), has a corresponding check box (**Prescribed values for u2**), which is selected by default.

Enter a value or expression for the prescribed value in the associated text field or click to clear the check box as required.

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 **Apply reaction terms on** list, select **All physics (symmetric)** to apply reaction terms bidirectionally on u as well as on any other dependent variables appearing in r . Select instead **Current physics (internally symmetric)** to apply reaction terms on all components of u or **Individual dependent variables** to apply reaction terms from each active component in the Dirichlet condition only on the corresponding component of u . The latter is the default setting and effectively implies a unidirectional constraint.
- Select the **Use weak constraints** check box to use weak constraints instead of the standard pointwise constraints.



For an overview of the differences between the constraint types and cases where a unidirectional constraint is required, see [Symmetric and Nonsymmetric Constraints](#).

Constraint

The **Constraint** boundary condition specifies an expression R which is constrained to be equal to zero on the selection, $R = 0$. By default, this is a bidirectional constraint, meaning that all variables in R are affected by reaction terms.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the boundaries, edges, or points to define or select **All boundaries**, **All edges**, or **All points** from the list.

CONSTRAINT

Enter a value or expression for the value of R in the constraint $R = 0$. For example, to constrain u to 2, enter $2-u$ in the field for R .



The sign in front of u in the constraint controls the sign of the implicit Lagrange multiplier μ , as well as the sign of reaction forces computed using the `reacf()` operator. For consistency with the way predefined physics interfaces implement constraints, write $2-u$ rather than $u-2$ in 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 **Apply reaction terms on** list, select **All physics (symmetric)** to apply reaction terms bidirectionally on any dependent variables appearing in R . This is the default method for enforcing the constraint. Select instead **Current physics (internally symmetric)** to apply reaction terms only on components of u or **Individual dependent variables** to apply reaction terms from each constraint component only on the corresponding component of the dependent variable vector u .
- Select the **Use weak constraints** check box to use weak constraints instead of the standard pointwise constraints.



For an overview of the differences between the constraint types and cases where a unidirectional constraint is required, see [Symmetric and Nonsymmetric Constraints](#).

Flux/Source

The **Flux/Source** boundary condition adds a flux or source g on the boundary:

$$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) = g - qu \text{ or } -\mathbf{n} \cdot \Gamma = g - qu$$

The first equation describes this boundary condition for a Coefficient Form PDE, and the second term describes it for a General Form PDE. The g term may contain a general expression of the dependent variables. The q coefficient simplifies the implementation of a *Robin boundary condition* by including a term on the form qu , where u is the dependent variable.

BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the boundaries, edges, or points to define or select **All boundaries**, **All edges**, or **All points** from the list.

BOUNDARY FLUX/SOURCE

Enter a value or expression for the value of the boundary flux or source g in the corresponding field or fields. 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 or fields. 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:

$$\mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) = 0 \text{ or } \mathbf{n} \cdot \Gamma = 0$$

BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries, edges, or points to define or select **All boundaries**, **All edges**, or **All points** as required.

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 antiperiodic so that $u(x_0) = -u(x_1)$ and can control which 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, right-click **Periodic Condition** and add a **Destination Selection** subnode to control the destination. By default it contains the selection that COMSOL Multiphysics has identified.



Also, in the **Model Builder** under **Model>Definitions**, a “read only” **Explicit** selection node is added and 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 u_1** check box. By default, the periodic condition applies to all dependent variables.



Periodic Boundary Conditions

Destination Selection

Right-click a **Periodic Condition** node to add the **Destination Selection** subnode and to change the selection for the destination. The selection that COMSOL Multiphysics makes appears as the default selection in the **Selection** list (as **Explicit 1**, for example).

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. By default it contains the selection that COMSOL Multiphysics has identified.

	The software usually automatically identifies the boundaries as either source boundaries or destination boundaries. By default it contains the selection that COMSOL Multiphysics identifies. A “read only” Explicit selection node () (found in the Model Builder under Model > Definitions) shows the selected destination boundaries.
	Periodic Boundary Conditions

Theory for the Wave Form PDE

The Wave Form PDE theory is described in this section.

Derivation of the Weak Form of the Wave Form PDE

Using [The Wave Form PDE](#) it is possible to solve one or several first-order wave equations; that is, PDEs of the form

$$d_a \frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{\Gamma}(u) = f \quad (16-10)$$

where u is the unknown, d_a the mass coefficient, f the source, and $\boldsymbol{\Gamma}$ the flux vector, which generally depends on u .

The numerical method consists of a discontinuous Galerkin (DG) method in space in combination with explicit Runge-Kutta time stepping. This combination of space and time discretization is particularly well-suited for wave problems due to a favorable CFL condition. This is true even when using a high-order polynomial ansatz for u .

In order to derive the weak form underlying the DG method, let $\{\Omega^e\}$ denote a mesh of the domain $\{\Omega\}$, with Ω^e denoting a single element. On this mesh, let V be the broken space

$$V = \bigoplus P^s(\Omega^e)$$

with P^s denoting the space of all polynomials of degree at most s on Ω^e .



These functions are continuous in the interior of each mesh element, but generally discontinuous across the element boundaries.

A basis for V is given by the nodal discontinuous Lagrange shape functions `shhwdisc`. This basis is tailor made for this type of DG method, and it has nearly optimal interpolation properties.

The starting point for deriving a weak form is to multiply the PDE with a test function $v \subset V$ and integrate over the domain, to yield

$$\int_{\Omega} d_a \frac{\partial u}{\partial t} v dA + \int_{\Omega} \nabla \cdot \Gamma v dA = \int_{\Omega} f v dA$$

The next step is to integrate by parts. Some care must be taken, since the integrands v and u are discontinuous functions across element boundaries and only continuous in the interior of each element. Therefore, the integrals are first written as a sum over the elements and then integration by parts is done on each element, which gives

$$\sum_e \int_{\Omega^e} d_a \frac{\partial u}{\partial t} v dA - \sum_e \int_{\Omega^e} (\Gamma \cdot \nabla v) dA + \sum_e \int_{\partial \Omega^e} (\mathbf{n} \cdot \Gamma^*) v ds = \sum_e \int_{\Omega^e} f v dA$$

where \mathbf{n} is the outward unit normal on the element. Further, Γ^* is the so-called numerical flux, which defines the flux vector on each element boundary. The flux vector is usually discontinuous because it depends on u .

The numerical flux defines how adjacent elements are connected and how continuous u will be. Different definitions of the numerical flux lead to different variants of DG methods.

The numerical flux implemented in COMSOL Multiphysics is the global Lax-Friedrichs flux:

$$\mathbf{n} \cdot \Gamma^* = \mathbf{n} \cdot \langle \Gamma \rangle + \tau[u]$$

where the angles $\langle \rangle$ and brackets $[]$ are the average and jump operators, respectively. Thus, on each element boundary this flux is simply the average of the flux on the two adjacent elements sharing the face, plus a penalty on any jumps of the solution. The penalty is needed for stability and is proportional to the parameter τ , which is assumed to be constant over the whole domain Ω . Using the definition of the Lax-Friedrichs flux, the weak form is obtained

$$\sum_e \int_{\Omega^e} d_a \frac{\partial u}{\partial t} v dA - \sum_e \int_{\Omega^e} (\Gamma \cdot \nabla v) dA + \sum_e \int_{\partial \Omega} (\mathbf{n} \cdot \langle \Gamma \rangle + \tau[u]) v ds = \sum_e \int_{\Omega^e} f v dA$$

What makes this DG method particularly attractive for explicit time stepping is the fact that the term

$$d_a \frac{\partial u}{\partial t}$$

yields a block diagonal mass matrix, where each block only involves the degrees of freedom on each element. As a consequence, there is no need to invert, or solve any linear system involving the global mass matrix. The inverse of the global mass matrix simply amounts to inverting the local mass matrix on each element. This is efficiently done using high performance routines such as BLAS or LAPACK.

A known drawback with explicit time stepping is the requirement on the time step, which has to be very small in order to obtain a stable numerical method. This is referred to as the CFL condition, which relates the largest possible time step k to the smallest mesh size h . For wave equations with unit wave speed, the CFL condition takes the form

$$k \leq C \frac{h}{p^2}$$

where p is the order of the shape functions and C a generic constant, typically 0.25.

As implemented in COMSOL, the *nodal discontinuous Lagrange shape functions* are the only set of shape functions defined for this interface. The associated element order can be chosen from the **Element order** list. The highest available order is four, and the default order is two.



The nodal discontinuous Lagrange shape functions can only be defined on triangular meshes in 2D or tetrahedral meshes in 3D.



See [Discretization](#) as defined for [The Wave Form PDE](#).

Time Explicit Integrator

After discretization in space, an explicit ordinary system of differential equations is obtained. The standard procedure is to integrate the DG system of equations with the explicit Runge-Kutta family of methods.

The combination is in the literature denoted RK-DG. Often, a $p+1$ order RK method is combined with p :th order shape functions. See [Ref. 1](#) for details. The new integrator

supports a CFL-based time step regulator. The stability limitation for the time step k is of the sort

$$k \leq \frac{C}{(1+2p)^2} \min \left| \frac{h}{\lambda \left(\frac{\partial \Gamma}{\partial u} \right)} \right|$$

where C is a moderate constant.

COMSOL Multiphysics can compute the largest time step k , based on the smallest mesh size h , the order of the shape functions, and the maximum wave speed in the domain. Under **Time stepping**, select **From expressions** to automatically compute the largest stable time step k . Under **Cell time scale expressions** add `wahw.wtc` as the variable for the **Estimate of Maximum Wave Speed** when using [The Wave Form PDE](#).



- [Time Explicit Solver](#)
- [Solver Overview](#)

Local Time Stepping

The time step restriction is directly proportional to the smallest mesh element size. The CFL condition is normally too restrictive on highly graded meshes. On such a mesh it is common that only few elements are small, yet these dictate the overall time step for the whole problem. As an example, think of a mesh with a small geometric feature somewhere or a mesh stemming from an adaptive computation.

In these cases one option is to use local time stepping, which allows the use of a larger time step based on the size of the majority of the elements. This is possible due to the element-wise nature of the DG scheme. The larger the spread is between the smallest mesh-element size and the ideal mesh-element size dictated by a points per wavelength argument, the more beneficial this technique is.

The basic problem with local time stepping, aside from stability issues, is to obtain high accuracy. Classical results only involve second order accuracy. This is not good enough for DG methods.

In order to perform local time stepping, the third order classical Adams-Bashforth (AB) method is implemented

$$u_{n+1} = u_n + \frac{k}{12}(23R(u_n) - 16R(u_{n-1}) + 5R(u_{n-2})) \quad (16-11)$$

here, u_n is the solution at time t_n , k is the time step, and $R(u_n)$ the weak form $f - \nabla \cdot \Gamma$ of the DG scheme (compare to the weak formulation of [Equation 16-10](#)), which includes the numerical flux.

The basic idea with local time stepping is as follows. First, the elements are divided into groups, based on their size. Typically, the groups are constructed such that the time steps k , $k/2$, $k/4$, and so forth, are stable time steps for each group of elements and where any coupling between groups is disregarded. This allows for easy synchronization of the solution at every full step. The right-hand-side vectors

$$R(u_n), R(u_{n-1}), \text{ and } R(u_{n-2})$$

are expensive to compute, thus these are naturally stored from previous time steps, so for each group of DOFs, their own history of right-hand sides are stored.

The main idea with local time stepping is to match the different groups with their own time step and thus save computational resources.



- [Time Explicit Solver](#)
 - [Solver Overview](#)
-

Reference for the Wave Form PDE User Interface

1. Jan S. Hesthaven and Tim Warburton, *Nodal Discontinuous Galerkin Methods—Algorithms, Analysis, and Applications*, Springer, 2008.

The Wave Form PDE



To enable this user interface, select **Options>Preferences>Show** from the main menu. Then under **Wave Form PDE**, click to select the **Show Wave Form PDE interface in Model Wizard** check box. Click **OK** and then go to the Model Wizard to add this interface.

The **Wave Form PDE** user interface (Δu), found under the **Mathematics>PDE Interfaces** branch (Δu) in the **Model Wizard**, solves PDEs with first-order derivatives in time and space using optimized algorithms with respect to speed and memory consumption. This section covers the formulation and settings pertaining to those equations.

When you add this user interface, these default nodes are also added to the **Model Builder—Wave Form PDE**, **Zero Flux**, and **Initial Values**. Right-click **Wave Form PDE** to add other nodes that implement other boundary conditions, for example.

Use the *wave form* for first-order hyperbolic PDEs. Assuming a scalar equation for the dependent variable u , these problems take the form

$$\begin{aligned} d_a \frac{\partial u}{\partial t} + \nabla \cdot \Gamma &= f && \text{in } \Omega \\ -\mathbf{n} \cdot \Gamma &= g && \text{on } \partial\Omega \end{aligned}$$

together with suitable initial data. The first equation is the PDE, the second the boundary conditions.

The terms d_a , Γ , f , and g are coefficients. They can be functions of both the spatial coordinates or time, and the solution u , but not the derivatives of u . The coefficients f and g are scalar, whereas Γ is the *flux vector*. The coefficient d_a is assumed to be nonzero throughout the domain Ω , and for all times.



The **Wave Form PDE** user interface also supports systems of equations. The interpretation of the coefficients are the same as for the scalar case; f and g are vectors with one component for each equation. Γ contains one flux vector for each equation, and d_a is a square (invertible) matrix.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `wahw`.

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.

UNITS

Select a **Dependent variable quantity**. **Dimensionless (I)** is the default and there are many other options to choose from. Select a **Source term quantity**. **None** is the default and there are many other options to choose from. Enter a **Unit**.

DISCRETIZATION

To display this section, click the **Show** button (☞) and select **Discretization**.

Due to efficiency reasons, there is only one **Shape function type** (finite element type) defined for this interface—the **Nodal discontinuous Lagrange** functions. The associated element order (the order of the shape function for the element) can be chosen from the **Element order** list. The highest available order is 10 in 1D and 2D models and 4 in 3D models, and the default order is two.



The Nodal Discontinuous Lagrange shape functions can only be defined on triangular meshes in 2D, or tetrahedral meshes in 3D.

DEPENDENT VARIABLES

Define the number of the dependent variables and the variable names. The default name for a scalar PDE variable is u .



The dependent variables and their names must be unique with respect to all other dependent variables in the model.

Add or remove dependent variables in the model and also change their names.

Enter the **Number of dependent variables**. Use the **Add dependent variable** (+) and **Remove dependent variable** (-) buttons as required.

-
- [Domain and Boundary Physics for the Wave Form PDE User Interface](#)
• [Show More Physics Options](#)
• [Theory for the Wave Form PDE](#)
• [Using Units](#)
-

Domain and Boundary Physics for the Wave Form PDE User Interface

The [Wave Form PDE](#) includes the following domain and boundary physics nodes:

- [Flux/Source](#)
- [Initial Values](#)
- [Interior Flux](#)
- [Interior Source](#)
- [Wave Form PDE](#)
- [Zero Flux](#)

Wave Form PDE

This is the default equation for a **Wave Form PDE** interface. Here the coefficients for a wave form PDE are specified with the following equation coefficients:

$$d \frac{\partial u}{\partial t} + \nabla \cdot \Gamma(u) = f$$

- d_a is the mass coefficient
- $\Gamma(u)$ is the conservative flux vector
- f is the source term

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.

DAMPING OR MASS COEFFICIENT

Enter a value or expression for the damping (or mass) coefficient d_a (SI unit: s/m^2). The default is $1\text{ s}/m^2$. If there are multiple dependent variables, there is a matrix of d_a component inputs.

CONSERVATIVE FLUX

Enter values or expressions for the components of the conservative flux vector $\Gamma(u)$ (SI unit: $1/m$). These components may depend on both the spatial and temporal coordinates, and the solution u , but not any derivatives of u . If there are multiple dependent variables, there is one $\Gamma(u)$ vector for each variable.

SOURCE TERM

Enter a value or expression for the source term f (SI unit: $1/m^2$). If there are multiple dependent variables, there is a vector of f component inputs. The default is $0\text{ (1/m}^2)$.

LAX-FRIEDRICH'S FLUX PARAMETER

The shape functions used are discontinuous and therefore require auxiliary constraints on faces between adjacent mesh elements to yield a meaningful (that is, continuous) solution approximation. This is accomplished by specifying a so-called numerical flux on each face. The numerical flux implemented is the (global) Lax-Friedrichs flux, which is defined as the average of the fluxes on neighboring elements plus the jump of the solution times a parameter τ , which is necessary for stability.

Enter a value or global expression for the parameter τ . Only one expression can be entered for each equation and each domain. The parameter is per default one, but should be set accordingly to the dominant eigenvalue of the flux Jacobian matrix

$$\bar{\lambda} = \max \left| \lambda \left(d_a^{-1} \left(\mathbf{n} \cdot \frac{\partial \Gamma}{\partial u} \right) \right) \right| \quad (16-12)$$

given the bound

$$0 \leq \tau \leq \frac{\bar{\lambda}}{2} \quad (16-13)$$

A so-called *central flux* is obtained for $\tau = 0$. This gives a non-dissipative method and gives an approximation that does not penalize jumps in between elements.

Selecting $\tau = \frac{\bar{\lambda}}{2}$ sets a maximally dissipative flux, which corresponds to a so-called *upwind flux* for smoothly varying parameters $d_a, \frac{\partial \Gamma}{\partial u}$.



Lax-Friedrichs Flux Parameter

ESTIMATE OF MAXIMUM WAVE SPEED

Enter a value or expression for the estimate of maximum wave speed W_g . The default is 0.

FILTER PARAMETERS

The filter provides higher-order smoothing of nodal discontinuous Galerkin formulations and is intended to be used for absorbing layers, but you can also use it to stabilize linear wave problems with highly varying coefficients. The filter is constructed by transforming the solution (in each global time step) to an orthogonal polynomial representation, multiplying with a damping factor and then transforming back to the (Lagrange) nodal basis. Select the **Activate** check box to use this filter.

The exponential filter can be described by the matrix formula

$$V \Lambda V^{-1}$$

where V is a Vandermonde matrix induced by the node points, and Λ is a diagonal matrix with the exponential damping factors on the diagonal:

$$\Lambda_{mm} = \sigma(\eta) = \begin{cases} 1, & 0 \leq \eta \leq \eta_c \\ e^{-\alpha \left(\frac{\eta - \eta_c}{1 - \eta_c} \right)^{2s}}, & \eta_c \leq \eta \leq 1 \end{cases}$$

where

$$\eta = \eta(m) = \frac{i_m}{N_p}$$

and N_p is the basis function and i_m the polynomial order for coefficient m . α (default value: 36), η_c (default value: 0.6), and s (default value: 3) are the filter parameters that you specify in the corresponding text fields. The damping is derived from an a spatial dissipation operator of order $2s$. For $s = 1$, you obtain a damping that is related to the classical 2nd-order Laplacian. Higher order (larger s) gives less damping for the lower-order polynomial coefficients (a more pronounced low-pass filter), while keeping the damping property for the highest values of η , which is controlled by α . The default values 36 for α correspond to maximal damping for $\eta = 1$. It is important to realize that the effect of the filter is influenced by how much of the solution (energy) is represented by the higher-order polynomial coefficients. For a well resolved solution this is a smaller part than for a poorly resolved solution. The effect is stronger for poorly resolved solutions than for well resolved ones. This is one of the reasons why this filter is useful in an absorbing layer where the energy is transferred to the higher-order coefficients through a coordinate transformation. See [Ref. 1](#) (Chapter 5) for more information.

α must be positive; $\alpha = 0$ means no dissipation, and the maximum value is related to the machine precision, $-\log(\varepsilon)$, which is approximately 36. η_c should be between 0 and 1, where $\eta_c = 0$ means maximum filtering, and $\eta_c = 1$ means no filtering, even if filtering is active.

Initial Values

The **Initial Values** node adds the initial values for the dependent variables to be specified. These serve as an initial condition for the transient simulation.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.



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 for \mathbf{u} u** (dimensionless). The default value is 0.

Zero Flux

The **Zero Flux** boundary condition is the default boundary condition and prescribes a zero flux across the boundary:

$$\mathbf{n} \cdot \Gamma = 0 .$$

It can be used on exterior boundaries only.

Flux/Source

The **Flux/Source** boundary condition can be used on exterior boundaries only. With this node the boundary condition is enforced according to:

$$-\mathbf{n} \cdot \Gamma = g + q u$$

where g and q can be specified. Here the dependent variable u in the right-hand side is evaluated on the inside (as seen from the domain where the PDE interface is defined). When the normal vector is used in expressions for curved boundaries, it is important that the mesh version of these vectors are used. The components of this vector are denoted `nxmesh`, `nymesh`, and `nzmesh`, respectively.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.



In this context, exterior means exterior to the PDE interface, which does not need to be an exterior boundary to the geometry. On such boundaries, the normal direction as defined by the normal vector variables does not necessarily have to be outward pointing.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

BOUNDARY ABSORPTION/IMPEDANCE TERM

Enter the flux term q (SI unit: 1/m). The default is 0 (1/m).

Interior Source

The **Interior Source** boundary condition can be used on interior boundaries only. With this node the boundary condition is enforced according to:

$$-\mathbf{n} \cdot \Gamma = -\mathbf{n} \cdot \Gamma_{LF} + g$$

where Γ_{LF} is the Lax-Friedrichs flux computed internally. Here the source g can be specified.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

Interior Flux

The **Interior Flux** boundary condition can be used on interior boundaries only. With this node the boundary condition is enforced according to:

$$-\text{down}(\mathbf{n}) \cdot \Gamma = g \text{ and } -\text{up}(\mathbf{n}) \cdot \Gamma = -g$$

where g can be specified. It is evaluated on the interior boundary, where up and down operators are supported. If a dependent variable is used in this expression (without up or down operators, an implicit mean operation is invoked taking the average of the up and down values).

To appreciate how this boundary condition works, consider a simple example of a one-way transport equation in 1D

$$u_t + au_x = 0$$

with appropriate initial data and boundary conditions, and where the parameter $a > 0$ jumps (it is a discontinuous function of x) at an internal interface.

The proper upwind numerical flux condition is not obtained by using the internal Lax-Friedrichs flux. It can be shown that this is obtained by the numerical flux

$$\begin{aligned}
 n \cdot \Gamma^* &= \frac{1}{a^l + a^r} (a^l n \cdot \Gamma^r + a^r n \cdot \Gamma^l + a^l a^r (u^l - u^r)) \\
 &= \frac{a^r a^l}{a^l + a^r} n ((u^r + u^l) + (n u^l - n u^r))
 \end{aligned}$$

where r and l denote the right and left side of the interface respectively.

Since the down side coincides with the left side in 1D ($n=1$), this condition can be expressed by setting

```
g=-2*down(a)*up(a)*down(u)/(down(a)+up(a)).
```

In general and in higher dimensions, one typically needs the down (and up) versions of the mesh normal to express these conditions. For example, when the sign of the normal is unknown (i.e which side of an interface COMSOL Multiphysics considers the up/down side) the above condition can be entered as

```
g = -down(a)*up(a)/
(down(a)+up(a))*dnx*(up(u)+down(u)+dnx*(down(u)-up(u)))
```

Here dnx means the down side normal ($\text{dnx} = -\text{unx}$).

It is important to use the mesh version of this vector in higher dimensions on curved boundaries. For example dnxmesh , dnymesh and dnzmesh denotes the x, y, and z components of mesh normal vector on the down side.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define. Only interior boundaries can be selected.

BOUNDARY FLUX/SOURCE

Enter the flux term g (SI unit: 1/m). The default is 0 (1/m).

About Auxiliary Equation-Based Nodes

The PDE User Interfaces provide an environment for specifying a model completely in terms of equations. In many cases, however, you may only be interested in adding an equation term or a constraint to the PDE implemented by a physics interface.



The auxiliary equation-based nodes are found under the context submenus **More>**, **Edges>**, and **Points>**. To display these in the context menu, click the **Show** button () on the **Model Builder** toolbar and select **Advanced Physics Options**.

These are the available nodes (listed in alphabetical order):

- [Auxiliary Dependent Variable](#)
- [Discretization \(Node\)](#)
- [Pointwise Constraint](#)
- [Weak Constraint](#)
- [Weak Contribution \(PDEs\)](#)
- [Weak Contributions on Mesh Boundaries](#)



- [The PDE User Interfaces](#)
- [Domain, Boundary, Pair, Edge, and Point Conditions for PDEs](#)

Weak Contribution (PDEs)

The **Weak Contribution** node is available in all interfaces and for all geometry levels, including the global level. The node adds an arbitrary contribution to the total system of equations. Its weak form expression may contain the `test()` operator acting on any dependent variable in the model, and therefore add contributions to any equation.

To create an independent weak form equation rather than a weak form contribution, add extra states (dependent variables), right-click the **Weak Contribution** node and add an [Auxiliary Dependent Variable](#). You can then use the auxiliary dependent variables

in the weak-form expression.



To add this node, click the **Show** button () and select **Advanced Physics Options**. Then, in the **Model Builder**, right-click the main interface node and, depending on the geometric entity level, select **More>Weak Contribution**, **Edges>Weak Contribution**, **Points>Weak Contribution**, or **Global>Weak Contribution**.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entity level (domains, boundaries, edges, or points) to define.

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, and 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, meaning that the settings are taken from the main equation in the interface.

If the check box is cleared, the following settings become available:

Integration Order

The **Integration order** specifies the desired accuracy of integration during discretization. Polynomials of at most the given integration order are integrated without systematic errors. For smooth constraints, a sufficient 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.

Integrate on Frame

The **Integrate on frame** setting determines which frame to base the integration on—**Spatial**, **Material**, **Mesh**, or **Geometry**. The default frame is the one used for the physics interface.



This section is available for 2D axisymmetric and 1D axisymmetric models



The option effectively controls the meaning of the numerical flux and natural boundary condition. By default, the **Multiply by $2\pi r$** check box is selected, which is consistent with the implementation in the physics interfaces. This convention defines all fluxes as per unit area and the natural boundary condition per length and full revolution. Click to clear the check box to work with fluxes and natural boundary conditions per length and radian.



[Weak Contribution \(ODEs and DAEs\)](#)

Weak Contributions on Mesh Boundaries

The **Weak Contribution on Mesh Boundaries** node is available on the domain level in all physics interfaces. To add this node, click the **Show** button () and select **Advanced Physics Options**. Then right-click the main PDE node and select **More>Weak Contribution on Mesh Boundaries** at the domain level.

This node is very similar to the [Weak Contribution \(PDEs\)](#) node on the boundary level, except that it requires a domain selection. Otherwise, the settings are identical. The domain selection is interpreted as all mesh element boundaries (edges in 2D and faces in 3D) in the interior of the selected domains.



Element edges or faces that are part of a real boundary are not included in this selection.

Use a **Weak Contribution on Mesh Boundaries** node to set up boundary conditions between every pair of adjacent mesh elements in its domain selection. To access values in the two adjacent elements, use the `up()` and `down()` operators. In practice, this

node must be used together with discontinuous shape functions to implement a discontinuous Galerkin method. You can also right-click to add an [Auxiliary Dependent Variable](#) subnode.



up and down (operator)

Auxiliary Dependent Variable

If a complete equation is specified on weak form in a [Weak Contribution \(PDEs\)](#) or [Weak Contributions on Mesh Boundaries](#) node and a new dependent variable that is not part of the physics interface is used, right-click either of these nodes to add an [Auxiliary Dependent Variable](#) subnode.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node; that is, the selection is automatically selected and is the same as for the interface. You can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.



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).

- **Initial value** for the dependent variable. The default is 0.
- **Initial time derivative** for the dependent variable. The default is 0.

Under **Discretization**:

- Except for the **Compute boundary fluxes** check box (which is not available), the rest of the settings are the same as for [The Coefficient Form PDE User Interfaces](#).
- The value type of auxiliary dependent variables is always **Complex** when **Split complex variables in real and imaginary parts** is activated in the [Compile Equations](#) node of any solver sequence used.



[Discretization Section Shape Function Types and Element Orders](#)

About Explicit Constraint Reaction Terms

In the [Weak Constraint](#) and [Pointwise Constraint](#) nodes you specify an expression, R , which is forced equal to zero. Optionally, you may also specify how reaction terms are applied. The default setting, **All physics (symmetric)**, applies reaction terms based on the constraint expression in a way that preserves the symmetry of a symmetric unconstrained system of equations. These reaction terms are uniquely defined by the symmetry requirement.

Selecting **User defined** from the **Apply reaction terms on** list lets you specify the constraint reaction terms explicitly, using a syntax borrowed from weak form modeling.

-
- [Constraint Reaction Terms](#)
- [About Weak Form Modeling](#)
-

CONSTRAINT REACTION TERM EXAMPLE

For example, in a three-variable problem for variables u_1 , u_2 and u_3 , specify the constraints (using the **Constraint expression** field in two separate [Weak Constraint](#) or [Pointwise Constraint](#) nodes)

$$\begin{cases} 0 = R_1(u_1, u_2) = 2u_1 - 3u_2 & \text{on } \partial\Omega \\ 0 = R_2(u_2, u_3) = 2u_2 + 3u_3 & \text{on } \partial\Omega \end{cases}$$

Note that both constraints involve both more than one variables and that to each constraint corresponds a Lagrange multiplier variable, μ_1 and μ_2 , respectively. The weak equation corresponding to these constraints is

$$\int_{\partial\Omega} \hat{\mu}_1 R_1 + \hat{\mu}_2 R_2 = \int_{\partial\Omega} \hat{\mu}_1(2u_1 - 3u_2) + \hat{\mu}_2(2u_2 + 3u_3) = 0$$

where μ_1 and μ_2 are the test functions corresponding to the Lagrange multipliers.

This integral equation must be respected for every possible value of the Lagrange multiplier test functions. The only difference between a weak constraint and a pointwise constraint in this respect is that the Lagrange multiplier test functions in a weak constraint are nonzero over the elements adjacent to each mesh node, while the pointwise constraint test functions are nonzero only at the nodes. Therefore, weak constraints are enforced in a local weighted average sense, while pointwise constraints are enforced exactly at the mesh nodes. For this discussion, the difference is not important.

The default reaction terms, applied symmetrically to all dependent variables in the model, are defined simply by switching the test operator (here denoted by the circumflex “ $\hat{}$ ”) from the Lagrange multipliers to the constraint expressions. Since the test operator is a linear differential operator, the weak form symmetric reaction term is

$$\begin{aligned} \int_{\partial\Omega} \hat{\mu}_1 \hat{R}_1 + \hat{\mu}_2 \hat{R}_2 &= \int_{\partial\Omega} \hat{\mu}_1 \left(\frac{\partial R_1}{\partial u_1} \hat{u}_1 + \frac{\partial R_1}{\partial u_2} \hat{u}_2 \right) + \hat{\mu}_2 \left(\frac{\partial R_2}{\partial u_2} \hat{u}_2 + \frac{\partial R_2}{\partial u_3} \hat{u}_3 \right) = \\ &\int_{\partial\Omega} \hat{\mu}_1(2\hat{u}_1 - 3\hat{u}_2) + \hat{\mu}_2(2\hat{u}_2 + 3\hat{u}_3) \end{aligned}$$

The user-defined **Constraint force expression** to enter in a Weak Constraint or Pointwise Constraint node to explicitly recreate these symmetric reaction terms may be identified as the expressions multiplying the Lagrange multipliers. The test operator is denoted `test()` in weak expression syntax. Therefore, the constraint force expression for constraint $R_1 = 2u_1 - 3u_2 = 0$ is `test(2*u1-3*u2)` or, equivalently, `2*test(u1)-3*test(u2)`. The corresponding expression for R_2 is `test(2*u2+3*u3)` or, after linearization, `2*test(u2)+3*test(u3)`.

As a general rule, anything that multiplies `test(u1)` appears as a source term in the u_1 equation. Similarly, coefficients of `test(u2)` and `test(u3)` are source terms in the u_2 and u_3 equations, respectively. The symmetric reaction terms from the R_1 constraint contains both a `test(u1)` and a `test(u2)` term. Therefore, its generalized reaction force affects both these equations, while the reaction terms from R_2 affect the u_2 and u_3 equations.

Now suppose that u_1 and u_2 in reality represent components of the same vector, \mathbf{u} , while u_3 can be seen as an external quantity which should affect, but not be affected by, the value of \mathbf{u} . The symmetric reaction term from constraint R_2 violates this assumption and must be modified. In a user-defined **Constraint force expression**, write for example just `test(u2)` to apply reaction terms only as a generalized reaction force in the u_2 equation and leave the u_3 equation unaffected. This corresponds to a weak form integral contribution

$$\int_{\partial\Omega} \mu_1(2\hat{u}_1 - 3\hat{u}_2) + \mu_2\hat{u}_2$$

 Nonsymmetric reaction terms do not always lead to a well-posed problem. If, in the example above, you would set the **Constraint force expression** also for the first constraint to `test(u2)`, the Lagrange multipliers become nonunique because only their sum enters the u_2 equation. At the same time there is no means to enforce any constraint on u_1 because that equation contains no Lagrange multiplier at all.

Pointwise Constraint

To add a **Pointwise Constraint** node on the domain, boundary, edge, or point level, click the **Show** button () and select **Advanced Physics Options**. Then, depending on the geometric entity level, select **More>Pointwise Constraint** at the domain or boundary level, **Edges>Pointwise Constraint**, or **Points>Pointwise Constraint** from the context menu. There is no global pointwise constraint option.

This node adds standard pointwise constraints, similar to the ones used by boundary conditions of a constraint type in the physics interfaces.



Use **Pointwise Constraint** nodes to add extra constraints to a physics interface and to assume complete control over constraint reaction terms and points of application.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

From the **Selection** list, choose the geometric entities (domains, boundaries, edges, or points) to which you want to add a pointwise constraint. The default is **Manual**, but you can also choose, depending on the geometric entity level, **All domains**, **All boundaries**, **All edges**, or **All points** in the model.

POINTWISE CONSTRAINT

These settings are similar to those for the [Weak Constraint](#), but pointwise constraints do not need explicit, named, Lagrange multipliers. Instead, implicit Lagrange multipliers are eliminated by the solvers, together with the degrees of freedom being constrained.

Select an option from the **Apply reaction terms on** list—**All physics (symmetric)** (the default) or **User defined**. For either option, enter a **Constraint expression**, which COMSOL Multiphysics constrains to 0. For example, entering $2-u+v$ constrains $u+v$ to the value 2.

If **User defined** is selected, enter also a **Constraint force expression**. Note that the constraint force expression must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the u equation with reaction terms.



- [About Explicit Constraint Reaction Terms](#)
- [Constraint Reaction Terms](#)

DISCRETIZATION

Select a **Shape function type** (finite element types)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

In most cases, use the same shape function type and order for the pointwise constraint as for the dependent variables being constrained. If dependent variables of different order appear in the constraint expression, select the highest order for the pointwise constraint. Notable exceptions are the Hermite and Argyris shape functions, which should be constrained by Lagrange elements of the corresponding order.

The **Frame type** list is available when there is more than one unique frame in the model. In this case, select **Spatial** or **Material** from the **Frame type** list. This affects only how derivatives of Lagrange multipliers are computed and in general does not make any difference at all.

Weak Constraint

To add a **Weak Constraint** node on a domain, boundary, edge, or point level, click the **Show** button () and select **Advanced Physics Options**. 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.

The **Weak Constraint** node adds an extra dependent variable, known as a Lagrange multiplier, and a weak equation, which together enforce the specified constraint.

 If the weak constraint is redundant in the sense that some other weak or pointwise constraint also controls the value of the constraint expression, the Lagrange multiplier becomes under-determined. This makes the discrete system of equations singular and therefore virtually impossible to solve, even if the solution for the main dependent variables is unique.

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

For a default node, the setting inherits the selection from the parent node; that is, the selection is automatically selected and is the same as for the interface. You can select **Manual** from the **Selection** list to choose specific domains, boundaries, edges, or points or select **All domains**, **All boundaries**, **All edges**, or **All points** as required.

WEAK CONSTRAINT

Select an option from the **Apply reaction terms on list**—**All physics (symmetric)** (the default) or **User defined**. For either option, enter a **Constraint expression**, which COMSOL Multiphysics constrains to 0. For example, entering $2 \cdot u + v$ constrains $u + v$ to the value 2.

If **User defined** is selected, enter also a **Constraint force expression**. The constraint force expression must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the u equation with reaction terms.



- [About Explicit Constraint Reaction Terms](#)
- [Constraint Reaction Terms](#)

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, meaning that the settings are taken from the main equation in the interface. If the check box is cleared, the following settings become available:

Integration Order

The **Integration order** is a positive integer that specifies the desired accuracy of integration during discretization. Polynomials of at most the given integration order are integrated without systematic errors. For smooth constraints, a sufficient integration order is typically twice the order of the shape function. For example, the default integration order for second-order Lagrange elements is 4.

When constraints may be discontinuous at points inside the mesh elements, raising the integration order may improve accuracy.

Integrate on Frame

The **Integrate on frame** setting determines which frame to base the integration on—**Spatial**, **Material**, **Mesh**, or **Geometry**. The default frame is the one used for the physics user interface.



This section is available for 2D axisymmetric and 1D 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 represents flux per length and full revolution.



When weak constraints are activated under **Constraint Settings** in other constraint-type boundary conditions, there is no automatic multiplication by $2\pi r$ for axial symmetry.

LAGRANGE MULTIPLIER

Enter a **Lagrange multiplier variable** (the default name is `1m`) and an **Initial value**. Change the name if required, for example because a variable name conflicts.

DISCRETIZATION

Select a **Shape function type** (finite element types)—**Lagrange** (the default), **Hermite**, **Discontinuous Lagrange**, **Nodal discontinuous Lagrange**, **Discontinuous scalar density**, **Bubble**, or **Gauss point data**.

Select an associated **Element order** (order of the shape function for the element). The default is to use **Quadratic Lagrange** elements.

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.

The **Frame type** list is available when there is more than one unique frame in the model. In this case, select **Spatial** or **Material** from the **Frame type** list. This affects only how derivatives of Lagrange multipliers are computed. These are normally not used in the constraint equations, but may be of interest for postprocessing.



The value type of auxiliary dependent variables is always **Complex** when **Split complex variables in real and imaginary parts** is activated in the [Compile Equations](#) node of any solver sequence used.



[Discretization Section Shape Function Types and Element Orders](#)

Discretization (Node)

To display the option, click the **Show** button () and select **Advanced Physics Options** then right-click the main physics interface node and select **Global>Discretization** to add extra **Discretization** nodes.

Use the **Discretization** nodes to specify multiple discretizations for the same physics, typically using different element orders. In study steps and the **Multigrid Level** subnodes, you select among the discretizations specified for each physics interface. This may be used to manually create a multigrid hierarchy, just to compare the effect of different discretizations in different studies.

The **Discretization** settings window usually has one section and it is the same as the **Discretization** section found in the main physics interface settings window.



- Show Advanced Physics Options—Context Menu Equation-Based Nodes
- Discretization Section Shape Function Types and Element Orders
- Advanced Physics Sections
- For information about how to specify the splitting of complex variables, see [Compile Equations](#).

Modeling with ODEs and DAEs



The ODE and DAE User Interfaces

Adding ODEs, DAEs, and Other Global Equations

When working on complex models, single named degrees of freedom, or *states*, may be needed to track and update information that is not logically related to any particular point in space. The evolution of these states is generally governed by equations, which are independent of space, but often time-dependent. 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 ODEs and DAEs User 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 for any of the physics interfaces.



To access the node, right-click the main interface in the **Model Builder** and select **Global>Global Equations**.

Use the Global Equations node 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

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.

-
- 
 - [Plot Groups and Plots](#)
 - [Derived Values and Tables](#)
-

Solving ODEs: An Example

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 `a*r-b*r*f-rt` and `-c*f+d*r*f-ft`, 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 .

-
-  [About Parameters, Variables, and Expressions](#)
-

Solving Algebraic and Transcendental Equations: An Example

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 real-valued root at $u = 1$.

- 1 In the **Model Builder**, click the [Global Equations](#) node.
- 2 In the **Global Equations** settings window, enter 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).
- 3 Solve this using a stationary solver.

- 4** In the **State variable u (Global Evaluation)** settings window, click the **Evaluate** button (), and the value of the root displays in the **Table** window.

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 for the node:

- 1** In the **Model Builder**, click the **Global Equations** node.
- 2** Enter **u** in the **Name** column and **exp(-u) - u** in the **f(u,ut,utt,t)** column (both entries on the same row).
- 3** Compute the solution.
- 4** In the **State variable u (Global Evaluation)** settings window, click the **Evaluate** button (), and the value of the root (rounded to 0.567) displays in the **Table** window.



Derived Values and Tables

Distributed ODEs and DAEs

For ODEs and DAEs in domains, on boundaries and edges, and at points, [The Distributed ODEs and DAEs User Interfaces](#) are available to solve the following ODE (or DAE):

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} = f$$

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 ODE and DAE User Interfaces

The Global ODEs and DAEs User Interface

The **Global ODEs and DAEs** interface ($\frac{d}{dt}$), found under the **Mathematics** branch (Δu) in the **Model Wizard**, allows adding ODEs or other global equations to a model. The main default node is **Global Equations**, used to define the global equations including the names of the variables (states), the required initial values, and an optional description. Right-click the **Global Equations and ODEs** node to add additional **Global Equations**, **Global Constraint**, or **Weak Contribution (ODEs and DAEs)** nodes.



- [Solving Algebraic and Transcendental Equations: An Example](#)
- [Adding ODEs, DAEs, and Other Global Equations](#)

About ODEs, Initial-Value Problems, and Boundary-Value Problems

ODEs (ordinary differential equations) are often divided into these two types:

- *Initial-value problems* (IVPs), where the solution u and its derivatives (often with respect to time) are specified in one point (in time) so that $u(0)$ and $u'(0)$ are known, so the system is assumed to start at a fixed initial point. The Global Equations node in COMSOL Multiphysics supports such IVPs described with an equation in the following form $f(u, \dot{u}, \ddot{u}, t) = 0$ including initial values for u and its derivatives.
- *Boundary-value problems* (BVPs), where the solution u have specified values at a pair of points such as $u(0)$ and $u(1)$; that is, the points 0 and 1 are regarded as boundary points of the domain for the problem. In COMSOL you can specify such a BVP as a stationary 1D PDE, where the 1D interval represents the independent variable and the u is the dependent variable in the interval.

Global Equations

A default **Global Equations** node is added to [The Global ODEs and DAEs User Interface](#). Right-click to add additional **Global Equations**, **Weak Contribution (ODEs and DAEs)**, and **Global Constraint** nodes.



In any other physics interface, click the **Show** button () and select **Advanced Physics Options**. Then right-click the physics interface and select **Global>Global Equations** to add a node directly, without needing to add a separate Global ODEs and DAEs interface.

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 written u_t , and the second time derivative of u is u_{tt} . 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. If some equations include time derivatives and others not, the system is a DAE (differential-algebraic equation).



Initial conditions are necessary for ODEs and DAEs. For the DAEs, it is important to specify a set of initial conditions consistent with the algebraic part of the system. Otherwise, the solvers may be forced to modify the initial values to make them consistent, or they may fail completely.

In the **Global Equations** table, each row corresponds to a named state; that is, it defines a single degree of freedom and one equation.



The selected row in the table of global equations may also be edited using the **Name**, **f(u,ut,utt,t)**, **Initial value (u_0)**, **Initial value (u_t0)**, and **Description** fields underneath the table.

In each column enter as follows.

- Enter the **Name** of the state variable. 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 right-hand 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 domain of definition 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**.
- To add another equation, make additional entries in first empty 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 (or be in a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®).

DISCRETIZATION

To display this section, click the **Show** button (>Show) and select **Discretization**.

The **Discretization** section for **Global Equations** is used to specify the **Value type** (Real or Complex) of the variables the **Split complex variables in real and imaginary parts** setting

is activated 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**.

Global Constraint

Right-click [The Global ODEs and DAEs User Interface](#) node and select **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.



In any other physics interface, click the **Show** button () and select **Advanced Physics Options**. Then right-click the physics interface and select **Global>Global Constraint**.

GLOBAL CONSTRAINT

Choose an option from the **Apply reaction terms on** list—**All physics (symmetric)** (the default) or select **User defined** to define reaction terms. For either choice, enter a **Constraint expression**. The default is 0. For example, entering $2-u+v$ constrains $u+v$ to the value 2. If **User defined** is selected, enter a **Constraint force expression**.



The **Constraint force expression** must use the `test()` or `var()` operator. For example, write `test(-u)` to enforce the constraint by modifying only the u equation with reaction terms. See [Constraint Reaction Terms](#) and [Pointwise Constraint](#) for more information.

Weak Contribution (ODEs and DAEs)

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



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.



See [The Wall Distance User Interface](#) for an example of how to write an ODE using a weak formulation.

The Distributed ODEs and DAEs User Interfaces

The interfaces for distributed ODEs and DAEs ($\frac{d}{dt}$)—**Domain ODEs and DAEs (dode)**, **Boundary ODEs and DAEs (bode)**, **Edge ODEs and DAEs (eode)**, and **Point ODEs and DAEs (pode)**—are found under the **Mathematics>ODE and DAE Interfaces** branch ($\frac{d}{dt}$) in the **Model Wizard**. These interfaces provide the possibility to solve distributed ODEs and DAE in domains, on boundaries and edges, and at points.

When any of these interfaces are added, these default nodes are also added to the **Model Builder**—**Distributed ODE** and **Initial Values**. Right-click the main interface node 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 used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `dode` (in domains), `bode` (on boundaries), `eode` (on edges), or `pode` (at points).



Under **Discretization**, and except for the **Compute boundary fluxes** check box (which is not available), the rest of the settings are the same as for [The Coefficient Form PDE User Interfaces](#).



- Discretization Section Shape Function Types and Element Orders
- Distributed ODE
- Initial Values (the same as for the PDE interfaces)

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:

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} = f$$

- e_a is the *mass coefficient*.
- d_a is a *damping coefficient* or a *mass coefficient*.
- f is the *source term*.



The settings are the same as for the [Coefficient Form PDE](#).

Algebraic Equation

This node adds an **Algebraic Equation** in an interface for distributed ODEs and DAEs. Specify an algebraic equation as

$$f = 0 \tag{16-14}$$

DOMAIN, BOUNDARY, EDGE, OR POINT SELECTION

The default setting is to include **All domains**, **All boundaries**, **All edges**, or **All points** in the model. To choose specific geometric entities, 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 User Interface

Use the **Events** user 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. Right-click the main node to add **Discrete States**, **Indicator States**, **Explicit Event**, or **Implicit Event** nodes from the **Global>** submenu.

INTERFACE IDENTIFIER

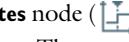
The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first interface in the model) is `ev`.



If you have the Batteries & Fuel Cells Module, see [Capacity Fade in a Lithium-Ion Battery](#) for the use of this interface and the **Discrete States**, **Indicator States**, and **Implicit Event** nodes (Model Library path **Batteries_and_Fuel_Cells_Module/Batteries/capacity_fade**).

Discrete States

Use the **Discrete States** node () 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** node () to define state variables that the solver uses to trigger implicit events, which occurs at a zero crossing of an indicator state variable.



This behavior implies, for example, that an event of the type $z > 0$, where z is an indicator state, can be detected accurately because the zero crossing of z is found. But if the condition (in [Event Conditions](#)) is formulated as, for example, $z > 1$, the event is not triggered accurately.

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](#) node.

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** node () specifies 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, Boundaries, Edges, or Points](#) 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** node () to specify an event that occurs when a 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, Boundaries, Edges, or Points](#) 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, Boundaries, Edges, or Points

The **Reinitialization on Domains**, **Reinitialization on Boundaries**, **Reinitialization on Edges**, or **Reinitialization on Points** node 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, boundaries, edges, or points that the event reinitializes.

The Wall Distance User Interface

The **Wall Distance** user interface () , found under the **Mathematics** branch () 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 user 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 used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `wd`.

DOMAIN SELECTION

The default setting is to include **All domains** in the model. To choose specific domains, select **Manual** from the **Selection** list.

PHYSICAL MODEL

Enter a **Smoothing parameter** σ_w in [Equation 16-17](#). 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 shares 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**.

-
- 
 - Show More Physics Options
 - Domain and Boundary Nodes for the Wall Distance User Interface and Initial Values
-

Domain and Boundary Nodes for the Wall Distance User Interface

The **Wall Distance User Interface** has these domain settings and boundary conditions available:

- Distance Equation
- Initial Values
- Wall



The default boundary condition, a homogeneous Neumann condition, applies to all boundaries that are not walls:

$$\nabla G \cdot \mathbf{n} = 0$$

This node does not display in the model tree.



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.

Distance Equation

The **Distance Equation** node adds to [Equation 16-17](#), 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

Select an option for the **Reference length scale** l_{ref} (SI unit: m)—**Automatic** (the default) to use [Equation 16-18](#) to compute l_{ref} or **Manual** to enter a value in the field. The default value is 1 m.

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.

INITIAL VALUES

Enter a value or expression for the initial value of the **Reciprocal wall distance** G (SI unit: 1/m). The default value wd.G0.

Wall

The **Wall** node implements the following boundary condition for walls:

$$G = G_0 = \frac{2}{l_{\text{ref}}}$$

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

Theory for Wall Distance

The Wall Distance User Interface theory is described in this section:

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*:

$$|\nabla D| = 1 \quad (16-15)$$

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 16-15 then transforms to

$$\nabla G \cdot \nabla G = G^4 \quad (16-16)$$

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:

$$\nabla G \cdot \nabla G + \sigma_w G(\nabla \cdot \nabla G) = (1 + 2\sigma_w)G^4 \quad (16-17)$$

where σ_w is a small constant. If σ_w is less than 0.5, the maximum error falls off 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 16-17](#) 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 16-17](#). C is 2 in the Wall Distance interface.

The effect of l_{ref} is loosely speaking that the distance to objects larger than l_{ref} is represented accurately, while objects smaller than l_{ref} appears to be further away than their exact geometrical distance. 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

$$\frac{\nabla G}{\sqrt{\max(\nabla G^2, \text{eps})}} \quad (16-18)$$

Reference for the Wall Distance User 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.

Curvilinear Coordinates

Introduction

A *curvilinear coordinate system* is a type of coordinate system where the coordinate lines can be curved. Using a curvilinear coordinate system can make it easier to, for example, define boundary conditions that follow a curved surface. The curvilinear coordinate systems in COMSOL Multiphysics® define a vector field and an associated base vector system using four different methods:

- A diffusion method, which solves Laplace's equation $\Delta u = 0$ and computes the vector field as $-\nabla u$. Solve for the vector field using a Stationary study.
- An elasticity method, which computes the vector field from the solution with the eigenvalue closest to zero for the following eigenvalue problem:

$$-\nabla \cdot (s(\nabla \cdot \mathbf{u})I + (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = \lambda \mathbf{u}$$

Solve for the vector field using an Eigenvalue study.

- A flow method, which solves the following equation for \mathbf{u} and p , and uses \mathbf{u} as the vector field:

$$\begin{aligned}\nabla \cdot [-pI + (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] &= 0 \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}$$

Solve for the \mathbf{u} and p using a Stationary study.

- A user-defined vector field. No study step is required.



Solve for the curvilinear coordinates (unless they are user-defined) in a separate study or separate study step, which you run before the study or study step where you solve for the physics that make use of the computed curvilinear coordinates. Simultaneously solving for the curvilinear coordinates and the physics that use them does normally not work.

The Curvilinear Coordinates User Interface

With the **Curvilinear Coordinates** user interface you can create a vector field \mathbf{v} and a base vector system (with basis vectors \mathbf{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3) that can be used by other physics to specify, for example, external currents or anisotropic material properties of a bundle of

wires or fibers. The resulting coordinate system can be—but does not have to be—curvilinear. Right-click the **Curvilinear Coordinates** node () to add one of the available methods for computing the vector field for the curvilinear coordinates:

- [Diffusion Method](#)
- [Elasticity Method](#)
- [Flow Method](#)
- [User Defined](#)

All the nodes for these methods have an initially empty selection. You can use more than one vector field computation method (of the same or different types), each one in a different set of domains.

The **Curvilinear Coordinates** user interface is available for 2D, 2D axisymmetric, and 3D geometries.

The **Curvilinear Coordinates** node's settings window contains the following sections:

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `cc`.

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.

EQUATION

See [Physics Nodes—Equation Section](#).

SETTINGS

The **Normalize vector field** check box is selected by default because a normalized vector field simplifies the use of curvilinear coordinates and a base vector system.

Select the **Create base vector system** check box to add a **Curvilinear System** node under **Definitions**, which is a Base Vector System node with a name that indicates that it is created by a **Curvilinear Coordinates** user interface and contains the base vectors from the curvilinear coordinate computation. Selecting this check box also adds a Coordinate System Settings subnode for specifying the second basis vector \mathbf{e}_2 (the software then forms the third basis vector as the cross product of the first and second basis vectors).

Diffusion Method

Add a **Diffusion Method** node to compute the vector field based on Laplace's equation $\Delta u = 0$ with the vector field \mathbf{v} defined as $\mathbf{v} = -\nabla u$ (divided by $|\nabla u|$ if normalized). This method is a scalar “potential method” resulting in an incompressible vector field and is useful for geometries that are smooth but leads to concentrations at sharp corners. To define the equation in the geometry you can add the following boundary conditions:

- [Wall](#) (the default boundary condition)
- [Inlet](#)
- [Jump](#)
- [Outlet](#)

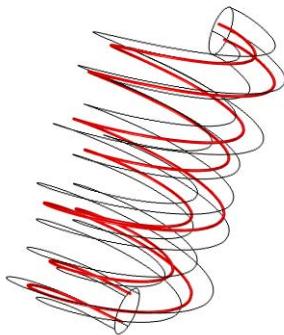


Figure 16-1: The computed velocity field in a helix using a Curvilinear Coordinates user interface with a Diffusion Method.

Elasticity Method

Add an **Elasticity Method** node to compute the vector field based on the solution with the eigenvalue closest to zero for the following eigenvalue problem:

$$-\nabla \cdot (s(\nabla \cdot \mathbf{u})\mathbf{I} + (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = \lambda \mathbf{u}$$

The vector field \mathbf{v} is the same as \mathbf{u} but normalized if normalization is selected.



The Elasticity method requires that you use an eigenvalue study.

This method is similar to the Coil Current Calculation approach used in the Multi-Turn Coil Domain feature in Magnetic Fields (in the AC/DC Module) and is useful for coil applications, for example. To define the equation in the geometry you can add the following boundary conditions:

- [Wall](#) (the default boundary condition)
- [Outlet](#)

In addition a default [Inlet](#) node is added because you must add one inlet for the geometry to define the positive direction of the curvilinear coordinate's vector field as indicated by the arrow that appears on the inlet boundary in the **Graphics** window.

Flow Method

Add a **Flow Method** node to solve the following equation for the vector \mathbf{u} and the scalar p and use \mathbf{u} as the vector field:

$$\begin{aligned}\nabla \cdot [-p\mathbf{I} + (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] &= 0 \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}$$

The vector field \mathbf{v} is the same as \mathbf{u} but normalized if normalization is selected. This approach is equivalent to computing the flow of an incompressible fluid—that is, creeping flow or Stokes flow. The Flow method is useful for geometries with nonconstant cross sections. To define the equation in the geometry you can add the following boundary conditions:

- [Wall](#) (the default boundary condition)

- Inlet
- Jump
- Outlet

User Defined

Add a **User Defined** node to specify the vector field **u** as user-defined expressions for its components. In the **User Defined** section, enter the expressions for those components in the text fields under **Vector field**. The vector field **v** is the same as **u** but normalized if normalization is selected. You can select any other coordinate system in the model from the **Coordinate system** list to use as the coordinate system for defining the vector field. The **Global coordinate system** is the default.

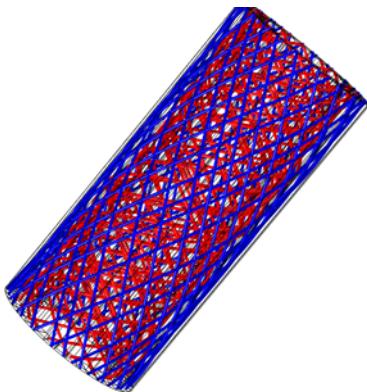


Figure 16-2: A model of arterial wall mechanics, using four user-defined Curvilinear Coordinates user interfaces and a cylindrical coordinate system to define the wall pattern.

Inlet

Use an **Inlet** node to define where the vector field starts. Select the boundaries to define as the vector field's inlet in the **Boundary Selection** section. For the **Diffusion Method** and **Elasticity Method** it sets **u** to 1 at the inlet. For the **Flow Method** you can choose, in the **Inlet** section, to define the type of inlet using the **Type** list:

- Choose **Normal velocity** to specify the normal velocity **u·n** as a velocity **u_n** (default value: 1, SI unit: m/s) in the **Velocity** field.
- Choose **Velocity field** to specify the components of the velocity field **u_{in}** in the text fields under **Velocity** (default: 0, SI unit: m/s). You can select any coordinate system

in the model from the **Coordinate system** list to use as the coordinate system for defining the velocity field. The **Global coordinate system** is the default.

Jump

Use a **Jump** node at a boundary to define a closed-loop vector field. You can add **Jump** nodes to interior boundaries. It is equivalent to a jump from 0 to 1 for the **Diffusion Method** and to a flow inlet and a flow outlet (with constant pressure) for the **Flow Method**; see **Inlet** above for the available settings. For the **Elasticity Method**, the jump condition is not applicable; instead, you can add an **Inlet** node to an interior boundary if needed.

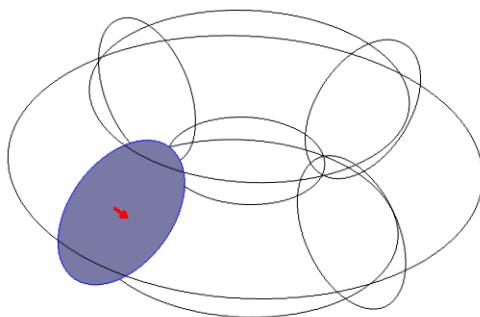


Figure 16-3: A Jump boundary condition applied to one of the interior boundaries of a torus, using a Diffusion method. The arrow indicates the direction of the vector field (the side where the value is 1).

Outlet

Use an **Outlet** node to define where the vector field ends. For the **Diffusion Method** this means that $u = 0$, for the **Elasticity Method** it sets $\mathbf{e} \times \mathbf{n}$ to 0. For the **Flow Method** it is a zero pressure and no stress condition: $p = 0$ and $(\nabla u + (\nabla u)^T) \cdot \mathbf{n} = 0$. Select the boundaries to define as the vector field's outlet in the **Boundary Selection** section.

Wall

The **Wall** node is the default boundary condition and defines the walls as boundaries where the normal component of the vector field is zero. It applies to all boundaries where you do not assign any of the other boundary conditions.

Coordinate System Settings

A **Coordinate System Settings** node is added by default if you have selected the **Create base vector system** check box in the **Curvilinear Coordinates** node's settings window. You use this node to specify the second basis vector for the created base vector coordinate system. The Curvilinear Coordinates user interface solves for first basis vector and then computes the third basis vector for a full orthonormal coordinate system as the cross product of the first and second basis vector:

$$\begin{aligned}\mathbf{e}_1 &= \frac{\mathbf{v}}{|\mathbf{v}|} \\ \mathbf{e}_2 &= \frac{\mathbf{v}_2 - (\mathbf{v}_2 \cdot \mathbf{e}_1)\mathbf{e}_1}{|\mathbf{v}_2 - (\mathbf{v}_2 \cdot \mathbf{e}_1)\mathbf{e}_1|} \\ \mathbf{e}_3 &= \mathbf{e}_1 \times \mathbf{e}_2\end{aligned}$$

The **Coordinate System Settings** node's settings window includes the following section:

SETTINGS

Here you define the direction of the second basis vector \mathbf{v}_2 by selecting one of the directions from the **Second basis vector** list: **x-axis**, **y-axis**, **z-axis** (in 3D), or **User defined**. If you select **User defined**, enter the components of \mathbf{v}_2 in the corresponding text fields under **Second basis vector**. You can select any other coordinate system in the model from the **Coordinate system** list to use as the coordinate system for defining the second basis vector. The **Global coordinate system** is the default.

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**.

Theory for the Sensitivity User Interface

About Sensitivity Analysis

The Sensitivity user interface is special in the sense that it does not contain any physics of its own. Instead, it is a tool for evaluating the sensitivity of a model with respect to almost any variable. The Sensitivity user interface is used together with a Sensitivity study step, which in turn controls the Sensitivity solver extension. Simple cases can be handled directly in the Sensitivity study step, while more advanced cases must be set up in a Sensitivity user interface prior to solving.

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 study step and corresponding user 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 user interface does not contain any physics it is not intended for use on its own. When the user 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 set of control variables can be specified. The Sensitivity user interface can perform these distinct tasks:

- Select control variables and set their values
- Define scalar objective functions



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.

The companion Sensitivity study step is responsible for:

- Choosing which objective functions and control variables to solve for
- Selecting a sensitivity evaluation method
- Selecting which study step to compute sensitivities for
- Setting up the Sensitivity solver extension

Theory for Stationary Sensitivity Analysis

Evaluating the sensitivity of a scalar-valued objective function $Q(\xi)$ with respect to the control variables, ξ , at a specific point, ξ_0 , can be rephrased as the problem of calculating the derivative $\partial Q / \partial \xi$ at $\xi = \xi_0$. In the context of a multiphysics model, Q is usually not an explicit expression in the control variables ξ alone. Rather, $Q(u(\xi), \xi)$ 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 $L(u(\xi), \xi) = 0$. If the PDE has a unique solution $u = L^{-1}(\xi)$, the sensitivity problem can be informally rewritten using the chain rule as that of finding

$$\frac{d}{d\xi} Q(u(\xi), \xi) = \frac{\partial Q}{\partial \xi} + \frac{\partial Q}{\partial u} \cdot \frac{\partial u}{\partial L} \cdot \frac{\partial L}{\partial \xi}$$

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 , $\partial Q / \partial u$ is an N -by-1 matrix, $\partial u / \partial L$ is an N -by- N matrix (since L^{-1} is unique) and $\partial L / \partial \xi$ 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:

$$\frac{d}{d\xi} Q(u(\xi), \xi) = \frac{\partial Q}{\partial \xi} + \frac{\partial Q}{\partial u} \cdot \left(\frac{\partial L}{\partial u} \right)^{-1} \cdot \frac{\partial L}{\partial \xi} \quad (17-1)$$

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 17-1](#), 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 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, using one of these methods:

- Select one of the [Forward Sensitivity](#) methods 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 (or forward numeric) 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 forward numeric method in addition requires two additional residual evaluations. 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.



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 [Studies and Solvers](#).

Issues to Consider Regarding the Control Variables

THE EFFECT OF DISCRETIZATION

The sensitivity analysis is always performed on the discretized system of equations. As already mentioned, the control variables 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 a 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.



If the plan is to use the sensitivities in an automatic optimization procedure, as is done through the Optimization user interface available with the 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

You can use the control variables directly to parameterize any aspect of the physics that is controlled by an expression. This applies 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 require special attention.

Control variables cannot be used directly in the geometry description. Instead, the model must be set up using a Deformed Geometry or Moving Mesh user interface to control the shape of the geometry. Then use control variables to control the mesh movement, effectively parameterizing the geometry. See [Deformed Geometry and Moving Mesh](#) for details about these physics interfaces and ALE in general.

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 differentiable or complex analytic 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 $Q = \text{real}(a \cdot \text{conj}(b))$, 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

$$Q(a + \delta a, b + \delta b) \approx Q(a, b) + \text{real}\left(\frac{\partial Q}{\partial a} \delta a + \frac{\partial Q}{\partial b} \delta b\right) \quad (17-2)$$

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 17-2. For that reason, use it in the definition of any time-average quantity set as objective function in a sensitivity analysis.

The Sensitivity User Interface

The **Sensitivity (sens)** user interface () , found under the **Mathematics>Optimization and Sensitivity** () branch in the **Model Wizard**, provides tools for adding advanced sensitivity evaluation to a stationary model. Basic problems only in terms of global scalar objective functions and model parameters can be set up directly in a Sensitivity study step, and therefore do not require the use of a Sensitivity user interface.



For a more extensive introduction to the mathematics implemented by this interface, see the [Theory for the Sensitivity User Interface](#).

The objective functions 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), which may be fields dependent on position in space or scalar quantities defined globally. 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](#)



Note that adding a Sensitivity study step to a study makes it possible to perform a sensitivity analysis directly at the study level. See [Sensitivity \(Study Step\)](#).

The main **Sensitivity** node's settings window contains the following section:

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is **sens**.

-
- 
 - Global Objective
 - Global Control Variable
 - Adding a Sensitivity User Interface to a Model
-



Sensitivity Analysis of a Communication Mast Detail: Model Library path
**COMSOL_Multiphysics/Structural_Mechanics/
mast_diagonal_mounting_sensitivity**

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 this node 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) to define.

PAIR SELECTION

If this node is selected from the **Pairs** menu, choose the pair to define. An identity pair has to be created first. Ctrl-click to deselect.



- Continuity on Interior Boundaries
 - Identity and Contact 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 by entering an objective expression.

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 Variable

Use the **Global Control Variable** node to 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 User Interface to a Model

You can add a **Sensitivity** user interface when creating a new model or at any time during modeling. For a new model, select physics user 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 physics user interfaces.

- 1 Expand the **Mathematics>Optimization and Sensitivity** node in the list of physics.
- 2 Select **Sensitivity** ().
- 3 Click the **Add Selected** button () underneath the list to add the selected physics user interface to the model. The physics user 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.

Deformed Geometry and Moving Mesh

This chapter explains how to use the user 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.

Deformed Mesh Fundamentals

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 user interface of which Solid Mechanics is part.

Deformed Geometry vs. Moving Mesh

There are two physics user interfaces implementing different types of deformed meshes: the Deformed Geometry () and the Moving Mesh () user interfaces. Both are found under the **Mathematics>Deformed Mesh** branch () in the **Model Wizard**.

In the Deformed Geometry user 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 user 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 user 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 user 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 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, and so forth, 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 deform, 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, using 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 you restart simulation, 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 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.

About 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 , ϕ , z). Before adding a Moving Mesh or Deformed Geometry user interface to a model, all four frames coincide and use the spatial coordinate names. Also all user interfaces based on solid mechanics include moving mesh functionality and by default behave much in the same way as a Moving Mesh user interface.

When a Moving Mesh or Solid Mechanics user 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 , Φ , Z). From this point, Eulerian and Lagrangian formulations behave differently because 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 user interface is added. At that point, a new geometry coordinate system is created and given a new set of independent variable names (by default X_g , Y_g , Z_g or R_g , Φ_{1g} , Z_g). 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 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 user 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 user 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 X_m , Y_m , Z_m or R_m , Φ_{1m} , Z_m). The original geometry coordinates are mapped and stored together with the new mesh such that any Deformed Geometry user 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.

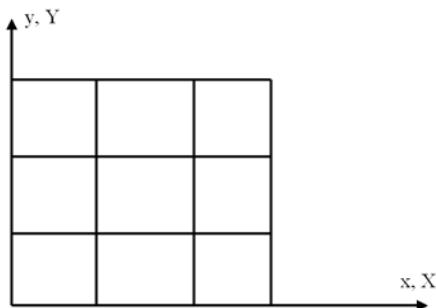


Figure 18-1: An undeformed mesh. In the reference configuration, which may be the actual configuration at a reference time or a hypothetical state, the spatial frame (x, y) and the material frame (X, Y) coincide.

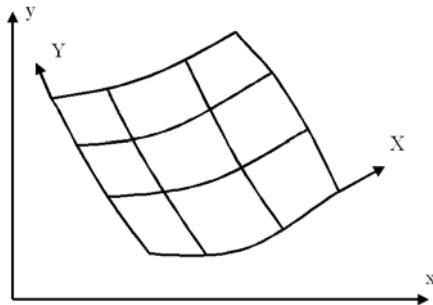


Figure 18-2: After deformation of the material, the spatial frame (x, y) remains the same, while the material coordinate system (X, Y) has been deformed, following the material. Meanwhile, the material coordinates of each material point remain the same but its spatial coordinates have changed.

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

$$x_0 = x(X_0, Y_0, t), \quad y_0 = y(X_0, Y_0, t) \quad (18-1)$$

These functions can be explicit transformations (expressions) or the solution to a mesh smoothing equation. The mesh node's material coordinates (X_0, Y_0) can in turn be seen as functions of an underlying system of geometry coordinates (X_g, Y_g) and a parameter, p , such that

$$X_0 = X(X_g, Y_g, p), \quad Y_0 = Y(X_g, Y_g, p) \quad (18-2)$$

with similar options for the transformations. 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

$$\begin{aligned}\mathbf{x} &= \mathbf{f}(\mathbf{X}, t) = \mathbf{f}(\mathbf{g}(\mathbf{X}_g, p), t) \\ \mathbf{X} &= \mathbf{g}(\mathbf{X}_g, p) \\ \mathbf{X}_m &= \mathbf{h}(\mathbf{X}_g, i)\end{aligned}$$

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 user 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 that 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 and Mesh 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 user 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 can formulate their equations in either material or spatial frame, as set by the **Frame type** setting found under **Discretization** in the physics user interface node's 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 u_x and u_y in the software.
- The variable is defined on the material frame and its derivatives with respect to the material coordinates are denoted u_X and u_Y in the software.

In a few cases both sets of derivatives exist, but normally 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 u_t in the software. For example, for u defined on the spatial frame:

$$u_t(x_0, y_0) = \frac{\partial u}{\partial t} \Big|_{x_0, y_0}$$

- The *mesh time derivative*, which is taken for a fixed point in the mesh:

$$u_{\text{TIME}}(X_m, Y_m) = \frac{\partial u}{\partial t} \Big|_{X_m, Y_m}$$

This derivative is denoted u_{TIME} 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:

$$u_t = u_{\text{TIME}} - u_x x_{\text{TIME}} - u_y y_{\text{TIME}}$$

where $(x_{\text{TIME}}, y_{\text{TIME}})$ 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

$$\frac{\partial^2 \dot{x}}{\partial X^2 \partial t} + \frac{\partial^2 \ddot{x}}{\partial Y^2 \partial t} = 0$$

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

$$J = \det(\nabla_X x)$$
$$I_1 = J^{-2/3} \text{tr}((\nabla_X x)^T \nabla_X x)$$

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 user 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, and the solver might then run into convergence problems. Sometimes an **Inverted mesh element** warning 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. You can also generate a new mesh 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.
- If you use a **Geometry shape order** larger than 1 in the Moving Mesh and Deformed Geometry user 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. 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 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 on 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 by reducing the absolute and relative tolerances for the time-dependent solver.
- Help the mesh deformation by sliding the boundary elements along with the 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.



To make the Deformed Configuration mesh represent the deformed geometry, use a Moving Mesh (ALE) user 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). Also, some of the physics in the Corrosion Module and Electrodeposition Module include a deformed geometry.

-
- 
 - Deformed Configuration
 - Solution (data set)
-

The following sections contain details about these steps and 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, **Stationary Solver 1>Parametric 1** under **Solver 1** and select **Stop Condition** () from the context menu. In the table under **Stop Expressions** enter a Boolean expression (to stop when the expression is true) or an expression that makes the solver stop when the expression becomes negative. For example, enter `mod1.ale.relVolMin<0.01` to stop before the minimum relative element volume becomes less than 0.01. The following predefined variables are useful for defining a stop condition and for monitoring the mesh deformation: the minimum relative element volume, `ale.relVolMin`; the minimum mesh quality, `ale.minqual`, and the local relative element volume, `ale.relVol`. See [Predefined Variables](#) below.



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** field and 0.01 in the **Stop for remeshing below mesh quality** field in the settings window for the **Automatic Remeshing** () node (under the **Time-Dependent Solver** node in the solver sequence).

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, try using 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**, for example.

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** node () under **Meshes**. The deformed configuration works as a new geometry but with restricted functionality. The settings window of the deformed configuration indicates which solution it was constructed from. Click the **Update** button to see the corresponding deformed configuration in the graphics.

REMESHING THE DEFORMED CONFIGURATION

Expanding a **Deformed Configuration** node () shows that a new mesh sequence has been added beneath it. This mesh sequence contains a **Size** node () and a **Reference** node (). The reference node refers to the original mesh sequence. This means that the new mesh sequence uses the same nodes 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 nodes from the original mesh sequence are copied to the new mesh sequence. You can also add and remove nodes in the new mesh sequence. If several meshes are needed on the deformed configuration (for the multigrid solver, for example), you can add an additional mesh sequence by right-clicking **Deformed Configuration** () and selecting **Mesh** ().

CONTINUE SOLVING WITH THE NEW MESH

- 1 In the settings window for the study step (for example, **Study 1>Step 1: 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 To solve for remaining times or parameters, right-click **Study 1** and select **Compute**.

REMESHING SEVERAL TIMES

You can 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.

Moving Mesh User Interface

With the **Moving Mesh (ale)** user interface () found under the **Mathematics>Deformed Mesh** branch (img alt="grid icon" data-bbox="468 191 491 211"/>) in the **Model Wizard**, you can create models 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.

When this user interface is added, these default nodes are also added to the **Model Builder—Fixed Mesh** and **Prescribed Mesh Displacement** (the default boundary condition). Right-click the **Moving Mesh** node to add other nodes that implement, for example, boundary conditions and current sources.

Predefined Variables

The Moving Mesh user interface includes the following predefined variables, which can be of interest, for example, to monitor the quality of the mesh and define 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. By plotting this quantity while solving, you can monitor how the mesh deforms and where it might become too distorted.
- The minimum relative element volume, `ale.relVolMin`, must be > 0 , otherwise the mesh elements are inverted. A suitable stop criterion using this variable is that the minimum relative element volume must be larger than a small positive number.
- 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).



[Sloshing Tank: Model Library path](#)
[COMSOL_Multiphysics/Fluid_Dynamics/sloshing_tank](#)

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern

`<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `ale`.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.



You do need to use a Moving Mesh user interface in domains for which the displacements appear among the dependent variables, for example, where a Solid Mechanics user interface is active.

FRAME SETTINGS

Specify the names of the spatial coordinates of the base frame for the user 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.



- Domain and Boundary Nodes in the Moving Mesh User Interface
- Smoothing Methods
- Show More Physics Options

Domain and Boundary Nodes in the Moving Mesh User Interface

The [Moving Mesh User Interface](#) includes these domain and boundary nodes:

- Fixed Mesh
- Free Deformation
- Prescribed Deformation
- Prescribed Mesh Displacement
- Prescribed Mesh Velocity
- Prescribed Normal Mesh Velocity
- Zero Normal Mesh Velocity
- Zero Normal Mesh Displacement

Fixed Mesh

Use the **Fixed Mesh** node 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 define.

Prescribed Mesh Displacement

Use the **Prescribed Mesh Displacement** node 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 define. For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the user interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

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 user 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

To display this section, click the **Show** button () and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Free Deformation

The **Free Deformation** node constrains the mesh displacement only by the boundary conditions on the surrounding boundaries. The displacement in the domain is obtained by solving a PDE.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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** node to define the deformation explicitly using expressions, or if you want the spatial coordinates to follow a deformation computed by, for example, a Solid Mechanics user interface. (You can also achieve the latter effect by excluding the domains where the Solid Mechanics user interface is defined from the domains where the Moving Mesh user interface is active.)

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

PRESCRIBED MESH DISPLACEMENT

Specify expressions that define the deformation in the **Prescribed mesh displacement** fields (dx , dy , and dz for 3D models, for example) (SI unit: m). The default gives no mesh displacement.

Prescribed Mesh Velocity

Use the **Prescribed Mesh Velocity** node on the boundary of domains with free displacement to specify the velocity of the boundary. The spatial frame in the adjacent domains moves in accordance with the velocity.

BOUNDARY SELECTION

From the **Selection** list, choose the boundaries to define.

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

To display this section, click the **Show** button () and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Prescribed Normal Mesh Velocity

Use the **Prescribed Normal Mesh Velocity** 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 node according to the following equation:

$$\frac{\partial \mathbf{X}}{\partial t} \cdot \mathbf{n} = v_0 + v_{\text{mbs}}$$

where v_0 is the desired normal mesh velocity, and v_{mbs} is a smoothing velocity according to:

$$v_{\text{mbs}} = \delta_{\text{mbs}} |v_0| h H$$

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.

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 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

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the user interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

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.

Deformed Geometry User Interface

Using the **Deformed Geometry (dg)** user interface () found under the **Mathematics>Deformed Mesh** branch () in the **Model Wizard**, you can study how the physics changes when the geometry changes as a function of a parameter. The only real difference between the Deformed Geometry and Moving Mesh user interfaces 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.

INTERFACE IDENTIFIER

The interface identifier is used primarily as a scope prefix for variables defined by the physics user interface. Refer to such interface variables in expressions using the pattern `<identifier>.<variable_name>`. In order to distinguish between variables belonging to different physics user interfaces, the identifier string must be unique. Only letters, numbers and underscores (`_`) are permitted in the **Identifier** field. The first character must be a letter.

The default identifier (for the first user interface in the model) is `dg`.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

 The Deformed Geometry user 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 Deformed Geometry user 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.



For detailed information about selecting geometric entities (domains, boundaries, edges, and points), see [Selecting and Visualizing in Models](#)

For more information about deformed meshes, see [Moving Mesh User Interface](#).



- [Domain and Boundary Nodes for Deformed Geometry](#)
- [Smoothing Methods](#)
- [Show More Physics Options](#)



Electrochemical Polishing: Model Library path

[COMSOL_Multiphysics/Electromagnetics/electrochemical_polishing](#)

Domain and Boundary Nodes for Deformed Geometry

The [Deformed Geometry User Interface](#) includes these domain and boundary nodes:

- | | |
|---|--|
| <ul style="list-style-type: none">• Free Deformation• Prescribed Deformation• Fixed Mesh• Prescribed Mesh Displacement | <ul style="list-style-type: none">• Prescribed Mesh Velocity• Prescribed Normal Mesh Velocity• Zero Normal Mesh Velocity• Zero Normal Mesh Displacement |
|---|--|

Fixed Mesh

Use the **Fixed Mesh** node to specify that the selected domains retain their original shape as defined by the geometry and original mesh. This is the default.

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the

interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

Prescribed Mesh Displacement

Add the **Prescribed Mesh Displacement** node on the boundaries of domains with free deformation. The material frame in the adjacent domain moves in accordance with the displacement.

BOUNDARY SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific boundaries or select **All boundaries** as required.

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

To display this section, click the **Show** button () and select **Advanced Physics Options**. Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Free Deformation

The **Free Deformation** node 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.



Smoothing Methods

DOMAIN SELECTION

For a default node, the setting inherits the selection from the parent node, and cannot be edited; that is, the selection is automatically selected and is the same as for the interface. When nodes are added from the context menu, you can select **Manual** from the **Selection** list to choose specific domains or select **All domains** as required.

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** node to define the deformation of the material frame explicitly using expressions.

DOMAIN SELECTION

From the **Selection** list, choose the domains to define.

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.

Prescribed Mesh Velocity

Use the **Prescribed Mesh Velocity** node 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 define.

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

To display this section, click the **Show** button () and select **Advanced Physics Options**.

Select the **Use weak constraints** check box to replace the standard constraints with a weak implementation.

Prescribed Normal Mesh Velocity

Use the **Prescribed Normal Mesh Velocity** 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 node according to the following equation:

$$\frac{\partial \mathbf{X}}{\partial t} \cdot \mathbf{n} = v_0 + v_{\text{mbs}}$$

where v_0 is the desired normal mesh velocity, and v_{mbs} is a smoothing velocity according to:

$$v_{\text{mbs}} = \delta_{\text{mbs}} |v_0| h H$$

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.

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 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.

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.

Studies and Solvers

This chapter describes the study types and solvers available in COMSOL Multiphysics®.

Studies and the Study Nodes

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 (top level), containing the least amount of detail, is the **Study** node () , which defines a Study branch. 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 that are included, and which meshes that are used. These study steps correspond to a part of a *solver configuration* (solver sequence), which is the next level of detail (there are also study steps that are used for cluster computing, for example, which correspond to a part of a *job configuration*). A solver configuration 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 that make it possible to control the solver settings in detail.

Solving a problem can amount to simply right-clicking the **Study** node for a predefined study type and selecting **Compute** (), which generates the default solver configuration 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 that 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 configuration 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 selected physics and from 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 that 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.

Study

A **Study** node holds the nodes that define how to solve a model. These nodes are divided into three broad categories:

- Study steps, which determines overall settings suitable for a certain study type.
- Solver configurations, which contains the solvers and related configurations for dependent variables to solve for, intermediate storage of solutions, and specific solver settings. It is only visible if it has contents or if you click the **Show** button () and select **Advanced Study Options**.
- Job configurations (distributed parametric jobs, batch jobs, and cluster computing), which is only visible if it has contents or if you click the **Show** button () and select **Advanced Study Options**.

The topics are described in the following sections. The main **Study** node has one section:

STUDY SETTINGS

The **Generate default plots** check box is selected by default so that plot groups with suitable default plots for the physics in the study are generated automatically when computing the solution. Clear this check box if you do not want any default plots.

The **Generate convergence plots** check box is selected or cleared by default based on the setting for generating convergence plots in the **Preferences** dialog box. Clear this check box if you do not want convergence plots to be generated during the solution process.

Overview of the Study Types

BASIC STUDY TYPES

COMSOL Multiphysics includes four basic study types:

- Stationary
- Time Dependent, Time Discrete, and Time-Dependent Modal
- Eigenfrequency or Eigenvalue
- Frequency Domain or Frequency Domain Modal

LIST OF AVAILABLE STUDY TYPES

The following study types are available. Some of the study types require certain add-on modules:

TABLE 19-1: 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 .
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 .
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. See Boundary Mode Analysis .
Coil Current Calculation		This study is available with the AC/DC Module using the Magnetic Fields interface coil node. See Coil Current Calculation .
Cyclic Voltammetry		This study is available with the Electroanalysis interface, which requires one of these modules: Batteries & Fuel Cells, Corrosion, Electrochemistry, or Electrodeposition. It adds a Cyclic Voltammetry study step, which sets up a time-dependent solver. See Cyclic Voltammetry .

TABLE 19-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Eigenfrequency		Similar to an Eigenvalue study but computes the eigenfrequencies instead of the eigenvalues. Adds an Eigenfrequency study step. See Eigenfrequency .
Eigenvalue		This study uses a formulation to compute eigenvalues and eigenmodes using an eigenvalue solver. Adds an Eigenvalue study step. See Eigenvalue .
Empty Study		An empty study that initially contains no study steps.
Frequency Domain		For a study in the frequency domain such as wave equations or frequency response analysis. Adds a Frequency Domain study step. See Frequency Domain .
Frequency-Domain, Perturbation		For studies of small oscillations about a bias solution. See Frequency-Domain, Perturbation .
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. See Frequency Domain Modal .
Frequency-Stationary		This study is available with, for example, the Induction Heating and Microwave Heating physics. See Frequency-Stationary .
Frequency-Transient		This study is available with the Induction Heating, Microwave Heating, Inductively Coupled Plasma, and Microwave Plasma interfaces. See Frequency-Transient .
Frozen Rotor		This study is available with the Rotating Machinery, Laminar Flow and Turbulent Flow interfaces. See Frozen Rotor .
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 .
Mean Energies		The Mean Energies study is available with the Boltzmann Equation, Two-Term Approximation interface in the Plasma Module. See Mean Energies .
Modal Reduced Order Model		To export the reduced-order model matrices for time-dependent wave problems using a modal solver, the Modal Reduced Order Model study adds an Eigenfrequency study step followed by a Modal Reduced Order Model study step. See Modal Reduced Order Model .

TABLE 19-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Mode Analysis		A mode analysis computes the modes for an acoustic or electromagnetic wave using an eigenvalue solver. Adds a Mode Analysis study step. See Mode Analysis .
Particle Trajectories		Available with the Particle Tracing Module. See Particle Trajectories (Study) .
Prestressed Analysis, Eigenfrequency		For computing eigenfrequencies that are influenced by a prior static load. Adds a Stationary study step and an Eigenfrequency study step. See Prestressed Analysis, Eigenfrequency .
Prestressed Analysis, Frequency Domain		For computing the response to harmonic loads fluctuating around a static preload. Adds a Stationary study step and a Frequency Domain study step. See Prestressed Analysis, Frequency Domain .
Reduced Electric Fields		The Reduced Electric Fields study is available with the Boltzmann Equation, Two-Term Approximation physics in the Plasma Module. See Reduced Electric Fields .
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 .
Stationary		For a stationary or steady-state situation where you can use a stationary solver. This study type is also used for optimization problems that are constrained with a stationary PDE. Adds a Stationary study step. See Stationary .
Stationary with Initialization		For stationary two-phase flow models that require an initialization of a level set or phase field function. Adds a Wall Distance Initialization study step followed by a Stationary study step. See Stationary with Initialization .
Stationary, One-Way Coupled		The Stationary, One-Way Coupled study is available with the Fluid-Structure Interaction multiphysics, which requires the MEMS Module or the Structural Mechanics Module. The study creates two study steps: Stationary, Fluid and Stationary, Solid. See Stationary, One-Way Coupled .

TABLE 19-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Stationary, One-Way Coupled with Initialization		The Stationary, One-Way Coupled with Initialization study is available with the Fluid-Structure Interaction multiphysics, which requires the MEMS Module or the Structural Mechanics Module, plus the CFD Module. See Stationary, One-Way Coupled with Initialization .
Stationary Plug Flow		For use with the Plug flow reactor type in the Reaction Engineering physics available with the Chemical Reaction Engineering Module. Adds a Stationary Plug Flow step that solves for the molar flow rate as function of reactor volume. See Stationary Plug Flow .
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 that are constrained with a time-dependent PDE. Adds a Time Dependent study step. See Time Dependent and Time Discrete .
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 solved for. It is useful, for example, for time-dependent electrochemistry in cases where the mesh deformation is small. It adds a Current Distribution Initialization study step and Time-Dependent, Fixed Geometry study step. See Time-Dependent with Initialization, Fixed Geometry and Time-Dependent with Initialization .
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. See Time-Dependent Modal .
Time Dependent, One-Way Coupled		The Time Dependent, One-Way Coupled study is available with the Fluid-Structure Interaction multiphysics, which requires the MEMS Module or the Structural Mechanics Module. The study creates two study steps: Time Dependent, Fluid and Time Dependent, Solid. See Time Dependent, One-Way Coupled .

TABLE 19-1: STUDY TYPES

STUDY TYPE	ICON	DESCRIPTION
Time Dependent, One-Way Coupled with Initialization		The Time Dependent, One-Way Coupled with Initialization study is available with the Fluid-Structure Interaction multiphysics, which requires the MEMS Module or the Structural Mechanics Module, plus the CFD Module. See Transient, One-Way Coupled with Initialization .
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. See Transient with Initialization .

PARAMETRIC SWEEPS, MULTIGRID LEVELS, BATCH, AND CLUSTER COMPUTING NODES

In addition, [Parametric Sweep](#) nodes can be added to a study to perform parametric variation on the other studies. To optimize parameters with respect to an objective function defined by other studies, add the Optimization study node (the Optimization study requires the Optimization Module). Add a [Multigrid Level](#) subnode to the other study step nodes to describe a geometric multigrid level used by the study. You can also add [Batch](#) and [Cluster Computing \(Study\)](#) nodes—or their counterparts for parametric studies, [Batch Sweep](#) and [Cluster Sweep](#), respectively—as required.



Adding Study Step Nodes



To locate and search all the documentation, in COMSOL Multiphysics, select **Help>Documentation** from the main menu and either enter a search term or look under a specific module in the documentation tree.

Adding Study Step Nodes

You can add a study step to a study by right-clicking the study node and then selecting the study type from the **Study Steps** submenu. If you select a study for your physics interfaces in the **Model Wizard**, a study step node appears under the study node that is added to the model when you click **Finish**.

The predefined study types correspond to the most commonly performed simulations for different physics. 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 to include.

Study Steps and Solver Configurations

Most study steps correspond to a part of a solver configuration that includes a solver for the specific problem:

- **Stationary:** Generates equations without time derivatives. Corresponds to a **Stationary solver**.
- **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 Solver** which is set to transform eigenvalues to eigenfrequencies.
- **Frequency Domain:** Generates stationary equations that are used for frequency sweeps. 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 directly to any part of a solver configuration. Instead, they either correspond to a part of a job configuration, or modify the behavior of another study step. 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 configuration, 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 configuration. 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.

- **Sensitivity** (): Specifies objective functions and control variables with respect to which sensitivity will be computed. Global scalar objective functions may be specified directly in the study step, and model parameters selected as control variables. In addition, the study step provides control over the sensitivity solver method and contributions to the sensitivity problem defined by a Sensitivity or Optimization interface.
- **Optimization** (): Used to solve PDE-constrained optimization problems. This study step allows direct definition of objective functions and selection of model parameters, including parameters that control the geometry, for optimization. It also provides detailed control over solvers and contributions to an optimization problem defined by an Optimization interface. This study type requires an Optimization Module license.
- **Batch** (): Creates a job that can be run without the graphical user interface and which stores the solution on disk. Generates a **Batch** job configuration.
- **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 configuration and a **Parametric** job configuration.
- **Cluster Computing** (): Used to solve the problem on a distributed-memory computer architecture. Generates a **Cluster Computing** job configuration and a **Batch** job configuration.
- **Cluster Sweep** (): Used to formulate a sequence of problems that arise when you vary some parameter in the model. The program computes the solution for each parameter tuple on a distributed-memory computer architecture. The results are stored on file and updated into the model. Generates **Cluster Computing**, **Batch**, and (if applicable) **Parametric** job configurations.

The following sections describe some common functionality in the study step nodes' settings windows. For specific information for each study type, see [Study Step Types](#).

Common Study Step Settings

The study steps form a solver configuration that computes the solutions for the study. The study step nodes' settings windows contain the following sections (in addition to specific study settings for each type of study step):

RESULTS WHILE SOLVING

Specify which plot to display while solving and also which probes to include.

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 that are compatible with the study type. See [Selecting Physics and Variables in the Study Steps](#).

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.

MESH SELECTION

Specify—for each geometry—which mesh to use for the study step.

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.



- [Specifying Values of Dependent Variables](#)
 - [Individual study steps described in Study Step Types](#)
-

Working with Studies and Solvers

Selecting Physics and Variables in the Study Steps

All study step settings windows contain a **Physics and Variables Selection** section, which you can use to control which physics (or even specific variables and physics nodes) to solve for. This can be useful for:

- Solving physics in a sequence, including different physics features in each step.
- Solving and comparing different “analysis cases” for a model (sharing the same geometry and material) by varying boundary conditions, sources, or variables without the need to enable and disable nodes in the physics and recompute the solution.

By default, you can select from the participating physics. To select individual physics features and variables, select the **Modify physics tree and variables for study step** check box.

SELECTING PHYSICS TO SOLVE FOR

The **Physics** column contains the names of all physics in the model. You can choose to not solve for one or more of the physics by clicking the button in the **Solve for** column (by default, a study solves for all physics). The button then changes to an to indicate that the physics is not solved for, but it still provides values for the degrees of freedom (dependent variables) according to the settings for values of variables not solved for (see [Specifying Values of Dependent Variables](#)). Click the button to solve for the physics. In the **Discretization** list you can specify which discretization to use. The default (and often the only choice) is **Physics settings**, which means that the study uses the discretization from the main physics node’s settings. Changing it affects the discretization order used by this study. To add another discretization, use a separate **Discretization** node in the physics. The leftmost column is usually empty but contains a warning () if the physics’ degrees of freedom are not solved for regardless of the setting in the **Solve for** column. This can be the case if the physics is not compatible with the study step.

SELECTING VARIABLES AND PHYSICS NODES TO INCLUDE

If the **Modify physics tree and variables for study step** check box is selected, you specify to select individual variables or physics features to include in the model that you solve. The **Physics and Variables Selection** section then contains a tree that is a copy of the

following parts of the model tree in the Model Builder (see Figure 19-1):

- **Variables** nodes under **Global Definitions**.
- **Variables** nodes under **Model>Definitions** for all Model branches.
- All physics nodes in the Model branches.

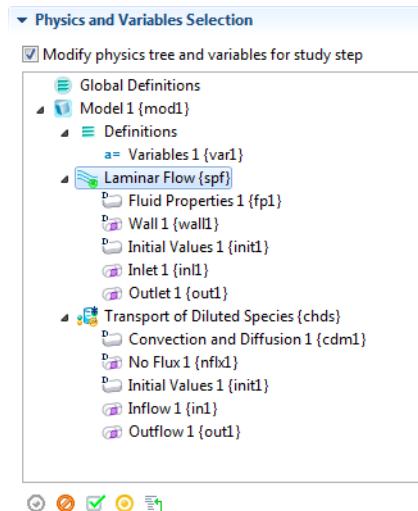


Figure 19-1: An example of a Physics and Variables Selection section tree when the Modify physics tree and variables for study step check box is selected.

It is possible to include or exclude all variables, physics, and physics nodes in a study step (that are not disabled in the model tree). Select one or more nodes in the tree and right-click or use the buttons at the bottom of the section (below the tree) to change their status. Click the **Go to Source** button () to move to the corresponding original node in the model tree. The following options are available:

Disabling and Enabling Physics and Variables Selection Nodes

Click **Disable** (or right-click to select from the context menu) to disable enabled nodes that are possible to disable. The contributions, conditions, or variables in a node that you disable are not included in the study when solving. You can also disable selected nodes by clicking the **Disable** button () underneath the tree. A disabled node is grayed out in the tree.

Click **Enable** (or right-click to select from the context menu) to enable disabled nodes. The contributions, conditions, or variables in a node that you enable are

included in the study when solving. You can also enable selected nodes by clicking the **Enable** button () underneath the tree.

When you right-click, the following context menu options mean that a node cannot be enabled or disabled:

- **Cannot be Disabled**—for default nodes in the physics interfaces.
- **Disabled in Model Builder**—for nodes that you have disabled in the Model Builder.
- **Not Applicable**—for physics nodes that are not applicable for the study type in the study step. The item in the tree is not available.

When solving, equations and variables are generated as if the disabled nodes in the tree were disabled in the Model Builder. This means that the nodes' selections override each other as if the nodes were disabled in the Model Builder.

Change of States and Override and Contribution Indicators

An asterisk displays in the upper-right corner of nodes for which the state has been changed in the study step's selection tree compared to the state in the Model Builder.

In this example, under the **Physics and Variables Selection** section, a Transport in Diluted Species interface () is disabled (grayed out), provides no degrees of freedom (red dot in the lower-right corner), and has a change of state indicated by the asterisk. The asterisk means the Laminar Flow physics in the Model Builder is not disabled. Also see [Figure 19-1](#) for another example. In general, any variable or physics node in the Model Builder that is disabled in any study step gets an asterisk in the upper-right corner. For physics this applies also when you have not selected the **Modify physics tree and variables for study step** check box and the physics is disabled in the **Solve for** column in the **Physics and Variables Selection** section.

The dynamic visual icon indicators for overridden and contributing nodes also appear in the tree in the settings window for the study steps when you have selected the **Modify physics tree and variables for study step** check box in a study step's settings window. When you select a physics node in the tree that appears, the override and contribution icon indicators appear in the same way as in the Model Builder when you select a physics node, but if you disable any physics node in the study step's tree, the icon indicators then show how the physics node override and contribute to the model when one or more physics nodes are disabled in the study step.

Options and States for the Physics

The following options are available for the main physics nodes under the **Physics and Variables Selection** tree. Right-click a node and select one of the following from the context menu or click the button beneath the tree (see [Figure 19-1](#)). Selecting these

options affects the entire physics user interface. Select:

- **Solve For** (the default setting) to solve for the physics, including all enabled physics nodes and the contributions, constraints, and variables that are added. This is similar to the button when you specify what physics to solve for without the selection tree.

A physics user interface in this state shows a small green circle in its lower-right corner to indicate that the study step solves for the degrees of freedom (dependent variables) in the physics. This is an example of a Laminar Flow physics with the green dot (.

- **Provide Degrees of Freedom** to not solve for the physics but provide degrees of freedom (dependent variables) and other physics variables using the settings for values of variables not solved for (see [Specifying Values of Dependent Variables](#)).

A physics in this state shows a small yellow square in its lower-right corner to indicate that the study step provides degrees of freedom but does not solve for the physics. In this example, a Laminar Flow physics is both showing that it provides degrees of freedom (yellow dot in the lower-right corner) and has a change of state indicated by the asterisk (.

- **Disable** to disable a physics. The physics does not contribute to the study and no variables, including the degrees of freedom (dependent variables) are included.

A disabled physics is grayed out and shows a small red square in its lower-right corner to indicate that the study step provides no degrees of freedom for it. In this example, a Transport of Diluted Species interface (is disabled (grayed out), provides no degrees of freedom (red dot in the lower right corner), and has a change of state indicated by the asterisk.

In addition, the physics can be in the following states:

- If the physics is disabled in the Model Builder, it appears grayed out and shows a small red icon in its lower-right corner. If you right-click, the context menu contains **Disabled in Model Builder**. In this case, none of the options above is available.
- If the physics is not applicable because it does not support the study step, then by default it has the **Provide Degrees of Freedom** setting, and you can also choose **Disable**. **Solve For** is not available.

Discretization Selection

You can also right-click a physics node in the selection tree to select the discretization. The discretizations appear at the bottom of the context menu (underneath the horizontal divider). In most case, the only option, and the default, is **Physics Settings**,

which takes the discretization from the physics user interface's settings window, but if you have added separate **Discretization** nodes, you can select from one of those instead of **Physics Settings**.



Key to Nodes and Toolbar Buttons



If you have the AC/DC Module, see [Electric Shielding: Model Library](#) path **ACDC_Module/Resistive_Devices/electric_shielding**.

Generating Solver Configurations and Job Configurations

When you have added the study steps to the study, a solver configuration (and maybe a job configuration) is generated after right-clicking the study and selecting **Compute** () or pressing F8.

The generated solver configuration 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 displays automatically if the **Job Configuration** node has content.

It is possible to have several solver configurations 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.



Solver Configurations and Job Configurations.

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.

Clearing, Deleting, Disabling, and Enabling a Solver or Solutions

Use a **Clear** function to keep the nodes and be able to recreate the solver configurations by computing again. Use a **Delete** function to completely remove a solver configuration.



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** (X) or press the Delete key. Click **Yes** 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** (X). Click **Yes** 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.

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 (S) and select **Compute** (=), or to press F8. This always runs the enabled solver configuration with a green border around its icon. However, if a solver configuration is disabled, you can still run it by right-clicking it and selecting **Compute** (=). If the study contains more than one study step, you can also right-click and select **Compute Selected Step** (=) to compute just that study step, or select **Compute to Selected** (=*) (for all study steps except the first one) or **Compute From Selected** (=*) (for all study steps except the last one) to compute only a part of the study steps in a study.

Sometimes the default settings in the study steps are not sufficient to specify the details of how to obtain a solution. For example, if you want to change a tolerance or use a different time-stepping method, you can do that in the attached solver configurations. To run such a solver configuration, which has been manually altered, right-click the solver configuration (S) and select **Compute** (=). Alternatively, enable the solver configuration, right-click the **Study** node (S), and select **Compute** (=).

If your solver configuration 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 **Compute** (=), or pressing F8, runs all solvers in the solver configuration. These tools are also available from the main toolbar.

By default, a study creates a **Solution** data set and plot groups with results plots suitable for the physics 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 and functions evaluated using the initial values. To make the initial values available for results evaluation and plotting, right-click the **Study** node () and select **Get Initial Value** () . By default this plots the initial values of the variables solved for as specified by the **Field** subnode () under a **Dependent Variables** node () for the first study step in the solver configuration. For each study step you have also access to **Get Initial Value for Step** () , which evaluates the initial value for that step. In comparison, **Get Initial Value** () for the study evaluates the first study step's initial values. These buttons are also available on the main toolbar.

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 under **Output** in the **Dependent Variables** node ().

Getting Results While Solving

The ability to get numerical results and plots of the intermediate solutions while solving can be useful for diagnosing problems and for monitoring the progress of the solution. 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 **Table** 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 **Table** window and a **Probe Plot** window.



You can use probes to tabulate values of interest during a large parametric simulation, for example. It can then be possible to keep only the last solution in memory during the parametric sweep, which potentially can significantly reduce memory requirements and the simulation time.

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. 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.

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.

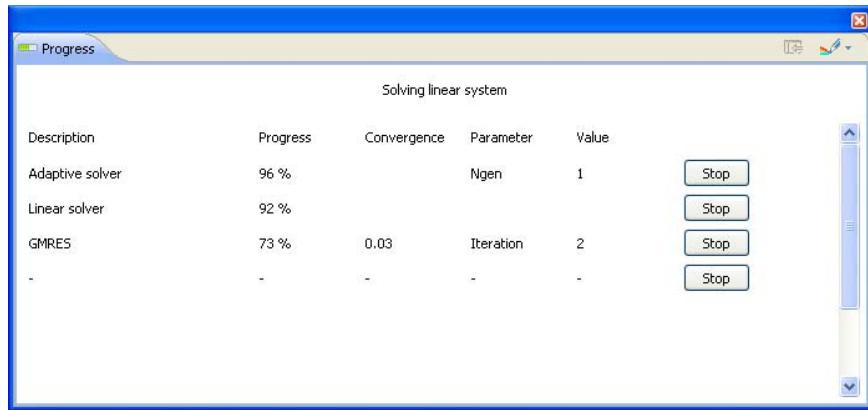


Figure 19-2: The Progress window for the solution of a linear stationary problem using the adaptive mesh refinement solver.

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 and Cancel 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. 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.



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.

When a solver starts working, it prints the number of degrees of freedom in the linear systems to be solved to the log. For certain problems, there are additional degrees of freedom involved in the discrete problem formulation that do not affect the size of the matrices assembled by the solver. These are called *internal degrees of freedom* and are displayed separately from the actual degrees of freedom in the log. For example, when solving plasticity problems in structural mechanics, plastic strains are represented by internal degrees of freedom.

When a solver has finished it reports the following information:

- The solution time (in seconds)
- The maximum amount of physical memory used (in MB)
- The maximum amount of virtual memory used (in MB)

In addition, the log includes the following information that is specific to the type of solver:

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 the Double dogleg solver (**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.



Convergence Criteria for Iterative Solvers

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 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 nonoptimality 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 **Levenberg-Marquardt** solver contains the following data:

- The number of Levenberg-Marquardt iterations (**Itns**).
- The current Levenberg-Marquardt factor (**ImFact**). 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.

CONVERGENCE 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 tool bar to copy the convergence data to a table in the **Table** window.



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). See [Convergence Criteria for Iterative 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**. The segregated solver shows one plot with one graph for each segregated step.

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. Each linear solver used has a separate plot window.

When using the parametric solver, the graphs for the different parameter steps are merged, and the convergence plots use the accumulated number of iterations. The graphs for the different nonlinear and linear solve steps are concatenated. The plot uses 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.

Specifying Values of Dependent Variables

When you have physics in a study step that you do not solve for but that provide degrees of freedom, you can specify how COMSOL Multiphysics handles the values of such degrees of freedom (dependent variables).

The settings window for all study types has the following section:

VALUES OF DEPENDENT VARIABLES

The settings in this section determine how the solver handles dependent variables that you do not solve for. This is applicable in, for example, a solver configuration where you only solve for a subset of the dependent variables in each step.

By default, COMSOL determines these values heuristically depending on the model as, for example, the specified initial values or a solution from an earlier study step. To specify the initial values of the dependent variables that you solve for, select the **Initial values of variables solved for** check box. To specify the values of dependent variables that you do not solve for, select the **Values of variables not solved for** check box. Then use the **Method** list to specify how to compute the values of variables not solved for. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object (a solution from a study step).

Use the **Study** list to specify what study to use if **Method** has been set to **Solution**:

- Select **Zero solution** to initialize all variables to zero.
- Select any other available study to use it as initial value.

Depending on the study type, you can choose different solutions from a list underneath the **Study** list:

- For a stationary study, from the **Selection** list, select **Automatic** (the default) to use the last (typically the only) solution, select **First** to use the first (typically the only) solution, select **Last** to use the last (typically the only) solution, select **All** to use all (typically just one) solutions from that study, select **Manual** to use a specific solution number that you specify, or select **I** to use the first (typically the only) solution. If you use a parametric continuation of the stationary study, there can be additional solutions to choose from.
- For a time-dependent study, from the **Time** list, select **Automatic** (the default) to use the solution for the last time, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Interpolated** to specify a time in the text field that opens and use the interpolated solution at that time, select **Manual** to use a specific solution number that you specify, or select one of the output times to use the solution at that time.
- For an eigenvalue study, from the **Selection** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that

study, select **Manual** to use a specific solution number that you specify, or select one of the eigenvalues to use the corresponding eigensolution.

- For a parametric or frequency-domain study, from the **Parameter value** list, select **Automatic** (the default) to use the first parameter value set or frequency, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the parameter value sets or frequencies to use the corresponding solution.

Remarks on Solver-Related Model Characteristics

THE IMPORTANCE OF A CORRECT JACOBIAN MATRIX

The solvers break down each problem—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.

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 converge 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 user-defined functions and then use that function in some PDE or boundary condition with arguments that depend on the solution.
- If you use a MATLAB M-file function or the External function interface, 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. A correct Jacobian can be computed if the function derivative is defined.
- If you use nonanalytic functions in a complex-valued problem, such as `real`, `imag`, `conj`, `abs`, or `arg`.

- If you use the `nojac` operator.



- [About Global and Local Definitions](#)
 - [MATLAB](#)
-

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 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 **Time-Dependent Solver** node's settings window, and in the **Advanced** section 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`, go to the **Stationary Solves>Advanced** subnode's settings window, and in the **General** section select the **Allow complex-valued output from functions with real input** check box. By default, COMSOL gives an error message if a real input to a function generates a complex output.

Study Step Types

The following sections contain information specific to each of the available study steps.

Stationary

A **Stationary** study node () corresponds to a stationary solver (default), a parametric solver, or an optimization solver. The Stationary study generates a solver that is used to solve a stationary problem.

Use the *parametric continuation solver* (which you activate by the **Continuation** check box in the **Study Extensions** section) to find the solution to a sequence of stationary PDE problems that arise when you vary some parameter of interest. This can be any parameter that defines an equation, boundary condition, material property, or similar property of the physics but not parameters that, for example, vary the geometry or mesh (for such a parameterization, use a *parametric sweep*). The parametric solver can also prove useful when it is difficult to get convergence in a nonlinear model. You can then introduce a parameter such that the solution is easy if the parameter is small. Then, to obtain the solution for the desired value of the parameter, slowly increase its value. This way, the nonlinear solver gets a good initial guess based on the solution for the previous parameter value.

Continuation always uses the parametric solver; that is, it cannot be used as a “for-loop.” The main advantage of the parametric solver over parametric sweep is that it can solve some problems considerably faster. The parametric solver analyzes the matrices and reuses those which are constant from step to step. If the Jacobian is constant, it can be factorized only once when using a direct solver, so each parameter value only lead to a back substitution. By default the **Parametric Sweep** study step chooses the continuation solver when possible, which means that you do not have to know the difference: the most efficient solver is chosen for the problem at hand.

STUDY SETTINGS



If you have a license for the Acoustics Module, MEMS Module, or Structural Mechanics Module, the **Study Settings** section includes an **Include geometric nonlinearity** check box.

If your model involves structural mechanics, select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear this check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

RESULTS WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** list. The software plots the data set of the selected plot group as soon as the results become available.

Use the **Probes** list to select any probes to evaluate. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (✚) buttons to make the list contain the probes that you want to see results from while solving.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. For each geometry listed in the **Geometry** column, select a mesh from the list of meshes in the **Mesh** column. Each list of meshes contains the meshes defined for the geometry that you find on the same row.

STUDY EXTENSIONS

This section contains some optional extensions of the study, such as a parametric continuation solver, load cases, and adaptive mesh refinement.

Parametric Continuation Solver

To use the parametric continuation solver, select the **Continuation** check box. Specify the parameter to vary in the **Continuation parameter** field. The parameter can occur in equations, constraints, or expressions but not as a geometry or mesh parameter, for example. The list you enter in the **Parameter values** field is a vector. It must contain a monotonically increasing or decreasing sequence of real parameter values, for example, `range(0,0.1,1.5)` or `0 10^(range(-3,2))`.

Load Cases

Select the **Define load cases** check box to define load cases as combinations of defined load groups, multiplied with optional weights (load factors), and constraint groups. Load cases are useful for efficiently solving for a number of cases with varying loads (and constraints) in the same model without the need to reassemble the stiffness matrix. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (+) buttons to make the list contain the load cases that you want to solve for. For each load case, click in the column for the load groups and constraint groups that you want to include in the load case. By default, no load groups and constraint groups are included (✗). Load groups and constraint groups that are included appear with a check mark (✓). Optionally, change the default weights for the load groups from 1.0 to another value in the corresponding **Weight** column (which is to the right of the load group that it is acting on). A weight of 1.5, for example, adds an extra 50% to the magnitude of the loads in the load group; a weight of -1 reverses the direction of the loads.



- Load Group and Constraint Group
- Using Load Cases

Adaptive Mesh Refinement

Select the **Adaptive mesh refinement** check box if you want to use adaptive mesh refinement. Select the geometry to use for the adaption from the **Adaption in geometry** list. Click the **Go to Source** button (↗) to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh Refinement subnode, its settings, and the adaptive mesh refinement algorithm.

Time Dependent

The **Time Dependent** study node (⌚) gives a solver with a **Time-Dependent Solver** node (⌚) or an **Optimization Solver** node (⌚).

STUDY SETTINGS

Specify the time interval for the output from the simulation in the **Times** field. You can type a monotonically increasing list of individual values, for example, 0 1 2 5 10 20; use the range operator, for example, `range(0,0.1,1.5)`, which gives time steps from 0 to 1.5 s with a step size of 0.1 s; or use any combinations of such input. Check the **Relative tolerance** setting if you want to override the relative tolerance suggested by the program. The tolerance settings control the internal time steps taken by the solver, so

selecting large time steps for the output times does not affect the accuracy in the time stepping.

When plotting the results from a time-dependent simulation you can choose to plot the solution at any of the times specified in the **Times** field. You can also plot an interpolated solution at any intermediate time. The interpolation used between times is a cubic Hermite spline; that is, the interpolation uses both the solution values and their time derivatives at two points: the closest output times before and after the time for which the interpolated solution is computed.



If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

RESULTS WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** and **Update at** lists. The software plots the data set of the selected plot group as soon as the results become available either at the times specified by the output times (from the **Times** field) or at a set of internal times defined by the solver. Select **Output times** (the default) or **Time steps from solver** from the **Update at** list.

Use the **Probes** list to select any probes to evaluate. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (✚) buttons to make the list contain the probes that you want to see results from while solving. For the probes you also select **Output times** or **Time steps from solver** (the default) from the associated **Update at** list at the bottom of the section.



Probes

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS



This section contains some optional extensions of the study. The options are mutually exclusive and only one of the check boxes can be selected.

Select the **Auxiliary sweep** check box to enable an auxiliary parameter sweep so that for each set of parameter values, a time-dependent problem is solved for. The auxiliary sweep is a multiparameter sweep with the time as the parameter at the innermost level; see [Parametric Sweep](#) for more information about parameter sweeps.

Select the **Adaptive mesh refinement** check box if you want to use adaptive mesh refinement. Select the geometry to use for the adaptive mesh refinement from the **Adaption in geometry** list. Click the **Go to Source** button () to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh Refinement subnode, its settings, and the adaptive mesh refinement algorithm.

Select the **Automatic remeshing** check box if you want the solver to remesh automatically when the quality of the mesh becomes poor in a time-dependent study. Select the geometry to use for the automatic remeshing from the **Remesh in geometry** list. With automatic remeshing active, the solver adds an **Automatic Remeshing** subnode under the **Time-Dependent Solver** node. In that subnode you specify the mesh quality expression that determines when to remesh.

Time Discrete



The settings available for the **Time Discrete** node () are some of the same settings available for the **Time Dependent** node.

The time-discrete solver is used to solve fluid dynamics problem using a projection method. Also, you can use the time-discrete solver to solve problems that have been discretized in time using the `prev` or `bdf` operator.

In the time-discrete solver you cannot use the `d` operator to get the time derivatives. That is why the `prev` operator is needed so that time derivatives can be written using the backward Euler method. However, the `prev` operator does not make it possible to define variables like this:



```
a=f(prev(a))
```

That is, you cannot use the time-discrete solver to solve incremental problems in time because a variable cannot be expressed in terms of itself. Such a formulation leads to a circular variable dependency. If you want to implement a variable that is dependent on itself, that variable must be a dependent variable that you solve for as a continuous ODE or PDE.

Time-Dependent Modal



The settings for the **Time-Dependent Modal** study node () are the same as those for the **Time Dependent** node.

Eigenfrequency

Use an **Eigenfrequency** study node () when you want to solve an eigenvalue problem for a set of eigenmodes and associated eigenfrequencies. An Eigenfrequency study node gives a solver with an **Eigenvalue Solver** node ().

STUDY SETTINGS

Use the **Desired number of eigenfrequencies** field to specify the number of eigenfrequencies you want the solver to return.

In the **Search for eigenfrequencies around** field, you can specify a value around which the eigenvalue solver should look for solutions to the eigenvalue equation



The eigenvalue solver can in some cases return more than the desired number of eigenfrequencies (up to twice the desired number). These are eigenfrequencies that the eigenvalue solver finds without additional computational effort.



If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select which mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS

Select the **Adaptive mesh refinement** check box if you want to use adaptive mesh refinement. Select the geometry to use for the adaption from the **Adaption in geometry** list. Click the **Go to Source** button () to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh

Refinement subnode, its settings, and the adaptive mesh refinement algorithm.

Eigenvalue

Use an **Eigenvalue** study node () when you want to solve an eigenvalue problem for a set of eigenmodes and associated eigenvalues. An Eigenvalue study node gives a solver configuration with an **Eigenvalue Solver** node ().

STUDY SETTINGS

Use the **Desired number of eigenvalues** field to specify the number of eigenvalues you want the solver to return.

In the **Search for eigenvalues around** field, you can specify a value around which the eigenvalue solver should look for solutions to the eigenvalue equation.

	The eigenvalue solver can in some cases return more than the desired number of eigenvalues (up to twice the desired number). These are eigenvalues that the eigenvalue solver finds without additional computational effort.
	If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model involves structural mechanics, then the Study Settings section includes an Include geometric nonlinearity check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS

Select the **Adaptive mesh refinement** check box if you want to use mesh adaption. Select the geometry to adapt the mesh for from the **Adaption in geometry** list. Click the **Go to Source** button () to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh Refinement subnode, its settings, and the adaptive mesh refinement algorithm.

Frequency Domain

The **Frequency Domain** study node () corresponds to a frequency sweep.

STUDY SETTINGS

Specify the frequencies to use for the frequency sweep. Type the frequencies in the **Frequencies** field using space-separated numbers or the **range** function.

Use the **Load parameter values** field to select a file with parameter values. You can browse your file system for files by clicking the **Browse** button. After selecting a file click the **Read File** button to load the parameter values into the **Frequencies** field.



If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model includes a physics interface that involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

RESULTS WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** list. The software plots the data set of the selected plot group as soon as the results become available.

Use the **Probes** list to select any probes to evaluate. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (☒), and **Add** (✚) buttons to make the list contain the probes that you want to see results from while solving.



Probes

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS

In addition to the frequency sweep, you can merge it with an auxiliary sweep over one or several extra parameters. Select the **Auxiliary sweep** check box to enable the extra parameter sweep. The auxiliary sweep is a multiparameter sweep with the frequency as the parameter at the innermost level; see [Parametric Sweep](#) for more information about parameter sweeps.

You enable the *asymptotic waveform evaluation (AWE) solver* by selecting the **Use asymptotic waveform evaluation** check box. The Frequency Domain study generates a solver configuration that is used to solve a stationary parametric problem or an asymptotic waveform evaluation problem.

Frequency Domain Modal

The **Frequency Domain Modal** study node (周恩) corresponds to a modal frequency sweep for systems with frequency-based loads.

STUDY SETTINGS

Specify the frequencies to use for the frequency sweep. Type the frequencies in the **Frequencies** field using space-separated numbers or the range function.

Use the **Load parameter values** field to select a file with parameter values. Click the **Browse** button to browse the file system. After selecting a file click the **Read File** button to load the parameter values into the **Frequencies** field.



If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

Multigrid Level

Click the **Show** button () and select **Advanced Study Options** and then right-click any of the **Study Step** nodes to add a **Multigrid Level** node (). This is a subnode to the other study step nodes and it describes a geometric multigrid level used by the study.

PHYSICS SELECTION

Select a **Physics interface**. You can also choose to include no physics by clearing the **Use in this study** check box (selected by default). In the **Discretization** list, you can specify

which discretization to use. Changing it affects the discretization order used by this study. You have to add the discretization you want to use in the physics.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

Parametric Sweep

Use a **Parametric Sweep** study node () to find the solution to a sequence of stationary or time-dependent problems that arise when you vary some parameters of interest. The parametric sweep can include multiple independent parameters directly for a full multiparameter sweep (solve for the first value of the first parameter combined with all values of the second parameter, then the second value of the first parameter combined with all values of the second parameter, and so on, or use a specified combination of parameter values). You can also add more than one **Parametric Sweep** node to create nested parametric sweeps. The program then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** nodes’ names.



It is only possible to use one of the Sensitivity, Optimization, or Parametric Sweep study step in any study.

The **Parametric Sweep** node’s settings window contains the following sections.

STUDY SETTINGS

Use the **Sweep type** list to specify the type of sweep to perform. The **Specified combinations** type (the default) solves for a number of given combination of values, while the **All combinations** type solves for all combination of values. Using all combinations can lead to a very large number of solutions.

Use the **Parameter names** and **Parameter value list** table to specify parameter names and values for the parametric solver. Use the **Add** button () to add a row to the table. Each row has one parameter name and a corresponding parameter value list. For the **Specified combinations** sweep type, the list of values must have equal length. When you click in the **Parameter value list** column to define the parameter values you can click the **Range** button () to define a range of parameter values.

If more than one parameter name have been specified the lists of parameter values is interpreted as follows: Assume that the parameter names are p1 and p2, and that p1 has the list 1 3 and p2 has the list 2 4. For the **Specified combinations** sweep type, the solver first uses p1 equal to 1 and p2 equal to 2. Thereafter, it uses p1 equal to 3 and p2 equal to 4. And when the sweep type in **All combinations**, the solver uses the following order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. Use the **Load from file** button () to browse to such a text file. The program appends the read names and values to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values, and a space separating the values. Click the **Save to file** button () to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®).

OUTPUT WHILE SOLVING

Select the **Plot** check box to allow plotting of results while solving. Then select what to plot from the **Plot group** list. The software plots the data set of the selected plot group as soon as the results become available.

Use the **Probes** list to select probes to update during the parametric sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (), **Move Down** (), **Delete** (), and **Add** () buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for parametric

sweep. Note that the control of tables and plot windows are controlled by the probe settings.



If a probe is updated on the **Parametric Sweep** level and also through another solution process (for example, a time-dependent simulation) this probe is not updated at the **Parametric Sweep** level. When the probes themselves (not the probe expression) depend on model parameters, the update of these probes is only correct for parameter sweeps that are done through outer parametric sweeps (not by a parametric solver). Outer parametric sweeps are performed by a **Parametric** node under **Job Configurations**. COMSOL Multiphysics currently does not autodetect model parameters in probes, so you might want to select **Off** from the **Use parametric solver** list in the **Study Extensions** section for the **Parametric Sweep** study node.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the parametric sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.



No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to modify the table data and make so-called response surfaces directly from the **Results** view toolbar **Surface Plot** button.

You can use the **Memory settings for jobs** and the **Keep solutions in memory** list to control how to store the solutions from the individual parametric sweep solutions. Select **All** to store all the parametric sweep solutions in memory, or select **Only last** to store only the last solution from the parametric sweep. When only the last solution is stored you can also select the **Save each solution as model file** check box. It stores the separate parametric sweep solutions and their corresponding models in separate MPH-files. Enter a filename in the **Filename** field or click **Browse** to choose a name and location for the model files. You can also use probes to collect some solution values of interest during the sweep rather than storing all solutions, which can save memory and solution time.

STUDY EXTENSIONS

From the **User parametric solver** list, select one of the following options:

- **Automatic** (the default) to generate a **Parametric** job configuration, unless the problem and parameters are such that the parametric sweep can be realized through a **Stationary Solver** with a **Parametric** solver node (), which is more efficient.
- **Off** to always generate a **Parametric** job configuration.

-
- | | |
|---|---|
|  | <ul style="list-style-type: none">• Probes• Stationary for information about the continuation parametric solver versus parametric sweeps.• Job Configurations• Using a Job Configuration to Store Parametric Results on File |
|---|---|

Batch

Use a batch job to start a COMSOL Multiphysics batch process that solves the current study on your computer. Click the **Show** button () and select **Advanced Study Options** and then right-click the **Study** node to add a **Batch** study node (). Once you have set the filename and directory, right-click the parent Study node and choose **Compute**  to start a COMSOL batch process that computes the current study. When the process has started the COMSOL Desktop® enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Detach Job** button in the **Progress** window. When the process finishes or you click the **Detach Job** button, an **External Process** node that represents the current running process appears under a **Batch Data** node, which in turn appears as a subnode of the **Batch** node under the **Job Configurations** node (). For more information, see [Batch](#), [Batch Data](#), and [External Process](#).

	You cannot create another Batch study, Batch Sweep study, Cluster Sweep Study, or Cluster Computing study in the same study at the same time.
---	---

BATCH SETTINGS

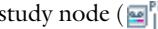
Specify the file where to store the model in the **Filename** field. Choose the directory where to store the model in the **Directory** field by typing it directly or clicking the **Browse** button to choose a batch directory.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that create graphical contents such as exporting to file. Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before it must inform that it is still running. Failure to do so means that the process is considered dead and a new process is started if not the maximum number of job restarts is reached.

Batch Sweep

Click the **Show** button () and select **Advanced Study Options** to enable this option.

Use the **Batch Sweep** study node () to find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest. If you want to make a full multiparameter sweep (solving first for the first value of the first parameter combined with all values of the second parameter, then for the second value of the first parameter combined with all values of the second parameter, and so on), you can add several **Parametric Sweep** nodes, one for each additional parameter except the outermost parameter. The program then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** nodes’ names. The **Batch Sweep** is always the outermost sweep. It starts multiple COMSOL Multiphysics batch processes—one for each parameter set in the **Batch Sweep** node—that solves the current study on your computer given the parameter set.

Click the **Show** button () and select **Advanced Study Options** and then right-click the **Study** node to add a **Batch Sweep** study node (). Once you have set the filename and directory, right-click the parent **Study** node and choose **Compute**  to start the COMSOL batch processes that computes the current study. When the processes has started the COMSOL Desktop® enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Detach Job** button in the **Progress** window. The **Batch Sweep** then no longer synchronizes the solutions and accumulated probe table. To reenable the synchronization use the **Show All Progress** button in the **Batch Data** node. When the process finishes or you click the **Detach Job** button, an **External Process** node, one for each parameter, representing the running processes appears under a **Batch Data** node, which in turn appears as a subnode

of the **Batch** node under the **Job Configurations** node (). For more information, see [Batch](#), [Batch Data](#), and [External Process](#).



You cannot create another Batch study, Batch Sweep study, Cluster Sweep study, or Cluster Computing study in the same study at the same time.

STUDY SETTINGS

The batch sweep is a multiparameter sweep with its parameters solved as a batch job; see [Parametric Sweep](#) for more information about parameter sweeps.

OUTPUT WHILE SOLVING

Use the **Probes** list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (), **Move Down** (), **Delete** (), and **Add** () buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so on) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not selected, the probes selected by the **Probes** selector are used.



No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to make so-called response surfaces directly from the **Results** view toolbar **Surface Plot** button.



Probes

BATCH SETTINGS

Specify the file where to store the model in the **Filename** field. Choose the directory where to store the model in the **Directory** field by typing it directly or clicking the **Browse** button to choose a batch directory.

Select the **Clear meshes** check box to clear the meshes before running the batch sweep. The default is to clear the meshes. Select the **Clear solutions** check box to clear the solutions before running the batch sweep. The default is to clear the meshes. Select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional postprocessing to be performed after the sweep has finished. The setting is similar to the **All** and **Last** setting in the **Memory settings for jobs** for **Parametric Sweep**. The default is to disable solution synchronization. Select the **Synchronize accumulated probe table** check box to synchronize the accumulated probes computed by the batch processes with the model. The accumulated probe synchronization is enabled by default. Select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases the solution synchronization and probe synchronization functionality should be used instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that create graphical contents such as exporting to file. Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before it must inform that it is still running. Failure to do so means that the process is considered dead, and a new process is started if the maximum number of job restarts is not reached.

Cluster Computing (Study)

Click the **Show** button () and select **Advanced Study Options** and then right-click the **Study** node to add a **Cluster Computing** node.

Use the **Cluster Computing** study node () when you want to submit COMSOL Multiphysics batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster. For more information, see [Cluster Computing \(Job Configuration\)](#) and its related functionality. It is also intended as a user interface for

setting up distributed COMSOL batch jobs on the computer where the COMSOL Desktop is running. If you are running in distributed mode interactively, it is usually not needed unless you want a simple way to distribute parametric sweep.



You cannot create another Batch study, Batch Sweep study, Cluster Sweep study, or Cluster Computing study in the same study at the same time.

Once you have specified the settings, right-click the main **Study** node and select **Compute** in order to start a COMSOL Batch process that solves the current study. Once the process is started the COMSOL Desktop® enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Detach Job** button in the **Progress** window. Once the process is finished or you click the **Detach Job** button, an **External Process** node is selected that represents the current running process. Click the **Save as Default** button () in the settings window's toolbar to save the current setting as default. If you are running COMSOL in distributed mode the model runs in the current process. In this case only the **Distribute parametric sweep** check box is selected.

BATCH SETTINGS



After making these settings, click the **Save as Default** () button on the settings window toolbar to save the current directory settings as the default preference.

Choose the **Cluster type—General** (the default), **HPCS 2008**, **WCCS 2003**, **OGS/GE**, or **Not distributed**:

General

Select **General** (the default) to configure to run on many types of clusters and schedulers, including Linux clusters.

- When **General** is selected, and you have started a multiprocessor daemon (MPD) on the computer, click to select the **MPD is running** check box.
- The entry in the **Host file** field specifies the host file used for the job. If left empty, MPD looks for a file `mpd.hosts` in the Linux home directory.
- Select the bootstrap server that MPI should use in the **Bootstrap server** setting.

- If your cluster is Linux and it requires that an ssh (secure shell) or an rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter details for the [Filename and Directory Settings for all Cluster Types](#) and [Use Batch License Settings for all Cluster Types](#).

HPCS 2008 or WCCS 2003

Select **HPCS 2008** to use the Windows HPC Server 2008 job scheduler to submit the batch job. Select **WCCS 2003** to use the Windows Compute Cluster Server 2003 job scheduler to submit the batch job.

When **WCCS 2003** or **HPCS 2008** is selected:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter details for the [Filename and Directory Settings for all Cluster Types](#) and [Use Batch License Settings for all Cluster Types](#).

OGS/GE

Select **OGS/GE** to use the Open Grid Scheduler/Grid Engine job scheduler to submit the batch job. When **OGS/GE** is selected:

- Select which bootstrap server that should be used by MPI using the **Bootstrap server** setting.
- If your cluster is Linux and it requires that an ssh (secure shell) or an rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol.
- Specify the **Number of slots** and the name of the scheduler queue in **Queue name**.
- Enter the **Queue name** for the cluster queue.
- Enter details for the [Filename and Directory Settings for all Cluster Types](#) and [Use Batch License Settings for all Cluster Types](#).

Not Distributed

Select **Not distributed** when you want to submit a batch job to a job scheduler without running a distributed job. Enter details for the [Filename and Directory Settings for all Cluster Types](#) and [Use Batch License Settings for all Cluster Types](#).

Filename and Directory Settings for all Cluster Types

Specify the file where to store the model in the **Filename** field. Choose the mounted file system directory where to store the model in the **Directory** field by entering it directly or by clicking the **Browse** button to choose a batch directory.

- If the batch job has another path to the directory, select the **Specify external COMSOL batch directory path** check box and enter the path to the external process (batch) directory in the **Directory** field or click **Browse**. The cluster job uses this path from the compute node to access the input file and write back the result. On Windows this must be a fully qualified UNC path, for example, `\head1\shared\clusterprojects`.
- If COMSOL is installed in another directory than where the COMSOL Desktop runs, select the **Specify external COMSOL installation directory path** check box and then specify the installation directory (click **Browse** or enter the path to the **Directory**). This can occur if you are submitting jobs to a job scheduler. Typically on Windows this is the UNC path to the COMSOL installation root directory for the compute nodes to access the required COMSOL binaries—for example, `\head1\shared\COMSOL43b`.

Use Batch License Settings for all Cluster Types

Select the **Use batch license** check box to run using batch licenses. Batch licenses can be used to run multiple batch jobs for different models that only depend on a parameter. Usually you should use the **Cluster Sweep** node.

REMOTE AND CLOUD ACCESS

Enable the **Run remote** check box if you want to run COMSOL on a remote machine using a remote start command such as ssh and using a file transfer program such as scp to transfer the files to and from the remote computer. This allows you to run on a

machine installed on your network without a client server connection or on a machine installed on a remote cloud.



COMSOL must be installed on the remote machine and all settings must be specified correctly in the **Specify external COMSOL batch directory path** and **Specify external COMSOL installation directory path** settings according to how COMSOL is installed on the remote machine and the remote machines working directories.

You must also be able to access the remote machine without a password using the access method selected. This can be achieved by using something that does not require a password, for example, ssh or similar.

Choose the method for starting COMSOL remotely in **Remote invoke command**. You can choose **None**, **SSH**, or **User** defined. For the **SSH** setting you can choose between using **SSH**, **Putty**, or a **User** defined SSH command. You can select the SSH commands installation directory in the SSH directory if the SSH command is not available on the PATH. The SSH key file directory is set in the **SSH key file** setting. You can set ports you want to forward in **Forward ports** and the host you want to forward the ports to in **Port host**. This is useful when you have the license manager installed locally, but the machine where COMSOL is running cannot access the license manager; for instance, if the machine is in the cloud. Specify the user name to use on the remote machine with SSH user. For the **User** defined **Remote invoke command** you can enter a command in the **Command** setting. In the **Command** setting any use of the keyword `{remotehost}` is replaced by the name of the remote host when COMSOL starts.

Choose the method for transferring files to the remote computer in **File transfer command**. You can choose **None**, **SCP**, or **User** defined. For the **SCP** setting you can choose between using **SCP**, **Putty**, or a **User** defined SCP command. You can select the SCP commands installation directory in **SCP directory** if the SCP command is not available on the PATH. The SCP key file directory is set in the **SCP key file** setting. Specify the user name to use on the remote machine with SCP user. For the **User** defined **File transfer command** you can enter a command in the **To remote command** for the transfer of files from the local computer to the remote machine and **From remote command** for the transfer of files from the remote machine to the local computer. In the settings any use of the keyword `{remotehost}` is replaced by the name of the remote host. Any use of the keyword `{localfile}` is replaced by the name of the local file, and any use of the keyword `{remotefile}` is replaced by the name of the remote file.

Use the **Remote hosts** to list the host you want to run on. If several hosts are listed COMSOL allocates a job on the first host that is free. Use the **Remote OS** to specify if the remote computer is running the same OS (**Native**) or is running **Linux** or **Windows**.



Running COMSOL on the Amazon™ Cloud (Amazon EC2™)

CLUSTER SETTINGS

If you are running a parametric sweep and want to distribute it by sending one parameter value to each compute node, select the **Distribute parametric sweep** check box. This requires that your study includes a parametric sweep.

STUDY EXTENSIONS

Select the **Use graphics** check box when the batch process should run results nodes that creates graphical contents such as exporting to file.

Enter the **Number of simultaneous jobs**. The default is 1. This is the maximum number of batch processes that are allowed to run simultaneously.

Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete.

Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before it must inform that it is still running. Failure to do so means that the process is considered dead, and a new process is started unless the maximum number of job restarts is reached.

-
- Working with Remote Servers
 - Remote Computing Preferences
 - Cluster Computing Preferences

Cluster Sweep

Click the **Show** button () and select **Advanced Study Options** to enable this option. Use the **Cluster Sweep** study node () to find the solution to a sequence of stationary or time-dependent PDE problems that arise when you vary some parameters of interest. The cluster sweep can include multiple independent parameters directly for a

full multiparameter sweep (solve for the first value of the first parameter combined with all values of the second parameter, then the second value of the first parameter combined with all values of the second parameter, and so on, or use a specified combination of parameter values). You can also add more than one **Parametric Sweep** node to create nested parametric sweeps. The program then treats the parametric sweeps as a “nested for-loop” and indicates the nested structure using indentations of the **Parametric Sweep** nodes’ names. The **Cluster Sweep** is always the outermost sweep. It starts multiple COMSOL Multiphysics batch processes, one for each parameter set in the **Cluster Sweep** node, that solves the current study on your cluster given the parameter set by submitting COMSOL batch jobs to a job scheduler that in turn runs the batch job on a second computer or cluster. For more information, see [Cluster Computing \(Job Configuration\)](#) and its related functionality. If you are running in distributed mode interactively, it is usually not needed.



You cannot create another Batch study, Batch Sweep study, Cluster Sweep study, or Cluster Computing study in the same study and at the same time.

Right-click the **Study** node to add a **Cluster Sweep** study node () to run a batch jobs on a cluster. Once you have specified the settings, right-click the main **Study** node and select **Compute** in order to start a COMSOL Batch processes that solves the current study for the given parameter sets. Once the process is started the COMSOL Desktop® enters progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Detach Job** button in the **Progress** window. Once the processes are finished or you click the **Detach Job** button, an **External Process** node (), one for each process, is shown that represents the current running processes. Click the **Save as Default** button () in the settings window’s toolbar to save the current setting as default. If you are running COMSOL in distributed mode the model runs in the current process.

STUDY SETTINGS

The cluster sweep is a multiparameter sweep with its parameters solved as a distributed batch job; see [Parametric Sweep](#) for more information about parameter sweeps.

OUTPUT WHILE SOLVING



This section is not available if the study also contains a Parametric Sweep study step, which then contains these settings.

Use the **Probes** list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (✚) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.



No plots are generated automatically from the accumulated probe tables. When a full variation has been accumulated, then the **Format: Filled** is available for the table (see the **Table** settings window). This format makes it possible to make response surfaces directly from the **Results** view toolbar **Surface Plot** button.

BATCH SETTINGS



Except for the information below, see [Batch Settings](#) (for the [Cluster Computing \(Study\)](#) node) for settings information.

Before Sweep

Under **Before sweep**, the **Clear meshes** check box is selected by default to clear the meshes before running the batch sweep. The **Clear solutions** check box is selected by default to clear the solutions before running the batch sweep. Click to clear one or both of the check boxes as required.

During Sweep

Under **During sweep**, click to select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional analysis to be performed after the sweep has finished.



The setting is similar to the **All** and **Last** setting in the **Memory settings for jobs** for **Parametric Sweep**.

The **Synchronize accumulated probe table** check box is selected by default to synchronize the accumulated probes computed by the batch processes with the model. Click to clear the check box if required.

After Sweep

Under **After sweep**, select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases the solution synchronization and probe synchronization functionality should be used instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently.

REMOTE AND CLOUD ACCESS

See [Remote and Cloud Access](#) described for [Cluster Computing \(Study\)](#).

STUDY EXTENSIONS

See [Study Extensions](#) described for [Cluster Computing \(Study\)](#).

Time-Dependent with Initialization, Fixed Geometry

Use the **Time-Dependent with Initialization, Fixed Geometry** study () when you want to exclude geometry deformation effects from your model. The study is similar to the [Time-Dependent with Initialization](#) study, with the difference that the second time-dependent study step does not solve for the geometry deformation dependent variables. This study adds a [Current Distribution Initialization](#) study step and [Time-Dependent, Fixed Geometry](#) study step to the study node. See the study steps for settings information.

This study is available with the:

- Electrodeposition Module for the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces.
- Corrosion Module for the **Corrosion, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces.

Time-Dependent with Initialization

The **Time-Dependent with Initialization** study node () can be used to perform transient simulations of electrochemical cells. The study adds a **Current Distribution Initialization** study step and **Time Dependent** study step to the study node. The Current Initialization step solves for the electrode and electrolyte potentials as well as all global ODE dependent variables. All other dependent variables in the model, such as concentrations and electrode deformation, are set to the initial values in this step. The Time Dependent step performs a transient simulation for all dependent variables in the model, using the result of the first study step as initial values. See the study steps for settings information. This study is available with the:

- Electrodeposition Module for the **Electrodeposition, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces.
- Corrosion Module for the **Corrosion, Tertiary Nernst-Planck** and **Electrodeposition, Secondary** interfaces.

Time-Dependent, Fixed Geometry



The settings available for this study step are described for the **Time Dependent** node.

The **Time-Dependent, Fixed Geometry** study step is added to the **Time-Dependent with Initialization, Fixed Geometry** study. Use the **Time-Dependent, Fixed Geometry** study node () to exclude the deformation/ALE (X, Y, Z) variables from the variables that are actually solved for by the study step. This is a suitable study step if you want to simulate a time-dependent electrodeposition or corrosion problem for cases where the mesh deformation is expected to be small.

Current Distribution Initialization



The settings available for this study step are described for the [Stationary](#) node.

The **Current Distribution Initialization** study node () is added to the [Time-Dependent with Initialization](#), [Fixed Geometry](#) and [Time-Dependent with Initialization](#) studies. You can use this study step as an initialization step for the electric and electrolyte potentials in a transient simulation of an electrochemical cell. It computes a stationary solution for the electrolyte and electric potentials, based on the initial values of the model. The Current Distribution Initialization study step is typically followed by a time-dependent study step that solves for all dependent variables.

Stationary Plug Flow



The settings available for this study are described for the [Time Dependent](#) node.

The **Stationary Plug Flow** study is available with the [Reaction Engineering](#) interface, which requires the Chemical Reaction Engineering Module.

Use the **Stationary Plug Flow** study node () to solve the equations set up by the predefined Plug flow reactor type. The reactor equations describe the molar flow rates (SI unit: mol/s) as function of reactor volume (SI unit: m³) under stationary conditions. Selecting the Reaction Engineering interface and the Stationary Plug Flow study in the Model Wizard sets up a plug flow reactor type by default.

Frequency-Domain, Perturbation



The settings for the **Frequency-Domain, Perturbation** node () are the same as those for the [Frequency Domain](#) node.

Use a perturbed frequency domain analysis to study small oscillations about a bias solution (small-signal analysis).

With the **Frequency-Domain, Perturbation** study and study step ()¹⁰⁰, 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 type), 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.

This study (and study step) is available for a variety of interfaces and licenses:

- As a study step with the **AC Impedance Stationary** and **AC Impedance Time Dependent** studies, which require the Batteries & Fuel Cells Module or the Electrodeposition Module.
- As a study step for the **Small-Signal Analysis, Frequency Domain** study, which requires the AC/DC Module or MEMS Module.
- As a study and study step with the Thin-Film Flow and Lubrication Shell physics, which require the CFD Module or MEMS Module.
- As a study and study step with the Film Damping Shell and Thin-Film Gas Flow physics, which require the MEMS Module.

-
- 
 - [Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis](#)
 - [Physics Exclusive and Contributing Node Types](#)
-

Small-Signal Analysis, Frequency Domain



Use the **Small-Signal Analysis, Frequency Domain** study () for perturbed frequency domain studies of small oscillation about a bias solution. The study creates two study steps, where the first (a **Stationary** study node) solves for a stationary problem, and the second step (a **Frequency-Domain, Perturbation** study node) solves for a harmonic perturbation in the frequency domain of the stationary solution.

This study is available with:

- Electric Currents and Magnetic Fields, which both require the AC/DC Module or MEMS Module
- Electric Currents-Shell and Magnetic and Electric Fields, which require the AC/DC Module



Harmonic Perturbation—Exclusive and Contributing Nodes

Prestressed Frequency Analyses Study

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.

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.



For different plot settings made available, see **Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings**.

Prestressed Analysis, Eigenfrequency

Use the **Prestressed Analysis, Eigenfrequency** study () to compute eigenfrequencies which are influenced by a prior static preload. It is necessary to have a geometric

nonlinearity active in order to perform this type of analysis. It adds a [Stationary](#) study step (Step 1) and an [Eigenfrequency](#) study step (Step 2), which is performed on the prestressed structure, to the **Study** node. The settings are described for the study steps. To perform this study, no additional forces need to be added to the physics settings as only the modes and mode frequencies are returned.

This study is available for:

- Solid Mechanics using the Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module
- Shell, Plate, and Truss using the Structural Mechanics Module
- Electromechanics using the MEMS Module



Harmonic Perturbation—Exclusive and Contributing Nodes

Prestressed Analysis, Frequency Domain

Use the **Prestressed Analysis, Frequency Domain** study node () to compute the response to harmonic loads fluctuating around a static preload. It is necessary to have a geometric nonlinearity active in order to take the influence on the stiffness from the pre-load into account. This study type performs a [Stationary](#) study step (Step 1) followed by an [Frequency Domain](#) analysis study step (Step 2), which is performed on the prestressed structure. See the study steps for settings. 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, 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.

This study is available for the:

- Solid Mechanics using the Structural Mechanics Module, Geomechanics Module, MEMS Module, or Acoustics Module
 - Electromechanics using the MEMS Module
-



Harmonic Perturbation—Exclusive and Contributing Nodes

AC Impedance Stationary

The **AC Impedance Stationary** study () creates two study steps, where the first (a [Stationary](#) study node) solves for a stationary problem, and the second step (a [Frequency-Domain, Perturbation](#) study node) solves for a harmonic perturbation in the frequency domain of the stationary solution. See the study steps for settings.

This study is available with:

- Battery with Binary Electrolyte, Lead Acid Battery, and Lithium-Ion Battery, which require the [Batteries & Fuel Cells](#) Module.
 - Secondary Current Distribution and Tertiary Current Distribution, Nernst-Planck, which are available for the [Batteries & Fuel Cells](#) Module and the [Electrodeposition](#) Module.
-



Harmonic Perturbation—Exclusive and Contributing Nodes

AC Impedance Time Dependent

The **AC Impedance Time Dependent** study () creates two study steps, where the first (a [Time Dependent](#) study node) solves for a time-dependent problem, and the second step (a [Frequency-Domain, Perturbation](#) study node) solves for a harmonic perturbation in the frequency domain of the solution at the last time step. See the study steps for settings.

This study is available with the:

- Battery with Binary Electrolyte, Lead Acid Battery, and Lithium-Ion Battery which require the Batteries & Fuel Cells Module
- Secondary Current Distribution and Tertiary Current Distribution, Nernst-Planck, which are available for the Batteries & Fuel Cells Module and the Electrodeposition Module



Harmonic Perturbation—Exclusive and Contributing Nodes i

Linear Buckling

A linear buckling analysis includes the stiffening effects from stresses coming from nonlinear strain terms. The stiffness coming from stresses and material defines an eigenvalue problem where the eigenvalue is a load factor that, when multiplied with the actual load, gives the critical load in a linear context.

When you add the **Linear Buckling** study (), two study steps are added. The first is a [Stationary](#) study step, and then followed by a **Linear Buckling** step as the second step. In this step, an eigenvalue solver computes the buckling modes and the associated critical load factors.

Another way to calculate the critical load is to include large deformation effects and increase the load until the solver fails because the load has reached its critical value.

The **Linear Buckling** study is available for the Solid Mechanics physics using the Structural Mechanics Module or the MEMS Module. It is also available with Shell, Plate, and Truss physics when using the Structural Mechanics Module.

STUDY STEP SETTINGS

Use the **Desired number of buckling modes** field to specify the number of buckling modes you want the eigenvalue solver to return.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

Mode Analysis

A **Mode Analysis** study node () (and study step) computes the modes for an acoustics or electromagnetic wave using an eigenvalue solver. When this study is added, it adds a **Mode Analysis** study step under the Study node.

The **Mode Analysis** study is available with Pressure Acoustics, Frequency Domain, Aeroacoustics, Frequency Domain, and Thermoacoustics, Frequency Domain in 2D and 1D axial symmetry. It is also available with Boundary Mode Acoustics and Boundary Mode Aeroacoustics in 2D axial symmetry and 3D, all of which require the Acoustics Module.

It is also available for 2D models with the Electromagnetic Waves, Frequency Domain physics, which requires the RF Module.

STUDY SETTINGS

- Enter a **Desired number of modes**. The default is 6.
- Select a method to **Transform—Effective mode index** (the default), **Out-of-plane wave number**, or **None**.
- Enter a value or expression in the **Search for modes around** field. The default is 1.
- Enter a value or expression for the **Mode analysis frequency**. The default is 1 GHz.

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS

Select the **Adaptive mesh refinement** check box to use adaptive mesh refinement. Then from the **Adaption in geometry** list, select an available geometry to use for the adaption. Click the **Go to Source** button () to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh Refinement subnode, its settings, and the adaptive mesh refinement algorithm.

Modal Reduced Order Model

A **Modal Reduced Order Model** study node () (and study step) uses the Modal Solver and its matrix export functionality to produce a modal reduced-order model. It uses the transient equation form for the export. The matrices can be accessed through the COMSOL API or in table format from **Derived Values>System Matrix**.

STUDY SETTINGS

Select the **Include geometric nonlinearity** check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

Stationary, One-Way Coupled

The **Stationary, One-Way Coupled** study node () creates two study steps, where the first, a **Stationary, Fluid (Study Step)** solves for the fluid flow variables, and the second step, **Stationary, Solid (Study Step)** solves for the solid deformation. The settings available for the study steps are the same as for the **Stationary** node. When additional physics is included, it is by default added to both study steps.

This study is available with the Fluid-Structure Interaction interface, which requires the MEMS Module or the Structural Mechanics Module.

Stationary, Fluid (Study Step)

The **Stationary, Fluid** study node () is added to the **Stationary, One-Way Coupled** or **Stationary, One-Way Coupled with Initialization** study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the **Stationary** node.

Stationary, Solid (Study Step)

The **Stationary, Solid** study node () is added to the **Stationary, One-Way Coupled** **Stationary, One-Way Coupled with Initialization** study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the **Stationary** node.

Stationary, One-Way Coupled with Initialization

The **Stationary, One-Way Coupled with Initialization** study () is for stationary, one-way coupled Fluid-Structure Interaction models using a turbulence model that requires the distance to the closest wall. The study creates three study steps:

- The first, **Wall Distance Initialization** solves for the distance to the closest wall.
- The second, a **Stationary, Fluid (Study Step)** solves for the fluid flow variables.
- The third step, **Stationary, Solid (Study Step)** solves for the solid deformation.

The settings available for the study steps are the same as for the **Stationary** node. When additional physics are included, it is by default added to the last two study steps.

The **Stationary, One-Way Coupled with Initialization** study is available with the Fluid-Structure Interaction physics, which requires the MEMS Module or the Structural Mechanics Module, plus the CFD Module.

HOW TO ADD THIS STUDY

- 1 Add a **Fluid-Structure Interaction** physics interface to the model.
- 2 On the **Fluid-Structure Interaction** settings window under **Physical Model**, select **RANS** as the **Turbulence model type**.
- 3 Select **Low Reynolds number k-ε** or **Spalart-Allmaras** as the **Turbulence model**.
- 4 In the **Model Builder**, right-click the root node and select **Add study**.
- 5 In the **Model Wizard** under **Preset Studies**, select the **Stationary, One-Way Coupled with Initialization** study.

Time Dependent, One-Way Coupled

The **Time Dependent, One-Way Coupled** study () creates two study steps, where the first, a **Time Dependent, Fluid (Study Step)** solves for the fluid flow variables, and the second step, **Time Dependent, Solid (Study Step)** solves for the solid deformation. The settings available for the study steps are the same as for the **Time Dependent** node. When additional physics is included, it is by default added to both study steps.

The **Time Dependent, One-Way Coupled** study is available with the Fluid-Structure Interaction interface, which requires the MEMS Module or the Structural Mechanics Module.

Time Dependent, Fluid (Study Step)

The **Time Dependent, Fluid** study step () is added to the **Time Dependent, One-Way Coupled** or **Transient, One-Way Coupled with Initialization** study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the **Time Dependent** node.

Time Dependent, Solid (Study Step)

The **Time Dependent, Solid** study step () is added to the **Time Dependent, One-Way Coupled** or **Transient, One-Way Coupled with Initialization** study, which is available for the Fluid Structure Interaction interface. The settings available for this node are the same as for the **Time Dependent** node.

Transient, One-Way Coupled with Initialization

The **Transient, One-Way Coupled with Initialization** study () is for time dependent, one-way coupled Fluid-Structure Interaction models using a turbulence model that requires the distance to the closest wall. The study node creates three study steps:

- The first, [Wall Distance Initialization](#) solves for the distance to the closest wall.
- The second, [Time Dependent, Fluid \(Study Step\)](#) solves for the fluid flow variables.
- The third, [Time Dependent, Solid \(Study Step\)](#) solves for the solid deformation.

The settings available for the study steps are the same as for the [Time Dependent](#) node. When additional physics is included, it is by default added to the last two study steps.

The **Transient, One-Way Coupled with Initialization** study is available with the Fluid-Structure Interaction physics, which requires the MEMS Module or the Structural Mechanics Module, plus the CFD Module.

HOW TO ADD THIS STUDY

- 1 Add a **Fluid-Structure Interaction** physics interface to the model.
- 2 On the **Fluid-Structure Interaction** settings window under **Physical Model**, select **RANS** as the **Turbulence model type**.
- 3 Select **Low Reynolds number k- ϵ** or **Spalart-Allmaras** as the **Turbulence model**.
- 4 In the **Model Builder**, right-click the root node and select **Add study**.
- 5 In the **Model Wizard** under **Preset Studies**, select the **Transient, One-Way Coupled with Initialization** study.

Stationary with Initialization

The **Stationary with Initialization** study () is for stationary two-phase flow models that require an initialization of a level set function or phase field function. It adds a [Wall Distance Initialization](#) study step followed by a [Stationary](#) study step to the Study node. The first step, the **Wall Distance Initialization** study step, is dedicated to solving for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface. The second step is an ordinary **Stationary** study step, but it excludes the reciprocal wall distance except for turbulent flow where, if the physics is put on a moving frame, the reciprocal wall distance is solved for in the stationary step as well.

This study is available with these physics user interfaces, which require a CFD Module or Heat Transfer Module:

- All types of Turbulent Flow, Low Re k - ϵ physics (Conjugate Heat Transfer and Non-Isothermal Flow).
- With the CFD Module, these physics user interfaces also have this study available: All types of Turbulent Flow, Spalart-Allmaras physics (Conjugate Heat Transfer, Non-Isothermal Flow, and High Mach Number Flow).

Transient with Initialization

The **Transient with Initialization** study () is for time-dependent two-phase flow models that require an initialization of a level set function or phase field function. It adds a [Wall Distance Initialization](#) study step followed by a [Time Dependent](#) study step to the Study node. The first step, the **Wall Distance Initialization** study step, is dedicated to solving for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface. The second step is an ordinary **Time Dependent** study step.

This study is available with these physics user interfaces, which require a CFD Module or Microfluidics Module:

- Laminar Two-Phase Flow and Turbulent Two-Phase Flow, Level Set
- Laminar Two-Phase Flow and Turbulent Two-Phase Flow, Phase Field
- Mathematics>Moving Interface>Level Set and Phase Field
- With the CFD Module or Heat Transfer Module, these interfaces also have this study available: all types of Turbulent Flow, Low Re k - ϵ physics (Conjugate Heat Transfer and Non-Isothermal Flow).
- With the CFD Module, these physics user interfaces also have this study available: all types of Turbulent Flow, Spalart-Allmaras physics (Conjugate Heat Transfer, Non-Isothermal Flow, and High Mach Number Flow).

Wall Distance Initialization

See the [Stationary with Initialization](#) and [Transient with Initialization](#) studies for availability by module.



The settings for this study are the same as for the [Stationary](#) and [Time Dependent](#) studies.

The **Wall Distance Initialization** study step is added to the [Stationary with Initialization](#) and [Transient with Initialization](#) studies. This first step is dedicated to solving for the reciprocal wall distance. In the case of turbulence models, this is the distance to the closest wall. In the case of two-phase flow it is the distance to the phase interface.

Boundary Mode Analysis

A **Boundary Mode Analysis** study node () 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. When selected, it adds a **Boundary Mode Analysis** study step followed by a [Frequency Domain](#) study step to the Study node.

This study is available with the Electromagnetic Waves or Microwave Heating interfaces, which both require the RF Module.

STUDY SETTINGS

- Enter a **Desired number of modes**. The default is 1.
- Select a method to **Transform—Effective mode index** (the default), **Out-of-plane wave number**, or **None**.
- Enter a value or expression in the **Search for modes around** field. The default is 1.
- Enter a **Port name**. The default is 1.
- Enter a value or expression for the **Mode analysis frequency**. The default is 1GHz.

If you have a license for any of the Acoustics Module, MEMS Module, or Structural Mechanics Module and your model involves structural mechanics, then the **Study Settings** section includes an **Include geometric nonlinearity** check box. Select this check box to enable a geometrically nonlinear analysis for the study step. Some physics force a geometrically nonlinear analysis, in which case it is not possible to clear the **Include geometric nonlinearity** check box. For further details, see the theory sections for the respective physics in the applicable modules' user guides.

PHYSICS AND VARIABLES SELECTION

See [Selecting Physics and Variables in the Study Steps](#) for details.

VALUES OF DEPENDENT VARIABLES

See [Specifying Values of Dependent Variables](#) for details.

MESH SELECTION

Select a mesh to use for each geometry in the study. First select the geometry from the **Geometries** list and then select the mesh from the **Mesh** list.

STUDY EXTENSIONS

Select the **Adaptive mesh refinement** check box if you want to use adaptive mesh refinement. Select the geometry to use for the adaption from the **Adaption in geometry** list. Click the **Go to Source** button () to move to the settings window for the geometry. See [Adaptive Mesh Refinement](#) for information about the Adaptive Mesh Refinement subnode, its settings, and the adaptive mesh refinement algorithm.



If you have the RF Module, see the model “Polarized Circular Ports” found in the Model Library by following the path **RF_Module/Tutorial_Models/polarized_circular_ports**.

Frequency-Stationary



See [Stationary](#) for all the settings.

The **Frequency-Stationary** study is available with the following physics:

- Induction Heating, which requires the AC/DC Module.
- Microwave Heating, which requires the RF Module.



See [Time Dependent](#) for all the settings.

The **Frequency-Transient** study is available with the following physics:

- Induction Heating, which requires the AC/DC Module.
- Microwave Heating, which requires the RF Module.
- Inductively Coupled Plasma, which requires the AC/DC Module and the Plasma Module.
- Microwave Plasma, which requires the RF Module and the Plasma Module.

Physics that support this study compute electromagnetic fields in the frequency domain and the temperature (electron temperature in the case of Inductively Coupled Plasma and Microwave Plasma) in the time domain.

Only use this study when the power transfer from the fields to any susceptible variables occurs at twice the angular frequency set by the study. In a large number of cases, the thermal time constant of an object of interest is substantially greater than the angular frequency of the electromagnetic radiation. In order to solve the problem in the time domain, tens or hundreds of thousands of RF cycles need to be computed by the solver before the problem evolves to the periodic steady-state solution. By solving for the fields in the frequency domain, the change in the fields over a single RF cycle does not need to be resolved and thus the periodic steady state solution is reached much more rapidly. This means that the transient, thermal response of an object is computed by this study, but any (small) fluctuations in temperature over any given RF cycle are not.

Mean Energies

Use the **Mean Energies** study () to enter an array of values for the mean electron energy. COMSOL Multiphysics computes the electron energy distribution function (EEDF) so that the mean electron energy is equal to the mean energy requested. This study conveniently allows data such as rate coefficients, Townsend coefficients, and electron transport properties to be computed as a function of the mean electron energy.

This study is available with the Boltzmann Equation, Two-Term Approximation interface, which requires the Plasma Module.

Reduced Electric Fields

Use the **Reduced Electric Fields** study () to sweep through a range of reduced electric fields. The study type is used with the Boltzmann Equation, Two-Term Approximation physics to compute electron transport properties and electron impact rate coefficients for a given reduced electric field. Specifying a range of values for the reduced electric field allows for these properties to be tabulated. This tabulated data can then potentially be used in a space-dependent plasma simulation. The reduced electric field is defined as the electric field divided by the background gas number density.

This study is available with the Boltzmann Equation, Two-Term Approximation physics, which requires the Plasma Module.

Particle Trajectories (Study)

The **Particle Trajectories** study () and study step have the same settings as the **Time Dependent** study step except that by default, only the Particle Tracing Module's physics user interfaces for particle tracing are active in the physics selection.

Coil Current Calculation



For information about the use of this study and its functionality, see the *AC/DC Module User's Guide*.

The **Coil Current Calculation** study () and study step is available for 3D models. The study is available with the Magnetic Fields interface and the Multi-Turn Coil Domain node (requires the AC/DC Module). Use this study to solve an eigenvalue problem for the current flow in a Multi-Turn Coil Domain node that gives the current density likely produced by a bundle of conductive wires. The best results are obtained when the coil has a constant cross section without sharp bends and bottlenecks. The eigenvalue problem is loosely based on the equation for incompressible fluid flow, with some modifications, and is solved by the specialized **Coil Current Calculation** study step.

Add an Automatic Current Calculation subnode to the Multi-Turn Coil Domain to set up the automatic computation of the current flow in the coil domain. The

boundary conditions for the current calculation study are specified using the Electric Insulation, Input, and Output subnodes available with the node.

Cyclic Voltammetry



The settings are described for the [Time Dependent](#) node.

Use the **Cyclic Voltammetry** study () together with the with the Electroanalysis user interface, which requires one of these modules: Batteries & Fuel Cells, Corrosion, Electrochemistry, or Electrodeposition. The study performs transient simulations of voltammetry experiments. When this study is added in the Model Wizard, a **Cyclic Voltammetry** study step is added to the Model Builder.

The **Cyclic Voltammetry** study step sets up a time-dependent solver. The initial and maximum time step solver settings are based on the settings in the Electroanalysis user interface, and a [Stop Condition](#) is added to the solver to stop the simulation at the end of the voltammetry cycling.



If the Electroanalysis physics does not contain any Electrode Surface node with active Cyclic Voltammetry settings, no Stop Conditions are added to the solver. Voltammetry simulations can also be performed using a Time Dependent study step.

Sensitivity (Study Step)

Use the **Sensitivity** study node () to add sensitivity analysis to a study. Use the Sensitivity study step to make a sensitivity analysis for an optional standard study step. Compared to the Sensitivity user interface, the Sensitivity study step provides the following capabilities:

- Model parameters can act as global control variables

- Sensitivity functions can be added at the study level
- Sensitivity functions and variables from physics user interfaces can be used on the study level.



It is only possible to use one of the Sensitivity, Optimization, or Parametric Sweep study step in any study.

The settings window for the **Sensitivity** node contains the following sections:

SENSITIVITY METHOD

Choose a method from the **Gradient method** list: one of the analytical methods **Forward** or **Adjoint** (the default). See [Choosing a Sensitivity Method](#) for information about the forward and adjoint methods. These methods have similar limitations as the gradient-based optimization methods (SNOPT and Levenberg-Marquardt). For example, nonanalytic functions are not correctly treated. Also, when model parameters are used in the geometry or mesh, the sensitivity is not correctly computed.

You choose the study step to use from the **Study step** list. The supported study step types are Stationary, Time Dependent, and Frequency Domain.

OBJECTIVE FUNCTION

You specify the objective function for the optimization problem in the table's **Expression** column. Enter any globally available expression that evaluates to a real number. Optionally, you can add a description in the **Description** column. Click the **Add Expression** () and **Replace Expression** () buttons to search through a list of predefined expressions.



For a sensitivity objective that is expressed in terms of the solution u of a PDE (or in terms of control variables), [Integration](#) is one example of how you can define a scalar objective as required by the sensitivity solver. The evaluation of the objective function is similar to [Global Variable Probe](#), so any variable that can be represented by a global variable probe is suited as an objective.

Multiple objectives

If you have defined more than one objective function, choose how to evaluate the overall objective: For sensitivity studies, only **Sum of objectives** is available.

Solution

Here you select the evaluation method of the objective function when several solutions are present, like for Time Dependent studies. For sensitivity studies, only **Auto** is available. The solver chooses the evaluation method based on the innermost study. For studies in the **Frequency Domain**, the contributions from all solutions are summed (equivalent to the **Sum of objectives** option). For a **Time Dependent** study, the optimization solver selects the last solution (final time).

CONTROL VARIABLES AND PARAMETERS

The table under **Control Variables and Parameters** is used to define control variables. In this table you can select all parameters defined in the **Global Definitions>Parameters** node's settings window through the **Add** () button.

From a list in the **Parameter name** column, select the parameter to define as a control variable.

Move control parameter rows up and down using the **Move Up** () and **Move Down** () buttons. To remove a control parameter, select some part of that parameter's row in the table and click the **Delete** button (). You can also save the definitions of the control parameters 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 control variables, use the **Load from File** button () and using the **Load from File** dialog box that appears. Data must be separated by spaces or tabs.



The tables to activate or deactivate objective functions and control variables in the model are only visible if those functions or variables are present in the model.



If you have the LiveLink™ for Excel®, you can also save and load control variables to and from Microsoft Excel Workbook (*.xlsx) files.

Optimization (Study Step)

The Optimization study step () specifies an optimization problem and controls the optimization solvers provided by the Optimization Module.



For information about the use of this study and its functionality, see the *Optimization Module User's Guide*.

Frozen Rotor



For information about the use of this study and its functionality, see the *CFD Module User's Guide*.

Use the **Frozen Rotor** study node () with the Rotating Machinery, Laminar Flow and Rotating Machinery, Turbulent Flow interfaces, which require the CFD Module. This is a special study type for rotating machinery and is only supported by such interfaces. The frozen rotor approach assumes that the flow in the rotating domain, expressed in the rotating coordinate system, is fully developed.



See [Stationary](#) for all the settings.

Solver Configurations

The Solver Configurations Node

The **Solver Configurations** node () contains all solver configurations defined for a study. The node's context menu has these options. Select:

- **Show Default Solver** () to show the solver nodes that correspond to the study step.
- **Clear Solutions** () to remove any solutions in the solver.
- **Delete Solvers** () to delete all solvers under the **Solver Configurations** node.
- From the main menu, click the **Show** button () and select **Advanced Study Options** and then right-click the **Solver Configurations** node to add a **Create Custom Solver** () to add a **Solver** node without any added solver settings or other nodes.



[Clearing, Deleting, Disabling, and Enabling a Solver or Solutions](#)

Show Default Solver

To display the solver that corresponds to the study steps in a study and the current physics settings, right-click the main **Study** node () and select **Show Default Solver** () from the context menu. Or right-click the **Solver Configurations** node and select **Show Default Solver**.

Adding Solver Nodes

To add solver nodes to a solver configuration, right-click a **Solver** node and then select a solver from the **Solvers** submenu.

Running a Solver Configuration

Running a *solver configuration* is tantamount to computing a solution. Like mesh nodes, solution nodes are not built automatically as they are added. To compute a solution, right-click the corresponding solver node and select **Compute**. Or right-click a study node and select **Compute** to compute the enabled solver configuration with a

green border around its icon, if such a solver configuration exist. If no solver configuration exists or if all sequences are disabled, a new solver configuration is generated and computed. If you have already generated a solver configuration for the study, you can compute some or all subnodes in it by selecting **Compute to Selected** (), **Compute from Selected** (), or **Compute** (). For example, right-click a **Dependent Variables** node and select **Compute to Selected** () to evaluate the initial values for the dependent variables (similar to the **Get Initial Value** and **Get Initial Value for Step** options for the main **Study** nodes and the study steps). **Compute** () runs the entire solver configuration.



Computing the Initial Values

Errors and Warnings

Problems encountered when running a solver or generating a mesh are treated in two different ways depending on if it is possible to avoid the problem and continue the operation or if the operation must be stopped. In the first case, a **Warning** node () appears under the node in the model tree that caused the problem. In the second case, an **Error** node () appears under the node in the model tree that caused the problem. For the solvers, and for multiple meshing warnings and errors, you find the **Warning** and **Error** nodes under the **Information** node ().

Solver Overview

Solver Configurations

You solve a model by computing a *solver configuration*, which is a scheme for computing a solution. A **Solver Configuration** node displays if it has content or if you have selected click the **Show** button () and select **Advanced Study Options** and added this node to the **Model Builder**. You can also solve a model by computing a study, which defines a sequence of solver configurations and, in some cases, a sequence of job configurations.

Loosely speaking a solver configuration consists of one or more **Solver** nodes (). Each Solver node consists of a sequence of subnodes specifying how to compute the solution. Typically, such a solver configuration contains information about which physics and geometry to use, which variables to solve for, and which solvers to use for the type of study to perform.

Solver Operation, Attribute, and Utility Nodes

There are three types of nodes that appear as subnodes to a Solver node: *operation node*, *attribute node*, and *utility nodes*:

- Operation nodes (typically solvers) produce solutions as output. In particular, the output from the last executed operation node is available for results analysis and visualization.
- Attribute nodes hold properties that control the behavior of operation features.
- Utility nodes handle special types of operations.

Some of the settings in subnodes are synchronized with the corresponding Study setting. They are grayed out by default and can only be controlled from the subnode by changing the **Defined by study step** setting to **User defined**.

-
- [Solution Operation Nodes and Solvers](#)
 - [Solution Attribute Nodes](#)
 - [Solution Utility Nodes](#)
-



AVAILABLE SOLUTION OPERATION NODES

Table 19-2 lists the available Solution Operation Nodes and Solvers.

TABLE 19-2: SOLUTION OPERATION NODES

NAME	ICON	PURPOSE
AWE Solver		Solve a parametric problem with asymptotic waveform evaluation.
Eigenvalue Solver		Solve an eigenvalue problem.
Modal Solver		Solve time-dependent or parametric problem with modal analysis.
Optimization Solver		Optimization parameters (requires the Optimization Module).
Plug Flow Solver		Solve a plug flow reactor model (requires the Chemical Reaction Engineering Module).
Stationary Solver		Solve a stationary problem.
Time-Dependent Solver		Solve a time-dependent problem using an implicit time stepping scheme.
Time Discrete Solver		Solve a time-dependent problem that has been discretized in time.
Time Explicit Solver		Solve a time-dependent problem using an explicit time-stepping scheme.
Dependent Variables		Handles the dependent variables solved for (initial values, scaling) and dependent variables not solved for (prescribed values).

You find these solvers on the **Solvers** menu when right-clicking a **Solver** node.

AVAILABLE SOLUTION ATTRIBUTE NODES

Table 19-3 lists the available Solution Attribute Nodes.

TABLE 19-3: SOLUTION ATTRIBUTE NODES

NAME	ICON	PURPOSE
Adaptive Mesh Refinement		Adaptive mesh refinement parameters.
Advanced		Advanced general solver parameters.
AMS		Handles settings for the auxiliary space Maxwell solver (AMS) preconditioner.

TABLE 19-3: SOLUTION ATTRIBUTE NODES

NAME	ICON	PURPOSE
Automatic Remeshing		Automatic remeshing parameters.
Coarse Solver		Handles settings for the coarse solver (when using Multigrid or Domain Decomposition).
Control Field		Handles settings for field variables that are acting as control variables. Control variables have a special status when using the Sensitivity or the Optimization solver. Used together with the Dependent Variables operation node.
Control State		Handles settings for state variables that are acting as control variables. Control (state) variables have a special status when using the Sensitivity or the Optimization solver. Used together with the Dependent Variables operation node.
Direct		Handles settings for a direct linear solver.
Domain Decomposition		Handles settings for the domain decomposition solver.
Domain Solver		Handles settings for the domain solver (when using Domain Decomposition).
Field		Handles settings for field variables. Each field variable needs a separate Field node. This attribute is used with the Dependent Variables operation node.
Fully Coupled		Handles parameters for a fully coupled solution approach.
Incomplete LU		Handles settings for an incomplete LU solver or preconditioner.
Iterative		Handles settings for an iterative linear solver or preconditioner.
Jacobi		Handles settings for a Jacobi linear solver or preconditioner.
Krylov Preconditioner		Handles settings for a Krylov-type linear solver or preconditioner.
Lower Limit		Parameters for imposing restrictions on degrees of freedom.

TABLE 19-3: SOLUTION ATTRIBUTE NODES

NAME	ICON	PURPOSE
Lumped Step		Available when using the segregated attribute node. This step is intended for speeding up the computation of any L ₂ -projections, stemming from the identity operator, appearing as single physics within a multiphysics problem.
Multigrid		Handles settings for a multigrid linear solver or preconditioner.
Parametric		Handles settings for parameter stepping. This attribute can be used together with the Stationary Solver.
Postsmoother		Handles settings for the postsmoother (when using Multigrid).
Presmoother		Handles settings for the presmoother (when using Multigrid).
Previous Solution		Previous parametric solution parameters.
SCGS		Handles settings for the SCGS (symmetrically coupled Gauss-Seidel) solver or preconditioner.
Segregated		Handles parameters for a segregated solution approach.
Segregated Step		Handles settings for a segregated step (when using Segregated).
Sensitivity		Sensitivity parameters.
SOR		Handles settings for an SOR-type linear solver or preconditioner.
SOR Gauge		Handles settings for an SOR Gauge-type linear solver or preconditioner.
SOR Line		Handles settings for an SOR Line linear solver or preconditioner.
SOR Vector		Handles settings for an SOR Vector-type linear solver or preconditioner.
State		Handles settings for state variables. A state is composed of a set of ODE variables. Used together with the Dependent Variables operation node.
Stationary Acceleration		Accelerates the solution process for nonlinear problems with a time-periodic stationary solution.
Stop Condition		Stop condition parameters.

TABLE 19-3: SOLUTION ATTRIBUTE NODES

NAME	ICON	PURPOSE
Time Parametric		Handles settings for parameter stepping. This attribute can be used together with the Time-Dependent Solver.
Vanka		Handles settings for a Vanka linear solver or preconditioner.

Applicable solution attribute nodes are available by right-clicking a solution operation node such as a **Time-Dependent Solver** node.

AVAILABLE SOLUTION UTILITY NODES

Table 19-4 lists the available **Solution Utility Nodes**:

TABLE 19-4: SOLUTION UTILITY NODES

NAME	ICON	PURPOSE
Adaptive Mesh Refinement		Stores the solution on the finest mesh from a mesh refinement procedure.
Assemble		Provides a way of accessing assembled matrices and vectors for further work in Java®.
Compile Equations		Compiles equations by specifying which study and study step to use and specifying representation of complex variables.
Store Solution		Stores the solution at this point of the solver configuration also after the solver configuration has been computed.
State Space		Provides a way of accessing state-space matrices for further work in Java.

Applicable solution utility nodes are available on the **Other** menu when right-clicking a **Solver** node.

Solution Operation Nodes and Solvers

The following sections describe the solver operation nodes and their settings in detail.

- Selecting a Stationary, Time-Dependent, or Eigenvalue Solver
- AWE Solver
- Eigenvalue Solver
- Modal Solver
- Optimization Solver
- Plug Flow Solver
- Stationary Solver
- Time-Dependent Solver
- Time Discrete Solver
- Time Explicit Solver
- Dependent Variables
- References for the Solution Operation Nodes and Solvers

Selecting a Stationary, Time-Dependent, or 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

$$e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

the stationary solver searches for a solution where $\partial u / \partial t = 0$. The time-dependent solver and the stationary solver 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:

$$\nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + \alpha u = \lambda d_a u - \lambda^2 e_a u$$

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.

AWE Solver

Use the **AWE Solver** () to perform fast-frequency parameter sweeps using asymptotic waveform evaluation (AWE).



This is an alternative way to perform parameter stepping to the one you get by using the **Stationary** solver node in conjunction with the **Parametric** subnode.

GENERAL

Use the **Parameter name** field to specify a parameter name. The use of several parameter names is not supported.

Use the **Parameter values** field to enter a vector of parameter values that define the parameter value span for the simulation. Exactly how the vector of parameter values is used by the solver is determined by the option **Parameters to store** in the **Output** section as described below.

An alternative to specifying parameter values directly in the **Parameters values** field is to specify them in a text file. You can use the **Load parameter values** field and the **Browse** button to specify such a text file. Use the **Read File** button to read the specified file. The read values appear in the **Parameters values** field.



Loading values from a file overwrites any values already present in this field. The format of the text files must be such that the parameter values appear one per row.

Use the **Expressions** field to specify a space-separated list of globally available scalar-valued expressions to be used for error estimation by the AWE algorithm.

TOLERANCES

In the AWE algorithm, the values of the expressions specified in the **Expressions** field in the **General** section are evaluated at one or more points of a parameter interval using certain expansions. The AWE algorithm is considered to have converged in that interval if the functional values resulting from the different expansions and evaluation points are similar enough. Use the:

- **Relative tolerance** field to specify to what relative tolerance the functional values must agree at the evaluation points.
- **Absolute tolerance** field to specify to what absolute tolerance the functional values must agree at the evaluation points.

EXPANSION SETTINGS

Use the **Evaluation points** field to specify a scalar or vector of values where the expressions defined by the **Expressions** field in the **General** section are to be evaluated. The evaluation points must be specified as a number between 0 and 1 because they are interpreted as being relative to the parameter interval under consideration. Entering a scalar value of 0.5 means that the expressions are evaluated at the midpoint of each interval. Use the:

- **Expansion size** list to specify the number of terms to include when performing Taylor expansions of the solution.
- **Expansion type** list to specify which expansion type to use when evaluating the solution at the different evaluation points:
 - Select **Padé** to compute Padé expansions based on the Taylor expansions. The Padé expansions are then used when evaluating the solution.
 - Select **Taylor** to use the Taylor expansion itself when evaluating the solution.

VALUES OF LINEARIZATION POINT

The problem solved by the AWE solver is assumed to be a linearization about a solution. You can specify such a solution (a linearization point) using the **Prescribed by** list. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point.

Use the **Solution** list to specify which solution to use if **Prescribed by** has been set to **Solution**. Select:

- **Zero** to use a linearization point that is identically equal to zero.
- Any other available solution to use it as linearization point. It can be the current solution in the sequence or a solution from another sequence or a solution that was stored with the **Store Solution** node. You select a stored solution by changing **Use** to the name of the stored solution.

To store the used linearization point in the output, select the **Store linearization point and deviation in output** check box.

OUTPUT

Use the **Parameters to store** list to control at what parameter values the solver stores a solution. Select:

- **Steps given** to store solutions at the parameter values entered in the **Parameter values** field in the **General** section.
- **Steps taken by solver** to store solutions at the parameter values where the AWE algorithm has performed an expansion.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

ADVANCED

By default the solver allows shorter intervals in the AWE algorithm than the relative tolerance (from the **Relative tolerance** field in the **Tolerances** section) times the length of the interval defined by the values in the **Parameter values** field in the **General section**. But if shorter intervals are detected, these intervals are not bisected and a warning is printed in the log. To modify the shortest allowed interval, select the **Minimal interval** check box and enter a limit for the interval length.

The **Accept short intervals** check box can be used to control how the solver handles intervals that are found to be too short. If this check box is cleared, the solver stops with an error if a too short interval is found. When the check box is selected the solver silently accepts short intervals.

Use the **Assembly strategy** list to control how the solver assembles quantities needed to compute a Taylor expansion. Select:

- **All** to assemble all quantities at once. This option is faster than **One**.
- **One** to assemble one quantity at a time. This option requires less memory than **All**.

Eigenvalue Solver

Use the **Eigenvalue Solver** () to find the solution to linear or linearized eigenvalue problems (also called eigenfrequency problems).

THE EIGENVALUE SOLVER ALGORITHM

Finite element discretization leads to the generalized eigenvalue system

$$\begin{aligned}(\lambda - \lambda_0)^2 EU - (\lambda - \lambda_0)DU + KU + N_F\Lambda &= 0 \\ NU &= 0\end{aligned}$$

where the solver evaluates E , D , K , N , and N_F for the solution vector U_0 ; λ denotes the eigenvalue; and λ_0 is the linearization point. If $E = 0$, it is a linear eigenvalue problem; if E is nonzero, it is a quadratic eigenvalue problem. To solve the quadratic eigenvalue problem, COMSOL Multiphysics reformulates it as a linear eigenvalue problem. After constraint handling, it is possible to write the system in the form $Ax = \lambda Bx$.

More general eigenvalue problems sometimes arise when boundary conditions or material properties are nonlinear functions of the eigenvalue. These cases can be handled as a series of quadratic eigenvalue problems. COMSOL treats general dependences on the eigenvalue by assembling a quadratic approximation around the eigenvalue linearization point λ_0 . Normally, iteratively updating the linearization point leads to rapid convergence.

Finding the eigenvalues closest to the *shift* σ is equivalent to computing the largest eigenvalues of the matrix $C = (A - \sigma B)^{-1}B$. To do this, the solver uses the ARPACK FORTRAN routines for large-scale eigenvalue problems (Ref. 8). This code is based on a variant of the Arnoldi algorithm called the *implicitly restarted Arnoldi method* (IRAM). The ARPACK routines must perform several matrix-vector multiplications Cv , which they accomplish by solving the linear system $(A - \sigma B)x = Bv$ using one of the linear system solvers.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to specify the properties below (in addition to the relative tolerance, which is always available).

Use the **Desired number of eigenvalues** field to specify the number of eigenvalues and eigenvectors (default: 6) that the solver should compute and store in the output.

The number in the **Relative tolerance** field (default 1.0×10^{-6}) controls the relative error in the computed eigenvalues.

The **Eigenvalue** transformation list you can select a transformation method for transforming the eigenvalues into another related quantity. You can always select **None** (the default, which keeps the original eigenvalues), but depending on the physics in the model, other transformations might also be available.

Use the **Search for eigenvalues around** field to search for eigenvalues close to the specified real or complex scalar. This value is often called the shift. Select the **Transform value** check box to transform the search value using the selected eigenvalue transformation. The value of the eigenvalue is set in the **Value** field.

VALUES OF LINEARIZATION POINT

Both for linear and nonlinear PDE problems the eigenvalue problem is that of the linearization about a solution. Such a solution is specified with the **Prescribed by** list. Select:

- **Initial expression** (the default) to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point.

Use the **Solution** list to specify which solution to use if **Prescribed by** is set to **Solution**:

- Select **Zero** (the default) to use a linearization point that is identically equal to zero.
- Select any other available solution to use it as linearization point.

To store the linearization point and the deviation from that linearization (instead of the total solution), select the **Store linearization point and deviation in output** check box.

If the eigenvalue itself appears nonlinearly, the solver reduces the problem to a quadratic approximation around an eigenvalue linearization point. Use the settings under **Value of eigenvalue linearization point** to specify such a scalar. Select the **Transform point** check box to transform the linearization point value using the selected eigenvalue transformation. You specify the value of the linearization point in the **Point** field (default value: 0).

OUTPUT

Use the **Scaling of eigenvectors** list to specify the scaling method used to normalize the eigenvectors. Select:

- **RMS** to use root mean square (RMS) normalization.

- **Max** to use maximum norm normalization.
- **Mass matrix** to scale the eigenvectors such that the modal masses become unity. If this scaling method is used the **Participation factor field** list displays. Select the field for which to compute the *mass participation factors* (typically a displacement field such as `mod1_u`).

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

ADVANCED

The eigenvalue solver is an iterative algorithm. Use the **Maximum number of eigenvalue iterations** field to limit the number of iterations (default: 300).

Use the **Dimension of Krylov space** field to control the algorithm's memory use. The default value of 0 means that the solver sets the dimension automatically to approximately twice the number specified in the **Desired number of eigenvalues** field in the **General** section.

LOG

The **Log** section contains logs of the eigenvalue solver results and properties of the assembled system, including the solver iterations and the total solution time. This log is stored in the Model MPH-file.

Modal Solver

Use the **Modal Solver** () to perform either parameter stepping (also called frequency response) or time stepping (also called transient response) using a reduced model. The model reduction uses precomputed eigenvalues and eigenvectors. For details about the algorithm, see [The Modal Solver Algorithm](#) below.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step, or select **User defined** to specify all settings locally.

Use the **Study type** list to select the basic study type. Select:

- **Frequency domain** to perform parameter stepping using a reduced model.
- **Time dependent** to perform time stepping using a reduced model.

Further Options for Frequency Domain

Use the **Parameter values** field to enter a vector of parameter values that define the parameter value span for the frequency-domain simulation. Click the **Range** button () to define a range of parameter values using the **Range** dialog box.

Exactly how the vector of parameter values is used by the solver is determined by the option **Parameter list type**.

An alternative to specifying parameter values directly in the **Parameters values** field is to specify them in a text file. You can use the **Load parameter values** field and the **Browse** button to specify such a text file. Click the **Read File** button to read the specified file. The read values appear in the **Parameters values** field.



Reading parameter values from a file overwrites any values already present in that field. The format of the text files must be such that the parameter values appear one per row.

Use the **Parameter list type** list to control how to interpret the parameter values entered in the **Parameter values** field. Select:

- **Frequency** (the default setting) to use the parameter values without modification.
- **Fraction** to multiply the parameter values by the absolute value of the largest eigenvalue in the reduced model divided by two.
- **Spread** to treat the parameter values as an interval around each eigenvalue in the reduced model. That is, the absolute value of each eigenvalue is multiplied by the parameter values and the resulting parameter value vectors are concatenated into one.

Use the **Linearity** list to specify the type of linear behavior. Select:

- **Linear** to use a linear solver with the same linearization point for both residual and Jacobian computation, which corresponds to one step in Newton's method.
- **Linear perturbation** (the default setting) to use a linear solver that computes the Jacobian in the same way as the **Linear** option but uses a zero solution when computing the residual. It is useful for small-signal analysis and similar applications where the variations around a linearization point are of interest.

Further Options for Time Dependent

Use the **Times** field to enter a vector of times that define the time span for the simulation. Click the **Range** button () to define time values using the **Range** dialog

box. Output from a simulation includes the times given in this field and the corresponding solutions.

Tolerance

Use the **Relative tolerance** field to enter a positive number (default value: 0.01).

Depending on the selection in the **Study type** list in the **General** section, the tolerance has the following meanings:

- When **Study type** has been set to **Frequency domain** the **Relative tolerance** is used as a termination tolerance for iterative linear system solvers and for error checking (if enabled) for direct linear system solvers.
- When **Study type** has been set to **Time dependent** the **Relative tolerance** is used by the solver in each time step to control the relative error. The absolute tolerance settings below work in the same way as for the time-dependent solver, but internally the full length absolute tolerance vector is transferred to the modes by the same transformation (projection) as is used to transform the problem to reduced form (the eigenmodes).

EIGENPAIRS

Use the **Solution** list to specify a solver configuration to be used when constructing the reduced model.

Use the **Use** list, present only for solution sequences with additional stored solutions, to specify a solution containing the modes to be used in the reduced model.

Use the **Eigenpairs** list to specify which of the eigenpairs present in the solution to include when constructing the reduced model. With the default setting, **All**, the solver uses all available eigenpairs. Select **Manual** to enter space-separated list of eigenpair numbers in the **Eigenpair numbers** field.

Use the **Damping ratios** field to enter either a scalar value or a space-separated list with values. The total number of entered values must be one or equal to the number of eigenpairs in the reduced model. If one number is entered, that value becomes the damping ratio for all eigenpairs. If the field is empty (the default), no damping is applied by the solver.

VALUES OF LINEARIZATION POINT

The problem solved by the **Modal** solver is assumed to be a linearization about a solution. You can specify such a solution (a linearization point) with the **Prescribed by** list. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point. Use the **Solution** list to specify which solution to use if **Prescribed by** has been set to **Solution**. Select:
 - **Zero** to use a linearization point that is identically equal to zero.
 - Any other available solution to use it as linearization point.

To store the used linearization point in the output, select the **Store linearization point and deviation in output** check box.

OUTPUT

The output from the solver can either be the solution, the reduced matrices, or both. Use the **Compute** list to specify **Solution**, **Solution and reduced matrices**, or **Reduced matrices**. If you selected **Solution** or **Solution and reduced matrices** you can select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory (this option is active by default). If you selected **Solution and reduced matrices** or **Reduced matrices** you can select one or several of **Stiffness matrix**; **Damping matrix**; **Damping ratio matrix**; **Mass matrix**; **Projection matrix**; **Load vector**; **Load vector, zero solution**; or **Load vector, deviation** to output the corresponding quantities.

ADVANCED

Use the **Load factor** field to enter a globally available scalar-valued expression (default: 1). The solver uses this expression to multiply the residual. The purpose is to facilitate the use of simple nonconstant Dirichlet boundary conditions (for frequency response) and simple nonconstant Neumann boundary conditions (for transient response).

THE MODAL SOLVER ALGORITHM

The purpose of the modal solver is to speed up certain simulations by performing a model reduction using eigenpairs. That is, the solution of the underlying system of equations is assumed to be a linear combination of parametric or time-dependent coefficients and a few dominant eigenvectors.

The equation of interest can be written as

$$E\ddot{u} + D\dot{u} + Ku = L \quad (19-1)$$

where E is the mass matrix, D is the damping matrix, and K is the stiffness matrix. The vector L is, in general, time dependent. The algorithm assumes that the matrices are constant. Either E or D can be identically zero. When performing frequency response studies, the vector L is assumed to be harmonic. Before solving [Equation 19-1](#), an eigenvalue analysis has to be performed. Let Φ_r be a matrix containing (right) eigenvectors from the eigenvalue analysis as columns. Let Φ_l be a matrix of the same size as Φ_r containing, for instance, the same vectors as Φ_r or left eigenvectors for the same setup as used to compute Φ_r . If all appearing Dirichlet boundary conditions are homogeneous, an approximation, u_m , of u can be written in terms of the eigenvectors:

$$u_m = \Phi_r q(t) \quad (19-2)$$

where $q(t)$ is a vector of unknown coefficients. Replacing u in [Equation 19-1](#) by $\Phi_r q(t)$ and premultiplying by Φ_l^T results in

$$\Phi_l^T E \Phi_r \ddot{q} + \Phi_l^T D \Phi_r \dot{q} + \Phi_l^T K \Phi_r q = \Phi_l^T L \quad (19-3)$$

The damping matrix D is present when performing the eigenvalue analysis (when it is nonzero). No matter if a nonzero D is present or not it is possible to add damping by giving damping ratios per mode. Denote the damping ratio for mode i ξ_i and let ω_i^2 be the corresponding eigenvalue. The damping is then added to

$$\Phi_l^T D \Phi_r \text{ as } \text{diag}(2\xi_i \omega_i (E_r)_{ii})$$

where E_r is the reduced mass matrix.

Modal Reduced Matrices—Time-Dependent Study

The Modal Solver can export matrices and the right-hand side for use in further simulations. The reduced matrices are the stiffness matrix $K_r = \Phi_l^T K \Phi_r$

the mass matrix

$$E_r = \Phi_l^T E \Phi_r$$

the damping matrix

$$D_r = \Phi_l^T D \Phi_r$$

the damping ratio matrix

$$D_{\text{ratio}} = \text{diag}(2\xi_i\omega_i(E_r)_{ii})$$

and the projection matrix Φ_r . For the transient problem the reduced right-hand sides are the load vector, zero solution

$$L_0 = \overline{\Phi}_l^T L|_{u=0}$$

the load vector

$$L_r = \overline{\Phi}_l^T L$$

and the load vector, deviation

$$L_p = L_r - L_0$$

Two simulations are needed for a transient problem if $L_p \neq 0$: one for L_p and one for $f(t)L_0$, where $f(t)$ is a time-dependent function that controls the load factor. The solution is then given by superposition of the two solutions. Note that the vector L_p can be nonzero for numerical reasons. If you believe this to be the case you can reduce the output times to get a more exact value for L_p . In most cases the solution is accurately computed by $f(t)L_0$.

Frequency Response

For a frequency response study the load L is assumed to be of the type

$$L(\Omega, t) = \tilde{L}(\Omega)e^{i\Omega t} \quad (19-4)$$

where Ω is the frequency of the forcing function. The steady-state solution of [Equation 19-3](#) is then of the form

$$q(t) = ce^{i\Omega t} \quad (19-5)$$

Use the expression of L from [Equation 19-4](#) and the expression of q from [Equation 19-5](#) in [Equation 19-3](#) and expand around Ω_0 , where Ω_0 is the first frequency given, to get

$$\left[\left(\frac{\Omega - \Omega_0}{2\pi} \right)^2 \overline{\Phi}_l^T \tilde{E} \Big|_{\Omega_0} \Phi_r - \left(\frac{\Omega - \Omega_0}{2\pi} \right) \overline{\Phi}_l^T \tilde{D} \Big|_{\Omega_0} \Phi_r + \overline{\Phi}_l^T \tilde{K} \Big|_{\Omega_0} \Phi_r \right] c = \overline{\Phi}_l^T \tilde{L}(\Omega) \quad (19-6)$$

An approximative solution to the original problem is given by $u_m = \Phi_r c$. No matter if a nonzero D is present or not it is possible to add damping by giving damping ratios

per mode. Denote the damping ratio for mode i ξ_i and let ω_i^2 be the corresponding eigenvalue. The damping is then added to

$$\overline{\Phi}_l^T \tilde{D} \Big|_{\Omega_0} \Phi_r \text{ as } \text{diag}\left(\frac{\xi_i \omega_i (E_r)_{ii}}{2\pi}\right)$$

where E_r is the reduced mass matrix. This assumes that no non-homogeneous Dirichlet boundary conditions are present.

The only type of parameter dependent Dirichlet boundary conditions that are supported are those that can be written as a scalar frequency dependent function times a constant vector (that is, the constraint vector M can be written as $M = m(\Omega)M_0$). For non-homogeneous Dirichlet boundary conditions a particular solution is needed. Therefore a particular solution is computed from the nonreduced equation

$$\begin{cases} Ku_p = 0 \\ Nu_p = M \end{cases} \quad (19-7)$$

The term

$$\overline{\Phi}_l^T \left(\left(\frac{\Omega - \Omega_0}{2\pi} \right)^2 \tilde{E} \Big|_{\Omega_0} u_p - \left(\frac{\Omega - \Omega_0}{2\pi} \right) \tilde{D} \Big|_{\Omega_0} u_p \right)$$

is then added to the right side of [Equation 19-6](#) to homogenize the original problem. Once an approximative solution, u_h , has been found to the homogeneous problem using modal analysis an approximative solution of the nonhomogeneous problem is given by

$$u_m = u_p + u_h$$

Modal Reduced Matrices—Frequency Response Study

The Modal Solver can export matrices and the right-hand side for use in further simulations. The reduced matrices are the stiffness matrix

$$K_r = \overline{\Phi}_l^T \tilde{K} \Big|_{\Omega_0} \Phi_r$$

the mass matrix

$$E_r = \overline{\Phi}_l^T \tilde{E} \Big|_{\Omega_0} \Phi_r$$

the damping matrix

$$D_r = \overline{\Phi}_l^T \tilde{D} \Big|_{\Omega_0} \Phi_r$$

the damping ratio matrix

$$D_{\text{ratio}} = \text{diag}\left(\frac{\xi_i \omega_i (E_r)_{ii}}{2\pi}\right)$$

and the projection matrix Φ_r .

For frequency response the load vector is the only vector needed. In that case

$$L_0 = L_r = \overline{\Phi}_l^T \tilde{L}(\Omega) \text{ and } L_p = 0$$

The right-hand side (rhs) is given by

$$\text{rhs} = f(\Omega)L_r$$

where f is a function of frequency that controls the load factor. The equation to solve is given by

$$A_r = \left(\frac{\Omega - \Omega_0}{2\pi}\right)^2 E_r - \left(\frac{\Omega - \Omega_0}{2\pi}\right) D_r - D_{\text{ratio}} + K_r = \text{rhs}$$

Note that no nonhomogenous Dirichlet boundary conditions are assumed.

Optimization Solver



The **Optimization Solver** () provides the settings for solving PDE-constrained optimization problems. This solver requires the Optimization Module. See the *Optimization Module User's Guide* for details.

Plug Flow Solver



The **Plug Flow Solver** () is a version of the **Time Dependent** solver specially designed for solving plug flow reactor models set up in the Reaction Engineering interface. This solver requires a license for the Chemical Reaction Engineering Module.

The **Plug Flow Solver** (Plug flow icon) is the default solver for **Stationary Plug Flow** study steps. It shares all settings with the **Time Dependent** solver, except it steps in *volume* instead of time. The correspondence between time, t , and volume, V , is given by $V = vt$, where v is the volumetric flow rate (SI unit: m^3/s). For descriptions of the available settings, see [Time Dependent](#).

Stationary Solver

Use the **Stationary Solver** (Stationary icon) to find the solution to linear and nonlinear stationary problems (also called static or steady-state problems).

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

The number in the **Relative tolerance** field (default: 0.001) is used for tolerance-based termination of iterative solver processes and for error checking (if enabled) for direct linear system solvers.



The termination tolerance used for iterative processes is also influenced by values specified in the **Tolerance factor** fields present in active **Fully Coupled**, **Segregated**, and **Segregated step** subnodes.

Use the **Linearity** list to specify whether to use a nonlinear or linear solver. Select:

- **Automatic** to perform an analysis that automatically detects if the problem can be solved with a linear solver approach.
- **Linear** to use a linear solver. This option uses the same linearization point for both residual and Jacobian computation and corresponds to one step in Newton's method.
- **Linear perturbation** to use a linear solver. This option computes the Jacobian in the same way as the **Linear** option but uses a zero solution when computing the residual. It is useful for small-signal analysis and similar applications where the variations around a linearization point are of interest.
- **Nonlinear** to use a nonlinear solver.

If you select **Linear** or **Linear perturbation** from the **Linearity** list, COMSOL Multiphysics assumes that the problem to be solved is a linearization about a solution.

You can specify such a solution (a linearization point) using the **Prescribed by** list. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes under **Physics** as linearization point.
- **Solution** to use a solution as linearization point. Use the **Solution** list to specify which solution to use if **Prescribed by** has been set to **Solution**. Select:
 - **Zero** to use a linearization point that is identically equal to zero.
 - Any other solution to use it as linearization point. It can be the current solution in the sequence or a solution from another sequence or a solution that was stored with the **Store Solution** node. You select a stored solution by changing **Use** to the name of the stored solution.

To store the used linearization point, select the **Store linearization point and deviation in output** check box.

See [About Linear Solvers vs. Nonlinear Solvers](#) below for more information about linear and nonlinear solvers.

OUTPUT

- Select the **Reaction forces** check box if you want to compute and store reaction forces in the output.
- The computation of boundary flux variables involves solving a system of equations to obtain a continuous field from nodal flux values. If you select the **Use lumping when computing fluxes** check box, this system of equations is lumped. The benefits of using this option is that it can avoid certain spurious oscillations in the computed flux field, and it is also slightly faster. Lumping is not suitable in 3D for shape functions of order higher than 1.

LOG

This section, which is initially empty, contains a log from the time stepping. It is not available when the stationary solver is a subnode to another solver. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is available also when reopening the model.

ABOUT DAMPED NEWTON METHODS

The nonlinear solver uses an affine invariant form of the damped Newton method as described in [Ref. 3](#). You can write the discrete form of the equations as $f(U) = 0$, where

$f(U)$ is the residual vector and U is the solution vector. Starting with the initial guess U_0 , the software forms the linearized model using U_0 as the linearization point. It solves the discretized form of the linearized model $f'(U_0)\delta U = -f(U_0)$ for the Newton step δU using the selected linear system solver ($f'(U_0)$ is the Jacobian matrix). It then computes the new iteration $U_1 = U_0 + \lambda \delta U$, where λ ($0 \leq \lambda \leq 1$) is the damping factor. Next the modified Newton correction estimates the error E for the new iteration U_1 by solving $f'(U_0)E = -f(U_1)$. If the relative error corresponding to E is larger than the relative error in the previous iteration, the algorithm reduces the damping factor λ and recomputes U_1 . This algorithm repeats the damping-factor reduction until the relative error is less than in the previous iteration or until the damping factor underflows the minimum damping factor. When it has taken a successful step U_1 , the algorithm proceeds with the next Newton iteration.

A value of $\lambda = 1$ results in Newton's method, which converges quadratically if the initial guess U_0 is sufficiently close to a solution. In order to enlarge the domain of attraction, the solver chooses the damping factors judiciously. Nevertheless, the success of a nonlinear solver depends heavily on a carefully selected initial guess, so you should provide the best value for U_0 , giving at least an order of magnitude guess for different solution components.

Termination Criterion

You specify the termination criteria in the settings window for a [Fully Coupled](#) or [Segregated](#) subnode to the **Stationary Solver** node. For **Termination criterion: Solution**, the nonlinear iterations terminate when the following convergence criterion is satisfied: Let U be the current approximation to the true solution vector, and let E be the estimated error in this vector. The software stops the iterations when the relative tolerance exceeds the relative error computed as the weighted Euclidean norm

$$\text{err} = \sqrt{\frac{1}{M} \sum_{j=1}^M \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\frac{|E_{i,j}|}{W_{i,j}} \right)^2}$$

Here M is the number of fields; N_j is the number of degrees of freedom in field j . The double subscript denotes degree of freedom index (i) and field (j) component. We let $W_{i,j} = \max(|U_{i,j}|, S_j)$, where S_j is a scale factor that the solver determines on the basis of the scaling method. You can select scaling method from the **Method** list in the

Scaling section of the **Dependent Variables** node's settings window, according to the following rules:

- For **Automatic**, S_j is the average of $|U_{i,j}|$ for all DOFs i for fixed j , times a factor equal to 10^{-5} for highly nonlinear problems or 0.1 otherwise.
- For **Manual**, S_j is the value given in the **Scale** field.
- For **Initial value based**, S_j is the average of $|V_{i,j}|$ for all DOFs i with fixed j , where $V = U_0$ is the solution vector corresponding to the initial value. In case all DOFs are zero for that particular field j , the total mean of $|V_{i,j}|$ for all i and j is used instead.
- For **None**, $W_{i,j} = 1$. In this case, err is an estimate for the absolute error.



The (automatically damped Newton) nonlinear solver only checks the convergence criterion if the damping factor for the current iteration is equal to 1. Thus, the solver continues as long as the damping factor is not equal to 1 even if the estimated error is smaller than the requested relative tolerance.

For **Termination criterion: Residual**, the nonlinear iterations terminate when the following convergence criterion is satisfied: The software stops the iterations when the relative tolerance exceeds the relative error computed as the weighted Euclidean norm

$$\text{err} = \sqrt{\frac{1}{M} \sum_{j=1}^M \frac{1}{N_j \tilde{W}_j^2} \sum_{i=1}^{N_j} |F_{i,j}|^2}$$

where F is the current residual and \tilde{W} are the weights determined by the first and, if applicable, also the second residual. Here, the double subscript denotes the degree of freedom index (i) and the field (j) component. The iterations can also terminate if the relative step size is in the range of a hundred machine epsilon and in addition a full Newton step is taken.

For **Termination criterion: Solution or residual**, the nonlinear iterations terminate when the relative tolerance exceeds the relative error computed as the minimum of the solution based error and the error given by the **Residual factor** times the residual based error above.

ABOUT LINEAR SOLVERS VS. NONLINEAR SOLVERS

Automatic Nonlinearity Detection

COMSOL 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 is 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 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.

Which Models are Nonlinear?

How do you determine if a problem is linear or nonlinear? Finding out is not always easy, but for most physics 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 physics. First, in the Heat Transfer user 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, Single-Phase Flow is always nonlinear unless $\rho = 0$, resulting in the linear Stokes equations.

Whether your problem is linear or nonlinear, the solvers break it down into one or several linear systems of equations. Therefore, the linear solver selection affects the solution time and memory requirements also for nonlinear models.

ABOUT PSEUDO TIME STEPPING

Pseudo time stepping is used in transport problem to stabilize the convergence toward steady state. Here an adaptive feedback regulator controls a CFL (Courant–Friedrichs–Lowy) number which is then used for pseudo time stepping. The CFL number starts from a moderate value (order one) and increases up to several orders of magnitude at convergence.

A simple multiplicative PID regulator for CFL regulation is used

$$\text{CFL}_{n+1} = \left(\frac{e_{n-1}}{e_n} \right)^{k_P} \left(\frac{\text{tol}}{e_n} \right)^{k_I} \left(\frac{e_{n-1}/e_n}{e_{n-2}/e_{n-1}} \right)^{k_D} \text{CFL}_n \quad (19-8)$$

where the regulator parameters k_P , k_I , and k_D are positive constants. Here e_n is the nonlinear error estimate for step n and tol is a given target error estimate.

- The first factor is nothing but a power of the current convergence rate (based on the last two steps), and is the most important part of this regulator. If the error is decreasing, the regulator increases the CFL number and if the error is increasing, the regulator decreases the CFL number. The strength of this coupling (and the rapidness of this effect) is controlled by the parameter k_P .
- The next factor is used to regulate the CFL number toward the requested target error estimate. A standard local error estimate regulation uses only a factor of this sort, but for this type of regulation the absolute level of the error is not that important. However, without this factor ($k_I=0$) the CFL number might drift even though the error level is fluctuating on the same level. This factor can also be used to select an absolute regime for the error where increasing the CFL number should be more difficult.
- The last factor is a derivative factor; it is affected by the change of the convergence rate.

A hard lower limit $\text{CFL}_n \geq 1$ is used, and to lower the risk of premature termination there is an extra requirement of not accepting convergence until $\text{CFL}_n \geq \text{CFL}_\infty = 10^4$.

Pseudo time stepping is available for stationary problems. In the coupled approach it functions together with the constant damped Newton solver. See the settings for the [Fully Coupled](#) and [Segregated](#) for related parameters.

Time-Dependent Solver

Use the **Time-Dependent Solver** () to find the solution to time-dependent problems (also called dynamic or unsteady problems) using the implicit time-stepping methods: BDF or generalized- α .

THE IMPLICIT TIME-DEPENDENT SOLVER ALGORITHMS

The finite element discretization of the time-dependent PDE problem is

$$\begin{aligned} 0 &= L(U, \dot{U}, \ddot{U}, t) - N_F(U, t)\Lambda, \\ 0 &= M(U, t) \end{aligned}$$

which is often referred to as the *method of lines*. Before solving this system, the algorithm eliminates the Lagrange multipliers Λ . If the constraints $0 = M$ are linear and time independent and if the constraint force Jacobian N_F is constant then the algorithm also eliminates the constraints from the system. Otherwise it keeps the constraints, leading to a differential-algebraic system.

In COMSOL Multipysics, the solvers *IDA* and *generalized- α* are available to solve the above ODE or DAE system:

- IDA was created at the Lawrence Livermore National Laboratory ([Ref. 4](#)) and is a modernized implementation of the DAE solver DASPK ([Ref. 5](#)), which uses variable-order variable-step-size backward differentiation formulas (BDF).
- Generalized- α is an implicit, second-order accurate method with a parameter α or ρ_∞ ($0 \leq \rho_\infty \leq 1$) to control the damping of high frequencies. With $\rho_\infty = 1$, the method has no numerical damping. For linear problems this corresponds to the midpoint rule. $\rho_\infty = 0$ gives the maximal numerical damping; for linear problems the highest frequency is then annihilated in one step. The method was first developed for second-order equations in structural mechanics ([Ref. 6](#)) and later extended to first-order systems ([Ref. 7](#)).

Because the time-stepping schemes used are implicit, a possibly nonlinear system of equations must be solved each time step. IDA includes a Newton solver to solve this nonlinear system of equations. The Newton solver, in turn, uses an arbitrary linear solver for the resulting linear systems. Alternatively, you can use the Newton solver to solve the nonlinear system. This gives you more control of the nonlinear solution process; it is possible to choose the nonlinear tolerance, damping factor, how often the Jacobian is updated, and other settings such that the algorithm solves the nonlinear

system more efficiently. When you select the generalized- α time-stepping algorithm, COMSOL uses the Newton solver.

The linearization of the above system used in the Newton iteration is

$$\begin{aligned} E\ddot{V} + D\dot{V} + KV &= L - N_F\Lambda \\ NV &= M \end{aligned}$$

where $K = -\partial L / \partial U$ is the stiffness matrix,

$$D = -\partial L / \partial \dot{U}$$

is the damping matrix, and

$$E = -\partial L / \partial \ddot{U}$$

is the mass matrix. When $E = 0$, D is often called the mass matrix.

When using IDA for problems with second-order time derivatives ($E \neq 0$), extra variables are internally introduced so that it is possible to form a first-order time-derivative system (this does not happen when using generalized- α because it can integrate second-order equations). The vector of extra variables, here U_v , comes with the extra equation

$$\dot{U} = U_v$$

where U denotes the vector of original variables. This procedure expands the original ODE or DAE system to double its original size, but the linearized system is reduced to the original size with the matrix $E + \sigma D + \sigma^2 K$, where σ is a scalar inversely proportional to the time step. By the added equation the original variable U is therefore always a differential variable (index-0). The error test excludes the variable U_v unless consistent initialization is on, in which case the differential U_v -variables are included in the error test and the error estimation strategy applies to the algebraic U_v -variables.

BDF VS. GENERALIZED- α

The BDF solver uses backward differentiation formulas with order of accuracy varying from one (that is, backward Euler) to five. BDF methods have been used for a long time and are known for their stability. However, they can have severe damping effects, especially the lower-order methods. Backward Euler severely damps any high

frequencies. Even if you are expecting a solution with sharp gradients, you might get a very smooth solution due to the damping in backward Euler.

The generalized- α (alpha) solver has properties similar to the second-order BDF solver but the underlying technology is different. It contains a parameter, called α in the literature, to control the degree of damping of high frequencies. Compared to BDF (with maximum order two), generalized- α causes much less damping and is thereby more accurate. For the same reason it is also less stable.

The implementation of the generalized- α method in COMSOL detects which variables are first order in time and which variables are second order in time and applies the correct formulas to the variables.

For most problems, generalized- α is an accurate method with good enough stability properties. Many physics user interfaces in COMSOL—for transport problems, for example—have generalized- α as the default transient solver. Some complicated problems, however, need the extra robustness provided by the BDF method. There are also some problem types, like ODE systems, that can benefit from the higher accuracy that high-order BDF methods provide.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. If you select User defined, you can specify the following settings:

- Use the **Times** field to enter a vector of times that define the time span for the simulation using the **Range** button () if needed (default: range(0,0.1,1)).
- Use the **Relative tolerance** field to enter a positive scalar number (default: 0.01). The solver uses this number to control the relative error in each time step.

ABSOLUTE TOLERANCE

Specify an absolute tolerance that is used by the solver to control the absolute error. The tolerance specified here is applied to all variables unless modified per variable by selecting another method than the global method for a variable.

Use the **Global method** list to select how the specified absolute tolerance is to be interpreted for the variables that use the global method (by default, all variables use the global method). Select:

- **Scaled** to let the absolute tolerance be applied to scaled variables.
- **Unscaled** to let the absolute tolerance be applied to unscaled variables.

In the **Tolerance** field you enter a positive number that is applied to either scaled or unscaled variables.

To specify the absolute tolerance individually for a variable, select the variable from the **Variables** list and modify the corresponding tolerance with the **Method** list. Select:

- **Scaled** to apply the specified tolerance to scaled variables.
- **Unscaled** to apply the specified tolerance to unscaled variables.
- **Use global** (the default) to apply the tolerance specified for the global tolerance.

If you select either **Scaled** or **Unscaled** from the **Method** list additional fields appear. Use the **Tolerance** field to modify the absolute tolerance for the selected variable.

If a problem of wave-equation type is being solved and if **Method** in the **Time Stepping** section has been set to **BDF**, then it is necessary to specify absolute tolerances for time derivatives of the variables as well. By default, the solver chooses a tolerance for these components. To manually enter a tolerance for a time derivative, select the **Tolerance, time derivative** check box and enter a tolerance in the associated field.



The **Method** setting (**Scaled** or **Unscaled**) that is selected for a variable applies also to its time derivative.

The absolute and relative tolerances control the error in each integration step. More specifically, let U be the solution vector corresponding to the solution at a certain time step, and let E be the solver's estimate of the (local) error in U committed during this time step. For the **Unscaled Method**, the step is accepted if

$$\left(\frac{1}{M} \sum_j \frac{1}{N_j} \sum_i \frac{|E_i U_i|}{A_{us,i} + R |U_i|}^2 \right)^{1/2} < 1$$

where $A_{us,i}$ is the unscaled absolute tolerance for DOF i , R is the relative tolerance, M is the number of fields, and N_j is the number of degrees of freedom in field j . The numbers $A_{us,i}$ are computed from a conversion of the input value A_k for the corresponding dependent variable k . For degrees of freedom for Lagrange shape functions or for ODEs these values are the same as entered (that is $A_{us,i} = A_k$) but for vector elements there is a field-to-DOF conversion factor involved.

For the **Scaled Method** and when you select **Update scaled absolute tolerance**, the step is accepted if

$$\left(\frac{1}{M} \sum_j \frac{1}{N} \sum_i \left(\frac{|E_i Y|}{A_{s,i} + R |Y_i|} \right)^2 \right)^{1/2} < 1$$

where $A_{s,i}$ is the scaled absolute tolerance for DOF i , M is the number of fields, R is the relative tolerance, N_j is the number of degrees of freedom in field j , and Y is the scaled solution vector. For dependent variables that are using the scaling method

Automatic, the numbers $A_{s,i}$ are computed from the input values A_k according to the formula

$$A_{s,i} = A_{k_i} (\beta + \|Y\|_{2,i}) \quad \beta = \begin{cases} (1 - e^{-\alpha j}) \|Y\|_{\infty, k_i} + e^{-\alpha j} & 0 < \|Y\|_{\infty, k_i} < 1 \\ 1 & \text{else} \end{cases}$$

where $\alpha = \frac{1}{5}$, j is the time-step iteration number $j = 0, 1, \dots$, and $\|Y\|_{2,k_i}, \|Y\|_{\infty, k_i}$ are the 2-norm and maximum norm of the dependent variable k_i , respectively. Here A_{k_i} is the converted input value A_k for the field k and DOF i . For dependent variables that are using another scaling method or when the **Update scaled absolute tolerance** check box is cleared, then $A_{s,i} = A_{k_i}$.



If the solution is smaller than the absolute tolerance, there is no accuracy at all.



For DAEs (differential-algebraic equations), you can exclude the algebraic equations from the error estimation so that the error is only based on the differential equations. See [Advanced](#) for information about excluding algebraic equations from the error estimate.

TIME STEPPING

Select a time-stepping method as well as various settings that apply to these methods.

You select a time-stepping method with the **Method** list. Select:

- **BDF** to use a backward differentiation formula.
- **Generalized alpha** to use the generalized- α method.
- **Initialization only** to compute consistent initial values only and then stop.

If a **Fully Coupled** or **Segregated** attribute node is attached to a **Time** node, the settings for the nonlinear systems solved by the time-stepping methods come from that node.

The time-stepping method **Generalized alpha** requires a **Fully Coupled** or **Segregated** attribute node.



The time-stepping method **BDF** can be used without a **Fully Coupled** or **Segregated** attribute node. In such a situation the **BDF** method uses an internal automatic nonlinear solver.

The **Steps taken by solver** list makes it possible to modify how the time-stepping methods **BDF** and **Generalized alpha** select their time steps. Select:

- **Free** to let the time-stepping method choose time steps freely. The times specified in the **Times** field in the **General** section are not considered when a time step is chosen.
- **Intermediate** to force the time-stepping method to take at least one step in each subinterval of the times specified in the **Times** field in the **General** section.
- **Manual** to override the automatic choice of time step with a manual choice.

The **Manual** option is only available for **Generalized alpha** and overrides the local error estimation made in each time step.

-
- The **Manual** option is only available for **Generalized alpha** and overrides the local error estimation made in each time step.
- **Strict** to force the time-stepping method to take steps that end at the times specified in the **Times** field in the **General** section. The solver takes additional steps in between these times if necessary.

Further options that apply to one or several combinations (as indicated at each bullet) of selections made in the **Method** and **Steps taken by solver** lists are:

- **Initial step** (not available when **Steps taken by solver** is set to **Manual**). By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.
- **Maximum step** (not available when **Steps taken by solver** is set to **Manual**). By default the solver chooses a maximum time step automatically. Select the **Maximum step** check box for manual specification of a maximum time step.
- **Time step** (only available when **Steps taken by solver** is set to **Manual**). Enter a manual time step specification as a scalar, a vector of times, or an expression containing global variables in the **Time step** field.

- **Allow complex numbers** (only available when **Method** is set to **BDF**). Select the **Allow complex numbers** check box to be able to solve problems that are not automatically determined to be complex valued in a correct way.
- **Maximum BDF order** (only available when **Method** is set to **BDF**). This setting controls the maximum allowed degree of the interpolating polynomial of the BDF method.
- **Minimum BDF order** (only available when **Method** is set to **BDF**). This setting can be used to prevent the solver from decreasing the order of the BDF method below 2
- **Event tolerance** (only available when **Method** is set to **BDF**). This setting can be used to set the event tolerance (default value: 0.01), which is used for root finding of event conditions when using implicit events; see [Implicit Event](#).



This does not apply to the start-up phase of the simulation.

- **Time step increase delay** (only available when **Method** is set to **Generalized alpha**, but not when **Steps taken by solver** is set to **Manual**). Select this check box and enter a positive integer in the field if you want the solver to be more restrictive when increasing the time step. Entering 0 results in the same behavior as clearing the check box.
- **Amplification for high frequency** (only available when **Method** is set to **Generalized alpha**). By entering a number between 0 and 1 you can control how much damping of high frequencies the solver provides. A value close to 0 results in efficient damping, while a number close to 1 results in little damping.
- **Predictor** (only available when **Method** is set to **Generalized alpha**). Select **Linear** to use linear extrapolation of the present solution to construct the initial guess for the nonlinear system of equations to be solved at the next time step. Select **Constant** to use the current solution as initial guess.

RESULTS WHILE SOLVING

Select the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depend on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time stepping method.

If applicable, use the **Probe** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

OUTPUT

Use the **Times to store** list to control at what times the solver stores a solution. Select:

- **Specified times** to store solutions at the times entered in the **Times** field in the **General** section.
- **Steps taken by solver** to store solutions at the time steps taken by the solver.



The selection made in the list **Steps taken by solver** in the **Time Stepping** section influences the output in this situation.

- Select the **Store reaction forces** check box if you want to compute and store reaction forces in the output.
- The computation of boundary flux variables involves solving a system of equations to obtain a continuous field from nodal flux values. If you select the **Use lumping when computing fluxes** check box, this system of equations is lumped. The benefit of using this option is that it can avoid certain spurious oscillations in the computed flux field, and it is also slightly faster. Lumping is not suitable in 3D for shape functions of order higher than 1.
- Select the **Store time derivatives** check box if you want to store time derivatives of the variables solved for in the output. Storing the time derivatives gives more accurate results when evaluating quantities that involve these time derivatives.
- Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.
- Select the **Store solution before and after events** check box to store two additional solutions every time an implicit or explicit event is triggered; see [The Events User Interface](#). The solutions before and after the reinitialization at the event are stored.

ADVANCED

Use the list **Singular mass matrix** to control whether the solver automatically determines if a system includes a differential-algebraic equation or not. Select:

- **Automatic** to make the solver look for zero-filled rows or columns in the mass matrix as a means of detecting a differential-algebraic equation.
- **Yes** if the model includes a differential-algebraic equation where the mass matrix has no zero-filled rows or columns.

Use the list **Consistent initialization** to control how the solver performs consistent initialization of differential-algebraic systems. Select:

- **Backward Euler** to perform consistent initialization using a small artificial step with the backward Euler method.
- **Off** to indicate that the initial values already are consistent, which means that the solver does not modify them.
- **On** to use a consistent initialization routine that is preferable to **Backward Euler** for index-1 differential-algebraic equations.



The **On** option is only available when **Time method** has been set to **BDF** at the same time as the internal nonlinear solver of the BDF method is used.

Use the list **Error estimation** to control how to treat algebraic degrees of freedom of a differential-algebraic system when estimating the time discretization error. Select:

- **Include algebraic** to include the algebraic degrees of freedom in the error norm.
- **Exclude algebraic** to exclude the algebraic degrees of freedom from the error norm.

LOG

This section, which is initially empty, contains a log from the time stepping. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is available also when reopening the model.



[The Implicit Time-Dependent Solver Algorithms](#)

Time Discrete Solver

Use the **Time Discrete Solver** () to find the solution to time-dependent problems (dynamic or unsteady problems) that have already been discretized in time using, for example, the `prev` operator or the `bdf` operator.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

Use the **Times** field to enter a vector of times that define the simulation's time span.

The time step is specified in the **Time step** field. Valid entries are a scalar, a vector of times, or an expression containing global expression variables.

Discretizing time derivatives using the `prev` operator or the `bdf` operator requires the solution at previous discrete times. How many previous time steps that should be stored is specified in the **Number of time discrete levels** field. If you, for example, use the first-order `bdf` operator (`bdf(u, 1)`), the solution at one previous time step is required. Using the second-order `bdf` operator (`bdf(u, 2)`) requires the solution at two previous time steps. The default value is 2.

Use the **Relative tolerance** field to enter a positive number. This number controls how accurately the nonlinear system of equations is solved in each time step. In general, the desired relative error in the solution should be entered here.

ABSOLUTE TOLERANCE

Here you can specify an absolute tolerance that the nonlinear solver uses to control the absolute error. The tolerance specified here is applied to all variables unless modified per variable by selecting another method than the global method for a variable.

Use the **Global method** list to select how the specified absolute tolerance is to be interpreted for the variables that use the global method (by default, all variables use the global method). Select:

- **Scaled** to let the absolute tolerance be applied to scaled variables.
- **Unscaled** to let the absolute tolerance be applied to unscaled variables.

In the **Tolerance** field you enter a positive number that is applied to either scaled or unscaled variables.

To specify the absolute tolerance individually for a variable, select the variable from the **Variables** list and modify the corresponding tolerance with the **Method** list. Select:

- **Scaled** to apply the specified tolerance to scaled variables.
- **Unscaled** to apply the specified tolerance to unscaled variables.
- **Use global** (the default) to apply the tolerance specified for the global tolerance.

If you select **Scaled** or **Unscaled** additional fields appear. Use the **Tolerance** field to modify the absolute tolerance for the selected variable.

RESULTS WHILE SOLVING

Check the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depend on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time-stepping method.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

OUTPUT

Use the **Times to store** list to control at what times the solver stores a solution. Select:

- **Specified times** to store solutions at the times entered in the **Times** field in the **General** section.
- **Steps taken by solver** to store solutions at the time steps taken by the solver.

When **Specified times** is selected, the solution to output is computed through interpolation. Therefore, the solution at previous time steps is not computed, which means that expressions with the `prev` and `bdf` operators cannot be used in analysis. Such expressions can only be used in analysis when you have selected **Steps taken by solver**.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

LOG

This section, which is initially empty, contains a log from the time stepping.

Time Explicit Solver

Use the **Time Explicit Solver** () to find the solution to time-dependent problems (also called dynamic or unsteady problems) using the family of Runge-Kutta explicit time-stepping schemes or the Adams-Bashforth 3 solver.

THE EXPLICIT TIME-DEPENDENT SOLVER ALGORITHMS

For the nodal discontinuous Galerkin method, it is natural and most efficient to use an explicit time-stepping method. Other situations where it can be advantageous is when using only particle tracing or wave problems together with so-called mass lumping.

Runge-Kutta methods

Explicit classical Runge-Kutta methods of order 1–4 are supported. Runge-Kutta 4 is the default choice for the discontinuous Galerkin method.

Adams-Bashforth methods

The third-order Adams-Bashforth multistep method (AB3) for $u_t = R(u)$ is

$$u_{n+1} = u_n + \frac{k}{12}(23R(u_n) - 16R(u_{n-1}) + 5R(u_{n-2}))$$

where u_n is the solution at time t_n , and k is the time step.

The time restriction for the discontinuous Galerkin method for wave problems is directly proportional to the smallest mesh element size. For the Wave Form PDE a suitable time step can be determined automatically: Select **From expressions** in the **Time stepping** list and specify the variable `wahw.wtc`, for example. In practice it is not uncommon that only a few elements are small, yet these elements dictate the overall time step for the problem. To remedy this problem, an Adams-Bashforth 3 (local) time-stepping scheme, which saves computational effort by time marching groups of elements of similar size independently, is available for the Wave Form PDE user interface.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. You can also select **User defined** to define all settings locally.

Use the **Times** field to enter a vector of times that define the time span for the simulation. Click the **Range** button () to define a range using the **Range** dialog box.

Use the **Method** list to specify the time-explicit method: **Adams-Bashforth 3**, **Adams-Bashforth 3 (local)** (available for the Wave Form PDE interface), or the classic **Runge-Kutta** family.

For Runge-Kutta, select the order of the time-stepping scheme from the **Order** list.

From the **Time stepping** list, for Runge-Kutta and Adams-Bashforth 3, specify **Manual** or time stepping **From expressions**, where the latter is useful for the Wave Form PDE. When you use **From expression**, a list of **Cell time scale expressions** appear, where you can add such expressions to define the time stepping. For explicit methods the largest stable time step can automatically be computed from an expression. Some physics user interfaces (Wave Form PDE, for example) defines such an expression in terms of an estimated maximum wave speed (defined by the interface) and the element size (`wahw.wtc`). Here the element order is also taken into account. The expression should in general represent a local cell time scale. For wave problems, the expression should be proportional to the time it takes for the fastest wave to pass one mesh element. Each expression given is evaluated on all mesh elements. The smallest value (time scale), over all elements and all expressions, dictates the time step used. If you select **User defined** from the **Defined by study step** list, you can use the **Add** button (+) and the **Delete** button (-) to add or delete rows in the list.

The time step is specified in the **Time step** field when **Time stepping** manual is selected. Valid entries are a scalar, a vector of times, or an expression containing global expression variables. The default value is 0.001 s ($1e^{-3}$ s).

Use the **Linear solver** list to select the linear solver to be used within the time stepping scheme to invert the mass matrix. Available linear solvers appear in the model tree. The default is to use the **Direct** linear solver. For cheap but approximate inversion of the mass matrix use the **Lumped** option.



This option can only be used together with a linear space discretization.

In rare cases, when the PDE is nonlinear, you may need to adjust the **Relative tolerance** (default value: 0.01).

RESULTS WHILE SOLVING

Check the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

Use the **Update at** list to specify at what times to plot the results. Select:

- **Output times** to update the plot at the times specified by the **Times** field.
- **Time steps from solver** to update the plot at the set of times used by the time-stepping method.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

OUTPUT

Use the **Times to store** list to control at what times the solver stores a solution. Select:

- **Specified times** to store solutions at the times entered in the **Times** field in the **General** section.
- **Steps taken by solver** to store solutions at the time steps taken by the solver.

When **Specified times** is selected, the solution to output is computed through interpolation. Therefore, the solution at previous time steps is not computed, which means that expressions with `prev` and `bdf` operators cannot be used in analysis. Such expressions can only be used in analysis when you have selected **Steps taken by solver**.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

Dependent Variables

The **Dependent Variables** node ( ) handles initial data and scaling for the dependent variables that you solve for as well as how to compute dependent variables not solved for. The methods are applicable to the dependent variables present as **Field** subnodes (  ) under the **Dependent Variables** node. This node automatically updates the **Field** nodes. So, if the study type for the solver changes or if you use a different study type, then the **Field** nodes change accordingly. See **Field** for details about the settings available for a **Field** node. You can plot and evaluate the initial values for the dependent variables by right-clicking the **Dependent Variables** node and choosing **Compute to Selected** or click the corresponding button () on the settings window's toolbar (this is similar to the **Get Initial Value** option for the main Study nodes).

The settings window for the **Dependent Variables** node contains the following sections:

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

INITIAL VALUES OF VARIABLES SOLVED FOR

Use the **Method** list to specify how to compute initial values for the dependent variables that you solve for. Select:

- **Initial expression** (the default) to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object. Use the **Solution** list to specify what solution object to use (directly or as part of the initial expression).
Select:
 - **Zero** to initialize all variables to zero.
 - Any available solution object to use it as initial value.

Depending on the study type for the selected solution object, you can choose different solutions from a list underneath the **Solution** list:

- For a stationary study, from the **Selection** list, select **Automatic** (the default) to use the last (typically the only) solution, select **First** to use the first (typically the only) solution, select **Last** to use the last (typically the only) solution, select **All** to use all (typically just one) solutions from that study, select **Manual** to use a specific solution number that you specify, or select **I** to use the first (typically the only) solution. If you use a parametric continuation of the stationary study, there can be additional solutions to choose from.
- For a time-dependent study, from the **Time** list, select **Automatic** (the default) to use the solution for the last time, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Interpolated** to specify a time in the text field that opens and use the interpolated solution at that time, select **Manual** to use a specific solution number that you specify, or select one of the output times to use the solution at that time.
- For an eigenvalue study, from the **Selection** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the eigenvalues to use the corresponding eigensolution.
- For a parametric or frequency-domain study, from the **Parameter value** list, select **Automatic** (the default) to use the first parameter value set or frequency, select **First**

to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the parameter value sets or frequencies to use the corresponding solution.

SCALING

Use the **Method** list to specify how to scale the variables solved for. Select:

- **Automatic** to get an automatically determined scaling (the default), which works well for most models. It is initially based on the magnitudes of the elements in the Jacobian and mass matrices. For nonlinear problems these scales are recomputed based on the magnitude of the solution iterate.
- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling if you know the order of magnitudes of the variables in advance. For instance, suppose that a model includes two fields `u` and `sigma`, and that the values of `u` are on the order of 10^{-4} , and the values of `sigma` are approximately 10^6 . To use this knowledge, type `u 1e-4 sigma 1e6` in the **Scale** field that appears. The solvers then internally use a rescaled solution vector for the degrees of freedom for `u` and `sigma`, which both are of the order 1. The scaling factors are given per field (or per state for ODEs) and are internally converted to degrees of freedom scaling factors. The units for the manual scales are the same as for the corresponding field quantity in the model's base unit system.
- **None** to get no scaling.



The automatic scaling in COMSOL Multiphysics does not work when using the nonlinear stationary solver and a field or state has an identically zero solution (the solver does not converge). In this case use **Manual** or **None**.

The scaling method also can be specified per variable in the settings window for the variable's **Field** node. For more information about scaling, see [Scaling of Variables and Equations](#) below.

VALUES OF VARIABLES NOT SOLVED FOR

These settings are only available if there are dependent variables in the model that you do not solve for. Then use the **Method** list to specify how to compute the values of variables not solved for. Select:

- **Initial expression** to use the expressions specified on the **Initial Values** nodes for the physics in the model.
- **Solution** to use initial values as specified by a solution object. Use the **Solution** list to specify what solution object to use if **Method** has been set to **Solution**. Select:
 - **Zero** to initialize all variables to zero.
 - Any other available solution object to use it as initial value.

Depending on the solution object to use, you can choose different solutions to use. If a solution has nodes for storing solutions in its sequence you can choose which solution to use using the **Use** list. The **Current** value is the value that the solution has at the moment the value is read. The other values are the values stored in the respective nodes of the sequence.

Depending on the study type for the selected solution object, you can choose different solutions from a list underneath the **Solution** list:

- For a stationary study, from the **Selection** list, select **Automatic** (the default) to use the last (typically the only) solution, select **First** to use the first (typically the only) solution, select **Last** to use the last (typically the only) solution, select **All** to use all (typically just one) solutions from that study, select **Manual** to use a specific solution number that you specify, or select **I** to use the first (typically the only) solution. If you use a parametric continuation of the stationary study, there can be additional solutions to choose from.
- For a time-dependent study, from the **Time** list, select **Automatic** (the default) to use the solution for the last time, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Interpolated** to specify a time in the text field that opens and use the interpolated solution at that time, select **Manual** to use a specific solution number that you specify, or select one of the output times to use the solution at that time.
- For an eigenvalue study, from the **Selection** list, select **Automatic** (the default) to use the first eigenvalue and its associated eigensolution, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that

study, select **Manual** to use a specific solution number that you specify, or select one of the eigenvalues to use the corresponding eigensolution.

- For a parametric or frequency-domain study, from the **Parameter value** list, select **Automatic** (the default) to use the first parameter value set or frequency, select **First** to use the first solution, select **Last** to use the last solution, select **All** to use all solutions from that study, select **Manual** to use a specific solution number that you specify, or select one of the parameter value sets or frequencies to use the corresponding solution.

You select whether to solve for a variable or not by left-clicking a **Field** subnode ( ) and then select or clear the **Solve for this field** check box in the settings window.

OUTPUT

Use the **Keep solution** list to specify which dependent variables to make available for results analysis and visualization. This is only applicable when you evaluate the dependent variables by right-clicking the **Dependent Variables** node and choosing **Compute to Selected**. Select:

- Initial values** (the default) to send the variables solved for, as specified by the **Field** nodes, as the solution data for results analysis.
- Variables not solved for** to send the variables not solved for, as specified by the **Field** nodes, as the solution data for results analysis.

SCALING OF VARIABLES AND EQUATIONS

If the dependent variables in a model have widely different magnitudes, the solver might have problems with the resulting ill-conditioned matrix. The scaling of the dependent variables also influences the weighted errors computed by the solvers. For instance, in a structural mechanics problem the displacements can be of the order of 0.0001 m while the stresses are 1,000,000 Pa (1 MPa). To remedy this situation, COMSOL internally rescales the variables so that a well-scaled system results.

The rescaling of the discretized linear system occurs before constraint handling. Assume that the degrees of freedom U_i are expressed terms of rescaled degrees of freedom \tilde{U}_i according to the formula

$$U_i = s_i \tilde{U}_i$$

where s_i are positive scale factors. Using a diagonal matrix S , the relation between U and \tilde{U} is $U = S\tilde{U}$, and you can write the rescaled linear system as

$$\begin{bmatrix} \tilde{K} & \tilde{N}_F \\ \tilde{N} & 0 \end{bmatrix} \begin{bmatrix} \tilde{U} \\ \tilde{\Lambda} \end{bmatrix} = \begin{bmatrix} \tilde{L} \\ \tilde{M} \end{bmatrix}$$

where

$$\Lambda = R\tilde{\Lambda} \quad \tilde{N}_F = S N_F R \quad \tilde{K} = S K S \quad \tilde{N} = R N S$$

and

$$\tilde{L} = S L, \quad \tilde{M} = R M$$

Here, R is a diagonal matrix of positive scale factors chosen such that the rows in the matrix N are of magnitude 1.

References for the Solution Operation Nodes and Solvers

1. P.E. Gill, W. Murray, and M.A. Saunders, *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Systems Optimization Laboratory (SOL), Stanford University, 2006.
2. P.E. Gill, W. Murray, and M.A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.
3. P. Deuflhard, “A Modified Newton Method for the Solution of Ill-conditioned Systems of Nonlinear Equations with Application to Multiple Shooting,” *Numer. Math.*, vol. 22, pp. 289–315, 1974.
4. A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward, “SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers,” *ACM T. Math. Software*, vol. 31, p. 363, 2005.
5. P.N. Brown, A.C. Hindmarsh, and L.R. Petzold, “Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems,” *SIAM J. Sci. Comput.*, vol. 15, pp. 1467–1488, 1994.
6. J. Chung, G.M. Hulbert, “A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized- α Method,” *J. Appl. Mech.*, vol. 60, pp. 371–375, 1993.

7. K.E. Jansen, C.H. Whiting, G.M. Hulbert, “A Generalized- α Method for Integrating the Filtered Navier–Stokes Equations with a Stabilized Finite Element Method,” *Comput. Methods Appl. Mech. Engrg.*, vol. 190, pp. 305–319, 2000.
8. The ARPACK Arnoldi package, www.caam.rice.edu/software/ARPACK.
9. P. Deuflhard, “A Stepsize Control for Continuation Methods and its Special Application to Multiple Shooting Techniques,” *Numer. Math.*, vol. 33, pp. 115–146, 1979.
10. R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Teubner Verlag and J. Wiley, Stuttgart, 1996.
11. R. Rannacher, “A Feed-back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples,” *East-West J. Numer. Math.*, vol. 4, pp. 237–264, 1996.
12. www.netlib.org/ode.

Solution Attribute Nodes

The following sections describe the solver attribute nodes—such as preconditioners, adaptive mesh refinement, and sensitivity analysis—and their settings.

Adaptive Mesh Refinement

This is an attribute node that adds adaptive mesh refinement. You can add an **Adaptive Mesh Refinement** node () together with the **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver** operation nodes.

THE ADAPTIVE SOLVER ALGORITHM

The adaptive solver performs the following iterative algorithm ([Ref. 10](#)):

- 1 Solve the problem on the existing mesh using the stationary or eigenvalue solver.
- 2 Evaluate the residual of the PDE on all mesh elements.
- 3 Estimate the error in the solution on all mesh elements. The computed error estimate is really an *error indicator* because the estimate involves an unknown constant (C above).
- 4 Terminate execution if it has made the requested number of refinements or if it has exceeded the maximum number of elements.
- 5 Refine a subset of the elements based on the sizes of the local error indicators.
- 6 Repeat these steps.

Time-dependent Adaption

An adapted solution at $t = t_n$ is mapped to the coarse base mesh. A new adapted mesh for the time interval $[t_n, t_{n+1}]$ is constructed by first computing a coarse solution on the base mesh in $[t_n, t_s]$, where t_s is the largest sample time and $t_{n+1} = 2t_n - t_{n-1}$. The error indicator is evaluated using the coarse solution at the given sample points.

In the case of automatic time interval control a measure of amount of refinement is computed and compared to a given requested value. If the computed value is too small or too large the interval length is increased or decreased, respectively, which results in a new t_{n+1} . If the interval length need to be changed the error indicator is sampled again using a new coarse solution. The comparison is done only once.

The new adapted mesh is obtained by using the error indicator sampled at given points in $[t_n, t_{n+1}]$, selecting a set of elements based on the element pick function, and then

finally refining these elements. The solution at t_n to the PDE problem on the previous adapted mesh for $[t_{n-1}, t_n]$ is then mapped to the new mesh for $[t_n, t_{n+1}]$ and time integration continues until the next mesh adaption takes place at t_{n+1} .

The simple measure used for determining the amount of refinement is

$$\rho = \frac{1}{2^p N} \sum_{i=1, \gamma(i) \neq 0}^N 2^{\gamma(i)}$$

Here γ is an N -vector of integers containing the number of times the element at that position should be refined, $p = \max_i \gamma(i)$, and N is the number of elements of the coarse base mesh.

- Now, the next interval length is decreased by a given factor if ρ is larger than 120% of the requested reference value. If ρ is smaller than 80% of the reference value it is instead increased. Otherwise the time interval length is kept the same.



Adaptive mesh refinement works with the mesh and equations defined in the domains (interior) of the geometry and does not consider meshes and equations on lower dimensions, such as surface meshes in shell models. In solid geometries, the adaptive mesh refinement of the interior mesh usually affects the surface mesh.

GENERAL

The software performs adaptive mesh refinement in one geometry only. Use the **Adaption in geometry** list to specify that geometry.

Use the **Maximum number of elements** field to specify the maximum number of elements in the refined mesh. If the number of elements exceeds this number, the solver stops even if has not reached the number specified in the **Maximum number of refinements** field.

Properties for the Stationary and Eigenvalue Adaptive Solver

Use the **Maximum number of refinements** field to specify the maximum number of mesh refinement iterations. The default value is 2 refinements.

Properties for the Time-Dependent Adaptive Solver

The following properties appear under **Time interval control**:

- The **Time interval length** can be controlled manually or automatically. Select **Manual** (default) or **Automatic**.
- The value in the **Interval reduction factor** field (default value: 0.5) determines how the solver reduces the time interval length. A value of 0.5 makes the interval length half of the previous interval length when reduced.
- By default, the solver determines the **Interval length** (unit: s) automatically (only available when **Time interval length** is **Manual**) using an interval length that gives a total of 10 intervals. The length of the time interval is the simulation time before a refinement of the mesh takes place. Click the check box to specify a user-defined time interval length in the field (default value: 1 s).
- The value in the **Interval growth factor** (only available when **Time interval length** is **Automatic**) edit field (default value: 2.0) determines how the solver increases the time interval length. A value of 2.0 makes the interval length twice as large as the previous interval length when increased.
- By default, the solver determines the **Initial interval length** (unit: s) automatically (only available when **Time interval length** is **Automatic**). The length of the initial time interval is the simulation time before the first refinement of the mesh takes place. Select the check box to specify a user-defined time interval length in the field (default value: 0.1 s).
- By default, the solver also determines the **Minimal interval length** (unit: s) automatically. The minimal length of the time interval is the shortest possible simulation time without performing a mesh refinement. Click to select the check box to specify a user-defined minimal interval length in the field (default value: 0.01 s).
- If **Time interval length** is **Automatic** the algorithm strives to assume the given value of Fraction of maximum refinement (default value: 0.2) by controlling the size of the time interval. A value of zero means no refinement of the base mesh and a value of one means refinement everywhere using Maximum element refinements. The shortening and lengthening of the interval is determined by the interval reduction and growth factors described below.

For the properties under **Mesh element control**; see [Mesh Refinement](#) below. Also, select or clear the **Convert to simplex mesh** check box (the default is to use this conversion). Mesh refinement is only possible for simplex meshes. If the original mesh is not a simplex mesh it can be converted to a simplex mesh by this selection.

ERROR ESTIMATION

Use the **Error estimate** list to control how the error estimate is computed.

Error Estimates for the Stationary and Eigenvalue Solvers

Select **L2 norm** to use the L_2 norm.

Select **Functional** to specify a globally available scalar-valued expression. This option adapts the mesh toward improved accuracy in the expression for the functional. (Only available when using **Adaptive Mesh Refinement** together with the **Stationary** solver.)

Further options regarding error estimation (available as indicated for each option) are:

- **Scaling factor** (only available when **Error estimate** is **L2 norm**). Use this field to enter a space-separated list of scaling factors s_l , one for each field variable (default: 1). The error estimate for each field variable is divided by this factor.
- **Stability estimate derivative order** (only available when **Error estimate** is **L2 norm**). The L_2 norm error estimate is based on a stability estimate for the PDE. Use this field to specify its order.
- **Functional** (only available when **Error estimate** is **Functional**). Use this edit field to specify a globally available scalar-valued expression to be used for the error estimate.
- **Adjoint solution error estimate** (only available when **Error estimate** is **Functional**). Use this list to select error estimate method in the adjoint solution: a recovery technique and a gradient-based method. Select **On** to enforce using the recovery technique, and select **Off** to use the gradient-based method. Select **Automatic** to let the solver check if the geometry only contains Lagrange basis functions. If so, the adjoint solution uses the recovery technique. Otherwise, it uses the gradient-based method.
- **Weights for eigenmodes** (only available when using **Adaptive Mesh Refinement** together with the **Eigenvalue** solver). For eigenvalue problems, the error estimate is a weighted sum of the error estimates for the various eigenmode. Use this field to enter a space-separated list of positive (relative) weights. The default value of 1 means that all the weight is put on the first eigenmode.

Error Estimates for the Time-Dependent Solver

For time-dependent adaptive mesh refinement you need to specify a user-defined **Error indicator**. Use the edit field to give the error indicator function used for the adaptive mesh refinement. A possible error indicator is the L_2 norm of the gradient of the dependent variables (for example, `sqrt(mod1.Tx^2+mod1.Ty^2)` for the temperature in a 2D heat transfer model). The gradient of the dependent variable is the default value for the error indicator in most physics.

A solution on the coarse base mesh is computed in the next time interval, and the error indicator is evaluated at the points specified in the **Sample points** field. In this way a new adapted mesh appropriate for the next time interval can be generated. The sample points must be specified as a number between 0 and 1 because they are interpreted as being relative to the time interval under consideration. Entering a scalar value of 0.5 means that the error indicator is evaluated at the midpoint of the interval. The default value is `range(0.0,0.1,1.0)`, which gives 11 sample points from 0 to 1.

The L_2 Norm Error Estimate

The L_2 norm error estimate relies on an assumption of a strong stability estimate for the PDE problem (satisfied, for example, for Poisson's equation over a domain with a smooth boundary). From such an assumption, it is possible to show that there is a constant C , such that the L_2 norm of the error, e_l , in the l th equation satisfies

$$\|e_l\| \leq C \|h^{q_l} \rho_l\|$$

where ρ_l is the residual in the l th equation and q_l is the stability estimate derivative order. The adaptive solver algorithm uses the following L_2 -norm error indicator:

$$\left(\int_{\Omega} \sum_l s_l^{-2} h^{2q_l} |\rho_l|^2 dA \right)^{\frac{1}{2}}$$

with the default value $q_l = 2$. This formula also introduces the scaling factors s_l for the residual with the default value $s_l = 1$. The local error indicator for a mesh element is

$$\sum_l s_l^{-2} h^{2q_l} \tau_l^2 A$$

where A is the area (volume, length) of the mesh element, and τ_l is the absolute value of the l th equation residual (one number per mesh element).

The Functional Error Estimate

The functional-based estimate relies on adjoint solution error estimation. Instead of approximating the error of the solution, the adaptive solver uses the approximation of the error of a certain error functional (Ref. 11). Under rather general assumptions, it is possible to show that the error e (of a functional) can be estimated as

$$|e| \leq \sum_l \|e_l^*\| \|\rho_l\|$$

where e_l^* and p_l are the error in the dual or adjoint solution to, and the residual for, the l th equation, respectively. The adaptive solver algorithm uses the following error indicator for a mesh element:

$$\sum_l w_l \tau_l A$$

where A is the area (volume, length) of the mesh element, and τ_l is the absolute value of the l th equation residual (one number per mesh element). Here w_l is an estimate of the adjoint solution error for the l th equation. This error is estimated in one of two ways. For both methods the sensitivity solver finds the discrete adjoint solution. If only Lagrange element shape functions are used, the solver uses the *ppr* technique to compute w_l as an element average of $|pprint(u_l^*) - u_l^*|$. Here u_l^* is the current computed adjoint solution for the l th equation. If not only Lagrange-element shape functions are used, then $w_l = h D_l$ where D_l is an element average of $|\nabla u_l^*|$. The global error printed in the solver log is the sum of the error indicator for all mesh elements.

MESH REFINEMENT

Use the **Refinement method** list to control how to refine mesh elements. Select:

- **Longest** to make the solver refine only the longest edge of an element. (Not available for 1D geometries.)
- **Mesh initialization** to generate a completely new mesh. (Not available for time-dependent adaptive mesh refinement.)
- **Regular** to make the solver refine elements in a regular pattern. (Not available for 3D geometries.)

By default, the software automatically determines the order of decrease in equation residuals on basis of the shape function orders in the geometry. To specify a residual order manually, select the **Residual order** check box and specify a nonnegative integer in the accompanying field. This option is not available for time-dependent adaptive mesh refinement.

Use the **Element selection** list to specify how the solver should select which elements to refine. Select:

- **Rough global minimum** to minimize the L_2 norm of the error by refining a fraction of the elements with the largest error in such a way that the total number of elements increases roughly by the factor specified in the accompanying **Element growth rate** field. The default value is 1.7, which means that number of elements increases by about 70%.

- **Fraction of worst error** to refine elements whose local error indicator is larger than a given fraction of the largest local error indicator. Use the accompanying **Element fraction** field to specify the fraction. The default value is 0.5, which means that the fraction contains the elements with more than 50% of the largest local error.
- **Fraction of elements** to refine a given fraction of the elements. Use the accompanying **Element fraction** field to specify the fraction. The default value is 0.5, which means that the solver refines about 50% of the elements.

For time-dependent adaptive mesh refinement, you can specify the maximum number of refinements of the mesh elements (default: 2) in the **Maximum element refinements** field.

RESTART

This section is available for the **Time-Dependent Solver** node.

After each mesh adaption the time integration is restarted and you can control the following time-stepping properties:

By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.

Use the **Consistent initialization** list to control how the solver performs consistent initialization of differential-algebraic systems by selecting **Off** (the default), **On**, or **Backward Euler**.

OUTPUT

This section contains information about the solution and mesh that contain the results from the adaptive mesh refinement.

PLOT WHILE SOLVING

This section is available for the **Stationary Solver** and **Eigenvalue Solver** nodes.

To plot the adaptive mesh refinement, select the **Plot while solving** check box. You can then select any existing plot group from the **Plot group** list to use for the plot.

Advanced

The **Advanced** node () is an attribute node that handles advanced settings for solver nodes such as **Stationary Solver**, **Time-Dependent Solver**, and **Eigenvalue Solver**.

GENERAL

Matrix Symmetry

Use the **Matrix symmetry** list to control how the solver handles matrix symmetry of linear system matrices. Select:

- **Automatic** to perform automatic symmetry detection. Both symmetric and Hermitian matrices can be detected.
- **Hermitian** to override the automatic symmetry detection and force the solver to assume that matrices are Hermitian.
- **Nonsymmetric** to override the automatic symmetry detection and force the solver to assume that matrices are nonsymmetric.
- **Symmetric** to override the automatic symmetry detection and force the solver to assume that matrices are symmetric.

Which Problems are Symmetric?

When the discretization of a PDE problem results in a symmetric Jacobian (stiffness) matrix (and a symmetric mass matrix for time-dependent or eigenvalue problems), you can often apply faster and less memory-consuming algorithms to solve the resulting linear systems. PDEs with symmetric discretization typically occur in models involving acoustics, diffusion, electromagnetics, heat transfer by conduction, and structural mechanics. In contrast, problems in fluid mechanics, convection-diffusion, and convection-conduction typically involve nonsymmetric Jacobian matrices.

If the model involves complex numbers you can distinguish between *symmetric* and Hermitian matrices. A *Hermitian matrix* A satisfies

$$\bar{A}^T = A$$

where T denotes the transpose and the bar denotes the complex conjugate.

COMSOL Multiphysics detects symmetry for symmetric and Hermitian matrices. To take advantage of the computational savings for models with symmetric matrices is to use a solver that utilizes the symmetry. The following linear system solvers and preconditioners do not take advantage of symmetric matrices:

- The Vanka preconditioner

- The incomplete LU preconditioner
- The algebraic multigrid solver/preconditioner



Selecting **Symmetric** for a problem that does not result in symmetric matrices leads to an incorrect solution.

Additional Solver Options

Even if variables are well scaled, equations can have very different scales. The **Row equilibration** check box is selected to balance the equations using row equilibration. Even when this check box is selected, row equilibration is not used in the following situations in order to preserve matrix symmetry:

- Automatic matrix symmetry detection is used and the system matrices are symmetric
- **Symmetric** or **Hermitian** is selected in the **Matrix symmetry** list
- The conjugate gradients or geometric multigrid solver is used
- The eigenvalue solver is used.

Use the **Null-space function** list to select a method for the computation of matrices needed for constraint handling (see [Elimination Constraint Handling](#) below). Select:

- **Automatic** to let the software automatically determine the most appropriate method (**Orthonormal** or **Sparse**).
- **Orthonormal** to compute the needed matrices using singular value decomposition.
- **Sparse** to handle constraint matrices with nonlocal couplings using a sparse algorithm.

By default, the solver chooses the number of mesh elements that are processed together during the assembly process (the block size). Select the **Assembly block size** check box for manual specification of a block size.

Select the **Allow complex-valued output from functions with real input** check box to control whether the solver treats such complex-valued output as an error or not.

The **Stop when undefined mathematical operation is detected** check box controls how the solver handles undefined mathematical operations such as division by zero.

REASSEMBLY

Select the **Manual control of reassembly** check box to be able to override the solver mechanism that automatically detects which quantities need to be reassembled. This

can be useful to improve efficiency in situations when the automatic mechanism is too sensitive and reassembles quantities that do not need to be reassembled.

Reassembly of Loads

When the **Manual control of reassembly** check box is selected you can control reassembly of the residual vector with the **Constant load** list. Select **On** (the default) or **Off** to instruct the solver to perform a reassembly process for the computation of the residual vector (when **Off**) or not.

The load (residual vector) is constant if the PDE and the Neumann boundary conditions are linear with time-independent coefficients and right-hand sides. For the discretized model, this means that the residual vector L depends linearly on U ($L = L_0 - KU - D\dot{U} - E\ddot{U}$), and that L_0 , K , D , and the mass matrix E are constant.

Even though **Constant load** has been set to **Off** some of the quantities that make up the residual might still be constant. Manual control of reassembly of these quantities can be controlled with the **Constant Jacobian**, **Constant damping or mass**, and **Constant mass** check boxes.

Select the **Jacobian constant** check box if the Jacobian matrix K is time independent. You can also choose this option if you want to use the same Jacobian throughout the time-dependent or nonlinear solver. This choice cuts down linear-system factorization/preconditioning time but causes more iterations because the Newton iteration is degraded into a fixed-point iteration.

Select the **Constant damping or mass** check box if the coefficients of the first-order time-derivative terms or the second-order time-derivative terms are time independent (often the case). In the discretized model, this means that the damping (sometimes called mass) matrix D or the mass matrix E is constant.

Reassembly of Constraints

When the **Manual control of reassembly** check box is selected you can control reassembly of the constraint residual with the **Constant constraint** list. Select **On** (the default) or **Off** to instruct the solver to perform a reassembly process for the computation of the constraint residual vector (when **Off**) or not.

The constraint is constant if the Dirichlet boundary conditions (constraints) are linear and time independent. For the discretized model, this means that the constraint residual M depends linearly on U ($M = M_0 - NU$) and that M_0 and N are constant. It is also assumed that the constraint Jacobian N is correct.

Even though **Constant load** has been set to **Off** the constraint Jacobian might still be constant. You can control the reassembly of that quantity with the **Constant constraint Jacobian** check box.

Select the **Constant constraint Jacobian** check box if the Dirichlet boundary conditions are linear with time-independent coefficients (not right-hand side). For the discretized model this means that N is constant.

ELIMINATION CONSTRAINT HANDLING

The constraint handling is, for simplicity, demonstrated for a stationary problem. The handling is similar for parametric, eigenvalue, and time-dependent problems. Consider the linear (scaled) algebraic system

$$\begin{bmatrix} K & N_F \\ N & 0 \end{bmatrix} \begin{bmatrix} U \\ \Lambda \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix}$$

The Lagrange multiplier vector Λ is typically undetermined, and COMSOL Multiphysics does not solve for it. Similarly, the constraint $NU = M$ often contains the same equation several times. To handle this problem, COMSOL turns to a constraint-handling method that uses elimination. The solver computes a solution U_d to the constraint $NU = M$ as well as a matrix Null, whose columns form a basis for the null space of N . For unidirectional constraints ($N_F \neq N^T$) a matrix Nullf is also computed, whose columns form a basis for the null space of N_F^T . Then it obtains the solution as $U = \text{Null } U_n + U_d$. Here U_n is the solution of $K_c U_n = L_c$, where

$$\left\{ \begin{array}{l} K_c = \text{Nullf}^T K \text{ Null} \\ L_c = \text{Nullf}^T (L - KU_d) \end{array} \right.$$

Here K_c is the eliminated stiffness matrix.

For eigenvalue and time-dependent problems, the corresponding eliminated D and E matrices are

$$D_c = \text{Nullf}^T D \text{ Null}, \quad E_c = \text{Nullf}^T E \text{ Null}$$

This is an attribute node that handles parameters for linear system solvers/preconditioners that use the auxiliary space Maxwell solver (AMS). You can use an **AMS** node () together with the **Iterative**, **Krylov Preconditioner**, and **Coarse Solver** attribute nodes. Right-click any of these nodes to add an **AMS** node.

The AMS solver uses the auxiliary space Maxwell solver preconditioner from the Lawrence Livermore National Laboratory linear solver/preconditioner library *hypre*, a software library of high performance preconditioners and solvers (Ref. 4). AMS provides edge finite element discretization of variational curl-curl problem stemming from stationary or time-dependent Maxwell's equations. The version of AMS available in COMSOL Multiphysics is designed for the lowest-order edge elements. For higher-order discretizations, use it together with the geometric multigrid (GMG) solver with the option **Lower element order first** and sufficiently number of levels so that AMS can work efficiently as a coarse solver. For details, see Ref. 5.

The AMS node's settings window contains the following section:

GENERAL

Enter the number of iteration of the AMS solver in the **Number of iterations** field (default: 2).

In the **Variables** field, add the applicable dependent variables that use vector elements (such as magnetic scalar potential) and that you want to include in the AMS solver. Use the **Move Up** () , **Move Down** () , **Delete** () , and **Add** () buttons to configure the list of variables.

From the **Cycle type** list, select one of the available AMS cycle types 1–14 (the default is cycle type 1, a multiplicative solver that should work well in most cases; see Ref. 5 for details). These cycle types are various combinations of smoothing and applications of algebraic multigrid on decomposed problems.

From the **Magnetostatics** list, select **Automatic** (the default), **On**, or **Off**. The automatic case determines magnetostatics by comparing the maximum row sum of absolute values for the projected matrix $T^T A T$ and A . Here T is the discrete gradient matrix; see documentation for **SOR Vector**. Magnetostatics is deduced if the projected matrix is negligible compared to A . If magnetostatics is deduced or chosen, AMS skips the subspace corrections associated with the projected matrix $T^T A T$.

From the **Divergence cleaning** list, select **Automatic** (the default), **On**, or **Off**. The automatic case is the same as for the determination of magnetostatics. In the

magnetostatic case, AMS should skip corrections associated with $T^T AT$ and use divergence cleaning of the right-hand side. You can also manually specify the magnetostatics and divergence cleaning settings. This can be useful if divergence cleaning has already been made or if you suspect that the automatic detection fails.

Automatic Remeshing

This attribute node adds automatic remeshing. You can add an **Automatic Remeshing** node () together with the **Time-Dependent Solver** operation node. It can be used together with the Moving Mesh interface to assure a satisfactory mesh quality throughout the simulation.

GENERAL

The software only performs automatic remeshing in one geometry. Use the **Remesh in geometry** list to specify that geometry if the model contains more than one geometry.

CONDITIONS FOR REMESHING

From the **Condition type** list you can choose between three different types of conditions for when remeshing should occur:

- **Mesh quality** (the default). This means that the solver remeshes when the mesh quality becomes less than a given limit. You can change the mesh quality expression in the **Mesh quality expression** field and change the limit in the **Stop when mesh quality is below** field (default value: 0.2).
- **Distortion**. This means that the solver remeshes when the distortion the mesh has undergone becomes larger than a given limit. You can change the expression that defined the distortion in the **Distortion expression** field and change limit in the **Stop when distortion exceeds** field (default: 2).
- **General**. This means that the solver remeshes when a logical condition becomes true. You specify the condition in the **Stop when condition is true** field.

The **Remesh at** setting determines which previous solution is used for the remeshing:

- When **Last output from solver before stop** is selected, the remeshing is done on the last solution that would have been stored by the solver if remeshing would not have occurred. This setting discards any solver progress done since the last output.
- When **Last step taken by solver before stop** is selected (the default), the remeshing is done using the solution from the last solver step before the condition for remeshing became fulfilled. Only the very last solver step, at which the condition was triggered,

is discarded. Typically this setting is preferred because then the progress of the automatic remeshing does not depend on the solver's list of output times.

REMESH

After each remeshing the time integration is restarted and you can control the following time-stepping properties:

By default the solver chooses an initial step automatically. Select the **Initial step** check box for manual specification of an initial step.

Use the **Consistent initialization** list to control how the solver performs consistent initialization of differential-algebraic systems. Select **Off** (the default), **On**, or **Backward Euler**.

OUTPUT

This section contains information about which solution and meshes that contain the results from the automatic remeshing node.

Coarse Solver

The **Coarse Solver** node () is an auxiliary attribute subnode used by the **Multigrid** and **Domain Decomposition** attribute nodes. This node does not have any settings. Instead, its purpose is to administrate coarse grid solvers for a multigrid solver. To add a solver, right-click the **Coarse Solver** node.

Control Field

The **Control Field** node is an attribute node that handles settings for field variables that are acting as control variables. Control variables have a special status when using the Sensitivity or Optimization solver. Each control field has a separate **Control Field** node. This attribute is used together with the [Dependent Variables](#) node.

GENERAL

The **Field components** section display the variable names for the fields components.

Use the **Solve for this field** check box to control whether to use this variable when solving a sensitivity or optimization problem. For other parts of the solution process, the control fields are held fixed. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved

for its values is determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.



A variable can still be solved for despite not being stored in output and vice versa.

Control State

The **Control State** node is an attribute node that handles settings for state variables that are acting as control variables. Control (state) variables have a special status when using the Sensitivity or Optimization solver. Each control state has a separate **Control State** node. This attribute is used together with the **Dependent Variables** node.

GENERAL

The **State Components** section display the variable names for the states components.

Use the **Solve for this state** check box to control whether to use this variable when solving a Sensitivity or Optimization problem. For other parts of the solution process, the control variables are held fixed. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for its values is determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.



A variable can still be solved for despite not being stored in output and vice versa.

Direct

The **Direct** node () is an attribute node that handles settings for direct linear system solvers. You can use a **Direct** node () together with solvers such as **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver**. The attribute can also be used together

with the **Coarse Solver** attribute when using multigrid linear system solvers. An alternative to the direct linear system solvers is given by iterative linear system solvers which are handled via the **Iterative** attribute node. Several attribute nodes for solving linear systems can be attached to an operation node, but only one can be active at any given time.

GENERAL

Use the **Solver** list to select a linear system solver. Select:

- **MUMPS** (multifrontal massively parallel sparse direct solver), the default solver.
- **PARDISO** (parallel sparse direct solver).
- **SPOOLES** (sparse object oriented linear equations solver).

All linear system solvers above work on general sparse linear systems of the form $Ax = b$ and use LU factorization on the matrix A to compute the solution x . In doing so, they use a preordering algorithm that permutes the columns of A to minimize the number of nonzeros in the L and U factors. Popular preordering algorithms include Minimum degree, Nested dissection, and Multisection. The MUMPS and SPOOLES solvers run distributed when running COMSOL Multiphysics in distributed mode (on clusters, for example). All linear system solvers benefit from shared memory parallelism (multicore processors, for example); however, MUMPS do so to a slightly lesser extent than PARDISO and SPOOLES.

Linear System Solver Selection Guidelines

The physics in the model selects a default linear system solver that usually is appropriate for the problem type, at least for single-physics models. If the default solver does not perform well, use the following guidelines to choose a linear system solver.

- 1 Try the PARDISO direct solver.
- 2 Try the MUMPS direct solver.
- 3 If the solver still runs out of memory or is too slow, use one of the iterative solvers GMRES, FGMRES, or BiCGStab. Select a preconditioner according to the guidelines in the section about the iterative solver.
- 4 If the system is positive definite and real symmetric or Hermitian, try the conjugate gradients iterative solver, which is more memory-efficient and sometimes faster than GMRES, FGMRES, and BiCGStab. Select a symmetric preconditioner.
Alternatively, try the SPOOLES direct solver. It often uses less memory but is less numerically stable. SPOOLES is also slower.

Which Models Are Positive Definite?

A model with a real symmetric or Hermitian system matrix is often also *positive definite*, which means that a number of efficient linear system solvers are applicable. Further, the simple preconditioners SSOR, SOR, SORU, Jacobi (diagonal scaling), and the multigrid solvers benefit from a positive definite matrix. A real symmetric or Hermitian matrix is positive definite if all its eigenvalues are positive.

For stationary problems, the system matrix is the Jacobian (stiffness) matrix A . This means that stationary models in diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix.

For time-dependent problems, the system matrix is of the form $A + \sigma B + \sigma^2 C$, where B is the damping matrix, C is the mass matrix, and $\sigma > 0$ is inversely proportional to the time step (if $C = 0$, then B is often called the mass matrix). Because these matrices are often positive definite, time-dependent models in diffusion, electromagnetics, structural mechanics, and heat transfer by conduction usually have a positive definite system matrix.

For eigenvalue problems, the system matrix is of the form $A - \sigma B + \sigma^2 C$, where σ is the *shift*—that is, the number around which the software searches for eigenvalues (specified in the **Search for eigenvalues around** field); the default is 0). Because A , B , and C are usually positive definite, eigenvalue problems in acoustics, diffusion, electromagnetics, heat transfer by conduction, and structural mechanics usually have a positive definite system matrix if $\sigma \leq 0$.

Elliptic and Parabolic Models

The classes of elliptic and parabolic models include the positive definite models. For such models, the efficient multigrid preconditioners often perform well. A simplified definition of these classes reads as follows. A system of stationary or eigenvalue second-order PDEs is *elliptic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix. A system of time-dependent PDEs has a time derivative term of the form $d_a \dot{u}$, where the mass coefficient d_a is often a positive definite matrix and the e_a coefficient is 0. Such a system is *parabolic* if the second-order terms in the PDE give rise to a positive definite Jacobian matrix.

Stationary or eigenvalue models in acoustics, convection-diffusion, electromagnetics, heat transfer, and structural mechanics are usually elliptic. Likewise, time-dependent models in convection-diffusion, electromagnetics, and heat transfer are often parabolic. The Navier-Stokes equations, wave-type equations, or formulations involving weak constraints are neither elliptic nor parabolic.

The MUMPS Solver and Its Options

The MUMPS solver works on general systems of the form $Ax = b$ and uses several preordering algorithms to permute the columns and thereby minimize the fill-in. MUMPS is multithreaded on platforms that support multithreading and also supports solving on distributed memory architectures through the use of MPI. The code is written in Fortran 90. For further details about MUMPS, see [Ref. 1](#).

Use the **Memory allocation factor** field to tell the package how much memory is required. MUMPS estimates how much memory the unpivoted system requires. The **Memory allocation factor** is used to tell MUMPS how much more memory the pivoted system requires.

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Automatic** (automatically selected by the MUMPS solver).
- **Approximate minimum degree**.
- **Approximate minimum fill**.
- **Quasi-dense approximate minimum degree**.
- **Nested dissection**.

Use the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not. Clear the check box to turn off the weight matching strategy.

The **Use pivoting** list controls whether or not pivoting should be used (default: **On**).

When pivoting is used, use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold (default value: 0.1). This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column.

Select the **Out-of-core** check box to store matrix factorizations (LU factors) as blocks on disk rather than in the computer's internal memory. When the check box is selected you can control the maximum amount of internal memory allowed for the blocks in the **In-core memory (MB)** field. The default is to store the blocks in RAM and not on disk. The MUMPS out-of-core solver stores the LU factors on the hard drive. This minimizes the internal memory usage. The cost is longer solution times because it takes longer time to read and write to disk than using the internal memory. You can specify the temporary directory where MUMPS stores the LU factors using the `-tmpdir` switch; see the *COMSOL Installation Guide* for details.

The PARDISO Solver and Its Options

The PARDISO solver works on general systems of the form $Ax = b$. In order to improve sequential and parallel sparse numerical factorization performance, the solver algorithms are based on a Level-3 BLAS update, and they exploit pipelining parallelism with a combination of left-looking and right-looking supernode techniques.

PARDISO is multithreaded on platforms that support multithreading. On distributed memory architectures the solver settings are changed to corresponding MUMPS settings if needed. The code is written in C and Fortran. COMSOL uses the PARDISO version developed by Olaf Schenk and collaborators ([Ref. 3](#)), which is included with Intel MKL (Intel Math Kernel Library).

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Minimum degree**
- **Nested dissection**
- **Nested dissection multithreaded** the default. It performs the nested dissection faster when COMSOL runs multithreaded.

Use the **Scheduling method** list to select the scheduling method used when factorizing the matrix. The two-level method is usually faster when you have many cores:

- **Auto** (the default) Selects one of the two algorithms based on the type of matrix.
- **One-level**
- **Two-level**

Use the **Row preordering** check box to control whether the solver should use a maximum weight matching strategy or not (the default is to use such row preordering). Clear the check box to turn off the weight matching strategy.

Use the **Bunch-Kaufmann** check box to control whether to use Bunch-Kaufmann pivoting or not (the default is to not use such pivoting). Select the check box to use Bunch-Kaufmann pivoting.

Use the **Multithreaded forward and backward solve** check box to run the backward and forward solves multithreaded. Doing so mainly improves performance when you have many cores and the problem is solved several times such as in eigenvalue computations and iterative methods.

Use the **Pivoting perturbation** field to control the minimum size of pivots (the pivot threshold ε). To avoid pivoting, PARDISO uses a pivot perturbation strategy that tests the magnitude of the potential pivot against a constant threshold of $\varepsilon = \alpha |PP_{\text{MPS}}D_rAD_cP|_\infty$, where P and P_{MPS} are permutation matrices, D_r and D_c are

diagonal scaling matrices, and $|\cdot|_\infty$ is the infinity norm (maximum norm). If the solver encounters a tiny pivot during elimination, it sets it to $\text{sign}(l_{ii})\epsilon|PP_{\text{MPS}}D_rAD_cP|_\infty$. The perturbation strategy is not as robust as ordinary pivoting. In order to improve the solution PARDISO uses iterative refinements.

Select the **Out-of-core** check box to store matrix factorizations as blocks on disk rather than in the computer's internal memory. When the check box is selected you can control the maximum amount of internal memory allowed for the blocks in the **In-core memory (MB)** field. The PARDISO out-of-core solver stores the LU factors on the hard drive. This minimizes the internal memory usage. The price is longer solution times because it takes longer time to read and write to disk than using the internal memory. You can specify the temporary directory where PARDISO stores the LU factors using the `-tmpdir` switch; see the *COMSOL Installation Guide* for details.

The SPOOLES Solver and Its Options

The SPOOLES solver works on general systems of the form $Ax = b$ using the multifrontal method and direct LU factorization of the sparse matrix A . When the matrix A is symmetric or Hermitian, the solver uses an LDLT version of the algorithm, which saves half the memory. SPOOLES uses several preordering algorithms to permute the columns and thereby minimize fill-in. SPOOLES is multithreaded on platforms that support multithreading and also supports solving on distributed memory architectures through the use of MPI. The code is written in C. COMSOL uses SPOOLES version 2.2 developed by Cleve Ashcraft and collaborators ([Ref. 2](#)).

Use the **Preordering algorithm** list to select one of the following preorderings:

- **Best of ND and MS** (the best of nested dissection and multisection)
- **Minimum degree**
- **Multisection**
- **Nested dissection**

Use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold (default: 0.1). This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column.

ERROR

The accuracy of the solution of the linear system can be controlled by choosing the appropriate option from the **Check error estimate** list. **No** implies no error checking, whereas **Automatic** or **Yes** does. The error check asserts that the relative error times a

stability constant ρ is sufficiently small. Use the **Factor in error estimate** field to manually set ρ . By default, error checking is enabled with the **Automatic** option meaning that the main solver has the responsibility for error management. No exceptions are thrown by the direct solver unless you choose the **Yes** option. If an error occurs in the main solver, warnings originating from the error checking in the direct solver appear.

Domain Decomposition

The **Domain Decomposition** node (☞) is an attribute node for the **Iterative** solver. Use the Domain Decomposition node to set up a domain decomposition solver using a Schwarz method. Domain decomposition divides the modeling domain into subdomains where each PDE in the subdomains is easier to solve. The total solution is then obtained by iterating between the computed solutions for each domain using as boundary conditions the currently known solutions from the other domains. The domain decomposition solver is efficient for distributed memory systems (cluster computing) and as a more memory-efficient alternative to a direct solver for large problems.

THE DOMAIN DECOMPOSITION SOLVER

The domain decomposition solver or preconditioner is a memory-efficient iterative algorithm for large problems where other methods are infeasible. The basic idea of the iterative (spatial) domain decomposition is as follows. Consider an elliptic PDE over a domain D and a partition $\{D_i\}$ such that

$$D = \bigcup D_i$$

Instead of solving the PDE on the entire D at once, the algorithm solves a number of subdomain problems for each subdomain D_i . If the subdomain D_i is adjacent to a boundary, its boundary conditions are used. On the interfaces between subdomains certain natural transmission conditions arise. It is known (Ref. 19) that the solution to the set of subdomain problems is equivalent to the original problem formulated over D . The solution can be found by iteratively solving each subdomain problem with all other domains fixed. The details to how this is done leads to various domain decomposition methods. One class of methods is the *overlapping Schwartz method*. There, the partition $\{D_i\}$ is allowed to grow such that each subdomain has a small overlap with its neighboring domains. The size of the overlap is an important parameter that determines the convergence rate for the method. To further accelerate the convergence rate a coarse problem is used. The coarse problem solved on the entire D should yield an estimate of the solution to the full problem on D .

Two practical properties of this method are:

- Control of maximum memory consumption independent of problem formulation:
Only a small part of the problem needs to be discretized and solved for at once.
- Coarse-grained concurrency: Disjoint problems can be solved concurrently on different cluster nodes.

Four domain decomposition methods are implemented based on this method:
additive, multiplicative, hybrid, and symmetric Schwartz methods.

GENERAL

Select a **Solver**—**Multiplicative Schwarz**, **Additive Schwarz**, **Hybrid Schwarz**, or **Symmetric Schwarz**.

For each of the following domain decomposition settings, enter values as required:

- **Number of iterations.** The default is 1.
- **Number of subdomains.** The default is 2. The subdomain partition is created from an element partition on the solver level.
- **Maximum number of DOFs per subdomain.** The default is 100,000 DOFs. The solver tries to not create subdomains larger than this and increases the number of subdomains to fulfill the target. The lowest value accepted is 1000.
- **Maximum number of nodes per subdomain.** The default is 1. This option is only relevant in cluster computations. Each subdomain is then handled by the selected number of nodes.
- **Overlap size.** The default is 2. Each subdomain in the initial (nonoverlapping) partition is extended via the connectivity of the stiffness matrix in a recursive algorithm.

Hierarchy Generation Method

Select a **Hierarchy generation method** (see [Geometric Multigrid](#) for details for all methods except **None**):

- **Lower element order first (any)** (the default)
- **Coarse mesh and lower order**
- **Lower element order first (all)**
- **Coarse mesh**
- **Lower element order and refine (all)**
- **Lower element order and refine (any)**

- **Refine mesh**
- **Manual**
- **None.** If **None** is selected, no coarse mesh is used in addition to the fine mesh. This can lead to severe reduction in convergence rate but saves memory,

If **Lower element order first (any)**, **Coarse mesh and lower order**, **Lower element order first (all)**, or **Coarse mesh** are selected then:

- Enter a value or expression for the **Mesh coarsening factor**. The default is 2.
- Use the **Add** (), **Move Up** (), **Move Down** (), and **Delete** () buttons to add and organize geometries to use in the **Use hierarchy in geometries** list.
- Select the applicable check box or boxes—**Assemble on all levels** (selected by default) or **Keep generated multigrid levels**.

If **Lower element order and refine (all)**, **Lower element order and refine (any)**, or **Refine mesh** are selected then:

- Select a **Refinement method**—**Split longest side** or **Regular refinement**.
- Use the **Add** (), **Move Up** (), **Move Down** (), and **Delete** () buttons to add and organize geometries to use in the **Use hierarchy in geometries** list.
- Select or clear the **Assemble on all levels** check box if required (selected by default),

If **Manual** is selected:

- Use the **Add** (), **Move Up** (), **Move Down** (), and **Delete** () buttons to add and organize multigrid levels in the **Use multigrid levels** and **Assemble on multigrid levels** lists.

For all methods:

- Select the **Use subdomain coloring** check box (selected by default) to use a coloring technique that leads to more efficient computations for the multiplicative and symmetric methods because they require the global residual to be updated after each subdomain. The coloring technique gives each subdomain a color such that subdomains with the same color are disjoint and can be computed in parallel before the residual is updated.
- Select the **Recompute and clear subdomain data** check box if desired. This is a computationally expensive option. The subdomain problems are factorized for each iteration and then cleared from memory. Only use this option if your model is too big to fit into main memory causing memory swapping to disk.

Domain Solver

The **Domain Solver** node () is an auxiliary attribute subnode used by the **Domain Decomposition** attribute node. This node does not have any settings. Instead, its purpose is to administrate domain solvers for a domain-decomposition solver. To add a solver, right-click the **Domain Solver** node.

Field

The **Field** node () is an attribute node that handles settings for field variables. Each field variable has a separate **Field** node. This attribute is used together with the **Dependent Variables** node.

GENERAL

The **Field components** section display the variable names for the fields components. Also, when internal extra variables are used, these are displayed here as **Internal variables**.

Use the **Solve for this field** check box to control whether to solve for the field (variable) or not. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for its values is determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.



A variable can still be solved for despite not being stored in output and vice versa.

SCALING

Control the scaling of a variable with the **Method** list. Select:

- **Automatic** to get an automatically determined scaling.
- **From parent** to use the scaling type selected in the **Method** list in the **Scaling** section of the corresponding **Variables** operation node.

- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling.
- **None** to get no scaling.

If you choose **Manual**, specify the scale by entering a value in the **Scale** field.

Specifying a **Method** for a variable here overrides the **Method** selected in the **Scaling** section of the corresponding **Variables** operation node unless **From parent** is selected.

Fully Coupled

This is an attribute node that handles parameters for a fully coupled solution approach. The **Fully Coupled** node () uses a damped version of Newton's method or a double dogleg method, and you can use it together with the **Stationary Solver** and the **Time-Dependent Solver** (only Newton's method is available with the time-dependent solver). An alternative to the fully coupled approach is given by the segregated solver, which you control with the **Segregated** node. Although several **Fully Coupled** and **Segregated** attribute subnodes can be attached to an operation node, only one can be active at any given time.

THE DOUBLE DOGLEG METHOD

The double dogleg method is available for stationary problems. It is a Newton trust region method and is as such able to adjust the direction as well as the step length when solving the nonlinear equation $F(u) = 0$, $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

In order to apply the double dogleg method, consider the minimization of the quadratic model

$$m_k(s) = \frac{1}{2} \|F_k + F'_k s\|^2 = \frac{1}{2} F_k^T F_k + (F'_k F_k)^T s + \frac{1}{2} s^T F'_k F_k s$$

subject to $\|s\| \leq \delta_k$. Here the size of the step s is required to be bounded by the trust region radius δ_k . Both the Cauchy point—that is, the minimizer of m in the steepest descent direction—as well as the Newton point are utilized. In each iteration the algorithm dynamically adjusts the size of the trust region depending on the predicted decrease of m compared to the actual one. The double dogleg step is then given by a certain convex combination of the Cauchy step (steepest descent direction) and the Newton step. For difficult problems you can choose to start the computation by a damped Newton step. Enter the damping factor between 0 and 1 in the **Initial damping**

factor field. The algorithm terminates if the norm of the scaled residual is less than the given tolerance, $\|SF_k\| \leq \text{tol}$. You can choose the type of scaling in the **Residual scaling** list. See the *Method and Termination* settings below.

GENERAL

Use the **Linear solver** list to select a solver for linear systems that appear in the corresponding solver configuration. The available solvers are attribute nodes of the types **Direct** and **Iterative**.

METHOD AND TERMINATION

Use the **Nonlinear method** list to control which damping factor to use in the damped Newton iterations:

- Select **Automatic (Newton)** to let the solver automatically determine a damping factor in each iteration of Newton's method.
- Select **Constant (Newton)** to manually specify a constant damping factor that is used in all iterations of Newton's method.
- The option **Automatic highly nonlinear (Newton)** is similar to **Automatic (Newton)** with the difference that this option can make the solver more careful when solving highly nonlinear problems. Try this option if the solver does not converge with **Automatic (Newton)**.
- Select **Double dogleg** to use the double dogleg nonlinear solver. This solver is only available for stationary problems.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Nonlinear method** list are:

- **Initial damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify a damping factor for the first Newton iteration. Default value: $1.0 \cdot 10^{-4}$.
- **Minimum damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). Default value: $1.0 \cdot 10^{-8}$. Use this field to specify the smallest allowed damping factor.
- **Restriction for step-size update** (not available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify a factor (default: 10) that limits how much the damping factor is allowed to change in a Newton iteration. The damping factor can change up or down by at most this factor.
- **Use recovery damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). If a damping factor smaller than the **Minimum damping factor** is

required, the nonlinear solver terminates when **Off** is selected in this list. When **On** is selected, the nonlinear solver takes a Newton step using the constant damping factor in the **Recovery damping factor** field. The default setting **Automatic** is equivalent to **On** for stationary problems and **Off** for time-dependent problems. For stationary parametric continuation problems, **Automatic** corresponds to **On** when solving for the first parameter value and **Off** when solving for subsequent parameter values.

- **Damping factor** (only available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify a constant damping factor (default: 1) for Newton's method.
- **Jacobian update** (only available when **Nonlinear method** is set to **Constant (Newton)**). The available options in this list are **Minimal** (the default), **On every iteration**, and **On first iteration**. The option **On every iteration** computes a new Jacobian for all iterations of Newton's method. The options **Minimal** and **On first iteration** are identical when used together with a **Stationary** operation node. They compute a new Jacobian for the first iteration of Newton's method only. The option **On first iteration** computes a new Jacobian for each new nonlinear system of equations when used with a **Time-Dependent Solver** while **Minimal** reuses the Jacobian for several nonlinear systems whenever deemed possible.
- **Residual scaling** (only available when **Nonlinear method** is set to **Double dogleg**). The available options in this list are **Field-wise** and **Uniform**. The option **Field-wise** scales the equations based on the field-wise sizes of the initial residual. When the option **Uniform** is selected the algorithm terminates on the relative residual based on the initial residual.

Select the **Limit on nonlinear convergence rate** check box (only available when used together with a **Time-Dependent Solver** and when **Nonlinear method** is set to **Constant (Newton)**) to force the nonlinear solver to terminate as soon as the convergence is estimated to be too slow. Enter a limit on the convergence rate in the accompanying field.

Use the **Termination technique** list to control how the Newton iterations are terminated. Select:

- **Tolerance** to terminate the Newton iterations when the estimated relative error is smaller than a specified tolerance.

- **Iterations or tolerance** to terminate the Newton iterations when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.
 - **Iterations** to terminate the Newton iterations after a fixed number of iterations.
-



Iterations is available when **Nonlinear method** is **Constant (Newton)**.

- **Pseudo time-stepping** (only available for **Stationary Solver** and when **Nonlinear method** is set to **Constant (Newton)**). Use the list to enable pseudo time stepping. When enabled use the fields to specify regulator parameters: **Initial CFL number**, **PID regulator-Proportional**, **PID regulator-Derivative**, **PID regulator-Integrative**, and **Target error estimate**. See [About Pseudo Time Stepping](#).

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are:

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field to limit the number of Newton iterations. When the maximum number of iterations have been performed Newton's method is terminated even if the tolerance is not fulfilled.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field to modify the tolerance used for termination of the Newton iterations. The actual tolerance used is this factor times the value specified in the **Relative tolerance** field in the **General** sections of the **Stationary Solver** and **Time-Dependent Solver** nodes.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). Use this field to specify a fixed number of iterations to perform.

Use the **Termination criterion** list to control how the Newton iterations are terminated for stationary problems (not available when **Termination technique** is set to **Iterations**). Select:

- **Solution** to terminate the Newton iterations on a solution-based estimated relative error.

- **Residual** to terminate the Newton iterations on a residual-based estimated relative error.
- **Solution or residual** to terminate the Newton iterations on the minimum of the solution-based and residual-based estimated relative errors. Here a scalar **Residual factor** (default: 100) multiplying the residual error estimate should be given.

RESULTS WHILE SOLVING

Select the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **None** (the default) to plot no probe results.
- **Manual** to manually choose which probe results to plot.

Incomplete LU

This is an attribute node that handles parameters for linear system solvers/preconditioners that use incomplete LU factorization. You can use an **Incomplete LU** node () together with the **Iterative, Krylov Preconditioner, Presmoothening, Postsmoother**, and **Coarse Solver** attribute nodes. Right-click any of these nodes to add an **Incomplete LU** node.

The Incomplete LU preconditioner performs an incomplete LU factorization of the system matrix A . That is, it drops small elements during the column-oriented Gaussian elimination (see Ref. 13 and Ref. 14). Thus it saves memory, and the resulting factors L and U are approximate. The resulting preconditioner is an approximation to A . The preconditioner supports threshold drop, fill-ratio drop, and threshold pivoting. It can optionally respect the nonzero pattern in the original matrix. The preconditioner accepts matrices in symmetric and Hermitian format but expands these to full storage before factorization.

GENERAL

In this section you specify the properties of the incomplete LU preconditioner.

Select a **Solver**. Select:

- **Incomplete LU** (the default) to use a solver designed specifically for incomplete LU factorization.
- **SPOOLES** to use the sparse object-oriented linear equations solver SPOOLES.

Drop Using

For **Incomplete LU**, select an option from the **Drop using** list to specify a drop rule.

Select:

- **Tolerance** (the default) to let the solver drop (neglect) elements that have small enough absolute values. Tune the sizes of the neglected elements either in the **Drop tolerance** field or using the accompanying slide bar. A larger tolerance neglects more elements.
- **Fill ratio** to let the solver keep a certain fraction of the elements. The elements with largest absolute values are kept. Tune the number of nonzero elements in the incomplete factorization using either the **Fill ratio** field or the accompanying field. A smaller fill ratio neglects more elements.

See also [Selecting a Drop Rule](#) below.

Drop Tolerance

For both **Incomplete LU** and **SPOOLES**, use the **Drop tolerance** field or the accompanying slide bar to tune the maximum allowed sizes of dropped (neglected) elements (default: 0.01). A smaller drop tolerance means that the preconditioner drops fewer elements and so the preconditioner becomes more accurate. This leads to fewer iterations in the iterative solver, but memory requirements and preconditioning time increase. A larger drop tolerance means that the preconditioner drops more elements and so memory use and preconditioning time decrease. In this case, however, the preconditioner becomes less accurate, which leads to more iterations in the iterative solver, or, if the drop tolerance is too high, to no convergence at all. Often it is most efficient to use as high a drop tolerance as possible; that is, choose it so that the iterative solver barely converges.

Respect Pattern

For **Incomplete LU**, by default the solver never drops elements in positions where the original matrix is nonzero. Clear the **Respect pattern** check box to allow the solver to drop also such elements.

Pivot Threshold

For both **Incomplete LU** and **SPOOLES**, use the **Pivot threshold** field to enter a number between 0 and 1 that acts as pivot threshold (default: 1). This means that in any given column the algorithm accepts an entry as a pivot element if its absolute value is greater than or equal to the specified pivot threshold times the largest absolute value in the column. The solver permutes rows for stability. In any given column, if the absolute value of the diagonal element is less than the pivot threshold times the largest absolute value in the column, it permutes rows such that the largest element is on the diagonal. Thus the default value 1 means that it uses partial pivoting.

Number of Iterations and Relaxation Factor

For **Incomplete LU**—once the approximate factors L and U have been computed—you can use the incomplete LU factorization as an iterative preconditioner/smooth. Here, $M = (LU)/\omega$, where ω is a relaxation factor, and L and U are the approximate factors. Use the **Number of iterations** field to specify how many iterations to perform (default: 1). The relaxation factor ω is similar to the one used by, for example, the **SOR** node. Specify such a factor in the **Relaxation factor** field (default: 1).

Preordering Algorithm

For **SPOOLES**, use the **Preordering algorithm** list to select one of the following preorderings:

- **Nested dissection** (NS, the default)
- **Minimum degree**
- **Multisection** (MS)
- **Best of ND and MS** (the best of nested dissection and multisection)

SELECTING A DROP RULE

The Incomplete LU preconditioner uses the *threshold drop rule* (the default) or the *fill-ratio drop rule*. The preconditioner drops (neglects) an element during the elimination phase if its absolute value is smaller than the Euclidean norm of the entire column times a *drop tolerance*. In contrast, the fill-ratio drop rule limits the number of nonzeros in the incomplete factors L and U , and it keeps the largest absolute values. The number of values it keeps depends on the number of nonzeros in the

corresponding column of the original matrix times a fill-ratio factor. There are two exceptions to these drop rules:

- The preconditioner never drops diagonal elements.
- The preconditioner optionally drops nonzeros in positions where the original matrix is nonzero. To make the preconditioner drop them, clear the **Respect pattern** check box in the settings for the Incomplete LU preconditioner.

The primary problem with setting up a preconditioner is the trade-off between resources (computer time and memory) and the preconditioner's quality. The computational cost of setting up a preconditioner with the Incomplete LU preconditioner is at least proportional to the number of nonzeros in the produced factors L and U . An advantage of using the fill-ratio drop rule is that you can estimate and limit the cost beforehand; the main disadvantage is that the quality of the preconditioner is typically not as good as using the threshold drop rule with a drop tolerance resulting in the same number of nonzeros. However, with the threshold drop rule there is no good way of estimating resource requirements beforehand.

Furthermore, there is no general formula for these drop rules that gives a drop tolerance or fill ratio that guarantees fast convergence for a certain iterative method. Therefore it is often necessary to rely on experiments and experience for this difficult and, from a performance point of view, important choice.

Iterative

This is an attribute node that handles settings for iterative linear system solvers. You can use the **Iterative** node () together with operation nodes such as **Eigenvalue Solver**, **Stationary Solver**, and **Time-Dependent Solver**. An alternative to the iterative linear system solvers is given by direct linear system solvers, which are handled via the **Direct** attribute node. Although several attribute subnodes for solving linear systems can be attached to an operation node, only one can be active at any given time.

The iterative solver iterates until a relative tolerance is fulfilled. You specify this tolerance in the **Relative error** field of the operation node that this attribute belongs to.

GENERAL

Use the **Solver** list to select an iterative linear system solver. Select:

- **GMRES** to use the restarted GMRES (Generalized Minimum RESidual) iterative method.
- **FGMRES** to use the restarted FGMRES (Flexible Generalized Minimum RESidual) iterative method.
- **BiCGStab** to use the BiCGStab (BiConjugate Gradient Stabilized) iterative method.
- **Conjugate gradients** to use the conjugate gradients iterative method.
- **Use preconditioner** to use the currently active preconditioner attached to this **Iterative** node as the linear system solver. This solver uses the settings for **Factor in error estimate** and **Maximum number of iterations** from the **Error** section of the corresponding **Iterative** node.

It is possible to roughly order the linear system solvers according to their memory usage and computational time per iteration (with least memory and time first):

- 1 Conjugate gradients
- 2 BiCGStab
- 3 GMRES
- 4 FGMRES

The solvers that require less memory and computational time per iteration typically are less robust and not applicable to all problem types. See also [The Iterative Solvers](#) below.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Solver** list are:

- **Number of iterations before restart** (only available when **Solver** is set to either **GMRES** or **FGMRES**). Use this field to specify how many iterations the solver performs until it restarts (the default is 50). There is no guarantee that a restarted GMRES converges for a small restart value. A larger restart value increases the robustness of the interactive procedure, but it also increases memory use and computational time. For large problems, the computational cost to produce a preconditioner of such quality that the termination criteria are fulfilled for a small number of iterations and for a small restart value is often large. For those problems it is often advantageous to set up a preconditioner with a somewhat lesser quality and instead increase the restart value or iterate more steps. Doing so typically increases the condition

number for the preconditioned system, so an increase in the error-estimate factor might be needed as well

- **Preconditioning** (not available when **Solver** is set to **FGMRES**). Use this list to specify whether to precondition the linear system matrix from the **Left** or from the **Right**. Normally, this setting does not significantly influence the convergence behavior of the selected solver. The default choice is left preconditioning. Normally, the two versions of GMRES have similar convergence behavior (see Ref. 8). If the preconditioner is ill-conditioned there could, however, be differences in the behavior. For the conjugate gradient method this choice only affects the convergence criterion and not the algorithm itself.

Selecting a Preconditioner

When using an iterative linear system solver you must select a preconditioner. The choice of preconditioner affects the number of iterations and the solver's eventual convergence. Preconditioning can consume more time and memory than the actual iterative solver itself. To choose a preconditioner, right-click the **Iterative** node and choose one of the following preconditioners from the context menu:

TABLE 19-5: SELECTING A PRECONDITIONER

PRECONDITIONER	USAGE
Incomplete LU	For nonsymmetric systems (the default preconditioner).
Multigrid—Geometric multigrid	For elliptic or parabolic systems.
Multigrid—Algebraic multigrid	For scalar problems or loosely coupled multiphysics problems of the elliptic or parabolic type.
AMS—auxiliary space Maxwell solver	For curl-curl problems stemming from stationary or time-dependent Maxwell's equations.
Jacobi (diagonal scaling)	For large positive definite models.
SOR	For elliptic problems without zeros on the diagonal. Typically better than Jacobi and still rather inexpensive.
SOR Gauge	For ungauged vector element formulations of Magnetostatics.
SOR Line	For the same problem class as for SOR but adopted to stretched/anisotropic meshes (for example, boundary layer meshes). More expensive than SOR.
SOR Vector	For large electromagnetics models using vector elements.

TABLE 19-5: SELECTING A PRECONDITIONER

PRECONDITIONER	USAGE
Krylov Preconditioner	For Helmholtz problems where the mesh does not fulfill the Nyquist criteria. It can be used on the coarse multigrid level or as a smoother.
Vanka	For large indefinite problems with zeros on the diagonal of the system matrix.
SCGS	For fluid-flow problems with linear elements.
Domain Decomposition	For large problems in a distributed-memory system or as an alternative to a direct solver.

Each preconditioner has its own settings; to adjust them, select the preconditioner node to open its settings window. If you want to solve a model without a preconditioner, disable all preconditioner nodes.

The Incomplete LU preconditioner, which is the default preconditioner, works in a more general context than the others, but it might be impractical because of its time and memory requirements; when they work, the multigrid preconditioners are always preferable. The SOR and Jacobi diagonal-scaling preconditioners use less time and memory but only ensure convergence of the iterative solver for positive definite problems. Problems with zeros on the diagonal are efficiently preconditioned with the Vanka preconditioner. To precondition electromagnetic problems that use vector elements for a PDE containing the curl-curl operator, use the SOR Vector preconditioner.

For details about the individual preconditioners, follow the links in the table above.

Preconditioner Selection Guidelines

The physics selects a default preconditioner that is usually appropriate for the problem type, at least for single-physics models. If the default does not perform well, select another one using the following guidelines:

- If the system is elliptic or parabolic (see [Elliptic and Parabolic Models](#)) use the geometric multigrid preconditioner.
- If you solve a fluid-flow problem with linear elements only, try the SCGS preconditioner. This is the default setting for most fluid-flow physics.
- If you solve an indefinite problem with zeros on the diagonal of the system matrix, such as the Navier-Stokes equations, try the Vanka preconditioner or the geometric multigrid preconditioner with Vanka or Incomplete LU as the smoother. With appropriate stabilization, it is possible in many cases to use SOR or SOR Line as a

GMG pre- and postsmoother instead of Vanka, which increases performance. This is the default setting in some fluid-flow physics.

- If the system is positive definite but so large that the other preconditioners run out of memory, try the SOR Vector as smoother.
- If you solve an electromagnetics problem using vector elements for a PDE containing the curl-curl operator, try the geometric multigrid preconditioner with the SOR vector presmoother and the SOR vector postsmoother, or try the SOR vector preconditioner. Alternatively, if the problem is real-valued stationary or time-dependent you can try the geometric multigrid (GMG) preconditioner with the SOR presmoother and the SOR postsmoother, and AMS as the coarse solver. AMS is designed for the lowest-order vector elements. For higher-order discretizations use GMG with the option **Lower element order first** and sufficiently number of levels such that AMS could be used efficiently as a coarse solver.
- Try the Incomplete LU preconditioner, which works for all linear systems. However, it requires the tuning of the drop tolerance (or fill ratio); it can run out of memory; and in many cases it is not the most efficient preconditioner.
- If the system is elliptic or parabolic and the system is a real-valued PDE for a single solution component (that is, a scalar problem) you can alternatively try the algebraic multigrid preconditioner.
- As an alternative to the multigrid solver and the use of a direct solver, the Domain Decomposition solver can be a memory efficient alternative and is a scalable solver well suited for use in a distributed memory system.

The Incomplete LU preconditioner and sometimes the multigrid preconditioners require some tuning to get fast convergence without running out of memory (see the sections about these preconditioners).

ERROR

The stopping criteria for the iterative solvers are based on an error estimate, which checks if the relative residual times a stability constant ρ is less than a tolerance. This tolerance is specified in the **Relative error** field of the operation node that this attribute belongs to.

Use the **Factor in error estimate** field to set ρ , which serves as a safety factor to avoid premature termination of the iterations due to, for example, ill-conditioning of the

matrix A or poor preconditioning (default: 400). A larger value of ρ increases the number of iterations, but decrease the chance that the iteration is terminated too early.



For information about the convergence criteria used by the iterative solvers and the **Relative tolerance** and **Factor in error estimate** fields, see [Convergence Criteria for Iterative Solvers](#).

Use the **Maximum number of iterations** field to enter a maximum number of iterations that the iterative solver is allowed to take (default: 10,000 iterations). When this number of iterations has been performed without reaching the specified tolerance the solver stops with an error message.

By choosing the appropriate option from the **Validate error estimate** list the error estimate for left preconditioned solvers can be validated. **No** implies no error estimate validation, whereas **Automatic** or **Yes** does. By default the validation is enabled with the **Automatic option** meaning that it is performed but preconditioner warnings are only issued if the iterative solver fails with an error. This setting is propagated recursively and applies to all children with left preconditioning.

THE ITERATIVE SOLVERS

The following section provides more detailed information about the iterative solvers:

The GMRES Iterative Solver

This linear system solver uses the restarted GMRES (generalized minimum residual) method (see [Ref. 6](#) and [Ref. 7](#)). This is an iterative method for general linear systems of the form $Ax = b$. For fast convergence it is important to use an appropriate *preconditioner*

The FGMRES Iterative Solver

This solver uses the restarted FGMRES (flexible generalized minimum residual) method ([Ref. 9](#)). The solver is a variant of the GMRES solver that can handle a wider class of preconditioners in a robust way. You can, for example, use any iterative solver as preconditioner for FGMRES. The downside with the method is that it uses twice as much memory as GMRES for the same number of iterations before restart. FGMRES uses right preconditioning and therefore has the same convergence criterion as right-preconditioned GMRES. If FGMRES is used together with a constant preconditioner such as the Incomplete LU preconditioner, then the FGMRES solver is identical to the right preconditioned GMRES solver.

The Conjugate Gradients Iterative Solver

This solver uses the conjugate gradients iterative method ([Ref. 6](#), [Ref. 10](#), and [Ref. 11](#)). It is an iterative method for linear systems of the form $Ax = b$ where the matrix A is positive definite and (Hermitian) symmetric. Sometimes the solver also works when the matrix is not positive definite, especially if it is close to positive definite. This solver uses less memory and is often faster than the GMRES solver, but it applies to a restricted set of models.

For fast convergence it is important to use an appropriate *preconditioner*, which should be positive definite and (Hermitian) symmetric.

The BiCGStab Iterative Solver

This solver uses the biconjugate gradient stabilized iterative method ([Ref. 6](#) and [Ref. 12](#)) for solving general linear systems of the form $Ax = b$. The required memory and the computational time for one iteration with BiCGStab is constant; that is, the time and memory requirements do not increase with the number of iterations as they do for GMRES. BiCGStab uses approximately the same amount of memory as GMRES uses for two iterations. Therefore, BiCGStab typically uses less memory than GMRES.

The convergence behavior of BiCGStab is often more irregular than that of GMRES. Intermediate residuals may even be orders of magnitude larger than the initial residual, which can affect the numerical accuracy as well as the rate of convergence. The iterations are restarted with the current solution as initial guess if the algorithm detects poor accuracy in the residual or the risk for stagnation.

In contrast to GMRES and conjugate gradients, BiCGStab uses two matrix-vector multiplications each iteration. This also requires two preconditioning steps in each iteration. Also, when using left preconditioned BiCGStab, an additional preconditioning step is required each iteration. That is, left preconditioned BiCGStab requires a total of three preconditioning steps in each iteration.

CONVERGENCE CRITERIA FOR ITERATIVE SOLVERS

When you use an iterative solver COMSOL estimates the error of the solution while solving. Once the error estimate is small enough, as determined by the convergence criterion

$$\rho |M^{-1}(b - Ax)| < \text{tol} \cdot |M^{-1}b| \quad (19-9)$$

the software terminates the computations and returns a solution. When you use a direct solver COMSOL can optionally make a check to determine if the above

convergence criterion is fulfilled after the solution step. If the error criterion is not met, the solution process is stopped and an error message is given.

The definitions of M for the various solvers are:

- For MUMPS, PARDISO, and SPOOLES, $M = LU$, where L and U are the LU factors computed by the solver.
- When using left-preconditioning with the iterative solvers GMRES, conjugate gradients, and BiCGStab, M is the preconditioner matrix.
- For the remaining iterative solvers, M is the identity matrix.

The convergence criterion in [Equation 19-9](#) states that the iterations terminate when the relative (preconditioned) residual times the factor ρ is less than a tolerance tol . For solvers where M is equal to the identity matrix, the iterations can sometimes terminate too early with an incorrect solution if the system matrix A is ill-conditioned. For solvers where M is not equal to the identity matrix, the iterations can sometimes terminate too early if M is a poor preconditioner. If the iterations terminate too early due to an ill-conditioned system matrix or a poor preconditioner, increase the factor ρ to a number of the order of the condition number for the matrix $M^{-1}A$. If ρ is greater than the condition number for the matrix $M^{-1}A$, the convergence criterion implies that the relative error is less than tol : $|x - A^{-1}b| < \text{tol} \cdot |A^{-1}b|$.

Jacobi

This is an attribute node that handles settings for the **Jacobi** () (or diagonal scaling) method. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute nodes.

The Jacobi method provides a simple and memory-efficient solver/preconditioner/smoothening based on classical iteration methods for solving a linear system of the form $Ax = b$. Given a relaxation factor ω (usually between 0 and 2), a sweep of the Jacobi (diagonal scaling) method transforms an initial guess x_0 to an improved approximation $x_1 = x_0 + M^{-1}(b - Ax_0)$, where $M = D/\omega$, and D is the diagonal part of A .

GENERAL

Settings When Not Used With Coarse Solver

Enter the **Number of iterations** to perform when this node is used as a preconditioner or smoother. This setting is not considered when the attribute is used as a linear system solver (with the **Use preconditioner** option in the **Solver** list of the **Iterative** attribute node). The solver then iterates until the relative tolerance specified by the

corresponding operation node is fulfilled rather than performs a fixed number of iterations.

Enter a **Relaxation factor** to specify a scalar relaxation factor ω . The allowed values of this factor are between 0 and 2. The default is 1.

Settings When Used with a Coarse Solver

If this node is used with a **Coarse Solver**, select a **Termination technique** to determine how to terminate the solver. Select **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used, or **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.

If **Fixed number of iterations** is selected enter a value for each of the following:

- **Number of iterations** to perform. The default is 10.
- **Relaxation factor** to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. The default is 1.

If **Use tolerance** is selected, enter a value for each of the following:

- **Relative tolerance** to specify the termination tolerance. The default is 0.1.
- **Maximum number of iterations** that the solver is allowed to take. When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver is automatically stopped with an error message.
- **Relaxation factor** to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. The default is 1.

Krylov Preconditioner

This is an attribute node that handles settings for Krylov-type methods. You can use the **Krylov Preconditioner** node () together with the **Iterative**, **Presmoothening**, **Postsmoother**, and **Coarse Solver** attribute nodes. This node contains settings similar to the ones for the **Iterative** attribute node. The difference is that this node is an auxiliary attribute node whereas **Iterative** handles settings for a main iterative solver. This node can also be used with another **Krylov Preconditioner** attribute node.

GENERAL

Select a linear system **Solver**. Select:

- **GMRES** to use the restarted Generalized Minimum RESidual iterative method.
- **FGMRES** to use the restarted Flexible Generalized Minimum RESidual iterative method.
- **BiCGStab** to use the BiConjugate Gradient Stabilized iterative method.
- **Conjugate gradients** iterative method.



These solvers can roughly be ordered according to their memory usage and computational time per iteration (with least memory and time first): Conjugate gradients, BiCGStab, GMRES, and then FGMRES. The solvers that require less memory and computational time per iteration typically are less robust and not applicable to all problem types.

Number of Iterations before Restart

When **GMRES** or **FGMRES** is selected as the **Solver**, enter a **Number of iterations before restart** to specify how many iterations the solver should take between each restart. A larger number increases robustness but also memory use and computational time.

Preconditioning

When **GMRES**, **BiCGStab**, or **Conjugate gradients** are selected as the **Solver**, select an option from the **Preconditioning** list to specify whether to precondition the linear system matrix from the **Left** or from the **Right**. Normally, this setting does not significantly influence the convergence behavior of the selected solver.

Termination Technique

For all **Solver** types, and to determine how to terminate the solver, select an option from the **Termination technique** list: **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the solver is used, **Use tolerance** to terminate the solver when a tolerance is fulfilled, or **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first. For the two latter cases, enter a **Relative tolerance** (default: 0.1).

Number of Iterations

For all **Solver** types, enter a **Number of iterations**. The default is 10 or, when a tolerance is used, 500.

Maximum Number of Iterations

When **Use tolerance** is selected as the **Termination technique**, enter a **Maximum number of iterations** that the solver is allowed to take. When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field, the solver is automatically stopped with an error message.

ERROR

Select an option from the **Validate error estimate** list—**Automatic** (the default), **Yes**, or **No**. By choosing the appropriate option from the **Validate error estimate** list the error estimate for left preconditioned solvers can be validated. **No** implies no error estimate validation, whereas **Automatic** or **Yes** does. By default the validation is enabled with the **Automatic option** meaning that it is performed but preconditioner warnings are only issued if the iterative solver fails with an error. This setting is propagated recursively and applies to all children with left preconditioning.

Lower Limit

The **Lower Limit** node () makes it possible to impose restrictions on degrees of freedom (field variables). These restrictions are checked after each substep of a segregated approach. The restriction is performed without any regards to the equations, so use this with care. Right-click a **Segregated** node () to add a **Lower Limit** node.

LOWER LIMIT

By default, no variables have active lower limits. To activate lower limits for field variables, use the **Lower limit (field variable)** field to specify the variables and their scalar lower limits as space-separated pairs: *field_variable_1 limit_value_1* *field_variable_2 limit_value_2*, and so on. For example, to impose a lower limit of 0.25 for the value of a field **u** in Model 1, type `mod1.u 0.25`.

Lumped Step

The **Lumped Step** node () is available when using the **Segregated** attribute node. This step is intended for speeding up the computation of any L_2 -projections, stemming from the identity operator, appearing as single physics within a multiphysics problem. Using the lumped step approximates the Jacobian matrix resulting from finite element discretization when solving the linear system for the unknown field variables. The approximation consists of replacing the Jacobian by a diagonal matrix

with row sums. In doing so, take care to ensure that the resulting approximate Jacobian is invertible (nonsingular).

Multigrid

The **Multigrid** solver () is used to set up a geometric multigrid (GMG) solver or an algebraic multigrid (AMG) solver. You can use the **Multigrid** solver together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute nodes. Right-click any of these nodes to add a **Multigrid** solver. For more information about the multigrid solvers and the multigrid algorithms, see [The Geometric Multigrid Solver/Preconditioner](#), [The Algebraic Multigrid Solver/Preconditioner](#), and [The Multigrid Algorithm](#) below.

GENERAL

Select a **Solver**—**Geometric multigrid** or **Algebraic multigrid**. For both options, enter the **Number of iterations** and select a **Multigrid cycle**—**V-cycle** (the default), **W-cycle**, or **F-cycle**.



The **Multigrid cycle** settings are the same as for the geometric multigrid (GMG) and algebraic multigrid (AMG) solvers.

Geometric Multigrid

The geometric multigrid solver uses a hierarchy of *multigrid levels* where each level corresponds to a mesh and a choice of shape functions. Thus, in addition to coarsening the mesh it is possible to construct a new “coarser” level by lowering the order of the shape functions. The number of degrees of freedom decreases when you go to a coarser multigrid level. The meshes for the different levels can be constructed manually or automatically. The automatic options use a coarsening algorithm to the fine mesh, which leads to meshes that are not aligned to each other. There is also an option to generate the finer meshes from the coarsest mesh by successive mesh refinements, which leads to aligned (nested) meshes. The manual option can be useful when you have a quadrilateral, hexahedral, or prism mesh, or when you for some other reason want to control details in the meshes. Select one of the following options from the **Hierarchy generation method** list to specify how to generate the multigrid levels:

- **Coarse mesh and lower order.** Combines lowering of the used shape function order and a coarsening of the mesh.

- **Lower element order first (all)**. Generates first a number of levels by lowering the order (by one) of all the used shape functions. If this is not possible the mesh is coarsened.
- **Lower element order first (any)**. The default method. Generates first a number of levels by lowering the order (by one) of the used shape functions. If there are no shape functions that can be lowered, the mesh is coarsened.
- **Coarse mesh**. Does not change the order.
- **Lower element order and refine (all)**. Generates a number of levels by lowering the order (by one) of all the used shape functions. If this is not possible the mesh is refined a number of times. The mesh solved for can with this method be a finer one than the one selected under the study node.
- **Lower element order and refine (any)**. Generates a number of levels by lowering the order (by one) of the used shape functions. If there are no shape functions that can be lowered, the mesh is refined. The mesh solved for can with this method be a finer one than the one selected under the Study node.
- **Refine mesh**. Does not change the order.
- **Manual**. Use this setting to select multigrid levels from the existing ones. You then specify the multigrid levels to use in the **Use multigrid levels** list and the multigrid levels to assemble the discrete differential operators on in the **Assemble on multigrid levels** list. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (+) buttons to configure the lists of multigrid levels.



See [Multigrid Level](#) for more information about multigrid levels.

For all hierarchy generation methods except Manual, additional settings are available:

- Enter the **Number of multigrid levels** to generate (the default is 1).
- In the **Use hierarchy in geometries** list, select the geometries to apply multigrid to. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (+) buttons to configure the list of geometries.
- The **Assemble on all levels** check box is selected by default to assemble the discrete differential operators on all meshes. Otherwise these operators are formed using the restriction and prolongation operators. Click to clear the check box as required.

When **Coarse mesh and lower order**, **Lower element order first (all)**, **Lower element order first (any)**, or **Coarse mesh** are selected from the **Hierarchy generation method** list:

- Select the **Keep generated multigrid levels** check box to save the meshes for all multigrid levels under the mesh node.
- Enter a **Mesh coarsening factor** to select the degree of coarsening to apply to the meshes when using mesh coarsening as multigrid hierarchy generation method. The higher this number is the more aggressive the mesh coarsening is. The default is 2.

When **Lower element order and refine (all)**, **Lower element order and refine (any)**, or **Refine mesh** are selected from the **Hierarchy generation method** list, select a **Refinement method** to refine the multigrid levels when using mesh refinement as multigrid hierarchy generation method. The options are:

- **Split longest side**. The default method. Elements are subdivided such that the longest side in each element is always split. This yields not so many new elements, while also preserving mesh quality.
- **Regular refinement**. Elements are subdivided in a regular manner.

Algebraic Multigrid

If **Algebraic multigrid** is selected, the following additional settings control the automatic construction of the multigrid hierarchy:

- Enter the **Number of multigrid levels** to select how many multigrid levels to generate. The default is 5. Also enter a **Maximum number of DOFs at coarsest level**. The default is 5000. Coarse levels are added until the number of DOFs at the coarsest level is less than the max DOFs at coarsest level or until it has reached the number of multigrid levels.
- Enter a value or use the slider to set the **Quality of multigrid hierarchy**. Higher quality means faster convergence at the expense of a more time consuming set-up phase. For instance, if the linear solver does not converge or if it uses too many iterations, try a higher value to increase the accuracy in each iteration, meaning fewer iterations. If the algebraic multigrid algorithm runs into memory problems, try a lower value to use less memory. The range goes from 1 to 10 where 10 gives the best quality. The default is 3.

THE GEOMETRIC MULTIGRID SOLVER/PRECONDITIONER

The geometric multigrid solver (GMS) or preconditioner is a fast and memory-efficient iterative method for elliptic and parabolic models. It performs one or several cycles of the geometric multigrid method. The classical multigrid algorithm

uses one or several auxiliary meshes that are coarser than the original (fine) mesh. The idea is to perform just a fraction of the computations on the fine mesh. Instead, it performs computations on the coarser meshes when possible, which leads to fewer operations. The size of the extra memory used for the coarser meshes and associated matrices is comparable to the size of the original data. This leads to an iterative algorithm that is both fast and memory efficient. See [Ref. 15](#) for more information.

THE ALGEBRAIC MULTIGRID SOLVER/PRECONDITIONER

The algebraic multigrid solver (AMG) or preconditioner performs one or several cycles of the algebraic multigrid method. This is similar to the geometric multigrid algorithm, the difference being that it constructs the multigrid levels directly from the finest-level system matrix A_0 . That is, it constructs the prolongations P_i from A_0 without using auxiliary meshes. It constructs the coarse level matrices A_i from A_0 with the Galerkin projection method. The advantage is that you need not bother about the coarse multigrid levels. The disadvantages are twofold:

- Algebraic multigrid does not work well for vector-valued PDEs in COMSOL’s implementation. That is, it handles only scalar PDEs.
- COMSOL’s implementation does not support complex-valued system matrices.

THE MULTIGRID ALGORITHM

To describe the multigrid algorithm, assume that you have $N + 1$ multigrid levels numbered from 0 to N , where 0 is the finest level (the level for which you seek the solution). To solve the linear system $A_0x = b$ (corresponding to level 0), the algorithm must reform the system matrices A_1, \dots, A_N for the coarse multigrid levels. It must also compute the *prolongation matrices* P_i that map a solution x vector on level i to the corresponding solution vector $P_i x$ on the next finer level $i - 1$.

The prolongation matrices are constructed using plain interpolation from one multigrid level to the other. The system matrices for the coarse levels can be constructed in two ways:

- By assembling A_i on the mesh of level i (the default method).
- By projection from the finer level: $A_i = P_i^T A_{i-1} P_i$. This is also called the Galerkin method. It typically leads to more nonzero elements in the system matrix A_i , but the convergence can be faster than in the default method.

The following algorithm describes one multigrid cycle:

- | The input to the algorithm is some initial guess x_0 to the solution of the system $A_0x = b$.

- 2 Starting with x_0 , apply a few iterations of a *presmoothening* to the linear system $A_0x = b$, yielding a more accurate iterate x_{0s} . Typically the presmoothening is some simple iterative algorithm such as SOR, but you can also choose any iterative solver.
- 3 Compute the residual $r_0 = b - A_0x_{0s}$. The presmoothening “smooths” the residual so the oscillations in r have such a long wavelength that they are well resolved on the next coarser level (1). Therefore, project the residual onto level 1 by applying the transpose of the prolongation: $r_1 = P_1^T r_0$.
- 4 If $N = 1$ use the *coarse solver* to solve the system $A_1x_1 = r_1$. The coarse solver is typically a direct solver such as MUMPS. The number of degrees of freedom on level 1 is less than for level 0, which means that solving $A_1x_1 = r_1$ is less expensive. If instead $N > 1$, solve the system $A_1x_1 = r_1$ (approximately) by recursively applying one cycle of the multigrid algorithm for levels 1, 2, ..., N . In both cases the obtained solution x_1 is called the *coarse grid correction*.
- 5 Map the coarse grid correction to level 0 using the prolongation matrix:

$$x_{0c} = x_{0s} + P_1x_1.$$
- 6 Starting with x_{0c} , apply a few iterations of a *postsmoother* to the linear system $A_0x = b$, yielding a more accurate iterate x_{0mg} . The default postsmoother is SORU (the version of SOR using the upper triangle of the matrix). The iterate x_{0mg} is the output of the multigrid cycle.

The cycle just described is called the V-cycle. The recursive call in step 4 (when $N > 1$) is also a V-cycle. For the W-cycle and the F-cycle, steps 1–6 above are the same but with the twist that the recursive call in step 4 is substituted with two multigrid calls for the coarser levels. For the W-cycle these two calls are recursive calls (W-cycle calls). For the F-cycle the first call is a W-cycle and the second a V-cycle.

For only two multigrid levels ($N = 1$) these cycles are the same because the algorithm uses the coarse solver in step 4. Also the amount of work on the finest level is the same for the different cycles. Normally the V-cycle is sufficient, but the W-cycle and the F-cycle can be useful for more difficult problems.

When using multigrid as a preconditioner, the action of this preconditioner is obtained by applying a fixed number of multigrid cycles. When using multigrid as a solver, the multigrid cycle repeats until it reaches convergence.

When using multigrid as a preconditioner for the conjugate gradients method for a symmetric matrix A , the preconditioning matrix M should also be symmetric. This requirement is fulfilled if the matrices M associated with the presmoothening and the postsmoothing are transposes of each other. For instance, this is the case if the

presmoother is SOR and the postsmoother is SORU, and if the same number of smoothing steps is used. This combination with two smoothing steps is the default.

Notes on the Efficiency of Smoothers

COMSOL performs smoothing on all but the coarsest multigrid level. A smoother should be computationally cheap and effective at reducing the part of the error that has a high spatial frequency on the mesh to which it is applied. Therefore, applying a smoother on several meshes with a hierarchy of mesh sizes results in a more efficient solver than if the smoother were applied only on the finest mesh.

The efficiency of the multigrid method with simple iterations as a smoother (such as the Jacobi and SOR iteration) hinges on the ellipticity of the underlying mathematical problem. For Helmholtz problems originating from an equation

$$-\nabla \cdot \left(\frac{1}{a} \nabla u \right) - \omega^2 u = f$$

or

$$\nabla \times \left(\frac{1}{a} \nabla \times \mathbf{E} \right) - \omega^2 \mathbf{E} = \mathbf{F}$$

the obtained linear problem is indefinite for large frequencies ω . For these problems, a simple iteration amplifies smooth eigenmodes if the mesh is too coarse and makes these methods unsuitable as smoothers. To determine when to use a simple iteration, apply the *Nyquist criterion*

$$h_{\max} < \frac{\lambda}{2} = \frac{\pi}{\omega \sqrt{a}}$$

which says that the mesh must have at least two mesh elements per wavelength. Thus, when using the geometric multigrid solver for these type of problems, ensure that this criterion is fulfilled on the coarsest mesh if simple iterations are used as a smoother. In situations where the criterion is not fulfilled on coarse meshes GMRES can be a suitable smoother (Ref. 18). However, this setting makes smoothers on all levels more expensive and might not always pay off compared to choosing a coarse grid that satisfies the Nyquist criterion. Note also that a smoother based on a Krylov preconditioner like GMRES requires the (outer) iterative solver to be set to FGMRES.

The **Parametric** node ( ) is an attribute node that handles settings for parameter stepping using a parametric solver. This node can be used together with a **Stationary Solver** node.

THE PARAMETRIC SOLVER ALGORITHM

The parametric solver performs a loop around the usual stationary solver in which it estimates the initial guess using the solution for the previous parameter value. If the nonlinear solver does not converge and you are solving for a single parameter, it tries a smaller parameter step; COMSOL Multiphysics determines the size of this step on the basis of the convergence speed of the Newton iteration using step-size selection criteria based on work in [Ref. 9](#).

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to modify the parameter table and sweep type.

Use the **Sweep type** list to specify the type of sweep to perform. The **Specified combinations** type (the default) solves for a number of given combination of values, while the **All combinations** type solves for all combination of values. Using all combinations can lead to a very large number of solutions.

Use the **Parameter names** and **Parameter value list** table to specify parameter names and values for the parametric solver. Use the **Add** button () to add a row to the table. Each row has one parameter name and a corresponding parameter value list. For the **Specified combinations** sweep type, the list of values must have equal length. When you click in the **Parameter value list** column to define the parameter values you can click the **Range** button () to define a range of parameter values.

Exactly how the parameter values are used by the solver is determined by the option **Parameters to store** in the **Output** section as described below. If more than one parameter name have been specified the lists of parameter values is interpreted as follows: Assume that the parameter names are p1 and p2, and that p1 has the list 1 3 and p2 has the list 2 4. For the **Specified combinations** sweep type, the solver first uses p1 equal to 1 and p2 equal to 2. Thereafter, it uses p1 equal to 3 and p2 equal to 4. And when the sweep type in **All combinations**, the solver uses the following order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. You can use the **Load from file** button () to browse to such a text file. The read names and values are appended to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values, and a space separating the values. Click the **Save to file** button () to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®).

Use the **Predictor** list to control how the initial value for the next parameter value is determined. Select:

- **Automatic** (the default) to let the parametric solver choose a constant or linear predictor based on the type of stationary solver (a constant predictor for segregated solvers and a linear predictor for fully coupled solvers).
- **Constant** to use the solution for the present parameter value as initial guess.
- **Linear** to compute the initial guess by following the tangent to the solution curve at the present parameter value.



This option is overridden and **Constant** used instead if you are solving for more than one parameter (that is, when you have entered more than one parameter name in the **Parameter names** field).

LOAD CASES

Select the **Define load cases** check box to define load cases as combinations of defined load groups, multiplied with optional weights (load factors), and constraint groups. Load cases are useful for efficiently solving for a number of cases with varying loads (and constraints) for the same model without the need to reassemble the stiffness matrix. Use the **Move Up** (), **Move Down** (), **Delete** (), and **Add** () buttons to make the list contain the load cases that you want to solve for. For each load case, click in the column for the load groups and constraint groups that you want to include in the load case. By default, no load groups and constraint groups are included (). Load groups and constraint groups that are included appear with a check mark (). Optionally, change the default weights for the load groups from 1.0 to another value in the corresponding **Weight** column (which is to the right of the load group that it is acting on). A weight of 1.5, for example, adds an extra 50% to the magnitude of the loads in the load group; a weight of -1 reverses the direction of the loads.

TUNING

By default the solver selects the parameter steps automatically based on the values entered in the **Parameter values** field in the **General** section. To be able to modify the default behavior select the **Tuning of step size** check box. Then the following options become available. Use the:

- **Initial step size** field to enter a positive number that determines the magnitude of the first parameter step.
- **Maximum step size** field to specify an upper bound on the parameter step size. Use this if you suspect that the solver tries to take unnecessarily long steps.
- **Minimum step size** field to specify a safeguard against too small parameter steps.

Use the **Use initial damping factor for all parameter steps** list to control the initial damping factor for the nonlinear solvers for the parameter steps. Select **On** to use the given **Initial damping factor** for the nonlinear solvers for all parameter steps. Select **Off** to use the initial damping factor only for the first parameter step. Select **Automatic** (default) to use the initial damping factor only for the first parameter step when the **Automatic (Newton)** or **Automatic highly nonlinear (Newton)** nonlinear solver method is used in the **Fully Coupled** solver node and use the initial damping factor in all steps for other solver combinations.

OUTPUT

Use the **Parameters to store** list to control at what parameter values the solver stores a solution. Select:

- **Steps given** to store solutions at the parameter values entered in the **Parameter values** field in the **General** section.
- **Steps taken by solver** to store solutions at all parameter values where the solver has computed a solution. This option can generate solutions in-between the values specified by the **Parameter values** field in the **General** section if the solver needs to take shorter steps than specified by the values in that field.

Select the **Store solution out-of-core** check box if you want the output solution to be stored on disk rather than in the computer's internal memory.

RESULTS WHILE SOLVING

Select the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select:

- **All** to plot all probe results.
- **Manual** to manually choose which probe results to plot.

CLUSTER SETTINGS

Select the **Distribute parameters** check box to distribute the parameters on several computational nodes. If the problem is too large to run on a single node you can enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a nondistributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

Postsmoother

The **Postsmoother** node () is an auxiliary attribute node used by the **Multigrid** attribute node. This attribute does not have any settings. Instead, its purpose is to administrate postsmoothers for a multigrid solver. To add a postsmoother, right-click the **Postsmoother** node.

Presmoother

The **Presmoother** node () is an auxiliary attribute node used by the **Multigrid** attribute node. This attribute does not have any settings. Instead, its purpose is to administrate presmothers for a multigrid solver. To add a presmoother, right-click the **Presmoother** node.

Previous Solution

The **Previous Solution** node () is an optional attribute node of the **Parametric** attribute node.

Use the **Variables** list to specify which variables to associate with the previous parameter value rather than the present one.

Use the **Linear solver** list to select a solver for the linear systems associated with the quantities specified by **Variables**. The available solvers are attribute node of the types **Direct** and **Iterative**.

SCGS

The **SCGS** node () is an attribute node that handles the SCGS (symmetrically coupled Gauss-Seidel) solver, which is useful as a preconditioner for solving the Navier-Stokes equations and similar fluid-flow problems. The SCGS iterative solver (smoother) works in a similar way as the blocked update of the Vanka solver, but it builds blocks based on the DOFs in each mesh element instead of blocks based on DOF connectivity to a Vanka variable. The advantage is that the blocks are smaller, allowing for storing their factorization once during the initialization phase (like SOR Line does) instead of factorizing on every update (like Vanka does by default).

Compared to other multigrid smoothers, SCGS provides better performance and is more robust, but it also requires somewhat more memory. SCGS only works for linear elements, and it is the default smoother for fluid-flow models with P1+P1 elements (linear elements for the velocity field and the pressure).

The solver includes three main methods:

- *Mesh elements*: Each mesh element corresponds to one SCGS block.
- *Mesh element lines*: Anisotropic mesh elements are grouped together in SCGS blocks along the direction of anisotropy, which gives better results for boundary layer meshes. Non-anisotropic mesh elements correspond to one SCGS block.
- *Mesh element lines and vertices*: Anisotropic mesh elements are grouped together in SCGS blocks, like above. The DOFs corresponding to non-anisotropic mesh elements are solved using vertex-based SCGS blocks, which consume less memory than element blocks. A separate relaxation factor can be set for the vertex pass.

The solver additionally has an option to use a Vanka hybrid step where Vanka blocks are first built and then SCGS blocks are built excluding the Vanka variable DOFs. This step makes it possible to run iterative solvers when using, for example, the Laminar Inflow boundary condition, independent of whether the mesh is anisotropic or not.

The **SCGS** node's settings window includes the following sections:

MAIN

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute node is being used as a preconditioner or smoother (default: 2).

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the **Coarse Solver** is used.
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

Settings When Used With Any Attribute Node

Use the **Relaxation factor** field to specify a scalar relaxation factor ω . The allowed values of this factor are between 0 and 2 (default: 0.8).

Use the **Block solver** list to specify how to solver the Vanka block linear systems by selecting one of these options:

- **Direct, stored factorization** (the default) to store the factorization. If two SCGS smoothers are used as presmoother and postsmoother of a Multigrid solver, with similar enough settings, they share the same stored factorization, which means that the only use half the memory.

- **Direct** to use a direct solver. The direct solver is slower than the default option to store the factorization, but it uses less memory.
- **Iterative, GMRES** to use the iterative method GMRES.

If you use the SCGS algorithm as preconditioner, or as smoother to a multigrid preconditioner when either of GMRES, Conjugate gradients, or BiCGStab is used as the linear system solver, use the **Direct, stored factorization** or the **Direct** option in the **Block solver** list in order to get a stationary preconditioner.



The **Iterative, GMRES** option in the **Block solver** list can be useful if you use the FGMRES method as linear system solver because it can handle preconditioners that are not stationary. The GMRES option can also be useful if you use the SCGS algorithm as a smoother to a multigrid solver because GMRES can in some cases be faster than the direct solver if set to a high tolerance, although this advantage is less pronounced with SCGS than Vanka due to the smaller block size used by SCGS.

When **GMRES** has been selected in the **Block solver** list the following options become available. Use the **Tolerance** field to specify the termination tolerance of GMRES (default: 0.02). Use the **Number of iterations before restart** field to specify how many iterations the solver should take between each restart (default: 100).

From the **Method** list, select one of the following methods (see above):

- **Mesh element lines and vertices** (the default)
- **Mesh elements**
- **Mesh element lines**

Select the Vanka check box and then use the **Variables** list to specify variables to include in a Vanka block approach.

Select the **Blocked version** check box (selected by default) to use a version of the SCGS method that is optimized for parallel computations.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations (default: 1) to perform for degrees of freedom not involved in the SCGS blocks.

Use the **Relaxation factor** field to specify a scalar relaxation factor for the iterations specified in the **Number of secondary iterations** field (default: 1). The allowed values of this factor are between 0 and 2 (default: 0.5).

Segregated

The **Segregated** node () is an attribute node that handles parameters for a segregated solution approach. This attribute makes it possible to split the solution process into substeps. Each substep uses a damped version of Newton's method. The attribute can be used together with the **Stationary Solver** and **Time-Dependent Solver** nodes. An alternative to the segregated approach is given by the coupled solver, which is handled with the **Fully Coupled** attribute node. Although several **Fully Coupled** and **Segregated** attribute nodes can be attached to an operation node, only one can be active at any given time.

To add substeps to a segregated iteration right-click the **Segregated** node. One segregated iteration consists of executing each active **Segregated Step** in the order they are shown in the model tree.



The convergence properties of a model might depend on the order of the segregated steps. You can move the Segregated Step nodes to change the order in which the solver runs each step.

TERMINATION CRITERION

For the **Solution** termination criterion: When termination of the segregated solver is based on the estimated error, it terminates if, for all the groups j , the error estimate is smaller than the corresponding tolerance,

$$\text{err}_{j,k} < \text{tol}_j$$

where the error estimate in segregated iteration k is

$$\text{err}_{j,k} = \max(\text{e}_{j,k}^N, \text{e}_{j,k}^S)$$

The number tol_j is the relative tolerance for the corresponding group.

$$\text{e}_{j,k}^N = \max_l (1 - \alpha_l) \left[\frac{1}{N_j} \sum_{i=1}^{N_j} \left(\frac{|(\Delta U^{l,j,k})_i|}{W_i^j} \right)^2 \right]^{1/2}$$

is an estimate of the largest damped Newton error. Here l is taken for all iterations in all substeps solving for the group j , α_l is the damping factor, $\Delta U^{l,j,k}$ is the Newton increment vector, and N_j is the number of DOFs. The weight factor W_i^j is described below. Moreover,

$$e_{j,k}^S = \left[\frac{1}{N_j} \sum_{i=1}^{N_j} \left(\frac{|(U^{j,k} - U^{j,k-1})_i|}{W_i^j} \right)^2 \right]^{1/2}$$

is the relative increment over one complete iteration k . In this expression, $U^{j,k}$ is the segregated solution vector for the group j , and $W_i^j = \max(|U_i^j|, S_i)$, where S_i is a scale factor that the solver determines from the settings in the **Scaling** section of the settings window for the **Dependent Variables** node, where the following choices are available in the **Method** list:

- For **Automatic**, S_i is the factor 0.1 times the average of $|U_m|$ for all DOFs m having the same name as DOF i .
- For **Manual**, S_i is the value for DOF i given in the **Manual scaling** field.
- For **Initial value based**, S_i is the factor 0.1 times the average of $|U_{0m}|$ for all DOFs m having the same name as DOF i , where U_0 is the solution vector corresponding to the initial value.
- For **None**, $W_i = 1$.

For the **Residual** termination criterion, the segregated solver terminates when the following convergence criterion is satisfied: for all the groups j , the error estimate is smaller than the corresponding tolerance, $\text{err}_{j,k} < \text{tol}_j$, where

$$\text{err}_{j,k} = \left[\frac{1}{\tilde{W}_j^2} \sum_{i=1}^{N_j} |(F^{j,k})_i|^2 \right]^{1/2}$$

where F is the current residual, and \tilde{W} are the weights determined by the first and, if applicable, the second residual. The iterations can also terminate if the relative solution based error is in the range of a hundred machine epsilon.

Pseudo Time Stepping

Pseudo time stepping is available in a stationary segregated approach as well; see the [About Pseudo Time Stepping](#) for a description of the CFL regulation. For the segregated solver the error estimate e_n in [Equation 19-8](#) is the arithmetic average of the errors in the different segregated groups.

GENERAL

Use the **Termination technique** list to control how the segregated iterations are terminated. Select:

- **Tolerance** (the default) to terminate the segregated iterations when the estimated relative error is smaller than a specified tolerance.
- **Iterations or tolerance** to terminate the segregated iterations when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.
- **Iterations** to terminate the segregated iterations after a fixed number of iterations.
- **Pseudo time-stepping** (only available for **Stationary Solver**). Use the list to enable pseudo time stepping. When enabled use the fields to specify regulator parameters: **Initial CFL number**, **PID regulator-Proportional**, **PID regulator-Derivative**, **PID regulator-Integrative**, and **Target error estimate**. See [About Pseudo Time Stepping](#).

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are (see [Termination Criterion](#) above):

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field to limit the number of segregated iterations (default: 10). When the maximum number of iterations have been performed the segregated method is terminated even if the tolerance is not fulfilled.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field to modify the tolerance used for termination of the segregated iterations. The actual tolerance used is this factor times the value specified in the **Relative tolerance** field in the **General** sections of the **Stationary Solver** and **Time-Dependent Solver** nodes.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). The default is 1. Use this field to specify a fixed number of iterations to perform.

Select the **Limit on nonlinear convergence rate** check box (only available when used together with a **Time-Dependent Solver** node) to force the nonlinear solver to terminate as soon as the convergence is estimated to be too slow. Enter a limit on the convergence rate in the accompanying field.

Use the **Termination criterion** list to control how the Newton iterations are terminated for stationary problems (not available when **Termination technique** is set to **Iterations**). Select:

- **Solution** to terminate the Newton iterations on a solution-based estimated relative error.
- **Residual** to terminate the Newton iterations on a residual-based estimated relative error.
- **Solution or residual** to terminate the Newton iterations on the minimum of the solution-based and residual-based estimated relative errors. Here a scalar **Residual factor** (default: 100) multiplying the residual error estimate should be given.

RESULTS WHILE SOLVING

Check the **Plot** check box to plot results as soon as they become available. The plotted expression and data set depends on the selected plot group in the **Plot group** list.

If applicable, use the **Probes** list to plot results from one or more probes. Select **All** to plot all probe results. The default, **None**, plots no probe results.

Segregated Step

The **Segregated Step** node () handles settings for one substep of a segregated iteration. This attribute uses a damped version of Newton's method and can be used together with a **Segregated** attribute node.

GENERAL

Use the **Variables** list to specify variables to be solved for in this segregated step.

Use the **Linear solver** list to select a solver for the linear systems associated with the quantities specified by **Variables**. The available solvers are attribute nodes of the types **Direct** and **Iterative**.

METHOD AND TERMINATION

Use the **Nonlinear method** list to control which damping factor to use in the damped Newton iterations. Select:

- **Automatic (Newton)** to let the solver automatically determine a damping factor in each iteration of Newton's method.
- **Constant (Newton)** to manually specify a constant damping factor that is used in all iterations of Newton's method.

- **Automatic highly nonlinear (Newton)** for highly nonlinear problems. This option is similar to **Automatic (Newton)** but can make the solver more careful when solving highly nonlinear problems. Try this option if the solver does not converge with **Automatic (Newton)**.
- For stationary problems select **Double dogleg** to use the double dogleg nonlinear solver.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Nonlinear method** list are:

- **Damping factor** (only available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify a constant damping factor for Newton's method.
- **Jacobian update** (only available when **Nonlinear method** is set to **Constant (Newton)**). The available options in this list are **Minimal**, **On every iteration**, and **On first iteration**. The option **On every iteration** computes a new Jacobian for all iterations of Newton's method. The option **Minimal** updates the Jacobian at least once and then only when the nonlinear solver fails during time or parameter stepping. The option **On first iteration** updates the Jacobian for the first subiteration for this segregated step. When used with a **Time-Dependent Solver** node there is also the option **Once per time step**. For this option the Jacobian is updated at least once per time-step. The option **Minimal** reuses the Jacobian for several nonlinear systems whenever deemed possible.
- **Initial damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify a damping factor for the first Newton iteration.
- **Minimum damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). Use this field to specify the smallest allowed damping factor.
- **Restriction for step-size update** (not available when **Nonlinear method** is set to **Constant**). Use this field to specify a factor that limits how much the damping factor is allowed to change in a Newton iteration. The damping factor can change up or down by at most this factor.
- **Use recovery damping factor** (not available when **Nonlinear method** is set to **Constant (Newton)**). If a damping factor smaller than the **Minimum damping factor** is required, the nonlinear solver terminates when **Off** is selected in this list. When **On** is selected, the nonlinear solver takes a Newton step using the constant damping factor that you enter in the **Recovery damping factor** field. The default setting **Automatic** is equivalent to **On** for stationary problems and **Off** for time-dependent problems. For stationary parametric continuation problems, **Automatic** corresponds to **On** when

solving for the first parameter value and **Off** when solving for the subsequent parameter values.

- **Residual scaling** (only available when **Nonlinear method** is set to **Double dogleg**). The available options in this list are **Field-wise** and **Uniform**. The option **Field-wise** scales the equations based on the field-wise sizes of the initial residual. When the option **Uniform** is selected the algorithm terminates on the relative residual based on the initial residual.

Use the **Termination technique** list to control how the Newton iterations are terminated. Select:

- **Tolerance** to terminate the Newton iterations when the estimated relative error is smaller than a specified tolerance.
- **Iterations or tolerance** to terminate the Newton iterations when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.
- **Iterations** to terminate the Newton iterations after a fixed number of iterations.



Iterations is available when **Nonlinear method** is set to **Constant (Newton)**.

Further options that apply to one or several selections (as indicated at each bullet) made in the **Termination technique** list are:

- **Maximum number of iterations** (only available when **Termination technique** is set to **Tolerance**). Use this field to limit the number of Newton iterations. When the maximum number of iterations have been performed Newton's method is terminated even if the tolerance is not fulfilled.
- **Tolerance factor** (not available when **Termination technique** is set to **Iterations**). Use this field to modify the tolerance used for termination of the Newton iterations. The actual tolerance used is this factor times the value specified in the **Relative tolerance** field in the **General** sections of the **Stationary Solver** and **Time-Dependent Solver** nodes.
- **Number of iterations** (not available when **Termination technique** is set to **Tolerance**). Use this field to specify a fixed number of iterations to perform.

Sensitivity

A **Sensitivity** solver node () solves a sensitivity analysis problem set up in a Sensitivity user interface.

GENERAL

In the **Objective** list, you specify the objective to use for the sensitivity analysis.

In the **Sensitivity method** list you can choose between the following options (see also [The Sensitivity Analysis Algorithm](#) below):

- **Adjoint**—The adjoint method solves for the derivatives of a single scalar objective function with respect to any number of sensitivity variables.
- **Forward**—The forward sensitivity method solves for the derivatives of all dependent variables and an optional scalar objective function with respect to a small number of sensitivity variables.



Sensitivity analysis for time-dependent problems is available in the Optimization Module.

THE SENSITIVITY ANALYSIS ALGORITHM

When you enable sensitivity analysis, the stationary solvers compute—in addition to the basic forward solution—the sensitivity of a functional

$$Q = Q(u_p, p) \quad (19-10)$$

with respect to the sensitivity variables p . The forward solution u_p is a solution to the parameterized discrete forward problem

$$L(u_p, p) = N_F \Lambda_p \quad M(u_p, p) = 0 \quad (19-11)$$

where Λ_p are the constraint Lagrange multipliers, or (generalized) reaction forces, corresponding to the constraints M . It is assumed that Q does not explicitly depend on Λ_p .

To compute the sensitivity of Q with respect to p , first apply the chain rule:

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \frac{\partial Q}{\partial u} \frac{\partial u}{\partial p} \quad (19-12)$$

In this expression, the sensitivity of the solution with respect to the sensitivity variables, $\partial u / \partial p$, is still an unknown quantity. Therefore, differentiate the forward problem in [Equation 19-11](#) formally with respect to p :

$$K \frac{\partial u_p}{\partial p} + N_F \frac{\partial \Lambda_p}{\partial p} = \frac{\partial L}{\partial p} + \frac{\partial N_F}{\partial p} \Lambda_p \quad N \frac{\partial u_p}{\partial p} = \frac{\partial M}{\partial p}$$

Here, $K = -\partial L / \partial u$ and $N = -\partial M / \partial u$ as usual. Assuming that the constraint force Jacobian N_F is independent of p (that is, $\partial N_F / \partial p = 0$), you can write the above relations in matrix form

$$J \begin{pmatrix} \frac{\partial u_p}{\partial p} \\ \frac{\partial \Lambda_p}{\partial p} \end{pmatrix} = \begin{pmatrix} \frac{\partial L}{\partial p} \\ \frac{\partial M}{\partial p} \end{pmatrix} \quad J = \begin{bmatrix} K & N_F \\ N & 0 \end{bmatrix} \quad (19-13)$$

Solve for the sensitivities $\partial u_p / \partial p$ and $\partial \Lambda_p / \partial p$, and plug them back into [Equation 19-12](#):

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \begin{pmatrix} \frac{\partial Q}{\partial u} \\ 0 \end{pmatrix}^T J^{-1} \begin{pmatrix} \frac{\partial L}{\partial p} \\ \frac{\partial M}{\partial p} \end{pmatrix} \quad (19-14)$$

This formula gives dQ/dp explicitly in terms of known quantities, but in practice it is too expensive to invert the matrix J .

If the number of individual sensitivity variables, p_j , is small, [Equation 19-13](#) can be solved for each right-hand side $[\partial L / \partial p_j \ \partial M / \partial p_j]^T$, and the solution is then inserted into [Equation 19-12](#). This is the *forward method*, which in addition to the sensitivity dQ/dp returns the sensitivity of the solution, $\partial u_p / \partial p$. The matrix J is in fact the same matrix as in the last linearization of the forward problem. The forward method therefore requires one additional back-substitution for each sensitivity variable.

If there are many sensitivity variables and the sensitivity of the solution itself, $\partial u_p / \partial p$, is not required, the *adjoint method* is more efficient. It is based on using auxiliary variables u^* and L^* , known as the *adjoint solution*, to rewrite [Equation 19-14](#):

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + (u^*)^T \begin{pmatrix} \frac{\partial L}{\partial p} \\ \frac{\partial M}{\partial p} \end{pmatrix}$$

$$J^T \begin{pmatrix} u^* \\ \Lambda^* \end{pmatrix} = \begin{pmatrix} \frac{\partial Q}{\partial u} \\ 0 \end{pmatrix}$$

In this form only one linear system of equations must be solved regardless of the number of sensitivity variables, followed by a simple scalar product for each variable. This is much faster than the forward method if the number of variables is large. The system matrix, which is solved for, is the transpose of the last linearization of the forward problem. This makes no significant difference for the iterative linear solvers. For the direct solvers, if J is symmetric or Hermitian, this makes no difference compared to the forward method, and the direct solvers can reuse the factorization. In the nonsymmetric case, MUMPS and PARDISO can reuse the factorization of J while SPOOLES needs to do a new factorization of J^T .

SOR

The **SOR** node () handles settings for the SOR (successive over-relaxation) iterative method. This attribute can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoother**, **Postsmoother**, and **Coarse Solver** attribute nodes. Right-click any of these nodes to add a **SOR** node.

- The SOR method provides a simple and memory-efficient solver/preconditioner/smooth based on classical iteration methods for solving a linear system $Ax = b$. Given a relaxation factor ω (usually between 0 and 2), a sweep of the SOR method transforms an initial guess x_0 to an improved approximation $x_1 = x_0 + M^{-1}(b - Ax_0)$, where the preconditioning matrix $M = L + D/\omega$, and D is the diagonal part of A , and L is the strictly lower triangular part of A . When $\omega = 1$ (the default), the Gauss-Seidel method is obtained.
- In the SORU method, $M = U + D/\omega$, where U is the strictly upper triangular part of A . The SOR and SORU methods use a more accurate approximation of the matrix, which leads to fewer iterations but slightly more work per iteration than in the Jacobi method.
- The SSOR (symmetric successive over-relaxation) method is one SOR sweep followed by a SORU sweep. The output x_1 for an input x_0 also comes from the above formula but with

$$M = \frac{\omega}{2-\omega} \left(L + \frac{D}{\omega} \right) D^{-1} \left(U + \frac{D}{\omega} \right).$$

When the system matrix A is symmetric, the SSOR method has the advantage that the preconditioning matrix M is symmetric. Symmetry of the preconditioner matrix is necessary when using the conjugate gradients iterative method. In such cases, the SSOR preconditioner is preferable to the SOR preconditioner.

GENERAL

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR algorithm to use. Select:

- **SSOR** (the default) to use the symmetric SOR algorithm, which in each iteration performs one **SOR** sweep followed by one **SORU** sweep.
- **SOR** to use the forward SOR algorithm.
- **SORU** to use the backward SOR (SORU) algorithm.

Use the **Relaxation factor** field to specify a scalar relaxation factor ω . The allowed values of this factor are between 0 and 2 (default: 1).

Select the **Blocked version** check box (selected by default) to use a blocked version of the SOR method that is optimized for parallel computations. M is then constructed from a column-permuted version of A .

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations (default: 2) to perform when this attribute is used as a preconditioner or smoother. This setting is not considered when the attribute is used as a linear system solver (with the **Use preconditioner** option in the **Solver** list of the **Iterative** node). The solver then iterates until it has established convergence or has reached the maximal number of iterations as specified by the corresponding **Iterative** node rather than performs a fixed number of iterations.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used (the default)

- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

SOR Gauge

The **SOR Gauge** node () handles settings for the SOR gauge iterative method. This is a method of SOR-type with added functionality useful as preconditioner/smoothening for, for example, 3D magnetostatics in the AC/DC Module discretized with vector elements. In short, the added functionality consists of divergence cleaning for degrees of freedom discretized with vector elements. This node can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmooth**, **Postsmooth**, and **Coarse Solver** nodes.

THE SSOR GAUGE, SOR GAUGE, AND SORU GAUGE ALGORITHMS

Magnetostatic problems are often formulated in terms of a magnetic vector potential. The solution of problems formulated with such a potential is in general not unique. Infinitely many vector potentials result in the same magnetic field, which typically is the quantity of interest. A finite element discretization of such a problem results in a singular linear system of equations, $Ax = b$. Despite being singular, these systems can be solved using iterative solvers, provided that the right-hand side of the discretized problem is the range of the matrix A . For discretized magnetostatic problems, the range of A consists of all divergence-free vectors. Even if the right side of the mathematical problem is divergence free, the right side of the finite element discretization might not be numerically divergence free. To ensure that b is in the

range of A , SOR gauge performs a divergence cleaning of the right side by using the matrices T and T^T similar to the algorithm for the [SOR Vector](#) iterative method. To this end, the system $T^T T \psi = -T^T b$ is first solved. Adding $T \psi$ to b then makes the numerical divergence of the right side small.

MAIN

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR Gauge algorithm to use. Each variant first performs one ordinary SOR iteration followed by one or several divergence cleaning iterations Select:

- **SSOR gauge** (the default) to perform an ordinary **SSOR** iteration followed by divergence cleaning.
- **SOR gauge** to perform an ordinary **SOR** iteration followed by divergence cleaning.
- **SORU gauge** to perform an ordinary **SORU** iteration followed by divergence cleaning.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2 (default: 1).

Select the **Blocked version** check box (selected by default) to use a version of the SOR Gauge method that is optimized for parallel computations.

Use the **Variables** list to specify variables to include in the divergence cleaning phase of an SOR Gauge iteration. By default, all vector degrees of freedom are included.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2). In addition to the initial divergence cleaning, the method performs a number of cleaning iterations in each linear solver iteration. You control the number of such divergence cleaning iterations in the **Number of secondary iterations** field.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used.
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of divergence cleaning iterations to perform for each main iteration (default: 1).

SOR Line

The **SOR Line** node () handles settings for the SOR line iterative method. This is a method of SOR type with added functionality useful for, for example, anisotropic meshes. It is a block SOR solver, where the blocks are formed from lines of nodes that are relatively close to each other. In addition, ordinary SSOR iterations are performed for all degrees of freedom after the SOR Line iterations have been performed. This node can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmoothening**, **Postsmoother**, and **Coarse Solver** nodes.

THE SOR LINE ALGORITHM

In regions where the mesh is sufficiently anisotropic, the algorithm forms *lines* of nodes that connect nodes that are relatively close to each other (Ref. 25). Thus, in a

boundary layer, a line is a curve along the thin direction of the mesh elements. A smoothing iteration does two things:

- Line update: Performs block SOR smoothing where each block consists of degrees of freedom located on a line. Due to the banded structure of each block matrix, this smoothing runs relatively fast.
- SSOR update: Performs a number of SSOR smoothing iterations on the whole mesh.

Like the SOR and Jacobi smoothers/preconditioners, the algorithm gives an error message if it finds zeros on the diagonal of the system matrix.

MAIN

Settings When Used With Any Attribute Node

Use the **Relaxation factor** field to specify a scalar relaxation factor that controls the damping of the block SOR smoothing steps. The allowed values of this factor are between 0 and 2 (default: 0.5).

User the **Line based on** list to control if the lines of nodes are based on the **Mesh** (the default) or on a **Matrix**. If you select **Matrix**, also define the **Maximum line length** (default: 20). This value determines the maximum length of the lines in number of DOFs for each block.

Use the **Multivariable method** list to control the line updates:

- If **Uncoupled** is selected, each block SOR smoothing step updates a set of degrees of freedom with the same name that are located on a line.
- If **Coupled** is selected (the default), each block SOR smoothing step updates all degrees of freedom located on a line.

For smoothing of the turbulence variables K and ϵ , **Coupled** is recommended.

Select the **Blocked version** check box (selected by default) to use a version of the SOR method that is optimized for parallel computations.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute is used as a preconditioner or smoother (default: 2).

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used (the default).
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver is automatically stopped with an error message.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations to perform after the SOR Line iterations have been performed (default: 1).

Use the **Relaxation factor** field to specify a scalar relaxation factor that controls the damping of the SSOR updates for the iterations specified in the **Number of secondary iterations** field. The allowed values of this factor are between 0 and 2 (default: 0.7).

SOR Vector

The **SOR Vector** node ( handles settings for the SOR vector iterative method. This is a method of SOR type with added functionality useful for electromagnetics problems involving the $\nabla \times (a \nabla \times \cdot)$ curl-curl operator and where you use *vector elements* (available primarily for electromagnetic wave simulations in the RF Module). In short, the added functionality consists of performing SOR iterations on an auxiliary linear system in addition to the ordinary SOR iterations. This node can be used as

preconditioner/smoothening together with the **Iterative, Krylov Preconditioner**, **Presmoothening**, **Postsmoothing**, and **Coarse Solver** nodes.

The SOR vector algorithm is an implementation of the concepts in Ref. 22 and Ref. 16. The algorithm applies SOR iterations on the main linear equation $Ax = b$ but also makes SOR iterations on a projected linear equation $T^T A T y = T^T b$. Here the projection matrix, T , is the discrete gradient operator, which takes values of a scalar field in the mesh vertices and computes the vector-element representation of its gradient. Loosely speaking, the argument for using this projection is the following: For example, let the linear equation $Ax = b$ represent the discretization of a PDE problem originating from the vector Helmholtz equation

$$\nabla \times (a \nabla \times \mathbf{E}) + c \mathbf{E} = \mathbf{F}$$

for the unknown vector field \mathbf{E} , where a and c are scalars, and \mathbf{F} is some right-hand side vector. Standard preconditioners/smoothers cannot smooth the error in the null space of the operator $\nabla \times (a \nabla \times \cdot)$. This null space is the range of the gradient operator. This algorithm adds a correction $\mathbf{E} \rightarrow \mathbf{E} + \nabla \phi$ to the standard SOR smoothed solution (or residual), where it computes ϕ from SOR iterations on a projected auxiliary problem. The projected problem is obtained by taking the divergence (or discretely $-T^T$) of the Helmholtz equation and plugging in the correction. You then obtain (for clarity, boundary constraints are disregarded)

$$-\nabla \cdot (c \nabla \phi) = -\nabla \cdot \mathbf{F}$$

which, if c is definite (strictly positive or strictly negative), is a standard elliptic type of equation for the scalar field ϕ .

When using this algorithm as a smoother for the multigrid solver/preconditioner, it is important—for the correct discrete properties of the projected problem—to generate nested meshes. Also it does assembly on all mesh levels (controlled by the multigrid **Assemble on all levels** check box). You can generate nested meshes through manual mesh refinements or do so automatically by selecting **Refine mesh** from the **Hierarchy generation method** list in the **Multigrid** node.

The projection matrix T is computed in such a way that non-vector shape functions are disregarded, and you can therefore use it in a multiphysics setting. It can also handle contributions from different geometries. Non-vector shape function variables are not affected by the correction from the projected system, and the effects on them are the same as when you apply the standard SOR algorithm.

MAIN

Settings When Used With Any Attribute Node

Use the **Solver** list to specify which variant of the SOR Vector algorithm to use. Select:

- **SSOR vector** to perform one ordinary SOR iteration on the main system followed by a number of SSOR iterations on an auxiliary (projected) system and then one ordinary SORU iteration. This is repeated in each SSOR vector iteration.
- **SOR vector** to perform one ordinary SOR iteration followed by a number of SOR iterations on an auxiliary system. This is repeated in each SOR vector iteration.
- **SORU vector** to perform a number of SORU iterations on an auxiliary system followed by one ordinary SORU iteration. This is repeated in each SORU vector iteration.

The algorithms perform these iterations to preserve symmetry as a preconditioner and also when used as symmetric presmoother and postsmoother in a multigrid setting.

Use the **Relaxation factor** field to specify a scalar relaxation factor. The allowed values of this factor are between 0 and 2. Default: 1)

Select the **Blocked version** check box (selected by default) to use a version of the SOR Vector method that is optimized for parallel computations.

Use the **Variables** list to specify variables to be included in the auxiliary system of the SOR Vector method.

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of main iterations (default: 2) to perform when this attribute is used as a preconditioner or smoother. For each main iteration, the algorithm makes a number of SOR iterations for the projected equation system; you set that number in the **Number of secondary iterations** field.

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** to perform a fixed number of iterations each time the **Coarse Solver** is used.
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of iterations to perform the auxiliary system for each main iteration (default: 1).

State

The **State** node is an attribute node that handles settings for state variables. A state is composed of a set of ODE variables. Each state has a separate **State** node. This attribute is used together with the **Dependent Variables** node.

GENERAL

The **State Components** section display the variable names for the states components. Also, when internal extra variables are used, these are displayed here as **Internal Variables**.

Use the **Solve for this state** check box to control whether to solve for the state or not. This setting is only available if the **Dependent Variables** node's setting **Defined by study step** is set to **User defined**. If the variable is not solved for its values is determined by the settings in the **Values of Variables Not Solved For** section of the corresponding **Dependent Variables** operation node.

Use the **Store in output** check box to control whether to store the variable in any output solution or not.



A variable can still be solved for despite not being stored in output and vice versa.

SCALING

Control the scaling of a variable with the **Method** list. Select:

- **Automatic** to get an automatically determined scaling.
- **From parent** to use the scaling type selected in the **Method** list in the **Scaling** section of the corresponding **Variables** operation node.
- **Initial value based** to get a scaling that is determined from the initial values. Use this if the components of the initial values give a good estimate of the order of magnitude of the solution.
- **Manual** to manually enter a scaling.
- **None** to get no scaling.

If you choose **Manual**, specify the scale by entering a value in the **Scale** field.

Specifying a **Method** for a variable here overrides the **Method** selected in the **Scaling** section of the corresponding **Variables** operation node unless **From parent** is selected.

Stationary Acceleration

The **Stationary Acceleration** subnode (⌚) can be useful to accelerate the solution process for nonlinear problems with a time-periodic stationary solution. You can add it as a subnode to all time-dependent solver nodes. Instead of time-marching the problem from start to finish, the **Stationary Acceleration** node solves for a number of periods and then extrapolates the solution forward in time based on the average solution and the average time derivative. This solution process is repeated until the average time derivative has reached steady state.



The **Stationary Acceleration** node can be used to speed up the solution process for some types of plasma models but it is not used by default for any physics.

STATIONARY ACCELERATION

In the **Variables** list, add the dependent variable for which you want to use stationary acceleration. Click the **Add** button () to open an **Add** dialog box that contains all dependent variables in the study. Select the variables that you want to add and then click **OK**. You can also delete variables from the list using the **Delete** button () and move them using the **Move Up** () and **Move Down** () buttons.

From the **Components** list, select the dependent variables for which the stationary acceleration performs the averaging and extrapolation. Select **All** (the default) to perform averaging and extrapolation for all variables, or select **Manual** to select the variables that you want to apply stationary acceleration from the list that appears.

In the **Frequency** field, enter the frequency of the periodic solution. The default value, 13.56 MHz, is a frequency that is commonly used for plasma processes.

In the **Stationary tolerance** field, enter the tolerance used to terminate the outer acceleration iterations, when the average time derivatives are small enough (default value: 0.01).

In the **Number of extrapolation cycles** field, enter the number of periodic cycles used to extrapolate the solution (default value: 50). The higher this number is the more the solution process is accelerated, but at the same time the process can lead to an unstable acceleration iteration process.

In the **Number of period averaging cycles** field, enter the number of cycles over which the stationary acceleration takes the average (default value: 5).

In the **Number of smoothing cycles** field you enter the number of cycles that the stationary acceleration solves for in each iteration of the acceleration scheme (default value: 10). The average is taken over the last cycles.

Stop Condition

The **Stop Condition** node () stops the solver when any of the specified conditions are fulfilled. It is an optional attribute subnode to the **Parametric** and **Time-Dependent Solver** nodes.

STOP EXPRESSIONS

Use the table to specify expressions for the conditions that define when the solver should stop. The solver evaluates the active expressions after each time step or parameter step. The setting in the **Stop if** column of each expression determines how it is evaluated. For **True ($>=1$)** (the default) the stepping stops if the real part is greater

or equal to one, which is useful when entering logic expressions that evaluate to a Boolean true or false (`mod1.EndTerminal(mod1.phis)<2.4`, for example). For **Negative (<0)** the stepping stops if the real part of the expression becomes negative (`mod1.z_pnt-0.0004`, for example). Another example of a stop condition is `timestep<0.04`, which makes the solver stop when the internal time step drops below 0.04 s (when the time-dependent solver hits a sharp transient, for example). To use such logical expressions, use the **True (>=1)** setting.

STOP EVENTS

Use the **Implicit event** table to specify at which events the solver should stop. All implicit events defined in the model automatically appear in the table. The stepping stops when any event marked as active is triggered.

The **Stop Events** section only appears when the **Stop Condition** node is an attribute of the **Time-Dependent Solver** operation node because events are only supported for time-dependent solvers.

OUTPUT AT STOP

Use the **Add solution** option to make the solver additionally store the corresponding solutions before and after the stop condition was fulfilled. Select one of the following options:

- **No** (the default) to not store any additional solutions. The last solution stored is the one normally stored by the solver before the stop condition was fulfilled.
- **Step before stop** to store the last step taken by the solver before the stop condition was fulfilled. No solutions are stored after this point even if they normally would be.
- **Step after stop** to store the solver step at which the stop condition was fulfilled. Any solutions up to this point are also stored as the normally would be.
- **Steps before and after stop** to store all solutions that would be stored by either of **Step before stop** and **Step after stop**.

Which solutions are normally stored by the solver depends on the **Times to store** setting for the **Time-Dependent Solver** node and on the **Parameters to store** setting for the **Parametric** node.

An example of using **Step before stop** would be to make sure to capture the last state of a simulation before a certain condition has been fulfilled, without having to store all of the solver steps up until this point. The setting **Step after stop** would similarly be used to capture the first state fulfilling a certain condition. When both the state before and after the condition are of interest, use the setting **Step before and after stop** to

capture the transition. If the stop condition was fulfilled by the reinitialization effect of an implicit event, **Step before stop** stores the solution before reinitialization and **Step after stop** stores the solution after reinitialization.

Select the **Add warning** setting to specify that the solver adds a warning when the solver has stopped due to a stop condition.

Time Parametric

The **Time Parametric** node ( 213) is an attribute node that handles settings for parameter stepping to add parametric sweeps. For each set of parameter values, a time-dependent problem is solved. This attribute can be used together with a **Time-Dependent Solver** or another time-dependent solver. The functionality when used as a subnode to a **Time-Dependent Solver** is similar to the **Parametric** as a subnode to a **Stationary Solver**, but continuation is not supported. The initial data is the same for all parameters.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step. Select **User defined** to modify the parameter table and sweep type.

Use the **Sweep type** list to specify the type of sweep to perform. The **Specified combinations** type (the default) solves for a number of given combination of values, while the **All combinations** type solves for all combination of values. Using all combinations can lead to a very large number of solutions.

Use the **Parameter names** and **Parameter value list** table to specify parameter names and values for the parametric solver. Use the **Add** button () to add a row to the table. Each row has one parameter name and a corresponding parameter value list. For the **Specified combinations** sweep type, the list of values must have equal length. When you click in the **Parameter value list** column to define the parameter values you can click the **Range** button () to define a range of parameter values.

If more than one parameter name have been specified the lists of parameter values is interpreted as follows: Assume that the parameter names are p1 and p2, and that p1 has the list 1 3 and p2 has the list 2 4. For the **Specified combinations** sweep type, the solver first uses p1 equal to 1 and p2 equal to 2. Thereafter, it uses p1 equal to 3 and p2 equal to 4. And when the sweep type in **All combinations**, the solver uses the following order for the parameter combinations: 1 2, 1 4, 3 2, and 3 4.

An alternative to specifying parameter names and values directly in the table is to specify them in a text file. You can use the **Load from file** button () to browse to such a text file. The read names and values are appended to the current table. The format of the text file must be such that the parameter names appear in the first column and the values for each parameter appear row-wise with a space separating the name and values, and a space separating the values. Click the **Save to file** button () to save the contents of the table to a text file (or to a Microsoft Excel Workbook spreadsheet if the license includes LiveLink™ for Excel®).

CLUSTER SETTINGS

Select the **Distribute parameters** check box to distribute the parameters on several computational nodes. If the problem is too large to run on a single node you can enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a non-distributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

Vanka

The **Vanka** node () handles settings for the Vanka iterative method. Formally, this method applies to saddle-point problems (that is, problems where the equilibrium solution is neither a maximum nor a minimum) as a preconditioner/smooth. The corresponding linear system matrix is indefinite and its diagonal often contains zeros. A typical example is the Navier-Stokes equations. Problems formulated with weak constraints are also of this type. In short, the method can be described as a block SOR method. Local coupling of certain degrees of freedom (typically the Lagrange multiplier degrees of freedom) determines the blocks. Ordinary SSOR iterations are performed for degrees of freedom not involved in the block method. This attribute node can be used together with the **Iterative**, **Krylov Preconditioner**, **Presmooth**, **Postsmoother**, and **Coarse Solver** attribute node. For more information about the Vanka method, see [The Vanka Algorithm](#) below.

MAIN

Settings When Not Used With Coarse Solver

Use the **Number of iterations** field to specify a fixed number of iterations to perform when this attribute node is being used as a preconditioner or smoother (default: 2).

Settings When Used With Coarse Solver

Use the **Termination technique** to select how to terminate the solver. Select:

- **Fixed number of iterations** (the default) to perform a fixed number of iterations each time the **Coarse Solver** is used.
- **Use tolerance** to terminate the **Coarse Solver** when a tolerance is fulfilled.
- **Iterations or tolerance** to terminate when the estimated tolerance is smaller than a specified tolerance or after a specified number of iterations, whichever comes first.

Further options that apply to the selection (as indicated at each bullet) made in the **Termination technique** list are:

- **Number of iterations** (available when **Termination technique** is set to **Fixed number of iterations** or **Iterations or tolerance**). Use this field to specify the fixed number of iterations to perform (default: 10).
- **Relative tolerance** (available when **Termination technique** is set to **Use tolerance** or **Iterations or tolerance**). Use this field to specify the termination tolerance (default: 0.1).
- **Maximum number of iterations** (only available when **Termination technique** is set to **Use tolerance**). Use this field to enter a maximum number of iterations that the solver is allowed to take (default: 500). When this number of iterations has been performed without reaching the tolerance specified in the **Relative tolerance** field the solver automatically stops with an error message.

Settings When Used With Any Attribute Node

Use the **Variables** list to specify variables to include in a Vanka block approach.

Use the **Block solver** list to specify how to solver the Vanka block linear systems:

- Select **Direct** (the default) to use a direct solver
- Select **Direct, stored factorization** to store the factorization. Storing the factorization makes the solver faster because the factorization is then not performed every update, but the storage uses more memory. If two Vanka smoothers are used as presmoother

and postsmoother of a Multigrid solver, with similar enough settings, they share the same stored factorization, which means that the only use half the memory.

- Select **GMRES** to use the iterative method GMRES.

If you use the Vanka algorithm as preconditioner, or as smoother to a multigrid preconditioner when either of GMRES, Conjugate gradients, or BiCGStab is used as the linear system solver, use the **Direct** or **Direct, stored factorization** option in the **Block solver** list in order to get a stationary preconditioner.



The **GMRES** option can be useful if you use the FGMRES method as linear system solver because it can handle preconditioners that are not stationary. The **GMRES** option can also be useful if you use the Vanka algorithm as smoother to a multigrid solver because GMRES can be a bit faster than the direct solver.

When **GMRES** has been selected in the **Block solver** list the following options become available. Use the **Tolerance** field to specify the termination tolerance of GMRES (default: 0.02). Use the **Number of iterations before restart** field to specify how many iterations the solver should take between each restart (default: 100).

Use the **Relaxation factor** field to specify a scalar relaxation factor ω . The allowed values of this factor are between 0 and 2 (default: 0.8).

Select the **Blocked version** check box (selected by default) to use a version of the Vanka method that is optimized for parallel computations.

SECONDARY

Use the **Number of secondary iterations** field to specify the number of SSOR iterations to perform for degrees of freedom not involved in the Vanka blocks.

Use the **Relaxation factor** field to specify a scalar relaxation factor for the iterations specified in the **Number of secondary iterations** field (default: 1). The allowed values of this factor are between 0 and 2 (default: 1).

THE VANKA ALGORITHM

The algorithm is a local smoother/preconditioner of Vanka type. It is based on the ideas in Ref. 17, Ref. 24, and Ref. 25. It is possible to describe it as a block SOR method, where the local coupling of the degrees of freedom (DOFs) determines the blocks. The important idea in this algorithm is to use Lagrange multiplier variables to

form the blocks. For illustration purposes, consider the Navier-Stokes equations. For these equations the pressure variable plays the role of Lagrange multiplier. The linearized equations on discrete form has the following structure:

$$A \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} S & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

where U and P are the velocity and pressure degrees of freedom, respectively. The algorithm loops over the Lagrange multiplier variable DOFs (here the pressure DOFs P_j) and finds the direct connectivity to this DOF. To do so, the algorithm locates the nonzero entries in the matrix column corresponding to P_j . The row indices of the nonzero entries define the DOFs U_k , and the algorithm forms a local block matrix based on this connectivity:

$$A_j = \begin{bmatrix} S_j & D_j^T \\ D_j & 0 \end{bmatrix}$$

One *Vanka update* loops over all P_j and updates

$$\begin{bmatrix} U_j \\ P_j \end{bmatrix} \leftarrow \begin{bmatrix} U_j \\ P_j \end{bmatrix} + \omega A_j^{-1} \left(\begin{bmatrix} F \\ G \end{bmatrix} - A \begin{bmatrix} U \\ P \end{bmatrix} \right)_j$$

where the $(.)_j$ denotes the restriction of a vector to the rows corresponding to the block j . ω is a relaxation parameter. The algorithm does not form the inverses of the block matrices explicitly. Instead, it computes the Vanka update with a LAPACK direct solver or a GMRES iterative method subroutine call. The GMRES method is the restarted GMRES without preconditioning. The algorithm relies on that it is possible to invert the submatrices A_j . If it is not possible, the algorithm gives an error message. A zero on the diagonal of A or A_j is not necessarily a problem for this updating strategy. In general, the Vanka update does not necessarily update all DOFs. This is the case for problems with weak constraints, where only a small subset of the problem's DOFs are directly coupled to the Lagrange multipliers for the constraints. Another example is the Navier-Stokes equations coupled to other equations, but where the coupling is not directly through the pressure variable (in the k - ϵ turbulence model, for example). The Vanka algorithm automatically detects DOFs that are not updated by the above Vanka updating procedure and performs, for each Vanka update, a number of SSOR sweeps for these DOFs. This part of the algorithm is the *SSOR update*; it only works for a submatrix that has a nonzero diagonal. Just as the SOR and Jacobi preconditioner

algorithms, this algorithm gives an error message if it finds zeros on the diagonal for the DOFs in the SSOR update.

References for the Linear System Solvers and the Preconditioners

1. <http://graal.ens-lyon.fr/MUMPS/>
2. www.netlib.org/linalg/spooles/
3. www.pardiso-project.org/
4. <https://computation.llnl.gov/casc/hypre/software.html>
5. https://computation.llnl.gov/casc/hypre/download/hypre-2.9.0b_usr_manual.pdf
6. Greenbaum, A., “Iterative Methods for Linear Systems,” *Frontiers in Applied Mathematics*, vol. 17, SIAM, 1997.
7. Y. Saad and M.H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856–869, 1986.
8. Y. Saad, *Iterative Methods for Sparse Linear Systems*, Boston, 1996.
9. Y. Saad, “A Flexible Inner-Outer Preconditioned GMRES Algorithm,” *SIAM J. Sci. Statist. Comput.*, vol. 14, pp. 461–469, 1993.
10. M.R. Hestenes and E. Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems,” *J. Res. Nat. Bur. Standards*, vol. 49, pp. 409–435, 1952.
11. C. Lanczos, “Solutions of Linear Equations by Minimized Iterations,” *J. Res. Nat. Bur. Standards*, vol. 49, pp. 33–53, 1952.
12. H.A. Van Der Vorst, “A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems,” *SIAM J. Sci. Statist. Comput.*, vol. 13, pp. 631–644, 1992.
13. J.R. Gilbert and S. Toledo, “An Assessment of Incomplete-LU Preconditioners for Nonsymmetric Linear Systems,” *Informatica*, vol. 24, pp. 409–425, 2000.
14. Y. Saad, *ILUT: A Dual Threshold Incomplete LU Factorization*, Report umsi-92-38, Computer Science Department, University of Minnesota, available from <http://www-users.cs.umn.edu/~saad>.

15. W. Hackbusch, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
16. R. Beck and R. Hiptmair, “Multilevel Solution of the Time-harmonic Maxwell’s Equations Based on Edge Elements,” *Int. J. Num. Meth. Engr.*, vol. 45, pp. 901–920, 1999.
17. S. Vanka, “Block-implicit Multigrid Calculation of Two-dimensional Recirculating Flows,” *Computer Methods in Applied Mechanics and Engineering*, vol. 59, no. 1, pp. 29–48, 1986.
18. H.C. Elman and others, “A Multigrid Method Enhanced by Krylov Subspace Iteration for Discrete Helmholtz Equations,” *SIAM J. Sci. Comp.*, vol. 23, pp. 1291–1315, 2001.

19. A. Toselli and O. Widlund, “Domain Decomposition Methods—Algorithms and Theory,” *Springer Series in Computational Mathematics*, vol. 34, 2005.

The COMSOL Multiphysics implementations of the algebraic multigrid solver and preconditioner are based on the following references:

20. K. Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD Report 70, GMD, 1999.
21. C. Wagner, *Introduction to Algebraic Multigrid*, course notes, University of Heidelberg, 1999.
22. R. Hiptmair, “Multigrid Method for Maxwell’s Equations,” *SIAM J. Numer. Anal.*, vol. 36, pp. 204–225, 1999.
23. D. J. Mavriplis, “Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flows,” *ICASE Report No. 98-7 (NASA/CR-1998-206911)*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1998.
24. V. John and G. Matthies, “Higher-order Finite Element Discretization in a Benchmark Problem for Incompressible Flows,” *Int. J. Numer. Meth. Fluids*, vol. 37, pp. 885–903, 2001.
25. V. John, “Higher-order Finite Element Methods and Multigrid Solvers in a Benchmark Problem for the 3D Navier-Stokes Equations,” *Int. J. Numer. Meth. Fluids*, vol. 40, pp. 775–798, 2002.

Solution Utility Nodes

The following sections describe the solver utility nodes and their settings in detail.

Adaptive Mesh Refinement

The **Adaptive Mesh Refinement** () utility node is a container for a solution obtained using the **Adaptive Mesh Refinement** attribute node. This utility node is added automatically (to an otherwise empty solver configuration) by the corresponding **Adaptive Mesh Refinement** attribute node. It is not possible to add this node manually, and it does not have any settings.



Adaptive Mesh Refinement (attribute node)

Assemble

The **Assemble** node () provides access to the raw data of any assembled matrix or vector. Right-click the solver node and select **Other>Assemble**. Select the appropriate check boxes for the matrices and vectors you want to inspect or modify and save the model as a Model Java-file. You can also display the values of these system matrix in a table using the **System Matrix** node () that you find under **Results>Derived Values**. The saved Java-file now contains code for assembling the selected matrices and vectors that can be used to access the matrix rows, columns, values, and so forth. For information about the eliminated system, see [Elimination Constraint Handling](#).

Note that some study types may require that additional parameters are defined. In order to get the expected matrices you have to specify these parameters manually. If you get an error about undefined variables you can define the variables in the **Global Definitions>Parameters** node. Note that the value you set is the one that is used in the expressions where the variable is found. Some examples of variables that may be needed are

- t, the requested output time

- **timestep**, the time step used by the solver, for time-dependent problems
- **freq**, the frequency to assemble the problem for, for frequency-dependent problems.

There are also other variables that might be needed: **phase** (the phase), **niterCMP** (the nonlinear iteration number), and **CFLCMP** (a pseudo-time-stepping control variable).

ELIMINATED OUTPUT

In this section you can choose to output matrices and vectors that are passed to the linear solvers—that is, where constraints have been eliminated—by enabling one or several of the following check boxes: **Eliminated load vector**, **Eliminated stiffness matrix**, **Eliminated damping matrix**, **Eliminated mass matrix**, **Constraint null-space basis**, **Constraint force null-space basis**, **Particular solution (ud)**, and **Scale vector**.

NON-ELIMINATED OUTPUT

In this section you can choose to output matrices and vectors that the solver assembles before the elimination step by enabling one or several of the following check boxes: **Load vector**, **Stiffness matrix**, **Damping matrix**, **Mass matrix**, **Constraint vector**, **Constraint Jacobian**, and **Constraint force Jacobian**.

OPTIMIZATION OUTPUT

In this section you can choose to output matrices and vectors assembled during optimization by enabling one or several of the following check boxes: **Optimization constraint Jacobian**, **Optimization constraint vector**, **Lower bound constraint vector**, and **Upper bound constraint vector**.

ADVANCED

If you want to assemble an eigenvalue problem you can set the **Eigenvalue name** (default: `lambda`) and the **Value of eigenvalue linearization point** by first selecting the **Set eigenvalue name** check box.

LOG

This section, which is initially empty, contains a log from the assembling.

Compile Equations

In the **Compile Equations** () node's settings window you specify which study and study step to use when computing the current solver configuration and compiling the equations to solve. The node displays the name of the selected study step: **Compile**

Equations: Stationary, for example. Right-click this node and select  **Statistics** to see the number of degrees of freedom for the solver (see [The Statistics Page](#) below).

STUDY AND STEP

Specify the study in the **Use study** list and the study step in the **Use study step** list. By default you get the parent study and its first study step.

SPLIT COMPLEX VARIABLES

Complex variables are by default represented by complex-valued degrees of freedom. By selecting the **Split complex variables in real and imaginary parts** check box, the representation of complex variables is changed to using separate real degrees of freedom for the real and imaginary parts. The split representation can improve convergence where nonanalytic functions of complex variables are used in equations. Using a split representation also makes it possible to avoid complex pollution (a small nonzero imaginary component) of real variables by specifying a *real* or *complex* value type for variables. If you use a split representation of complex variables, specify the value type of dependent variables in the **Discretization** sections in the settings windows for the main physics nodes.



The split representation enables a correct evaluation of Jacobians for the following operators: `real`, `imag`, `conj`, `abs`, and `realdot`.

The Statistics Page

Use the **Statistics** page to view statistics about a solver, its dependent variables, and their number of degrees of freedom. This can be done before solving the problem and is useful for determining which variables are the most costly to compute and store. To open the **Statistics** page for a solver, right-click the **Compile Equations** node () and select  **Statistics**. This page contains the following section:

NUMBER OF DEGREES OF FREEDOM

Here you see a list of the dependent variables and their number of degrees of freedom (DOFs) as well as the total number of DOFs. The list includes both variables solved for and variables not solved for.



Computing this statistics requires a computation of the size of the assembled finite element model. This can take some time for large models.

Input Matrix

The **Input Matrix** node () can be used to create the raw data of an assembled matrix or vector from Java®. Right-click the **Eigenvalue Solver**, **Stationary Solver**, or **Time-Dependent Solver** nodes and select **Input Matrix**. Select the appropriate check boxes for the matrices and vectors you want to input and save the model as a Model Java-file. The saved Java-file now contains code for inputting the selected matrices and vectors. For information about the eliminated system, see [Elimination Constraint Handling](#).

INPUT

In this section you can choose to input matrices and vectors that are passed to the linear solvers by enabling one or several of the following check boxes: **Load vector**, **Stiffness matrix**, **Damping matrix**, **Mass matrix**, **Constraint vector**, **Constraint Jacobian**, and **Constraint force Jacobian**.



The matrices and vectors that you input using the **Input Matrix** node replace the corresponding matrices and vectors in the assembled system from the model in COMSOL Multiphysics.

Store Solution

This is a utility node whose purpose is to make it possible to access intermediate solution results. By default, the software only stores the solution at the last computed node of a solver configuration. Add a **Store Solution** node () anywhere in a solver configuration to make the software store the solution at that point of the sequence in addition to the solution at the last computed node. You can use the solution from a

Store Solution node when analyzing the results and as initial values (via a study-type node) for other computations.

GENERAL

This section contains the name of the solver sequence that the **Store Solution** node is using.

LOG

This section is initially empty. It contains information if the **Store Solution** node is used to store parametric sweep data.

State Space

The **State Space** node () provides access to the raw data of a PDE in state-space form. To create state-space data, right-click the solver node and select **Other>State Space**. Specify the input and output and the state-space matrices and vectors that you want to access. Then save the model as a Model Java-file. The saved Java®-file now contains code for assembling the selected matrices and vectors that can be used to access the matrix rows, columns, values, and so forth.

INPUT

In the **Input parameters** field, enter all parameters that affect the model as space- or comma-separated entries.

OUTPUT

The state-space node assembles matrices that describe a model as a dynamic system when **Off** is selected from the **Static** list:

$$\begin{aligned} M\dot{c}x &= MAx + MBu \\ y &= Cx + Du \end{aligned}$$

If you select **On** from the **Static** list, a static linearized model of the system is described by

$$y = (D - C(MA)^{-1}MB)u$$

In the **Output expressions** field, enter all expressions that are to be evaluated as output from the model as space- or comma-separated entries. Select any of the **MA**, **MB**, **D**, and **C** check boxes, and if **Static** is set to **Off**, any of the **MC**, **Null**, **ud**, or **x0** check boxes. **Null** is the PDE constraint null-space matrix, and **ud** is a particular solution fulfilling the

constraints. **x0** is the initial data. The solution vector U for the PDE problem can then be written

$$U = \text{Null}x + ud + u0$$

where $u0$ is the linearization point, which is the solution stored in the sequence once the state-space export node is run. The input linearization point is stored in the sequence after the state-space node is run.

LOG

This section, which is initially empty, contains a log from the run of the **State Space** node. This log is stored in the Model MPH-file. Select the **Keep warnings in stored log** to keep warning messages in this log so that the information in those warnings is available also when reopening the model.

Harmonic Perturbation, Prestressed Analysis, and Small-Signal Analysis



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 these study types for details about availability by module and physics:

- Frequency-Domain, Perturbation
- Small-Signal Analysis, Frequency Domain
- Prestressed Analysis, Eigenfrequency
- Prestressed Analysis, Frequency Domain
- AC Impedance Stationary
- AC Impedance Time Dependent



- 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

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 nodes, and this relates to the

Harmonic Perturbation node, which can be added to a wide variety of physics interface nodes (for example, the **Electric Potential** and **Electric Ground** nodes for the **Electric Currents** interface).



For plot settings made available by using this study, see [Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings](#).



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](#).

Harmonic Perturbation—Exclusive and Contributing Nodes

A physics node that is exclusive has a **Harmonic Perturbation** subnode. This subnode 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.



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.

Nodes 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 node 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 subnode. This subnode cannot, in general, be a full copy of the original node 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 nodes 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.

-
- 
 - [Physics Exclusive and Contributing Node Types](#)
 - [Key to Nodes and Toolbar Buttons](#)
 - For different plot settings made available, see [Harmonic Perturbation—Exclusive and Contributing Nodes](#).
-

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.

Job Configurations

The **Job Configurations** node () is visible when you click the **Show** button () and select **Advanced Study Options**. The node automatically displays if the **Job Configuration** node has content.

There are three categories of job configurations, each of which corresponds to a model node:

- Parametric
- Batch
- Cluster Computing (Job Configuration)

The first two categories share the common set of subnodes listed in [Table 19-6](#). In addition, a **Batch** node has a default **Batch Data** subnode that in turn stores **External Process** nodes, which contain information about batch jobs. **Cluster Computing (Job Configuration)** nodes support no subnodes and must point to a **Batch** node.

TABLE 19-6: PARAMETRIC AND BATCH SUBNODES

NAME	ICON	PURPOSE
Job Configuration		Runs a Parametric, Batch, or Cluster Computing job node.
Solver		Runs a solver configuration.
External Class		Runs the main method of an external Java® class.
Geometry Sequence		Runs a geometry sequence.
Meshing Sequence		Runs a mesh sequence.
Save Model to File		Stores a model in the state that it is at that point in the job configuration.
Plot Group		Runs a Plot Group node.
Derived Value		Runs a Derived Values node.
Export to File		Runs an Export node and saves it on file.

TABLE 19-7: JOB CONFIGURATIONS NODE - CONTEXT MENU OPTIONS

NAME	ICON	PURPOSE
Show Default Solver		Shows the default job configuration node (if any) that corresponds to the study step nodes in the study.
Parametric		Adds a Parametric node.
Batch		Adds a Batch node and an associated Batch Data subnode.
Cluster Computing (Job Configuration)		Adds a Cluster Computing node.
Delete Solvers		Deletes all jobs under the Job Configurations node.

Parametric

Parametric () is one of the main nodes. It has the ability to loop over a given set of parameters. For each set of parameters it runs the sequence defined by its subnodes. You can combine the sequence with other **Batch**, **Parametric**, or **Cluster Computing** sequences in a hierarchical way by adding a job configuration that points to another node. You can, for instance, create a **Parametric** node that does a LiveLink™ update and then runs a **Cluster Computing** node that in turn runs a second **Parametric** sequence on another node.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

You define the parameters in the **Parameter names** and **Parameter values** fields. The parameters can be loaded from file by clicking the **Read File** button when you have selected the file through the **Load Parameter Values** dialog box, which you open by clicking **Browse**. You can add a **Stop condition** that is evaluated after each solution. Once the condition evaluates to a negative value the **Parametric** node is stopped.

RESULTS WHILE SOLVING

Check the **Plot** check box to allow plotting of results while solving. Select what to plot and when from the **Plot group** list. The data set of the selected plot group is plotted as soon as the results become available.

Use the **Probes** list to select any probes to evaluate. Use the **Accumulated probe table** to accumulate data during a sweep. The accumulation is over solver variation (time, frequency, and so forth) and variation over the parametric sweep. For independent variation of parameters you can use the accumulated table with the **Format: Filled** to change the table data into a matrix format that can be used for response surface plots.

ERROR

Errors are usually stored in the **Error** table. If you want to get the error message at once, select the **Stop if error** check box.

CLUSTER SETTINGS

You can distribute the sweep on several computational nodes by selecting the **Distribute parameters** check box. If the problem is too large to run on a single node, enable the **Maximum number of groups** field to use the nodes' memory more efficiently. In this case the same parameter is solved for by several nodes that cooperate as if running a non-distributed sweep. The number of nodes that cooperate is equal to the maximum of the total number of nodes divided by the **Maximum number of groups** setting and 1. So if the total number of nodes is 12 and the **Maximum number of groups** is 3, 3 groups with 4 nodes each cooperate.

Job Configuration

You can add a **Job** node () to run another job configuration. You select the sequence to run from the **Run** list. Recursive calls are detected and cause errors.



Using a Job Configuration to Store Parametric Results on File

Solver

The **Solver** node () runs a solver configuration. Select the sequence to run from the **Sequence** list. The default behavior is to run all solver configurations. Store a copy of the solution once it has run by selecting the **Keep all solutions** check box. The name of the copy is generated from the current sequence name and parameter values. The default is to clear previous solutions. To disable it, clear the **Clear previous** check box.

External Class

The **External Class** node () runs the main method of an external compiled Java® class file. You add it by right-clicking a **Job Configurations** node and selecting it from the **Other** submenu. Before the external class is called the system property `cs.currentmodel` is set to the name of the model that is calling the external class. You can set the name of the class file in the **Filename** field. Arguments can be passed to the main method with **Input**.

Geometry Sequence

The **Geometry Sequence** node () runs a geometry sequence. You add it by right-clicking a job configuration node and selecting it from the **Other** submenu. You select the sequence to run from the **Run** list. The default is to run all geometry sequences.

Meshing Sequence

The **Meshing Sequence** node () runs a mesh sequence. You add it by right-clicking a **Job Configurations** node and selecting it from the **Other** submenu. You select the sequence to run from the **Run** list. The default is to run all meshing sequences.

Save Model to File

The **Save Model to File** node () stores a model in the state it is at that point in the job configuration.

GENERAL

The **Overwrite previous model files** check box is selected by default. This means that previous models with the same name are overwritten. If a parametric sweep is running, the model is given a unique name based on the current parameter values. The **Add parameters to filename** check box is selected by default. Click to clear it to use the same name for each parameter value. Enter a **Filename** including its network path, or click **Browse** to navigate to the location on your network where you want to store the model.

OUTPUT

The names of the saved models are stored in the table under **Output** where the **Filename** and **Parameters** are listed. Open a saved model in a new instance of COMSOL Multiphysics by selecting an **Open file** from the list or by clicking the **Open** button.

Plot Group

The **Plot Group** node () runs a sequence of plot groups, creating a plot in the Graphics window. You add it by right-clicking a **Job Configurations** node and selecting it from the **Results** submenu. You select the sequence to run from the **Run** list. The default setting, **All**, runs all plot group nodes. You can use an **Export to File** node to store the resulting plot in a file.

Derived Value

The **Derived Value** node () runs a Derived Values node defined in the **Results** branch of the model tree. You add it by right-clicking a **Job Configurations** node and selecting it from the **Results** submenu. The computed value is stored in a **Result** table under the **Derived Value** node.

GENERAL

You select the **Derived Values** node to run from the **Run** list. The default behavior (**All**) is to run all **Derived Values** nodes.

RESULT

The **Table** setting decides which **Table** under **Results** to store the computed values in. The default is **New** for a new table. Clear the **Clear previous** check box if you want to add the computed values in each step of the job instead of clearing the values from an earlier sweep. This is useful if you want to add the values for additional parameters to an already existing table. The **Parameters** column in the **Result** table contains the parameters that computed the numerical value, the **Value** column contains the numerical value, and the **Derived values** column contains the name (tag) of the **Derived Values** node that computed the numerical value. The information about the derived values nodes is useful when you have selected **All** from the **Run** list and the computed values come from different **Derived Values** nodes.

Export to File

The **Export to File** node () runs an Export node defined in the **Export** branch of the model tree. You add it by right-clicking a job configurations node (**Batch**, for example) and selecting it from the **Results** submenu. The file is stored with a unique name that is generated from the current parameter values and the filename that the Export node has set.

GENERAL

Use the **Run** setting to select the Export node to run. The default behavior (**All**) is to run all Export nodes.

FILE

In the **File** section you set if COMSOL Multiphysics should overwrite files with the same name or if an error should occur, for example. If you clear the **Clear previous** check box, the job adds the values in each run instead of clearing the previous value. If you clear the **Add parameters to filename** check box, COMSOL does not create unique filenames for the current parameter values. This is useful if you want to start batch jobs with different parameter values from the command line and use the resulting file for further postprocessing.

OUTPUT

In the **Output** section you find the names of the files created during a sweep. You can select to **Open** a file by clicking the button. The file then opens in a web browser. The **Parameters** column in the **Output** table contains the parameter names, and the **Filename** column contains the corresponding filename.

Batch

The **Batch** node () is the main node for running batch jobs. Batch jobs run in separate processes. You can therefore continue working in the COMSOL Desktop® once a batch job is run. Model changes in the COMSOL Desktop after the batch job is submitted do not affect the model in the batch job. As in the **Parametric** node the batch job is defined by a number of subnodes. The batch job then runs each subnode. Use the **Save as Default** button in the toolbar to save the current directory setting as the default directory for batch files. The **Batch** job has a special **Batch Data** subnode, which collects **External Process** subnodes containing job status information.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

You set the number of cores the batch job should use in the **Number of cores** field. The default behavior is to use all available cores. If you set **Number of simultaneous** jobs to more than 1, several jobs can run at once. When you are running more than 1 job at once it is important that the product of **Number of cores** and **Number of simultaneous** jobs does not exceed the number of cores available on the computer. Otherwise you

experience performance degradation. When you run multiple batch jobs on your computer COMSOL Multiphysics makes sure this does not happen if you are using the automatic setting. You can set a **Start time** if you want the batch process to start a later time. Select the hour to start the run in **Start time**. Select the **Use graphics** check box when the batch process should run results nodes that creates graphical contents such as exporting to file. Enter the **Number of job restarts**. The default is 0. This is the maximum number of times the job can be restarted if it fails to complete. Enter a value for the **Alive time (seconds)**. The default is 300 seconds. This is the longest time the process is allowed to run before it must inform that it is still running. Failure to do so means that the process is considered dead and a new process is started if not the maximum number of job restarts is reached.

FILES

Set the **Filename** of the model. If the batch job is generated from a parametric sweep a unique name that depends on the parameter names and values is created. The default is to overwrite any previous models with the same name. Disable the default by clearing the **Clear previous model** check box. Select the **Clear meshes** check box to clear the meshes before running the batch sweep. The default is to not clear the meshes. Select the **Clear solutions** check box to clear the solutions before running the batch sweep.

Specify the **Directory** to store the model. If the batch job has another path to the directory you select the **Specify external COMSOL batch directory path** check box and enter the path to the batch **Directory** or **Browse** for the path. If COMSOL is installed in a different directory from where the batch job runs, enable the **Specify external COMSOL installation directory path** and specify the install directory (click **Browse** or enter the path to the **Directory**). This can occur if you are submitting jobs to a job scheduler with the **Cluster Computing** node. Click the **Save As Default** button () in the settings window to save the current directory setting as the default directory for batch files.

SYNCHRONIZATION

Select the **Synchronize solutions** check box to synchronize the solutions computed by the batch processes with the model. This allows additional postprocessing after the sweep has finished. The default is to disable solution synchronization. Select the **Synchronize accumulated probe table** check box to synchronize the accumulated probes computed by the batch processes with the model. The accumulated probe synchronization is enabled by default. Select the **Output model to file** check box to enable that all batch processes save the models to file. In most cases, use the solution synchronization and probe synchronization functionality instead because otherwise the data ends up in one file for each process and cannot be postprocessed efficiently.

Use the **Probes** list to select probes to update during the batch sweep. The default is **All**, which selects all probes for plotting and tabulation of probe data. Select **Manual** to open a list with all available probes. Use the **Move Up** (↑), **Move Down** (↓), **Delete** (✖), and **Add** (✚) buttons to make the list contain the probes that you want to see results from while solving. Select **None** to disable probe updating for batch sweep.

Select the **Accumulated probe table** check box to activate the accumulation of probe updates for both the variation on the solver level (time, frequency, and so forth) and on the batch sweep level. Use the **Output table** to select where to put the data. Select the **Use all probes** check box if all the model probes should be accumulated in the table. If not checked the probes selected by the **Probes** selector is used.

CLIENT SETTINGS

The **Batch** node can also be used as a client to drive a server on another machine. You enable the client functionality by selecting the **Client** check box. You can then set the **Host name** (default: localhost) and **Port number** of the server for the batch job (default: 2036) to connect to. This number is the default port number. If the server you want to connect to is using another port, then edit this number accordingly.

Batch Data

The **Batch Data** node (⌚) contains information about the batch processes that have been started by the **Batch** node (⌚). Each process is represented by an **External Process** (⌚) subnode. Click the **Attach Job** button (⌚) on the settings window's toolbar to display the progress of all the external processes. The GUI enters a progress mode in order to follow the progress of the external processes. You can at any time stop the progress mode by clicking the **Detach Job** button.



The External Process Window

External Process

The **External Process** nodes (⌚) under a **Batch Data** node contain information about the batch processes that have been started by the **Batch** node. Each **External Process** node is associated with a started batch job. The **Start command** field contains the command that was used to start the batch job. The **Filename** contains the filename of the model that is used in the batch job. When the batch job has finished you can open

it using the **Open** button. You **Update log**, **Attach Job** of the external job, **Cancel** the process, **Stop** the process, **Clear status** of the external process, and **Rerun** the job, by choosing the **Operation** and **OK** buttons. The status of the operation is viewed in the status. The log is updated when you choose **Update log**. When **Attach Job** is clicked, the COMSOL Desktop® enters a progress mode to follow the progress of the external process. You can at any time stop the progress mode by clicking the **Detach Job** button.



The External Process Window

Cluster Computing (Job Configuration)

The **Cluster Computing** node () is useful when you want to submit a batch job to a job scheduler want to run the model in distributed mode as a batch job. When you have specified the cluster computing settings, click the **Save as Default** button () in the settings window's toolbar to save the current setting as default. These settings are saved to the **Options>Preferences>Cluster computing** section.

GENERAL

Use the **Defined by study step** list to specify if the settings are synchronized with the corresponding study step.

If you want to use a job scheduler or do some other operations before or after the job you set the command line in the **Prepend command** and **Postpend command** fields. You can define a default command line with the system properties `cs.precmd` and `cs.postcmd`. If the command line contains `{nn}` or `{perhost}` they are replaced by the values in the **Number of nodes** field and **Number of processes** field, respectively.

From the **Batch job** list, select which batch job to submit. Click the **Go to Source** button () to move to the settings window for the selected **Batch** node.

CLUSTER SETTINGS



After making these settings, click the **Save as Default** () button on the settings window toolbar to save the current directory settings as the default preference.

Choose the **Cluster type**—**General** (the default), **HPCS 2008**, **WCCS 2003**, **OGS/GE**, or **Not distributed**:

General

Select **General** (the default) to configure to run on many types of clusters and schedulers, including Linux clusters.

- When **General** is selected, and you have started a multiprocessor daemon (MPD) on the computer, click to select the **MPD is running** check box.
- The entry in the **Host file** field specifies the host file used for the job. If left empty, MPD looks for a file `mpd.hosts` in the Linux home directory.
- Select which bootstrap server that should be used by MPI using the **Bootstrap server** setting.
- If your cluster is Linux and it requires that an ssh (secure shell) or an rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol.
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (default is 1 node).
- Enter the **Number of processes on host**. The default is 1.

HPCS 2008

Select **HPCS 2008** to use the Windows HPC Server 2008 job scheduler to submit the batch job.

- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (the default is 1 node).
- Select a **Node granularity**—**Node** (the default), **Socket**, or **Core**. **Node** allocates one process on each host, **Socket** allocates one process on each socket, and **Core** allocates one process on each core.
- The **Exclusive nodes** check box is selected by default. Click to clear if you want to run on nodes shared by other users.

Under **Advanced**:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- Set the names of **Requested nodes**. The job scheduler only allocates jobs on the nodes listed by you.

- Enter the **Node group**. The job scheduler only allocates jobs on the nodes belonging to the group.
- Enter the minimum required **Cores per node**. The default is 0. The job scheduler only allocates jobs to nodes with at least as many cores as set.
- Enter the minimum required **Memory per node (MB)**. The default is 0. The job scheduler only allocates jobs to nodes with at least as much memory as set.
- Enter the **Runtime (minutes)** before the job is canceled. The default is **Infinite**.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Select a **Priority**—**Highest**, **Above normal**, **Normal** (the default), **Below normal**, or **Lowest**—for the scheduled job.

WCCS 2003

Select **WCCS 2003** to use the Windows Compute Cluster Server 2003 job scheduler to submit the batch job.

- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Enter the **Number of nodes** (physical nodes) to use (the default is 1 node).
- The **Exclusive nodes** check box is selected by default. Click to clear as required.

Under **Advanced**:

- The entry in the **Scheduler** field is the IP address of the enterprise adapter of the head node or the DNS name of the head node. The default is `localhost`.
- Set the names of **Requested nodes**.
- Enter the **Runtime (minutes)** before the job is canceled. The default is **Infinite**.
- The entry in the **User** field is the user account that COMSOL uses for submitting the job. You provide the password in a separate command window that opens at execution time with the possibility to save the credentials.
- Select a **Priority**—**Highest**, **Above normal**, **Normal** (the default), **Below normal**, or **Lowest**—for the scheduled job.

OGS/GE

Select **OGS/GE** to use the Open Grid Scheduler/Grid Engine job scheduler to submit the batch job.

When **OGS/GE** is selected:

- Select the **Bootstrap server** that should be used by MPI.
- If your cluster is Linux and it requires that an ssh (secure shell) or an rsh (remote shell) is installed in a uncommon directory, use the **Rsh** field to set the rsh communication protocol.
- If you must provide extra arguments to MPI use the **Additional MPI arguments** field.
- Select a **Slot granularity**—**Host**, **Slot**, or **Manual**—to specify if COMSOL should parallelize on the physical **Host** level or on the OGS/GE-allocated **Slot** level. For **Host** and **Slot** specify the **Number of slots** to allocate. The **Manual** setting can be used to control the granularity more. In this case set the number of computational nodes to use in the **Number of nodes**. For **Slot** and **Manual** the number of processes on each node is set in the **Number of processes on host** field; usually this is 1.
- Enter the **Queue name** to set the name of the Sun Grid Engine.
- The Sun Grid Engine priority is set in the **Priority value** field. The default is 0.

Not Distributed

Select **Not distributed** when you want to submit a batch job to a job scheduler without running a distributed job.

REMOTE AND CLOUD ACCESS

See [Remote and Cloud Access](#) described for [Cluster Computing \(Study\)](#).

Using a Job Configuration to Store Parametric Results on File

As an example of the use of a job configuration, add a **Parametric** node () under **Job Configurations** () to run a parametric study, storing the model and associated data and plots for each parameter step to individual files. This is useful if you, for example, want to:

- Avoid large model files while running large parametric sweeps.
- Store information in individual output files instead of in one large file.
- Control several file outputs directly from the COMSOL Desktop®: Model MPH-files, data files (text files), and image files.

The following steps describe the most important parts of setting up a job configuration to accomplish the desired file output:

- 1 Start with a model that does not contain any parametric sweep and define your outputs. Then add them to the **Export** node (by right-clicking and adding **Data**, **Plot**, and image nodes as desired) and assign each data or plot output to a file.
- 2 Create a parametric sweep study by right-clicking the study node and adding a **Parametric Sweep**. Add the parameters you want to sweep over in Parameters under Global Definitions. Now add the parameters in **Parameter names** in the parametric sweep node you created and set the **Parameter values** you want to sweep over. In some cases COMSOL Multiphysics chooses to use the more efficient parametric solver when sweeping. Set the **Use parametric solver** under **Study Extensions** to **Off** if you want to avoid this. Use the **Keep solutions in memory** setting **Only last** to conserve memory. In that case use the Accumulated Probe table to store the data you want to use for later processing. You can also choose to store the models on file by enabling the **Save each solution as model file** and then enter a filename in the **Filename** field, or click **Browse** to choose a name and location for the model files. The models created during the simulation can then be found in a **Save model to File** node under **Job Configurations**. The default settings are configurable in the Preferences dialog box.
- 3 If you want to create output files with the parametric sweep for further analysis, right-click the main study node and select **Show Default Solver**. Select the parametric node under **Job Configurations** that corresponds to the parametric sweep you want to export data from. You can see which parametric sweep the node corresponds to from the **Defined by study step** list. Note that if you run a stationary study, you need to switch off **Use parametric solver** in the **Study Extensions** section (see above) to get the node under **Job Configurations**.
 - Right-click the **Parametric** node and select **Results>Plot Group** if you want to run a **Plot Group** for each parameter value in the sweep. In most cases use the **Plot** settings under **Results While Solving** (in the study step's settings window) instead.
 - Right-click the **Parametric** node and select **Results>Derived Value** to run a **Derived Values** node for each parameter value in the sweep. This functionality is similar to probes but is useful if you have already set up a **Derived Values** node that you want to use during the sweep. The values are stored in a table (similar to probes) for further processing.
 - Right-click the **Parametric** node and select **Results>Export to File** if you want to export data to a file for each parameter value in the sweep, running an **Export** node under **Results**. Note that after the sweep, the files created are listed in the node and can be opened in a browser using the **Open** button.

ADVANCED JOB CONFIGURATIONS

Because a job configuration defines a sequence of steps you can create highly advanced models that use different solvers and sweeps as input to each other and performs different types of postprocessing during a sweep. Here are some suggestions:

- Create a parametric job configuration that uses two solvers. The first solver is used as input to the second solver. One way to create such a sweep is to create two studies. Let the second study use the first as input from the **Values of Dependent Variables**. Create a **Parametric Sweep** in the first study and select to keep **Only last** solution. Run **Show Default Solver** in both studies and enter the **Job Configurations** node of the first study. Right-click the **Parametric** node and add a **Solver**. Select the new solver and set it to run the second solver from the second study. Also check the **Keep all solutions** settings. Right-click the **Parametric** node and select **Run**. You can add further functionality to the sequence such as **Export to File** to suit your needs.
- Another possibility is to create a parametric sweep that runs a **Cluster Computing** node. This gives you similar functionality as the distributed parametric sweep, but the results are stored in separate files. Note that this requires several licenses, one for each process running simultaneously. One way to create this is to add a **Cluster Sweep** node by right-clicking the study. This node creates such a sweep automatically. It also sets up synchronization of solutions and accumulated probe tables when the synchronizations is enabled. Another, more complicated, way is to add a **Cluster Computing** node by right-clicking the study. Note that you need the **Advanced Study Options** enabled. If you added the **Cluster Computing** node then run **Show Default Solver** and select the **Job Configurations** node. Right-click and add a **Parametric** node. Right-click the new **Parametric** node and enable it. Right-click again and add a **Job** node. Set the job sequence to point to the **Cluster Computing** node. This sets up the new **Parametric** node to start a new process for each parameter. After you have set the parameters that you want to compute for, right-click and select **Run**. The resulting models are stored in **External Process** nodes under the batch job run by the cluster computing node. In order to get exported data during the runs you can use the functionality **Export to File** described above or enable the synchronization of solutions and accumulated probe tables in the **Batch** node.
- Create a parametric sweep that runs a solver and a class file that uses the COMSOL API to modify the solution. This can be useful if you want COMSOL to communicate with another program. You do this by adding a **Parametric Sweep** node to the study that you want to use. Run **Show Default Solver** and enter the **Job Configurations** node. Right-click the **Parametric** node and select **Other>External Class**. You can also modify the **Parametric** node by, for example, adding a **Stop condition**.

Results Analysis and Plots

This chapter describes the functionality for visualization and analysis of simulation results in COMSOL Multiphysics®.

Postprocessing and Analyzing Results

The **Results** branch in the COMSOL Multiphysics model tree contains tools for postprocessing and analyzing the results from your simulations, including visualizations, animations, and data analysis. The **Results** branch groups the tools into the following categories:

- **Data Sets** contain the source of data for plotting, for example, by indicating a solution and geometry or by transforming another data set (for combining solutions or evaluating data along a cut line, for example). See [Data Sets](#).
- **Plot Group** and plots: A plot group is a collection of plots to display simultaneously in the **Graphics** window. The plot groups include 1D plots (graphs), 2D plots (surface plots, for example), and 3D plots (volume plots, for example) with many different plot types and options. You can enable or disable plots in a plot group to determine the most applicable final image for your model or project. The physics create suitable default plots grouped in descriptive plot groups. See [Plot Groups and Plots](#).
- **Derived Values** and **Tables**: Derived values define the evaluation of, for example, values of integrals, values of variables in specific locations, and global variables. The evaluation results are stored in Table nodes under **Tables** and displayed in the **Table** window. See [Derived Values and Tables](#).
- **Export**: You can export a variety of data, images, and animations. See [Exporting Data and Images](#).
- **Reports**: Create reports as HTML and Microsoft Word documents that contain settings, selections, comments, plots, and other information about the model. See [Reports](#).
- **Views**: To display the **Views** node under **Results** (for 2D or 3D models), click the **Show** button () and select **Advanced Results Options**. See [User-Defined Views](#).

Results



Results in the *COMSOL API Reference Manual*

About the Results Branch

The **Results** branch contains tools and functionality for postprocessing and visualizing of the results. The main **Results** node contains all the nodes that you create for such purposes. The **Result** node's settings window contains the following section:

RESULT SETTINGS

The default is to update all plots automatically when you, for example, recompute the solution, click the plot node, or change the color table for a plot. Clear the **Automatic update of plots** check box to keep the plots unchanged until you explicitly update the plot using the **Plot** button (☞), for example. This can be useful, for example, for large models with complex plots where you do not want to update the plots directly when opening the model or when solving. A blue asterisk in the upper-right corner of the plot node's icon indicate that the plot is not updated (☞*).



There are preference settings that you can use to avoid automatic updates of plots when opening or creating models. See [Results Preferences](#) in the section about preference settings.

THE MAIN RESULTS ANALYSIS AND VISUALIZATION TOOLS

During results analysis and visualization there are these main operation types:

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.

Plot Groups and Plots A plot group is a collection of plots 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 user interfaces

create suitable default plots grouped in descriptive plot groups. Use a combination of data sets and plot groups to create cross-section plots.

Derived Values and Tables Derived values define the evaluation of integrals, maximum and minimum values, values of variables in points, and values of global variables. The evaluation results are stored in tables and displayed in the Table window.

Exporting Data and Images Export data, images, and animations from plot groups to files or use a player to visualize dynamic data.

Reports Create reports for documenting models as reports with text and images on an HTML format for easy viewing in a web browser or as Word files.

To display the **Views** node under Results, click the **Show** button () and select **Advanced Results Options**.

Common Results Node Settings

Under **Results** there are common sections on the settings windows. [Table 20-1](#) provides cross references to the information relevant to these nodes under Results, although the same section may also be available for other nodes throughout COMSOL. For the **Coloring and Style** section, see [Table 20-2](#).

COMMON BUTTONS ON THE SETTINGS WINDOWS

The following buttons are available on many of the settings windows and are mostly self explanatory. These are not explicitly described or explained for every node.

- In general, use the **Move Up** (), **Move Down** (), or **Delete** () buttons and the fields under tables to edit the table contents. Or right-click a table cell and select **Move Up**, **Move Down**, or **Delete**.
- At any time during plot creation, click the **Plot** button () to preview a data set or plot. Or right-click the node and select **Plot**.
- Click the **Add to Selection** (), **Remove from Selection** (), and **Clear Selection** () buttons when working with geometric entities in the selection windows and when required.
- Click the **Range** button () to define a range.

- Click the **Go to Source** button () to move to the node that the selection in the list next to the button refers to.
- Click the **Evaluate** button () or right-click the **Derived Values** node and select **Evaluate All** () or **Clear and Evaluate All** ().



- [Going to the Source Node](#)
- [About Selecting Geometric Entities](#)
- [Key to Nodes and Toolbar Buttons](#)
- [Entering Ranges and Vector-Valued Expressions](#)

LINKS TO COMMON SETTINGS WINDOW DESCRIPTIONS

TABLE 20-1: DETAILS FOR THE COMMON SETTINGS SECTIONS

SETTINGS WINDOW SECTION	LINK TO MORE INFORMATION
Arrow Positioning	Arrow Positioning
Axis Data	Entering Axis Data for a Data Set
Color (3D plot group, Far Field plots)	Expressions and Predefined Quantities
Coloring and Style	See Table 20-2 below and Defining the Coloring and Style
Data (for Plots)	Selecting a Data Set for Plots
Data (for Derived Values and Export)	and Vector Inputs for Parametric Solver and Parametric Sweep Studies
Data Series Operation	Data Series Operation Settings for a Derived Value
Data for Parametric Solver and Parametric Sweep studies	Vector Inputs for Parametric Solver and Parametric Sweep Studies
Element Filter	Defining Element Filters
Expression or Expressions	Expressions and Predefined Quantities
Inherit Style	Inheriting Style Options
Integration Settings	Integration Settings for a Derived Value
Legends	Legends
Levels	Defining the Number of Levels
Node Properties	Node Properties for Reports

TABLE 20-1: DETAILS FOR THE COMMON SETTINGS SECTIONS

SETTINGS WINDOW SECTION	LINK TO MORE INFORMATION
Parametric Solver and Parametric Sweep studies	Vector Inputs for Parametric Solver and Parametric Sweep Studies
Plane Data	Defining Plane Data for a Data Set
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
Selection	About Selecting Geometric Entities
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 (ID plots)	Expressions and Predefined Quantities

COLORING AND STYLE

TABLE 20-2: 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

TABLE 20-2: CROSS REFERENCES FOR THE COMMON COLORING AND STYLE SETTINGS SECTIONS

SECTION	CROSS REFERENCE
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

	<ul style="list-style-type: none"> • Plot Groups and Plots • Derived Values and Tables • Studies and Solvers • Entering Ranges and Vector-Valued Expressions
--	--

Selecting a Data Set for Plots

Almost every plot type's settings window includes a **Data** section where you select a **Data set** from a list of available and applicable data sets. **From parent** (the default) 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 data set node that the selection in the list next to the button refers to.

FOR SOME ID PLOTS

Under **Data** select a **Data set**. 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.
-



Data Sets

Entering Axis Data for a Data Set

Revolution 1D and 2D data sets: Specify the revolution axis or point by a method based on the space dimension.

- For a **Revolution 1D** data set, enter a value in the **x** field to specify the revolution point.
- For a **Sector 2D** data set, enter values for both the **x** and **y** coordinates (SI unit: m).
- For a **Mirror 2D**, **Revolution 2D**, or **Sector 3D** data set, 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 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 (for the 2D data sets), and **x**, **y**, and **z** coordinates for the 3D data set (SI unit: m).
 - If **Point and Direction** is selected, enter **Point** and **Direction** vectors for **x** and **y** coordinates (for 2D data sets), and **x**, **y**, and **z** coordinates for the 3D data set (SI unit: m).

Vector Inputs for Parametric Solver and Parametric Sweep Studies

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, the associated parameter values are listed.

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 nodes, the time and eigenvalue settings are referred to as the inner solutions. Thus, in a graph plot for a parametric eigenvalue solution, for example, the “Solution” 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.

- Under **Data**, for time-dependent **Parametric Sweep** studies also select an option from the **Select via** list—**Stored output times** or **Interpolated times** (time-dependent models only).

The time steps cannot be selected directly because the different parametric solutions could have different time steps. Selecting **Stored output times** plots all of them, or select **Interpolated times** to get the same interpolated times for every parameter.

- If **Interpolated times** is selected, enter **Times** or click the **Range** button () to select and define specific times.

DERIVED VALUES

For **Parametric Solver** and **Parametric Sweep** studies, the **Parameter values** section lists the associated parameter values.

For **Parametric Sweep** studies select an option from the **Select via** list: **Stored output times** or **Interpolated times** (time-dependent models only). If the **Parametric Sweep** study has multiple inner solutions, such as time steps, select **Inner solutions** or **Outer solutions** from the **Table rows** list to control which solutions appear in the table rows.

The time steps cannot be selected directly because the different parametric solutions could have different time steps. Selecting **Stored output times** plots all of them, or select **Interpolated times** to get the same interpolated times for every parameter.

- If **Interpolated times** is selected, enter **Times** or click the **Range** button () to select and define specific times.

- When available, from the **Table rows** list, select **Inner solutions** or **Outer solutions**. These options are available when there is a Parametric Sweep problem with dynamic inner solutions (that is to say, time, eigenvalue, or parametric solutions).
 - If **Inner solutions** is selected, when the **Evaluate** button () is clicked the results table displays the *dynamic value* (for example, time, eigenvalue, or parametric) solutions in rows.
 - If **Outer solutions** is selected, when the **Evaluate** button () is clicked the results table displays the *parameters* in rows.



- Entering Ranges and Vector-Valued Expressions
- Derived Values and Tables

Expressions and Predefined Quantities

When plotting and evaluating results, COMSOL provides a large number of predefined quantities that are specific to the physics in the model as well as general quantities for the geometry, coordinate systems, and mesh.

COMSOL does not limit the results calculations to predefined quantities; you can plot and evaluate any function by entering the corresponding expression. You can combine numbers, parameters, mathematical constants, physical constants, variables, unary operators, and binary operators. 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. Any user-defined parameters appear as predefined quantities under **Definitions**.

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** list with geometry variables such as spatial coordinates and entity indexes.
- A **Mesh** list with mesh variables such as mesh element size and mesh quality.

EVALUATION OF UNDEFINED QUANTITIES

During the evaluation of expressions, by default COMSOL 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 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 (such as for a harmonic perturbation or a small-signal analysis), several options are available for how to evaluate the expression in the **Expression evaluated for** list.

Defining Plane Data for a Data Set

Select a **Plane type—Quick** (the default) to specify planes orthogonal to the coordinate axes or **General** to specify general planes. 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.

If **Quick** is selected:

- From the **Plane** list, select **xy-planes**, **yz-planes** (the default), **zx-planes**, **yx-planes**, **zy-planes**, or **zx-planes** as the set of planes orthogonal to the coordinate axes applicable for the model geometry. Specify the transverse coordinate by entering the location along the transverse coordinate axis.
- Enter the **x**-, **y**-, or **z-coordinates** in the field based on the **Plane** selection.
 - If **xy-planes** or **yx-planes** is selected, enter the **z-coordinates** (SI unit: m).
 - If **yz-planes** or **zy-planes** is selected, enter the **x-coordinates** (SI unit: m).
 - If **zx-planes** or **zx-planes** is selected, enter the **y-coordinates** (SI unit: m).

If **General** is selected:

- Select a **Plane entry method—Three points** or **Point and normal**. Enter **x**, **y**, and **z** coordinates.

For the **Mirror 3D** data set, select **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.

- If **Three points** is selected, enter **Point 1**, **Point 2**, and **Point 3** in the **x**-, **y**-, and **z**-coordinate fields (SI unit: m).
- If **Point and normal** is selected, enter **Point** (SI unit: m) and **Normal** (dimensionless) data in the **x**-, **y**-, and **z**-coordinate fields.

For the **Cut Plane** data set, 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. The **Distances** field refers to a direction that is normal to the cut plane.

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.



Using Special Formats and Symbols in Titles

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.

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 20-15](#), COMSOL 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 20-3: VALID HTML TAGS

HTML TAG	DESCRIPTION
 	Enclosed text is rendered using a bold font.
<I> </I>	Enclosed text is rendered using an italic font.
	Enclosed text is rendered in subscript with the enclosed text slightly lower than the surrounding text.
	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 20-4: VALID GREEK SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\ALPHA	A	\alpha	α
\BETA	B	\beta	β
\GAMMA	Γ	\gamma	γ
\DELTA	Δ	\delta	δ

TABLE 20-4: VALID GREEK SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\EPSILON	E	\epsilon	ϵ
\ZETA	Z	\zeta	ζ
\ETA	H	\eta	η
\THETA	Θ	\theta	θ
\IOTA	I	\iota	ι
\KAPPA	K	\kappa	κ
\LAMBDA	Λ	\lambda	λ
\MU	M	\mu	μ
\NU	N	\nu	ν
\XI	Ξ	\xi	ξ
\OMICRON	O	\omicron	\omicron
\PI	Π	\pi	π
\RHO	P	\rho	ρ
\SIGMA	Σ	\sigma	σ
\TAU	T	\tau	τ
\UPSILON	Y	\upsilon	υ
\PHI	Φ	\phi	ϕ
\CHI	X	\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 symbols:

TABLE 20-5: VALID MATHEMATICAL SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\approx	\approx	\bullet	\bullet
\sim	\sim	\partial	∂
\prop	\propto	\nabla	∇
\neq	\neq	\prod	\prod
\equiv	\equiv	\sum	\sum
\leq	\leq	\sqrt	\sqrt
\geq	\geq	\int	\int

TABLE 20-5: VALID MATHEMATICAL SYMBOL COMMANDS

COMMAND	SYMBOL	COMMAND	SYMBOL
\>	>>	\oplus	\oplus
\<<	<<	\otimes	\otimes
\plusmin	\pm	\larrow	\leftarrow
\infinity	∞	\rarrow	\rightarrow
\deg	$^\circ$	\larrow	\leftrightarrow
\cdot	\cdot	\ldarrow	\Leftarrow
\times	\times	\rdarrow	\Rightarrow

Arrow Positioning



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.
-



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 (

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 for plots and 5 for data sets). Otherwise, enter the values of the contour **Levels** or click the **Range** button (img alt="Range button icon" data-bbox="485 645 515 660") 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 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



[Color Tables](#) in the *COMSOL API Reference Manual*

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**.

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.



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.



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.



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; the maximum is 10,000 markers) 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).



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**.

Comet tail is available with the **Particle Tracing**, **Particle Tracing with Mass**, and the **Particle Trajectories** plots and requires a license for the Particle Tracing Module. See [Particle Tracing](#), [Particle Tracing with Mass](#), and [Particle Trajectories](#).



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.



Custom refinement applies to the base data set. The number of elements in the model can therefore increase radically if the plot uses, for example, a revolve data set, since 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](#)).

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, uxx is identically 0 if u is defined using linear elements. COMSOL Multiphysics 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 smooths 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 functionality 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.

Integration Settings for a Derived Value

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 Multiphysics 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.



For 2D axisymmetric models, **Volume Integration** and **Surface Integration** are available. COMSOL multiplies the expression (integrand) with $2\pi r$ prior to integration to compute the corresponding volume or surface integral if you do the next step.

- For **Surface Integration**, select the **Compute volume integral** check box.
- For **Line Integration**, select the **Compute surface integral** check box.

Data Series Operation Settings for a Derived Value

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**



The integrals and averages are computed using the trapezoidal method.

Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings



For details about the solvers and studies, including the availability by module, see [Harmonic Perturbation—Exclusive and Contributing Nodes](#).

For the plots that incorporate harmonic perturbation, small-signal analysis, or prestressed analysis, additional settings display in a variety of plot windows, in an **Expression evaluated for** list:

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.



See [Built-In 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 that 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 `lntotal` 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 `lntotalavg` 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 `lntotalrms` 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 `lntotalpeak` operator.

Node Properties for Reports

When applicable, the **Node Properties** section provide settings for including the node properties for all model nodes. Select the **Include author**, **Include data created**, and **Include version** check boxes to include those properties. For **Comments**, the default setting—**From referenced node**—takes the comments from the node in the model; select **Custom** add other comments or **None** for no comments.

Data Sets

About Data Sets

Data sets 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, you can create new data sets from other data sets. You add data sets to the **Data Sets** branch () under **Results**.

All plots refer to data sets; the solutions are always available as default data sets.

Characteristics of a data set include:

- A 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.



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.



See [Table 20-6](#) for links.

Adding 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 20-6](#) for links to all the types of data sets.

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 (, you can 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.

Overview of Data Set Types

The following table lists the available data set types, including links to the description of the data set's properties and settings:

TABLE 20-6: DATA SET TYPES

LINK	ICON	DESCRIPTION AND PLOT USE
Average or Integral	 	Creates a data set that computes the average of another data set, for example to plot the average.
Contour		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 Cut Line 3D	 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		Plots on cut planes are made on 3D data sets and can be visualized in either 2D or 3D plot groups.
Cut Point 1D, Cut Point 2D, and Cut Point 3D	 3D  2D  ID	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.

TABLE 20-6: DATA SET TYPES

LINK	ICON	DESCRIPTION AND PLOT USE
Edge 2D and Edge 3D	 3D	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.
	 2D	
Function 1D, Function 2D, and Function 3D		Create a data set that can evaluate functions on a domain.
Isosurface		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.
Join		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 and Minimum	 Max	Creates a data set that computes the maximum or minimum of another data set.
	 Min	
Mesh		To make a mesh available for visualization and results analysis. Use a Mesh node in a plot group to get the actual plot.
Mirror 2D and Mirror 3D	 3D	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.
	 2D	
Parameterized Curve 2D and Parameterized Curve 3D	 3D	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.
	 2D	
Parameterized Surface		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.

TABLE 20-6: DATA SET TYPES

LINK	ICON	DESCRIPTION AND PLOT USE
Parametric Extrusion 1D and Parametric Extrusion 2D		To extend another data set by using a parameter, such as time, as a dimension.
Particle		To visualize particle traces computed by a Particle Tracing interface.
Revolution 1D and Revolution 2D		Use a revolution data set to visualize a 2D or 1D axisymmetric solution in 3D and 2D, respectively.
Sector 2D and Sector 3D		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		To make solutions available for visualization and results analysis. Solvers create Solution data sets automatically.
Surface		Plot and evaluate a value on surfaces (boundaries) in 3D. Use this data set to create plots for data on surfaces.



For settings that are common to many Result nodes' settings windows, see [Common Results Node Settings](#).

Average or Integral

Select an **Average** () or **Integral** () 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.

DATA

Select a **Data set** for the data to compute the average or integral. To compute the average or integral for 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**.

SETTINGS

Select an integration **Method—Auto** (the default), **Integration**, or **Summation**.

- **Auto**—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 sums up the values. Reaction force variables are predefined in the structural mechanics interfaces. This is selected instead of Integration for reaction forces when the automatic selection of integration method is active.

For **Integration order**, COMSOL automatically determines an appropriate integration order for the expression. The default is 4. Click to select the check box to make a different entry in the field. COMSOL then performs the integration elementwise using numeric quadrature of the selected order.

Select a **Geometry level—Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. The default means the highest geometry dimension for the data in the data set; typically volumes in 3D, surfaces in 2D, and lines in 1D.



For an integral evaluation data set example see [Flow Past a Cylinder](#): Model Library path **COMSOL_Multiphysics/Fluid_Dynamics/cylinder_flow**.

Contour

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. Only 2D data sets can be used.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, and **Levels**.

Cut Line 2D and Cut Line 3D

Use a **Cut Line 2D** () or **Cut Line 3D** () data set to create lines through 2D or 3D geometries to visualize along the line. All plots and results analysis nodes available in 1D are available for cut line data sets as well as 3D plots and results analysis nodes for edges. These data sets are also used to create cross-section line plots.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.

LINE DATA

Use **Line entry method** to specify the cut line either by **Two points** or by a **Point and direction**.

- If **Two points** is selected, enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D) in the **Point 1** and **Point 2** fields ((SI unit: m). If **Point and direction** is selected, enter **x** and **y** coordinates (2D) or **x**, **y**, and **z** coordinates (3D) in the **Point** (SI unit: m) and **Direction** fields.
- The **Bounded by points** check box is selected by default to constrain the line between the defined points.
- 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. Otherwise, leave the snapping off (the default setting) to avoid the additional computational cost.

For **Cut Line 2D**, also select the **Additional parallel lines** check box to define multiple lines for plotting or evaluation, for example. Then enter **Distances** from the original line in the field. 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.

ADVANCED

Under **Space variable**, you can change the name of the space variable for the cut line's coordinate from its default value (dn1x , for example). The space variable name shows in the **Table** window when displaying the data.



For a Cut Line 3D example, see [Thin-Film Resistance: Model Library](#) path **COMSOL_Multiphysics/Electromagnetics/thin_film_resistance**.

Cut Plane

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 nodes 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data** and **Plane Data**.

ADVANCED

Under **Space variables**, you can change the name of the space variables **x** and **y** for the cut plane's coordinates from their default values (cp11x and cp11y , for example). These names appear as column titles for the coordinate columns in the **Table** window when displayed in numerical results from a Cut Plane data set.

Select the **Define normal variables** check box to create variable names for the cut plane's normal vector. Then under **Normal variables**, enter or edit the variable names for the

components of the normal vector—**nx**, **ny**, and **nz**. The default names are **cpl1nx**, **cpl1ny**, and **cpl1nz**, respectively.

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**.

Cut Point 1D, Cut Point 2D, and Cut Point 3D

Use a **Cut Point 1D** (), **Cut Point 2D** (), or **Cut Point 3D** () data set to plot and evaluate a value in a certain point along time or along a parametric solution and to create cross-sectional point plots. The choice of 1D, 2D, or 3D only controls the type of input the data set accepts. For example, a Cut Point 1D can only be added to 1D data sets, Cut Point 2D can only be added to 2D data sets, and so forth. Any of these can be used to make a point graph plot along time and so forth.

Add a **Cut Point 1D** data set for one point as cross-section data. Add a **Cut Point 2D** or **Cut Point 3D** data set to, for example, plot the value in a certain point along time or along a parametric solution and use the data set in its original dimension. Useful ways to visualize and display data in cut points are through **Point Evaluation** nodes ( [8.85](#) [e-12](#)) under **Derived Values** and **Point Graph** nodes () under **ID Plot Group** nodes.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.

POINT DATA

To specify the point data coordinates using the **Entry method** list:

- Select **Coordinates** (the default) to enter the coordinates. For Cut Point 1D, enter an **x** coordinate, for Cut Point 2D, enter **x** and **y** coordinates, and for Cut Point 3D enter **x**, **y**, and **z** coordinates.
- Select **From file** to enter or browse to a text file with the cut point data. Enter the path and filename in the **Filename** field, or click **Browse** to browse to the file.
- Select **Grid** to enter grid coordinates for gridded data. For Cut Point 1D enter an **x** coordinate, for Cut Point 2D enter **x** and **y** coordinates, and for Cut Point 3D enter **x**, **y**, and **z** coordinates.
- Select **Regular grid** to specify the number of points in each direction. Enter **Number of x points**, **Number of y points** (Cut Point 2D and 3D), and **Number of z points** (Cut Point 3D only). The default value is 10 points in each direction.

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. Otherwise, leave the snapping off (the default setting) to avoid the additional computational cost.



For a Cut Point 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](#).



For a Cut Point 2D example, see [2D Heat Transfer Benchmark with Convective Cooling](#): Model Library path [COMSOL_Multiphysics/Heat_Transfer/heat_convection_2d](#).



For a Cut Point 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/Microwave_Heating/microwave_oven](#).

Edge 2D and Edge 3D

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. 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data** and **Selection**.

Function 1D, Function 2D, and Function 3D

Adding an **Function 1D**, **Function 2D**, or **Function 3D** node () 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).

FUNCTION

Select the function to create a data set for—**None**, **All**, or any of the defined functions in the model. Select **All** to make the Function data set point to all functions in the list rather than to a specific function, which can be useful, for example, to plot several functions together in the same plot group.

PARAMETER BOUNDS

Available fields are based on the function dimension. Enter a **Name**. The **First parameter**, **Second parameter** (Function 2D and 3D), and **Third parameter** (Function 3D only) default names are **s**, **t**, and **u** respectively. For each parameter, enter a **Minimum** lower bound (the defaults are 0) and a **Maximum** upper bound (the defaults are 1) for the first, second, and third dimension of the domain.

RESOLUTION

Enter a **Resolution**. This is the number of points into which each dimension is discretized. The defaults are 1000 (Function 1D), 100 (Function 2D), and 30

(Function 3D), and the valid range is between 2 and 1,000,000 points. A high resolution might require significant computational resources.



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**.

Isosurface

Use an **Isosurface** () data set, found under the **More Data Sets** submenu, 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.



- Go to [Common Results Node Settings](#) for links to information about the **Data**, **Expression**, and **Levels** sections.
- [Data Sets](#)

Join

Use a **Join** () data set to combine 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 or to create the sum over a parametric sweep that contain a few solutions. The Join data set has predefined methods to combine the solutions to get the difference, norm of 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.

DATA 1 AND DATA 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.

Select which solutions to use from the **Solutions** lists:

- Select **All** (the default) to use all solutions in the data set.
- Select **One** to use one of the available solution in a time-dependent, parametric, or eigenvalue solution from the list of solutions that appear underneath the **Solutions** list.

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.

COMBINATION

In the **Method** list select a method for combining the solution data sets:

- Select **Difference** (the default) to combine the data sets using a difference: `data1-data2`.
- Select **Norm of difference** to combine the data sets as `abs(data1-data2)`, where `data1` and `data2` are the results of evaluating the expression in the two source data sets. For complex-valued data this corresponds to the Euclidean norm of the difference.
- Select **Explicit** to combine the data sets using an explicit expression with the “operators” `data1` and `data2` in, for example, a plot node’s **Expression** field. This can be useful to compare two different dependent variables in two different data sets such as two solutions from two different models using the same geometry. For example, to plot the sum of the variable `u` from the first data set and the variable `v` from the second data set, type `data1(u)+data2(v)`. The scope for `data1` is the model to which the solution data set under **Data 1** belongs, and similarly for `data2`.
- Select **General** to combine the data sets using a general expression in `data1` and `data2` that you type into the **Expression** field that appears. The default, `data1-data2`, is identical to the **Difference** method. This method is useful for combining data sets with similar solution data (from a mesh convergence study, for example) using another method than a difference, product, quotient, or sum.
- Select **Product** to combine the data sets using a product: `data1*data2`.
- Select **Quotient** to combine the data sets using a quotient: `data1/data2`.
- Select **Sum** to combine the data sets using a sum: `data1+data2`.

Maximum and Minimum

Select the **Maximum** evaluation ( MAX) or **Minimum** evaluation ( MIN) data set, found under the **More Data Sets** submenu, to create a data set that computes the maximum or minimum of another data set.

DATA

Select a **Data set** for the data to compute the maximum or minimum. To compute the maximum or minimum for 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**.

SETTINGS

Select a **Geometry level—Taken from data set** (the default), **Volume**, **Surface**, **Line**, or **Point**. The default means the highest geometry dimension for the data in the data set: typically volumes in 3D, surfaces in 2D, and lines in 1D.

Select an **Element refinement** (default: 2; the element refinement is the number of partitions of an element edge) to adjust the accuracy of the minimum or maximum values.

Mesh

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** node in a plot group to get a plot of the actual mesh.

MESH

Select the **Mesh** to use as the data from the list.

Mirror 2D and Mirror 3D

Use a **Mirror 2D** () data set 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. Use a **Mirror 3D** () data set to extend a solution defined on

one side of a plane to the other side of a plane. Both are selected from the **More Data Sets** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Axis Data** (Mirror 2D), and **Plane Data** (Mirror 3D).

ADVANCED

Select the **Define variables** check box to create a **Positive side indicator** variable. The default, `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. Under **Space variables**, enter or edit the variable names for the mirrored coordinate system. Enter or edit the **x**, **y**, and **z** (Mirror 3D only) variable names in the respective fields. The default names are `mir1x`, `mir1y`, and `mir1z`, respectively.



- For a Mirror 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 Mirror 2D example, and if you have the Plasma Module, see [DC Glow Discharge](#): Model Library path **Plasma_Module/Direct_Current_Discharges/positive_column_2d**.
-

Parameterized Curve 2D and Parameterized Curve 3D

Use a **Parameterized Curve 2D** () or **Parameterized Curve 3D** () data set 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.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.

PARAMETER

Enter a **Name** and the **Minimum** and **Maximum** range of the parameter curve.

EXPRESSIONS

Enter functions for the coordinates of the parameter. For Parameterized Curve 2D enter values in the **x** and **y** fields. For Parameterized Curve 3D, enter values in the **x**, **y**, and **z** fields.

RESOLUTION

Enter the number of subdivisions of the parameter range. The default **Resolution** is 1000, and the valid range is between 2 and 1,000,000 subdivisions. A high resolution might require significant computational resources.

 For a Parameterized Curve 3D example, and if you have the Heat Transfer Module, see [Radiative Heat Transfer in Finite Cylindrical Media: Model Library path Heat_Transfer_Module/Verification_Models/cylinder_participating_media](#).

Parameterized Surface

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.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.

PARAMETERS

Enter a **Name** and a range of the parameter in the **Minimum** and **Maximum** fields for both the **First parameter** and **Second parameter** fields.

EXPRESSIONS

Enter functions for coordinates of the two parameters in the **x**, **y**, and **z** fields.

RESOLUTION

Enter the number of subdivisions of the parameter ranges. The default **Resolution** is 200, and the valid range is between 2 and 1,000,000 subdivisions. A high resolution might require significant computational resources.

Parametric Extrusion 1D and Parametric Extrusion 2D

Use a **Parametric Extrusion 1D** () or **Parametric Extrusion 2D** () data set to extend another data set by using a parameter, such as time, as a dimension. Select these from the **More Data Sets** submenu.

SETTINGS

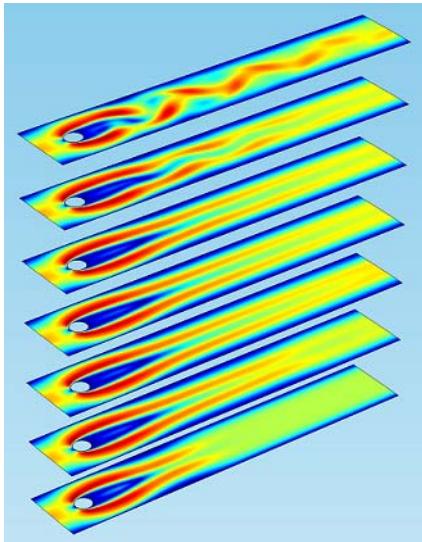
Select the **Level scale factor** check box to edit the field. The default is 1. The **Separate levels** check box is selected by default.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.



For a 1D Parametric Extrusion example, see [The KdV Equation and Solitons](#): Model Library path **COMSOL_Multiphysics/Equation-Based_Models/kdv_equation**.



A Parametric Extrusion 2D data set used here to visualize the solution to a time-dependent flow past a cylinder. The plot shows the velocity field magnitude at different time steps.

Particle



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 makes it possible to compute, for example, the number or fraction of particles in a given domain or on a given boundary during results processing. Visualize the particles using a plot of the particle trajectories in the original dimension, as a Poincaré map, or as a 2D phase portrait.

PARTICLE SOLUTION

Select a **Solution** from the list of solution data. 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_<id>` where `<id>` is the interface identifier for the specific Particle Tracing interface.

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 identifier `<mod1>`, and **Dependent Variables** `qx`, `qy`, and `qz` for the interface, the correct expression to enter is `mod1.qx`, `mod1.qy`, and `mod1.qz`.



If you have the AC/DC Module and the Particle Tracing Module, see [Magnetic Lens](#): Model Library path **ACDC_Module/Particle_Tracing/magnetic_lens**.

Revolution 1D and Revolution 2D

Use a **Revolution 1D** 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. Use a **Revolution 2D** 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data** and **Axis Data**.

REVOLUTION LAYERS

From the **Number of layers** list, choose **Normal** (the default), **Fine**, **Coarse**, or **Custom**. The predefined settings adapt the number of layers to the chosen revolution angle, which minimizes the time to plot the revolved geometry for revolution angles that are less than 360 degrees. If **Custom** is selected, enter the number of **Layers** about the revolution axis (default value: 50).

For all choices, enter the **Start angle** (SI unit: deg) for the revolved model. The default is 0 degrees. Enter the **Revolution angle** (SI unit: deg), to revolve the model to see into the geometry in degrees. The default is 360 degrees, that is, a full revolution. Enter 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 or to the xz -plane and then rotated about the z -axis using the start angle and revolution angle.

ADVANCED

For **Revolution 2D**, from the **Map plane to** list, select a plane to map the axisymmetric solution to—**xy-plane** (the default) to map the rz -plane to the xy -plane and then rotate it about the y -axis, or select **xz-plane** to map the rz -plane to the xz -plane and then rotate it about the z -axis.

For **Revolution 1D** and **Revolution 2D**, select the **Define variables** check box to create variable names for the space and angle variables in the revolved geometry. Then under **Space variables**, enter or edit the variable names for the revolved coordinate system. Enter or edit the **x**, **y**, and **z** (Revolution 2D only) variable names in the respective fields. The default names are **rev1x**, **rev1y**, and **rev1z**, respectively.

Under **phi**, enter or edit the variable name for **phi**. Phi is the name of the angle variable in the revolved coordinate system. The default name is **rev1phi**.

For example, the angle variables can be useful to enter Cartesian components of axisymmetric vector fields (such as `ht.tfluxr*cos(rev1phi)` for the x -component of a heat flux from a 2D axisymmetric heat transfer model, where `ht.tfluxr` is the radial component of the total heat flux).

Sector 2D and Sector 3D

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. It is also possible to invert the phase (change the sign) when rotating or reflecting.

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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data** and **Axis Data**.

SYMMETRY

In the **Number of sectors** field enter any integer greater than or equal to 2 (the default is 2) to define the number of sectors in the full geometry. When the **Number of sectors** entered is an even number, select a **Transformation—Rotation** (the default) or **Rotation and reflection**. For an uneven number of sectors, only rotation is available and it requires no additional user input.

If **Rotation and reflection** is selected:

- For **Sector 2D**, enter **x** and **y** coordinates for the **Direction of reflection axis**
- For **Sector 3D** enter **x**, **y**, and **z** coordinates for the **Radial direction of reflection plane**.

ADVANCED

Select the **Define variables** check box to create variables for the **Sector number** and the **Space variables**:

- The **Sector number** is an integer number from 0 to the number of sectors minus 1. Each sector has a unique sector number: 0, 1, 2, and 3 for a sector data set that includes four sectors, for example. The default variable for the sector number is **sec1number**, where **sec1** is the data set node's tag.
- The **Space 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 variable names (the number of which are based on space dimension) are **sec1x** for the **x** coordinate, **sec1y** for the **y** coordinate, and **sec1z** 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.

Azimuthal Mode Number

When the **Number of sectors** is odd or **Rotation** is selected as the **Transformation** (cyclic symmetry), also enter the **Azimuthal mode number**. The default is 0.



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 .

Invert Phase When Rotating and Invert Phase When Reflecting

These check boxes are available in various combinations as follows:

- When **Rotation** is selected as the **Transformation** and the **Number of sectors** is a multiple of 2 (an even number), choose the **Invert phase when rotating** check box to make the phase of the solution change in alternate sectors.
- When **Rotation and Reflection** is selected as the **Transformation** and the **Number of sectors** is a multiple of 2 (an even number), choose the **Invert phase when reflecting** check box to make the phase change in alternate sectors. When the **Number of sectors** is a multiple of 4, choose the **Invert phase when rotating** check box to make the phase change in alternate sectors.

Solution

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.



Solution data sets do not contain the solution but instead refer to a solution stored in a solver node.



- See Also**
- [Deformed Configuration](#)
 - [Remeshing a Deformed Mesh](#)

SOLUTION

- Select a **Solution** to make available for visualization and results analysis.

- If there is more than one model, select the geometry to perform visualization and results analysis on by selecting the corresponding **Model** from the list.
- Select the **Frame** to evaluate the coordinates in—**Mesh**, **Material**, **Geometry**, 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.
- Enter a value for the **Solution at angle (phase)** (SI unit: deg). The default is 0 degrees and evaluates 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.
- Enter a **Scale factor** (the default is 1; that is, no scaling) to multiply the solution by a real-valued scale factor.

Surface

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.



Go to [Common Results Node Settings](#) for links to information about the **Data** section.

PARAMETERIZATION

Specify how to parameterize the surface. Choose from one of these options in the **x- and y-axes** list to specify what the local *x*- and *y*-axes represent:

- **Surface parameters** (the default) uses the parameters of the 3D surface.
- **xy-plane** is the local *x*- and *y*-axes representing the global *xy*-plane.
- **yz-plane** is the local *x*- and *y*-axes representing the global *yz*-plane.
- **zx-plane** is the local *x*- and *y*-axes representing the global *zx*-plane.
- **yx-plane** is the local *x*- and *y*-axes representing the global *yx*-plane.
- **zy-plane** is the local *x*- and *y*-axes representing the global *zy*-plane.
- **xz-plane** is the local *x*- and *y*-axes representing the global *xz*-plane.
- If **Expression** is selected, 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.

Plot Groups and Plots

-
- 
 - See [Table 20-7](#) for a summary of all the available plot types, including links to each plot in this guide.
 - [Creating Cross-Section Plots and Combining Plots](#)
-

About the Plot Groups

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, 3D, and polar plots and then create individual or a series of plots in a plot group. Several plot groups of the same type can also be used in a model. Information in the form of data and images can be used to generate a report or exported.

Attributes can also be added as subnodes to a plot to modify the plot's behavior—**Deformation** () attributes deform a plot (to illustrate, for example, structural deformation), **Color Expression** () attributes modify the color of a plot, and filtering to only include parts of the plot is available using the **Filter** attribute () . You can also select appropriate color tables for the plots' color expressions based on your audience and what you plan to do with the final analysis.



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.

-
- 
 - At any time during plot creation, click the **Plot** button () to visualize a data set or plot.
 - 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 time-related settings only display on the interfaces for time-dependent models.
-

The physics user 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 delete 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** node's settings windows.

Adding Plots To Plot Groups

Under **Results** () in the **Model Builder** (), right-click the **Plot Group** node and select an option from the context menu to plot the graphs listed in [Table 20-7](#). Each plot group can have several plots combined to create a meaningful representation of the data.

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.

Also add attributes to 2D and 3D Plot Groups 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.

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
Arrow Line		Plot a vector quantity as arrows on lines or edges (3D).	Color Expression, Deformation, and Filter
Arrow Surface		Visualize a vector quantity in arrows.	Color Expression, Deformation, and Filter
Arrow Volume		Visualize a vector quantity as arrows in a volume.	Color Expression, Deformation, and Filter
Contour		Visualize a scalar quantity as a contour plot.	Color Expression, Deformation, Filter, and Height Expression (2D only)

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
Coordinate System Volume, Coordinate System Surface, and Coordinate System Line	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	3D 2D 1D	Plot the value of a global variable. This plot is available for the RF Module and the Acoustics Module.	none
Global	1D	Plot a global scalar quantity as a function of time or a parameter.	Color Expression
Histogram	1D	Plot a histogram showing the distribution of a quantity.	none
Isosurface	3D	Plot a scalar quantity as an isosurface plot.	Color Expression, Deformation, and Filter
Line Graph	1D	Plot a scalar quantity along a geometric line. The line can be an edge in the geometry, a parameterized curve, or a cut line.	Color Expression
Line	3D 2D	Plot a quantity on lines, boundaries (2D), or edges (3D).	Deformation, Filter, and Height Expression (2D only)

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
Max/Min Volume, Max/Min Surface, and Max/Min Line	 Volume  Surface 2D  Surface 3D  Line 2D  Line 3D	Plot the maximum and minimum of an expression and the points where they are attained within the geometry. Found on the More Plots submenu.	Deformation
Matrix Histogram		Plot a precomputed matrix as a 2D histogram. For rainflow counting, for example, to be able to visualize how the stress amplitudes and mean stresses are distributed. This plot requires the Fatigue Module.	none
Mesh		To display a mesh.	Deformation and Filter
Multislice	 3D	Display a scalar quantity on slices in multiple directions inside a 3D domain. Found on the More Plots submenu.	Deformation and Filter
Nyquist	 1D	Display a Nyquist plot, which shows the magnitude and phase of a frequency response.	Color Expression
Particle Tracing	 3D  2D	Visualize the trajectory of a massless particle subject to a flow field. Found on the More Plots submenu.	Color Expression and Deformation

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
Particle Tracing with Mass	 3D  2D	Make a particle tracing plot considering the particle's mass. Found on the More Plots submenu.	Color Expression and Deformation
Particle Trajectories and Filter Node for Particle Trajectories	 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, Filter, Deformation
Phase Portrait	 2D	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	 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
Point Graph	 1D	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.	Color Expression
Principal Stress Volume	 3D	Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter
Principal Stress Surface	 3D  2D	Plot principal stress and strain in structural mechanics models. Found on the More Plots submenu.	Color Expression, Deformation, and Filter

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
Scatter Surface Scatter Surface and Scatter Volume		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		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		Display a scalar quantity on slices inside a 3D domain.	Deformation and Filter
Streamline		Plot a vector field as a streamline plot.	Color Expression and Deformation
Surface		Display a quantity on a 2D domain or a 3D boundary.	Deformation, Filter, and Height Expression (2D plots)
Table Graph		Table plots display data from a table with one line per output column.	none
Table Surface		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		To display a quantity inside a 3D domain.	Deformation and Filter

TABLE 20-7: PLOT TYPES

NAME AND LINK	ICON	DESCRIPTION	PLOT ATTRIBUTES
PLOT ATTRIBUTES			ADD TO THESE PLOTS
Color Expression		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
Deformation		Deform the plot according to a vector quantity, for example, the displacement field in structural mechanics.	All plots except scatter plots.
Filter and Filter Node for Particle Trajectories		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 Expression		Height attributes introduce a 3D height on a 2D surface plot.	2D line plot, 2D surface plot, 2D table surface plot
CROSS SECTION PLOTS			
*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.	
*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.	
<p>* 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.</p>			

Color Coding for Plot Groups and Plot Types

The plot groups and plot types are color coded based on space dimension. See [Figure 20-1](#) for an example.

- All the 1D and polar plots selected from **ID Plot Group**  and **Polar Plot Group**  nodes are green.
- All the 2D plots selected from the **2D Plot Group**  node are pink.
- All the 3D plots selected from the **3D Plot Group**  node are blue.

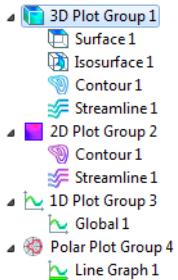


Figure 20-1: An example of the plot group and plot color coding.

Working with Plot Windows

This section explains how to add plot windows, specify the window to plot in, and lock plot windows.

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 Multiphysics® uses for general visualization.

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](#).



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 nodes available in 1D are available for Cut Line data sets as well as 3D plots and results analysis nodes 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 nodes 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](#).

The following sections give examples how to create cross-section plots.



- [Plot Groups and Plots](#).
- See [Table 20-7](#) for links to all the 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 in general while creating plots.

TABLE 20-8: 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 ID Plot Group with a Line Graph that uses this data set.
	Select Second Point for Cut Line	Click these buttons to plot a cross-section of data between two points.

TABLE 20-8: PLOTTING AND CROSS-SECTION TOOLBAR

BUTTON	NAME	USE AND RESULT
	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.
	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.
	Select Second Point for Cut Plane Normal	Click these buttons to plot a cross-section of data between the two points along the plane.
	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.

TABLE 20-8: PLOTTING AND CROSS-SECTION TOOLBAR

BUTTON	NAME	USE AND RESULT
	Isosurface	Click this button to add an Isosurface plot to a 3D Plot Group.
	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.

-
- | | |
|---|--|
|  | 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.



- Plot Groups and Plots
- See [Table 20-7](#) for links to all the plots.

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 ID** 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 Add a **ID Plot Group** (). Select **Cut Point ID** as the **Data set**. If required, right-click and **Rename** the plot group, for example, **Cross-Section-ID Point**.

3 Add a **Point Graph** to this 1D plot group and select **Cut Point ID** as the **Data set** or **From parent** (the default) to use the same data set as the plot group it belongs to.

4 Continue to define the **Point Graph** as required.

5 Click the **Plot** button ()**,** right-click the node and select **Plot**, or press F8.

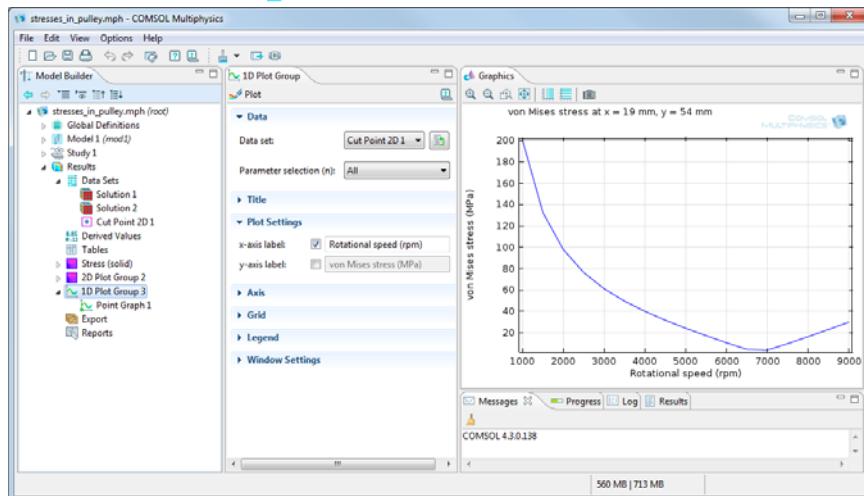
CREATING A 2D CROSS-SECTION PLOT USING A CUT POINT DATA SET

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)`.

- Add a **ID Plot Group** (Cut Point 2D as the **Data set**.
- 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.
- Continue to define the **Point Graph** as required.
- Click the **Plot** button (img alt="Plot button icon" data-bbox="385 265 415 285) right-click the node and select **Plot**, or press F8.

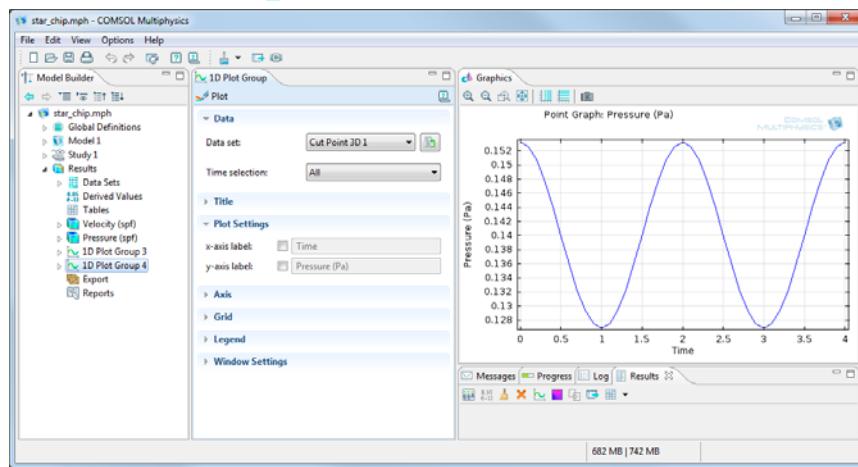


CREATING A 3D CROSS-SECTION PLOT USING A CUT POINT DATA SET

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.

- 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)`.
- Add a **ID Plot Group** (img alt="ID Plot Group icon" data-bbox="375 775 405 795). Select **Cut Point 3D** as the **Data set**. Click the **Go to Source** button (img alt="Go to Source button icon" data-bbox="285 795 315 815) to move to the node that the selection in the list next to the button refers to.

- 3 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.
- 4 Continue to define the **Point Graph** as required.
- 5 Click the **Plot** button (), right-click the node and select **Plot**, or press F8.



-
- 
 - [Plot Groups and Plots](#)
 - See [Table 20-7](#) for links to all the plots.
-

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** Add a **ID Plot Group** (). Right-click the plot node and select **Cut Line 2D** as the **Data set**.
- 3** Add a **Line Graph** and keep **From parent** to use the same data set as the plot group it belongs to. Settings under **y-Axis Data** and **x-Axis Data** determine the quantity on those axes.
- 4** Continue to define the **Line Graph** as required.
- 5** Click the **Plot** button () , right-click the node and select **Plot**, or press F8.

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 **ID Plot Group** with a **Line Graph** are added to the **Model Builder**.
- 4** Adjust the cut line as required by clicking the buttons, then click 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 **ID 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.



See Table 20-7 for links to all the plots.

3D Cross-Section Line Plots

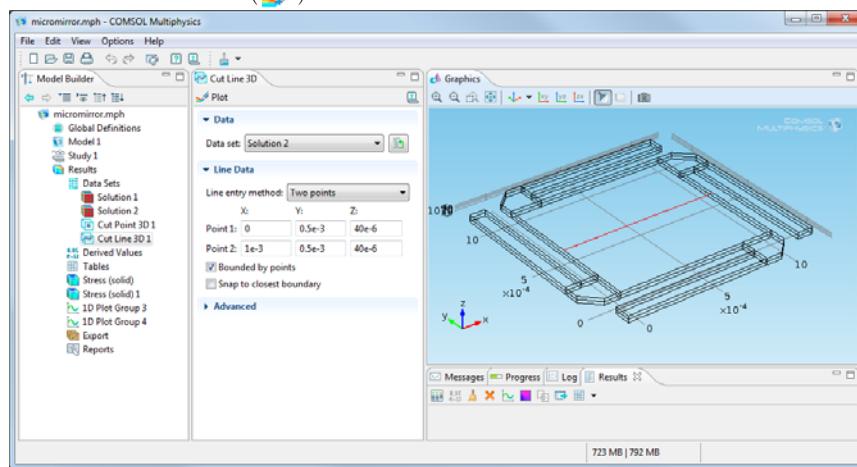
CREATING A 3D CROSS-SECTION PLOT USING A CUT LINE DATA SET

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 ().

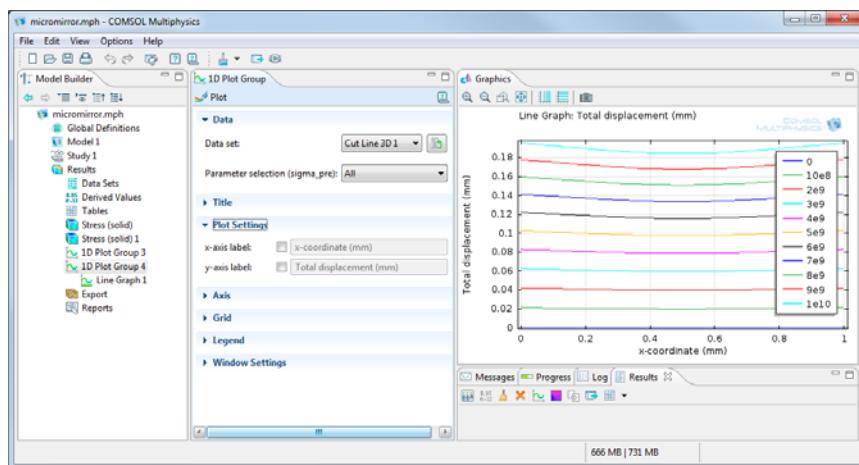


3 Add a **ID Plot Group** (). Right-click the plot node and select **Cut Line 3D** as the **Data set**.

4 Add a **Line Graph**. For the data set, **From parent** uses the same data set as the plot group it belongs to. Settings under **y-Axis Data** and **x-Axis Data** determine the quantity on those axes.

5 Continue to define the **Line Graph** as required. See [Line Graph](#).

6 Click the **Plot** button () right-click the node and select **Plot**, or press F8.



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 **ID 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 **ID 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 **ID 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 **ID 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 **ID 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.
-



See [Table 20-7](#) for links to all the plots.

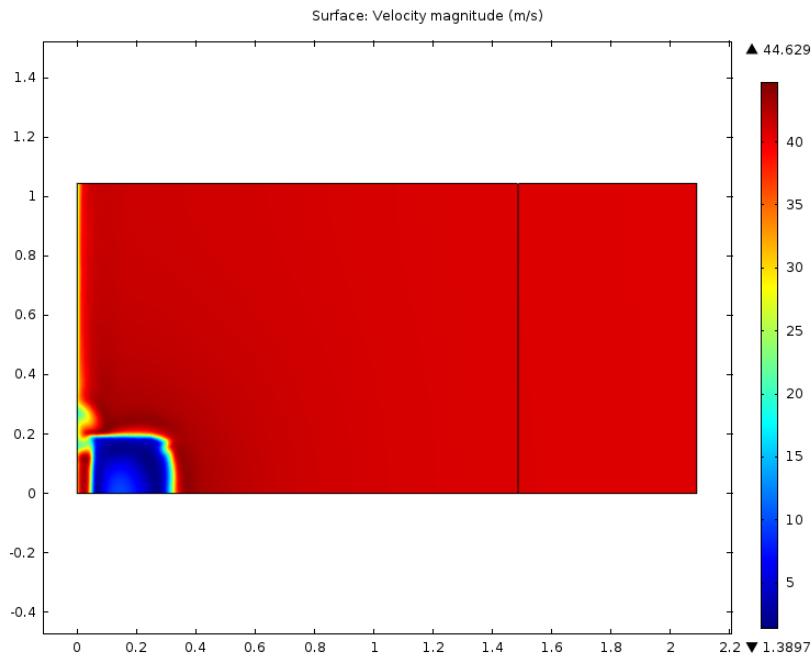
3D Cross-Section Surface Plot

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** Add a **2D Plot Group** (). Right-click the plot node and select **Cut Plane 3D** as the **Data set**.
- 3** Add a **Surface** plot and keep **From parent** to use the same data set as the plot group it belongs to.
- 4** Continue to define the **Surface** plot as required.
- 5** 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

- I 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:

- 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.



See [Table 20-7](#) for links to all the plots.

1D Plot Group and Polar Plot Group



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.



The time-related settings only display for time-dependent models.

Use a **1D Plot Group** () to plot graphs of, for example, a value that varies with time, or a frequency spectrum, using options to use FFT with a time-dependent solution. You can also use 1D plot groups to create cross-section plots. Normally the plot axes (x -axis and y -axis) use linear scaling but depending on the plotted data it might be beneficial to display the graphs using a log scale on the x -axis or y -axis. From the **Graphics** window's toolbar, click the **x-Axis Log Scale** () and **y-Axis Log Scale** () buttons to use a log scale on the x -axis and y -axis, respectively. Click again to return to a linear scale. For a log scale on an axis, COMSOL tries to show values like $10^{[\text{integer}]}$ (for example, 10^1 or 10^{-2}); if it is not possible to show more than three ticks like $10^{[\text{integer}]}$, the software instead shows regular numerical values on the axis such as 1, 2, 5, 10, 20, 50, 100; and finally, if these regular values are not possible to display, the axes show uniformly distributed regular values.

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.

DATA

Select a **Data set**. Depending on the type of data, also specify, for example, the time or frequency selection.

Parametric Sweep Studies

For **Parametric Sweep** studies also select an option from the **Select via** list—**Stored output times** or **Interpolated times**.

- If **Stored output times** is selected, the **Times** section is auto-filled with information from the selected **Data set**. If **Interpolated times** is selected, enter **Times**.

Load Cases

For some solution **Data sets**, you can select the **Load case** to use in the plot group (if you have defined load cases in the model).

Solution Data Sets

For some **Solution Data sets**, select a **Parameter selection (freq)**—**All**, **First**, **Last**, **From list**, or **Manual**.

- If **From list** is selected, select the **Parameter values** from the box that displays.
- If **Manual** is selected, enter **Parameter indices (1-91)** (the actual indices depend on the number of solutions). Or click the **Range** button () to define an **Integer Range**.

TITLE

The **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.

PLOT SETTINGS

Manually enter axis labels by selecting the **x-axis label** and **y-axis label** check boxes.

AXIS

Select the **Manual axis limits** check box to edit the limits already assigned based on the data set. For 1D Plot Groups, this is for the **x minimum**, **x maximum**, **y minimum**, and **y maximum**. For Polar Plot Groups, this is for the **r minimum** and **r maximum**.

For the 1D Plot Group, and as required, select one or all of these check boxes: **Preserve aspect ratio**, **x-axis log scale**, and **y-axis log scale**. When **Preserve aspect ratio** check box is selected, the distances on the *x*-axis and *y*-axis are kept equal.

GRID

Select the **Manual spacing** check box to edit the fields. For 1D Plot Groups, this is for the **x spacing** and **y spacing** fields. For Polar Plot Groups, this is for the **r spacing** and **θ spacing** (SI unit: degrees) fields.

For 1D Plot Groups, also specify extra grid points on *x*-axis and *y*-axis in the **Extra x** and **Extra y** fields. For Polar Plot Groups, this is for the **Extra θ** and **Extra r** fields.

LEGEND

Specify the position of the legends for the plots in the plot group. From the **Position** list, select **Upper right** (the default), **Upper left**, **Middle right**, **Middle left**, **Lower right**, or **Lower left**. Legends in all plots in the plot group use this position.

WINDOW SETTINGS

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.

INTERACTIVE (ID PLOT GROUP)

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 **ID 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.

-
- 

- [Plot Groups and Plots](#)
 - See [Table 20-7](#) for a summary of all the available plot types, including links to each plot described in this guide.
-

2D Plot Group and 3D Plot Group

Use a **2D Plot Group** () to combine one or more 2D plots into and visualize the plots simultaneously. Use a **3D Plot Group** () to combine one of more 3D plots into one to visualize the plots simultaneously.



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.



The time-related settings only display for time-dependent models.

DATA

Select a **Data set**. From the lists below select the solution to use. For example, for **Parametric Sweep** studies select a **Parameter value** as required. For time-dependent problems, select a **Time**.

TITLE

The **Title type** is automatically generated by default. Select **Custom**, **Manual**, or **None** as required.

PLOT SETTINGS

- 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. This can be useful for scatter plots, for example, where the axes represent other quantities than the *x* and *y* directions.
- By default, the plot does not include hidden objects (geometric entities that are hidden in the selected **View**). To include such hidden objects in the plot, select the **Show hidden objects** check box.
- 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**.

COLOR LEGEND

Specify the location of the color legends (color scales) in the plots for this plot group. From the **Position** contains the following positions for the color legends:

- Select **Alternating** to position the first color legend to the right of the plot, the second color legend to the right of the plot, and so on.
- Select **Bottom** to position the color legends horizontally at the bottom of the plot window.
- Select **Left** to position the color legends to the left of the plot.

- Select **Left double** to position the color legends to the left of the plot with two color legends positioned on top of each other (tiled vertically).
- Select **Right** to position the color legends to the right of the plot. This is the default position.
- Select **Right double** to position the color legends to the right of the plot with two color legends positioned on top of each other (tiled vertically).



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.

WINDOW SETTINGS

- 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.

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.



- [Plot Groups and Plots](#)
- See [Table 20-7](#) for a summary of all the available plot types, including links to each plot described in this guide.

Arrow Line

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** (□) 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 plots.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

Arrow Surface

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 plots.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Arrow Volume

Use an **Arrow Volume** () plot to 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Arrow Positioning**, **Coloring and Style**, and **Inherit Style**.

Contour

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](#), [Filter](#), or (2D only) [Height Expression](#) subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Levels**, **Quality**, **Inherit Style**, and, except for some details below, **Coloring and Style**.

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.

Coordinate System Volume, Coordinate System Surface, and Coordinate System Line

Use the **Coordinate System Volume** (), **Coordinate System Surface** (2D and 3D) , and **Coordinate System Line** () plots 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), as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots from the **More Plots** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

COORDINATE SYSTEM

For **Coordinate System Surface** in 2D and **Coordinate System Volume** from the **Source** list you can choose 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 variable, which include coordinate transforms to and from the added coordinate systems and physical quantities that are tensors, such as the thermal conductivity. The selected variable (for example, `ht.k`), then appears next to **Matrix variable**. Each row of the matrix is plotted as a vector. The first row is plotted in red, the second row in green, and the third row, if any, in blue.

For other coordinate system plots, select another **Coordinate system** to plot. The default is **None** and the list contains any additional coordinate systems that the model includes.

POSITIONING

This section is available for **Coordinate System Volume** and **Coordinate System Surface** (2D) 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).

Far Field

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 far field of an electromagnetic field or acoustic pressure field.

- 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 Graph](#), 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.



Go to [Common Results Node Settings](#) 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**.

EVALUATION

1D Plot Group and Polar Plot Group

Under **Angles**, enter the φ **resolution**. The default is 50.

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).

Under **Center**, enter a value for **z-coordinate** (SI unit: mm). The default is 0.

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 and 3D Plot Groups

Under **Angles**, enter the θ **resolution**. The default is 10. Enter the ϕ **resolution**. The default is 20.

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 **Table** window.



Select **View>Table** to open the **Table** window.

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)

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 mm.

Global

Use a **Global** (plot to graph a scalar quantity as a function of time or a parameter. Add a **Color Expression** subnode as required. Right-click a **ID Plot Group** or **Polar Plot Group** to add this plot type.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**) **Data**, **Title**, **Coloring and Style**, and **Legends**. Also see the section [Expressions and Predefined Quantities](#).

X-AXIS DATA (ID 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 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). Select a **Unit** for the phase angle.
- **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 inner solutions, select an option from the **Solutions** list—**Inner** or **Outer**.

- If **Inner** is selected, and for time-dependent studies, the **Times** steps are plotted on the x-axis and one line per parameter is included in the graph (as listed in the **Data>Parameter values** section on this page).
- If **Outer** is selected, one line in the graph is plotted for each inner solution and the **Parameter values** are plotted on the x-axis.



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.



[Small-Signal Analysis, Prestressed Analysis, and Harmonic Perturbation Plot Settings](#)

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. For the 2D Plot Group, select this from the **More Plots** submenu.

-
- 
 - Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Legends**, and **Quality**.
 - For a 2D histogram based on a precomputed matrix of data, see [Matrix Histogram](#).
-

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.

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.

Isosurface

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 **Deformation**, **Color Expression**, or **Filter** subnodes as required. Right-click a **3D Plot Group** to add this plot type.



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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Levels**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Line Graph

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). Add a **Color Expression** subnode as required. Right-click a **ID Plot Group** or **Polar Plot Group** to add this plot type.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **y-Axis** (or **r-Axis**), **Title**, **Coloring and Style**, **Legends**, and **Quality**.

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

For **Parametric Sweep** studies, for each pair of outer solution or inner solutions, one line is plotted on the graph. For example, if there are 10 outer solutions and each outer solution has five inner solutions, then 50 lines are drawn. The number of inner solutions can vary between outer solutions.

Select **Arc length** or **Reversed arc length** from the **Parameter** list to visualize along an arc length in the direction of the arc or the reversed direction of the arc, respectively, or select **Expression** to visualize along, for example, a coordinate expression. If **Expression** is selected, go to [Expressions and Predefined Quantities](#)

Line

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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

Matrix Histogram



This plot is available with the Fatigue Module.

Use the **Matrix Histogram** plot in 2D () when you have a precomputed matrix that you want to visualize as a 2D histogram. For example, in a fatigue analysis, you can use it for rainflow counting to be able to visualize how the stress amplitudes and mean stresses are distributed. Using this plot can then help to see how the actual damage is distributed between the different stress levels. If a large fraction of the total damage is

caused by loads that occur only a few times, the statistical sample of the loads may be too small to reach any good conclusions.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, and **Coloring and Style**.

EXPRESSION

Click the **Replace Expression** button () to select the matrix variable to use as input for the matrix histogram. The matrix histogram plot uses precomputed matrix variables only.

From the **Unit** list, select any applicable unit for the histogram plot. Select the **Description** check box to customize or enter a description of the plot.

AXES

From the **Unit** list, select an applicable unit for the x -axis and y -axes of the histogram. The matrix contains data points in the (x,y) -plane where the x - and y -values are stresses. This means that the x - and y -values can have any pressure unit, for example, Pascal. By changing the axes unit, you choose how the x - and y -axes in the plot are interpreted.

Max/Min Volume, Max/Min Surface, and Max/Min Line

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 the **Table** window (underneath the **Graphics** window if using 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 plots from the **More Plots** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, and **Inherit Style**.

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**.

Mesh

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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Element Filter**, and **Shrink Elements**.



You can also create a Mesh plot by right-clicking the **Mesh** node and selecting **Plot** ().

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**.



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

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 from the **More Plots** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

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.

Nyquist

Use **Nyquist** () 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. Add a [Color Expression](#) subnode as required. Right-click a **ID Plot Group** or **Polar Plot Group** to add this plot type.

Except where noted below, see [Global](#) 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.



1D Plot Group and Polar Plot Group

Particle Tracing



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.



Go to [Common Results Node Settings](#) 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**.



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 Results Node Settings](#) for the **Comet tail** settings links.

PARTICLE POSITIONING

For 3D models, enter the initial position of particles in the **x**, **y**, and **z** fields. 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.
- If **Boundary coordinates** is selected, select an item from the **Named selection** list and select an **Entry method—Number of points** or **Boundary parameters**.
 - 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**.

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.

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



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 the position x and one for the 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.



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 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



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.

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 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:

$$\dot{x} = v(t, x)$$

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, if an expression is specified for the particle mass that depends on time, the result are incorrect.

Axisymmetric Models



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

$$\ddot{r} = \frac{F_r}{m} + r\dot{\phi}^2 \quad \ddot{\phi} = \frac{F_\phi}{rm} - \frac{2\dot{r}\dot{\phi}}{r} \quad \ddot{z} = \frac{F_z}{m}$$

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 $r\dot{\phi}$ as $\dot{\phi}$ has the dimension radians/time.

-
- 
 - Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Coloring and Style**, **Quality** (Resolution and Recover only), and **Inherit Style**.
 - See [Particle Tracing](#) for **Particle Positioning**, **Release**, **Quality** (ODE solver settings), and **Advanced** settings.
 - See [Particle Tracing in Fluid Flow](#) for more information about predefined expressions for drag-driven particle movement that are available for particle tracing in fluid-flow models.
-



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 outer space. Go to [Common Results Node Settings](#) for the **Comet tail** settings links.

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](#). Go to [ODE Solver Settings—Relative Tolerance](#), [ODE Solver Settings—](#)

[Absolute Tolerance](#), and [ODE Solver Settings—Step Size](#) for details.

ADVANCED

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](#). Go to [Advanced—Termination](#) and [Advanced—Instantaneous Flow Field](#) for details.

Particle Trajectories



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](#) subnode 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

Filter Node for Particle Trajectories

You can right-click a **Particle Trajectories** plot node () 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 has the following section:

PARTICLE SELECTION

From the **Particles to include** list, select the particle types or subset to include in the particle trajectories plot:

- **All** (the default) to include all particles.
- **Primary** to include primary particles only.
- **Secondary** to include secondary particles only.
- **Logical expression** to include a subset of particles that fulfill the logical expression that you enter in the **Logical expression for inclusion** field. For example, `pt.V>1` only includes particles with a velocity larger than 1, and `x>0` only includes particles in areas where the *x* coordinate is positive.

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 certain number of particles, which you specify in the **Number of particles** field. The default is to render 100 particles.

Phase Portrait



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 from the **More Plots** submenu.

 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.


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).

 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.

 Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Coloring and Style**, and **Inherit Style**.

Poincaré Map

 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Inherit Style**, and **Coloring and Style**.

Point Graph

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. Add a [Color Expression](#) subnode as required. Right-click a **ID Plot Group** or **Polar Plot Group** to add this plot type.



See [Global](#) for these settings: **x-Axis Data** or **θ Angle Data**. Then go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Legends**, **y-Axis** (or **r-Axis**) **Data**, and **Coloring and Style**.

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.

Principal Stress Volume

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 from the **More Plots** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Principal Components**, **Title**, **Positioning**, **Coloring and Style**, and **Inherit Style**.

Principal Stress Surface

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 from the **More Plots** submenu.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Title**, **Inherit Style**, **Principal Components**, **Positioning**, and **Coloring and Style**.

Scatter Surface and Scatter Volume

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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Radius**, **Color**, **Coloring and Style**, and **Inherit Style**.

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).

Slice

Use a **Slice** () 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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.

PLANE DATA

Under **Plane Data**, select a **Plane Type**—**Quick** (the default) 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**. 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).

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.

Streamline

Use a **Streamline** plot in 2D (Deformation or **Color Expression** subnodes as required. Right-click a **2D Plot Group** or **3D Plot Group** to add these plots.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Coloring and Style**, **Quality**, and **Inherit Style**.

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:

- 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



The **Selection** section is available for some data sets when you select **On selected boundaries** 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.

ADVANCED

Define the 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

- | Under **Streamline Positioning**, from the **Positioning** list, select **On selected boundaries**.



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

- | 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). 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

- | 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.

- | 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.

 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.

 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.



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.

Surface

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.

-
- 
 - Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, and **Inherit Style**.
 - **2D Plot Group** and **3D Plot Group**
 - [Plot Groups and Plots](#)
-

Table Graph

Add a **Table Graph** () plot to display data from a table with one line per output column. Right-click a **ID Plot Group** or **Polar Plot Group** to add this plot type. First define a table to plot. This plot is also available by selecting **Table Graph** () from the **Table** window's toolbar.

-
- 

Go to [Common Results Node Settings](#) for links to information about these sections: **Legends** and **Coloring and Style**.
-

DATA

Select a **Table**. 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

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 **Table Surface** () from the **Table** window's toolbar.



Go to [Common Results Node Settings](#) for links to information about these sections: **Title**, **Range**, **Coloring and Style**, and **Inherit Style**.



When created from the **Table** window 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.

DATA

Select a **Table**. 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

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.



Go to [Common Results Node Settings](#) for links to information about these sections: **Data**, **Expression**, **Title**, **Range**, **Coloring and Style**, **Quality**, **Element Filter**, **Shrink Elements**, and **Inherit Style**.

Color Expression

Use a **Color Expression** node () to add coloring (according to an expression that you define) to the shapes or lines defined by a plot. Add this to these plot types—Line Graph, Point Graph, Global, Nyquist Plot, Arrow Volume, Arrow Surface, Arrow Line, Contour, Isosurface, Particle Trajectories, and Streamline.

In the **Model Builder**, add and define a plot group. Right-click the plot node (for example, **Streamline**) and select **Color Expression**.



Go to [Common Results Node Settings](#) for links to information about these sections: **Expression**, **Title**, **Range**, and **Coloring and Style**.

Deformation

Add a **Deformation** node () to deform the plot according to a vector quantity, for example, the displacement field in structural mechanics. You can add a deformation 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**.

SCALE

Select the **Scale factor** check box to edit the default value for the scale factor.

 Using a scale factor of 1 and equidistant displacements in the x , y , or z direction, you can plot several instances of the geometry side by side to, for example, visualize the solution at some times or for some parameter values. In such a plot you would typically specify the data set in each plot individually and turn off the color legends and titles for each separate plot.

 Go to [Common Results Node Settings](#) for links to information about these sections: **Expression** and **Title**.

Filter

You can add a **Filter** () subnode 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**.

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.

Height Expression

The **Height Expression** subnode () introduces 3D height on a 2D surface plot, 2D table surface plot, 2D contour plot, or 2D line plot. Add it to make the height of the plot represent a scalar quantity. 2D Surface, 2D Contour, 2D Line, and 2D Table Surface plots support the height expression attribute. In the **Model Builder**, add and define a 2D **Surface**, **Contour**, **Line**, or **Table Surface** plot then right-click the plot node and select **Height Expression**.



As a subnode to 2D **Histogram** () plots, the **Height** node settings window does not have an **Expression** section and only **Automatic**, **Manual**, and **None** are available as a **Title type**.

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, see [Expressions and Predefined Quantities](#).

TITLE

Select a **Title type**—**Automatic** (the default), **Custom**, **Manual**, or **None**.

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.

Derived Values and Tables

About Derived Values

You can 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](#)) to define evaluations of numerical results—globally, in a point, or integrated quantities. For 2D and 3D plots, you can also get numerical results directly in a table by clicking the plot.

For all derived values you can 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, you can 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 provide 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.



Common Results Node Settings

GETTING NUMERICAL RESULTS DIRECTLY

 For 3D and 2D 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 a ray projected from the point clicked hits the geometry.

 COMSOL displays the value at that point along with the point's coordinates in a row in an **Evaluation 2D** or **Evaluation 3D** table in the **Table** window. Each click adds a row to the table. From the **Table** window you can plot or copy the table data to a clipboard like any other table.



See Table 20-9 for links to the data sets.

Derived Value Types

In the **Model Builder**, under **Results**, right-click **Derived Values** (). Select an option from the list and continue defining each derived value (see Table 20-9).

TABLE 20-9: DERIVED VALUE TYPES

LINK TO SECTION	ICON	DESCRIPTION
Point Evaluation		To evaluate expressions or variables defined in a point.
Global Evaluation		To evaluate the numerical value of a global variable.
Global Matrix Evaluation		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.
Particle Evaluation		To evaluate an expression for all, or a subset of, the particles in a particle tracing model.
System Matrix		To evaluate an Assemble or Modal node to a table.
AVERAGE SUBMENU	See Volume Average , Surface Average , and Line Average	
Volume Average		To evaluate an average over a set of domains in 3D models.
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	See Volume Integration , Surface Integration , and Line Integration .	
Volume Integration		To evaluate an integral over a set of domains in 3D models.
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 20-9: DERIVED VALUE TYPES

LINK TO SECTION	ICON	DESCRIPTION
MAXIMUM SUBMENU		See Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum.
Volume Maximum		To evaluate the maximum over a set of domains in 3D models.
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		See Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum.
Volume Minimum		To evaluate the minimum over a set of domains in 3D models.
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.

Editing and Organizing Results Tables

When a **Table** () is generated and displays in the **Table** window, a variety of editing options are available as listed in [Table 20-10](#) and in [Figure 20-2](#).

TABLE 20-10: TABLE WINDOW EDITING BUTTONS

BUTTON AND FUNCTION	DESCRIPTION
Full Precision ()	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.

TABLE 20-10: TABLE WINDOW EDITING BUTTONS

BUTTON AND FUNCTION	DESCRIPTION
Copy Table and Headers to Clipboard (	Click the button or right-click anywhere in the table and select this option from the context menu. You can then paste the table's data and headers in a spreadsheet, for example.
Export (	Click to export the table to a text file in a spreadsheet format or to a Microsoft Excel Workbook (*.xlsx) if the license includes LiveLink™ for Excel®. When saving to a Microsoft Excel Workbook, an Excel Save dialog box opens where you can specify the sheet and range and whether to overwrite existing data and include a header.
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).
Copy Selection and Headers to Clipboard	Select the rows to copy, then right-click anywhere in the table and select this option from the context menu ( .
Copy Table to Clipboard	Right-click anywhere in the table and select this option from the context menu ( .
Copy Table and Headers to Clipboard	Right-click anywhere in the table and select this option from the context menu ( .



The Table window's toolbar control settings that are specific to the Table window in general and not to a specific table.

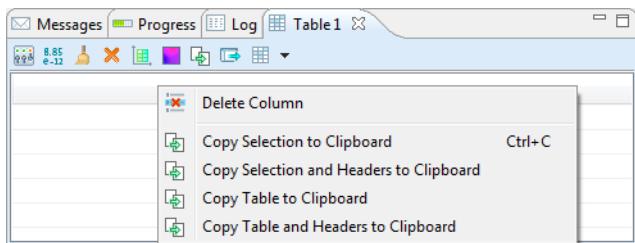


Figure 20-2: Context menu and button options for editing results tables.

Volume Average, Surface Average, and Line Average

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) and from the **Average** submenu 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** node and displayed in the **Table** 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.



Go to [Common Results Node Settings](#) for information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operations**.



For a line average example, see [Tubular Reactor](#): Model Library path **COMSOL_Multiphysics/Chemical_Engineering/tubular_reactor**.

Volume Integration, Surface Integration, and Line Integration

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** (  ) and from the **Integration** submenu 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 **Table** 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.



Go to [Common Results Node Settings](#) for information about these sections: **Data**, **Selection**, **Expression**, **Integration Settings**, and **Data Series Operation**.



For volume integration examples, and if you have the:

- AC/DC Module, for both a volume and surface integration example, see [Eddy Currents](#): Model Library path **ACDC_Module/Inductive_Devices_and_Coils/eddy_currents**.
- RF Module, see [Microwave Oven](#): Model Library path **RF_Module/Microwave_Heating/microwave_oven**.



For a surface integration example see [Effective Diffusivity in Porous Materials](#): Model Library path **COMSOL_Multiphysics/Diffusion/effective_diffusivity**.



For line integration examples, and if you have the:

- AC/DC Module, see [An RFID System](#): Model Library path **ACDC_Module/Inductive_Devices_and_Coils/rfid**.
 - CFD Module, see [Filling of a Capillary Channel—Level Set](#): Model Library path **CFD_Module/Multiphase_Tutorials/capillary_filling_ls**.
 - Microfluidics Module, see [Filling of a Capillary Channel—Level Set](#): Model Library path **Microfluidics_Module/Two-Phase_Flow/capillary_filling_ls**.
 - Heat Transfer Module, see [Cavity Radiation](#): Model Library path **Heat_Transfer_Module/Verification_Models/cavity_radiation**.
-

Volume Maximum, Volume Minimum, Surface Maximum, Surface Minimum, Line Maximum, and Line Minimum

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** ([8.85](#)) and from the **Maximum** or **Minimum** submenus select:

- **Volume Maximum** (**MAX**) or **Volume Minimum** (**MIN**) 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 **Table** window.
- **Surface Maximum** (**MAX**) or **Surface Minimum** (**MIN**) to evaluate a maximum or minimum value over a set of domains in 2D, 2D axisymmetric, or boundaries in 3D.
- **Line Maximum** (**MAX**) or **Line Minimum** (**MIN**) to evaluate a maximum or minimum value over a set of domains in 1D, boundaries in 2D, or edges in 3D.

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.

Select an **Element refinement** (default: 5; the element refinement is the number of partitions of an element edge) to adjust the accuracy of the minimum or maximum values.



Go to [Common Results Node Settings](#) for information about these sections: **Data**, **Selection**, **Expression**, and **Data Series Operation**.



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**.

Point Evaluation

Use **Point Evaluation** ([8.85 e-12](#)) to define the evaluation of a variable or an expression in a point. The result is stored in a **Table** node and displayed in the **Table** window.



- Go to [Common Results Node Settings](#) for information about these sections: **Data**, **Expression**, and **Data Series Operation**.
- [Table](#)

Global Evaluation

Use a **Global Evaluation** ([8.85 e-12](#)) to define the evaluation of the numerical value of a global variable.



- Go to [Common Results Node Settings](#) for information about these sections: **Data**, **Expression**, and **Data Series Operation**.
- [Table](#)

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 **Table** window then displays all values for all frequencies in a parametric sweep.

-
- 
 - Go to [Common Results Node Settings](#) for information about these sections: **Data** and **Expression**.
 - [Table](#)
-

DATA SERIES OPERATION

Select an option for operations on the data series for the inner solutions—typically 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 are NaNs, clear this check box.

TRANSFORMATION

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** (available for the AC/DC Module and RF Module). 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.
- **From S to Z**. For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is 1 Ω .
- **From Y to S**. For this transformation, also specify the **Characteristic admittance** (SI unit: S). The default value is 1 S.
- **From Y to Z**.

- **From Z to S.** For this transformation, also specify the **Characteristic impedance** (SI unit: Ω). The default value is $1\ \Omega$.
- **From Z to Y.**



The transformation operations are only applicable for square matrices.

System Matrix

Use a **System Matrix** () derived values node to evaluate a matrix defined in an **Assemble** or **Modal Solver** node to a table.

SOLUTION

Select a **Solution** from the list to specify which **Solver** branch to tabulate the system matrix for. From the **Solver feature** list, select the **Assemble** or **Modal Solver** node that computes the system matrix.

OUTPUT

Select a **Matrix—Stiffness matrix** (the default), **Damping matrix**, **Mass matrix**, or any other system matrix that the selected **Assemble** or **Modal Solver** node computes.

First select the system matrices to assemble or compute in the settings window for the **Assemble** or **Modal Solver** node and then compute the solution so that the solver stores these system matrices. Selecting a system matrix that has not been computed and stored results in an error.

Select a **Format—Sparse** (the default) or **Filled**. The system matrices can become very large but are usually sparse (most matrix elements are zero). If **Filled** is selected, the preference setting for the maximum filled matrix size (the default is 100) prevents the creation of a table with a very large matrix. Typically, when this is an **Assemble** node, a filled matrix output can only be used for very small models or when using reduced matrices.



When the System Matrix is a **Modal Solver**, then the matrices of interest are typically small and filled.

-
- 
- Solver Overview and Solution Utility Nodes
 - Derived Values and Tables
-

Particle Evaluation

Use a **Particle Evaluation** ( ) derived values node to define the evaluation of the numerical value of quantities computed by one of the particle tracing interfaces. This node requires the Particle Tracing Module.

- 
- Go to [Common Results Node Settings](#) for information about these sections: **Data** and **Expression**.
 - [Derived Values and Tables](#)
-

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 occurs. 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.

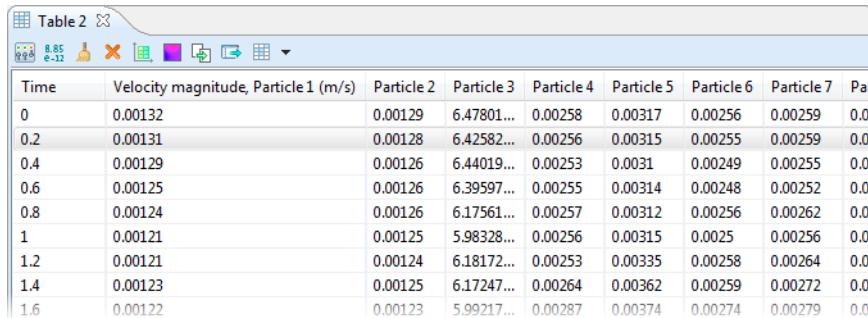
EVALUATION

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>Table** to open the **Table** 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	Pa
0	0.00132	0.00129	6.47801...	0.00258	0.00317	0.00256	0.00259	0.0
0.2	0.00131	0.00128	6.42582...	0.00256	0.00315	0.00255	0.00259	0.0
0.4	0.00129	0.00126	6.44019...	0.00253	0.0031	0.00249	0.00255	0.0
0.6	0.00125	0.00126	6.39597...	0.00255	0.00314	0.00248	0.00252	0.0
0.8	0.00124	0.00126	6.17561...	0.00257	0.00312	0.00256	0.00262	0.0
1	0.00121	0.00125	5.98328...	0.00256	0.00315	0.0025	0.00256	0.0
1.2	0.00121	0.00124	6.18172...	0.00253	0.00335	0.00258	0.00264	0.0
1.4	0.00123	0.00125	6.17247...	0.00264	0.00362	0.00259	0.00272	0.0
1.6	0.00122	0.00123	5.99217...	0.00287	0.00374	0.00274	0.00279	0.0

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 () to display as many significant digits as possible. If required, the precision level can be changed from the **Options>Preferences>General** menu.

Table

Tables can store the results of **Derived Values** and results from probes, for example. The result are displayed in the **Tables** window, which by default is located below the **Graphics** window. To add a **Table** node, if not already created, right-click the **Tables** () node and select **Table**. You can also add a table by, in the right corner of, for example, a **Volume**, **Surface**, or **Line Integration** page, clicking the **Evaluate (New Table)** button (). This evaluates the node and stores the result in a new table. This button is also used as the **Evaluate** button.

A **Table** node is added under **Tables**. Click the node to display a table with the selected integration node's description and values in the **Table** 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 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** (grid icon) node and select **Delete All** (✖). To clear the contents of all tables, right-click the main **Tables** (grid icon) node and select **Clear All**.

DATA

If applicable (for accumulated probe tables, for example) select a data format from the **Format** list—**All** (the default) or **Filled**. **All** displays 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. Filled tables can only be produced by studies that have **All combinations** selected in their parametric sweeps. The filled tables make it possible to retrieve data for a pair of parameters on a matrix format and can be used to make response surface plots. You can get filled tables from some probes or from using derived values when all solutions are selected. Basically, you get filled tables when the input parameters in a parametric sweep (such as time) represent a full outer product. In some cases, such as eigenvalue solutions or if the solver is interrupted, the data is not filled. Also, if you modify the table by, for example, evaluating more than once into the same table or delete columns, the table data is no longer filled.

- 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.

Click the **Import** button to import data from, for example, a spreadsheet. In the **Import** dialog box that opens, select a data file to import.

STORAGE

If desired you can store the table data on file instead of or in addition to storing it in the model. 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. This makes it possible to, for example, keep track of more data using probes while solving without having to put all the data in the model.
- 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.



The imported data replaces any existing data in the table.

Exporting Data and Images

Export 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 as image, movie, or data files 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 20-11](#).

TABLE 20-11: EXPORT TYPES

LINK TO SECTION	ICON	DESCRIPTION
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 web site.
Data		Exports numerical data to file. Data export operates directly on data sets. It is also possible to export mesh data.
Mesh		To export a mesh defined by a data set to file.
Table		To export the contents of a table to file.
1D Image, 2D Image, or 3D Image		To export plot images from a 1D Plot Group, 2D Plot Group, and 3D Plot Group.
Player		To create interactive animations.
Plot		To export a plot from a 1D Plot Group, 2D Plot Group, or 3D Plot Group.

About the Sectionwise Data Format for Data Export

When exporting data on the sectionwise data format, the program evaluates the entered expressions at a number of points in each mesh element. For example, evaluating in Lagrange points of order 1 means that the expressions are evaluated at the vertices of each mesh element. When a vertex is shared by more than one mesh element (as is typically the case) this means that the expressions are evaluated several times at that coordinate, but using the shape functions in the different mesh elements. The values of these evaluations at the same point may or may not be equal, depending on the expression being evaluated. In particular, derivatives are typically discontinuous across mesh element boundaries and usually have different values.

Once all the evaluations have been made, the data are checked for duplicate values (that is, evaluations with the same coordinates and the same values of the expressions). Such duplicates are removed before the data is exported to file. With smoothing turned on, a smoothed variant of the derivative is evaluated, which is continuous across mesh element boundaries, so in such cases there are many duplicates. When evaluating at Gauss points, the evaluation points are always in the interior of mesh elements, so there are never any duplicates.



To avoid the removal of duplicates, you can export several expressions to the same file, and then the values of all expressions must agree to be considered duplicates. Another way to ensure that no duplicates are removed is to add the variable `meshelement` to the list of expressions.



- [Data](#)
- See [Table 20-11](#) for links.

Animation

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. Use this node, for example, to export multiple images for different time steps or eigenvalues.



Use Windows Media Player to play AVI files.

SCENE

Select a **Subject**, which is one of the plot groups previously defined, or **None**.

OUTPUT

Select an **Output type**—**Movie** (the default) to generate a single movie file containing all the images; or **Image sequence** to generate multiple image files, one for each frame.

- 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 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 **image** is entered, select **.png** as the file type, and if there are 11 frames in the movie, 11 files are created: **image01.png**, **image02.png**, ..., **image11.png**.

- If **Movie** is selected, select a **Format** for the movie: **GIF** (the default), **Flash**, or **AVI**. For any movie format, enter a path and include a **Filename**. Or click **Browse** and navigate to where you want to **Save** the output. Also enter a number of **Frames per second** (the default is 10).

Enter information into these fields based on the **Format** for the movie:

- 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**, a scalar value between 0 and 1. The default is 0.75.
- 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.

AVI is a file format that can contain video encoded in different ways.

 However, the AVI format is not supported on Windows XP. For movies using this format, you might also need to change the codec used for animations (in the **Graphics** section of the **Preferences** dialog box).

ANIMATION EDITING

Control how the software creates the frames for the animation sequence. Select a **Sequence type**—[Stored Solutions](#) (the default), [Result Parameter](#), or [Dynamic Data Extension](#). Define the **Sequence type** parameters as required.



Each selected animation sequence component creates a frame in the movie or an individual image file.

Stored Solutions

The default, **Stored solutions**, is useful to animate time-dependent solutions or 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:

From the **Loop over** list, select the steps or parameter values to **Loop over—All solutions** (the default), or if applicable, a parameter or combination of parameters, or any parameter in a **Parametric Sweep** study, or the **Time**. If **All solutions** is chosen, a list appears that contains all combinations of parameter values and times (if applicable).

- **Parametric Sweep Study:** For parametric sweep studies where there are multiple inner solutions (for example, a parameter sweep around a time-dependent solution), **Inner solutions** and **Outer solutions** are also available, typically corresponding to **Time** and parameter values, respectively. Typically, an animation shows variations looping over a parameter, frequency, or time; animating all solutions might be useful to get an overview of all solutions that the model contains.
 - If you choose to loop over **Inner solutions**, select the **Parameter value** set to animate, then the time steps, which you can select, using the **Select via** list, as **Stored output times** (a **Time** list of all stored times), or **Interpolated times** (a text field where you can specify any times within the time range directly). See [Volume Integration](#), [Surface Integration](#), and [Line Integration](#) and [Global](#) for more information about the inner and outer solutions.
 - If you choose to loop over **Outer solutions** (that is, the parameters from the parametric sweep), the **Parameter values** list contains all combinations of parameter values. Select as required, then select an option from the **Inner type** list—**First**, **Last** (the default), or **All**.
- If a **parameter or combination of parameters** are selected to loop over, choose an option from the **Parameter selection** list: **All** (the default), **From list** to select from a

list of all parameter combinations, or **Manual**, to enter a range of parameter value indices directly (or click the **Range** button ()).

If the model includes other parametric sweeps or frequency sweeps, specify the value of those parameters for the frames in the animation in separate **Parameter value** lists. Also, if the model includes a time-dependent solution, select a time step from the **Time** list, or select **Interpolated** to specify any time within the time span in the text field that appears.

- If you select to loop over the **Time**, for time-dependent problems, choose an option from the **Time selection** list—**All** to use all time steps, **From list** to select from a list of all time steps, **Manual** to enter a range of times as indices directly, or select **Interpolated** to enter **Times**. If the model contains a parametric sweep, select an option from the **Parameter value** list.

Result Parameter

Use a **Result parameter** to animate the changes resulting from a sweep of the values for a defined global parameter (found under **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, choose a **Parameter** from the list (or select **None**), which contains all global parameters, and define an interval for the parameter values using the **Start** and **Stop** fields.

Dynamic Data Extension

Use a **Dynamic data extension**, for example, to animate the dynamics of an eigenmode in an eigenfrequency or eigenvalue solution. In such a dynamic data extension, the full harmonic cycle (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, and 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 ($\text{Re}(e^{i\alpha})$, 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.

FRAMES

- If the selected **Sequence type** is **Stored Solutions**, choose a **Frame selection**— **All** (to play all solutions in the stored solution) or **Number of frames**. For any sequence type, or if number of frames is selected here, enter the **Number of frames**. The default is 25 frames.
- From the **Size** list, select **Manual** (the default) or **Current**, which uses the current size of the **Graphics** window.
- If **Manual** is chosen, 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 pixels (width) by 480 pixels (height).
- If required, select the **Record in reverse order** check box.

LAYOUT

- By default, the **Title**, **Legend** (1D graphs) or **Color legend** (2D plots), **Axes**, and **Logotype** (1D and 2D plots) or **Title**, **Color legend**, **Grid**, **Axis orientation**, and **Logotype** (3D plots) parts of the graphics are included. To edit the default, select the **Include** check box and click to clear or select one or several of the available check boxes.
- Enter a **Font size** (pt) for the text in the animation frames. The default is 9 pt.
- Select a **Background—Color** (the default) or **Current**. If **Color** is selected, click **Color** 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.

ADVANCED

If needed, adjust settings for the resolution, antialiasing, and synchronization of the scales in the animation frames.

- Enter a **Resolution** for the images in the animation. The default is 96 DPI.
- Select the **Antialiasing** check box to reduce staircase-like lines and to smooth lines and edges.
- By default, the **Synchronize scales between frames** check box is selected, which means that all frames in the animation use the same color scale, isosurface levels, deformation scale, and so on. This synchronization makes areas with the same solution values keep the same color, for example, during the entire animation.

Click to clear the check box to make the scales and levels adapt to the solution in each frame. This can be useful, for example, for time-dependent simulations of

transient phenomena where the magnitude of the solution changes significantly during the time stepping. With the synchronization active it can then be difficult to distinguish small variations in the solution.

Click the **Export** button () in the settings window or right-click the node and select **Export**. The animation 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.



Parametric Sweep and Solver Overview

Data

Use **Data** () to export numerical data to a 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 export the data in spreadsheet or sectionwise formats.

To export data, you can 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**.



The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.



- Go to [Common Results Node Settings](#) for information about these sections: **Data** and **Expressions**.
 - For detailed information about the formats used, see [Data Formats](#) in the *COMSOL API Reference Manual*.
-

OUTPUT

Enter a **Filename** including a path to save the data file to your computer or click **Browse** and navigate to where you want to **Save** the output. For example, navigate to the desktop and enter a **File name** in the **Export Data** window. It is saved as a .txt file.

Select the **Points to evaluate in**—**Take from data set** (the default), **From file**, **Grid**, or **Regular grid**. Depending on the selected type of points to evaluate in, various settings are available.

Take from Data Set

The default **Take from data set** uses the data points for the data in the data set. Select a **Data format**—**Spreadsheet** (the default) or **Sectionwise**. Spreadsheet data is useful to use the data in spreadsheet applications and sectionwise data format is useful for unstructured interpolation because it contains the exact mesh used to perform the interpolation.

For either choice, select a **Space dimension**—**Take from data set** (the default), **Global, 0, 1, 2, or 3**. Then select a **Geometry level**—**Take from data set** (the default), **Volume**, **Surface**, **Line**, or **Point** (availability is based on the model space dimension).

If **Spreadsheet** is selected (and if required), choose the **Transpose** check box to transpose the data from columns to rows.

From File

If **From file** is selected, it uses coordinates from a data file. Then enter a **Coordinate filename** for a text file with the coordinates for the data output, or click **Browse** to locate the file.

Grid or Regular Grid

If **Grid** or **Regular grid** is selected, it uses a grid to define the points to evaluate in. Select a **Data format**—**Spreadsheet** (the default) or **Grid**. Spreadsheet data is useful for using the data in spreadsheet applications, whereas the grid data format is more compact and can be useful to store data that can be imported into another model.

If **Grid** is selected as the **Data format**, also specify the **x**, **y**, and **z** coordinates for the grid points in the fields, 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).

ADVANCED

- The **Include header** and **Full precision** check boxes are selected by default. Click to clear the check box if you do not want to include a header, or to limit the precision in the output to six significant figures (and which provides an output that contains all significant figures for data stored as double-precision numbers).
- By default the data is unsorted. Select the **Sort** check box to sort the data by increasing **x**, **y**, and **z** coordinates.

- The **Evaluate in** list is only available for data from Solution data sets. From the **Evaluate in** list select **Lagrange points** (the default) or **Gauss points** to specify where COMSOL evaluates the data—the nodes of the Lagrange elements or in the Gauss points for the Gaussian quadrature, respectively.
- Select a **Resolution—Normal** (default), **Finer**, **Fine**, or **Custom**. If **Custom** is selected, enter a **Lagrange-element node-point order** (the default is 1). Use a higher node-point order for a finer resolution.
- Select a data **Smoothing** method—**None**, **Internal** (the default, for smoothing within domains but not across interior boundaries), or **Everywhere**.

Click the **Export** button () in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **Output** section.



Expressions and Predefined Quantities

Mesh

Use the **Mesh** () node 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. The mesh data can also be exported 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:

DATA

Select a **Data set**. The **Data set** list contains the solution and mesh data sets previously defined. Select **None** to not export any mesh data.

OUTPUT

Select a **File type** from the list of available file formats. 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. You can export the mesh to a COMSOL Native file (binary **.mphbin** or text **.mphtxt**) or to an STL file (**.stl** or text **.stl**) (if a 3D data set is selected).

Click the **Export** button () in the settings window or right-click the node and select **Export**.

Table

Use the **Table** () node 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** ().

TABLE

Select a **Table**. The **Table** list contains the all tables in the model. By default, the selection is the one from which you have selected **Add Table to Export** or the first available table if you have added the **Table** node directly under **Export**. Select **None** to not export any table data.

OUTPUT

Enter a **Filename** including a path to save the table data as a file to your computer or click **Browse** and navigate to where you want to **Save** the output and specify the file type as a text file (*.txt), CSV file (*.csv), data file (*.dat), or Microsoft Excel workbook (*.xlsx) from the **Save as type** list. When you save the table data as a Microsoft Excel workbook, you can also specify a **Sheet** and **Range** (by default, those text fields are empty; the program then saves all data), and by default the **Overwrite** check box is selected.

ADVANCED

By default, the **Full precision** check box is selected and exports the table data using full precision for double precision floating-point data. Click to clear the check box to export the data using the table display precision instead.

Click the **Export** button () in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **Output** section.

1D Image, 2D Image, or 3D Image

Use the **1D Image**, **2D Image**, and **3D Image** () nodes to export plot images as PNG, BMP, EPS, TIFF, GIF, or JPEG image files.

To export a plot right-click **Export** () and select **ID image**, **2D Image**, or **3D Image** () or right-click any **Plot Group** node, for example, **3D Plot Group** or **ID Plot Group**, and select **Add Image to Export**. Click the **Image** node under **Export**.

 Use the **Animation** node, for example, to export multiple images for different time steps or eigenvalues.

The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

SCENE

Select a **Subject**. The list contains the 1D, 2D, or 3D plot groups previously defined. **ID Image** uses data from 1D Plot Groups, **2D Image** uses data from 2D Plot Groups, and **3D Image** uses data from 3D Plot Groups.

Under **Scene**, for **2D Image** and **3D Image** models, select a **View—From plot group** (the default) to use the view from the plot group settings, or select a another view from the list (if available and previously defined).

On the settings window toolbar, click the **Refresh** button () to refresh the **Graphics** window to get a preview of the image to export. This is useful when the **Subject** or **View** selection is changed.

IMAGE

Choose a **Size—Manual** to specify the image size manually or **Current** to use the current size of the **Graphics** windows. For either **Size** option, the **Antialiasing** check box is selected by default to reduce staircase-like lines and to smooth lines and edges. The rest of these settings are available if **Manual** is selected.

- Select a **Unit** of dimension—**Pixels (px)** (the default), **Millimeters (mm)**, or **Inches (in)**.
- Select the **Lock aspect ratio** check box to keep the original image width and height.
- In the **Width** and **Height** fields, enter the number of pixels, millimeters, or inches for the final image size.
- Enter a **Resolution**. The default is 96 DPI (dots per inch).

FILE

Choose an image file **Format**—PNG (the default), BMP, JPEG, TIFF, GIF, or EPS (1D only). If you choose the JPEG format you can also control the quality of the image using a quality measure (scalar number) between 1 and 100 (a higher number

represents a higher quality). The default value is 92. Select the check box next to **Quality** to enter another quality number.



JPEG is a format that uses “lossy compression,” so using a low quality measure can make the exported image differ from the original image.

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. The text entered in the **File name** field is used for all the images generated. For example, if `image` is entered, select `.png` as the file type, and then there are 11 frames in the movie and 11 files are created: `image01.png, image02.png, ..., image11.png`.

LAYOUT



When importing an image with a transparent background to another Windows application, first save the image as a file rather than saving it to the clipboard. In some cases the transparent background is not preserved if you copy an image via the clipboard.

One additional **Background** option is available for the Image export when using a PNG file format—**Transparent**. Otherwise, see [Animation](#) for the rest of the settings.

Click the **Export** button () in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **File** section.

Player

A **Player** () creates interactive animations directly in the COMSOL Desktop. To export a player, in the **Model Builder**, right-click **Export** () and select **Player**.

SCENE

Select a **Subject** for the player; the list has the plot groups previously defined, or choose **None**.

ANIMATION EDITING AND ADVANCED

See [Animation](#) for the settings.

FRAMES

- If the selected **Sequence 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.
- Enter the **Number of frames** (if applicable). The default is 25 frames.
- To preview individual frames, enter the **Frame number** or select it using the slider. Observe the geometry in the **Graphics** window to see the **Shown frame** number.

PLAYING

Use this section to adjust some settings that affect the playing of the recorded plots. In the **Display each frame for** field, enter the time to display each frame (in seconds) to control how fast the player runs (default value: 0.1 s). Select the **Repeat** check box to replay the sequence of plots repeatedly instead of playing it just once.

- In the settings window, click **Generate Frame** () (or right-click the **Player** node) to create a series of frames (if more than one **Frame number** is selected).
- Right-click the **Player** node and select **Play** (). Watch each **Frame number** cycle from beginning to end in the **Graphics** window. Use the buttons on the **Graphics** window to **Play** (), **Stop** (), and **Next** () and **Previous** () to cycle through the animation.

Plot

Right-click **Export** () and select **Plot** () to export a plot from a plot group. Or right-click any plot, for example, the **Slice** plot in a 3D Plot Group, and then select **Add Plot Data to Export**.



The nodes are numbered sequentially. To make it easier to organize, you can right-click and **Rename** the node.

PLOT

Select a **Plot group** from the list, which contains any previously defined plot groups. Select a **Plot** to export its data. **Plot Groups** can contain one or more individual plots.

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.

From the **Data format** list, select **Spreadsheet** (the default), **Sectionwise**, **STL Binary File (*.stl)**, or **STL Text File (*.stl)** (last two options are only available for **Volume**, **Surface**, **Slice**, **Multislice**, **Isosurface**, and **Far Field** plots). 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. The filename extension is automatically adjusted according to the data format.



Always check the filename extension after selecting the data format.

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.



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.

ADVANCED

- The **Include header** and **Full precision** check boxes are selected by default. Click to clear the check box if you do not want to include a header, or to limit the precision in the output to six significant figures (and which provides an output that contains all significant figures for data stored as double-precision numbers).
 - By default the data is unsorted. Select the **Sort** check box to sort the data by increasing x , y , and z coordinates.
-



The advanced settings are ignored if you export a plot to an STL file.

Click the **Export** button () in the settings window or right-click the node and select **Export**. The **Messages** window confirms where the files are exported as specified in the **File** section.

Reports

About the Report Generator

The Report Generator is a tool for reporting and documenting models created in COMSOL. It creates a record of the entire model including all the settings made during the modeling process. The report is an overview of the model and includes model properties, geometry, physics, mesh, studies, and results and visualization.

Several reports can be created for each model, and you can 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.

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**) and 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.

Report 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:

- **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 available report components.



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.

The Report Node

The main **Report** node  contains information about the formatting and defaults for the report. Click the **Preview Selected**  or **Preview All**  button to show a preview of the report in the **Preview** window. Click the **Write** button  in the toolbar for the **Report** settings window to create a report. The **Write** option is also available by right-clicking any node in the report. Selecting **Write** from any report node's context menu generates the entire report.

LEVEL OF DETAIL FOR NEW NODES

This section, which is collapsed by default, contains the setting for the level of detail in the new nodes that are added to the report. It is independent of the level of the template used to create the initial report contents. From the **Use default settings for** list, select **Brief**, **Intermediate** (the default), or **Complete** to specify a level of detail for new report nodes that is the same as the corresponding report templates use.

FORMAT

You can select to create a report in one of the following formats, which you choose from the **Output format** list:

- **HTML**, for creating the report as an HTML file for display in a web browser.
- **Microsoft Word**, for creating the report as a docx file for use as a document in Microsoft Word (version 2007 or later).

When adding a new report, the initial setting for the output format is that of the last report you wrote (or previewed).

For HTML reports, select the **Open in browser** check box to view the report in a web browser as soon as it is available. Select the **Optimize for printing** check box to generate images with resolution adapted for printing rather than screen display.

For reports in the docx format, select the **Open finished report** to directly open the generated report in Microsoft Word.

The Report Generator stores the report in a file with the chosen name and by default gives it the extension **.html** or **.docx**. For HTML reports, it stores images included in the report and the style sheet in a subdirectory with the same name as the report plus the suffix **_files**. Reports in Microsoft Word format are self contained.

When generating a report, you need to specify its name and where to store the file. First, use the **Report filename** list to specify whether to link the report name to the name of the model's MPH-file (**From model**) or to specify it independently (**Custom**).

If **From model** is selected, enter the directory for the report in the **Report directory** field or click **Browse** to launch the **Specify Report Directory** dialog box and browse to the desired location. The default directory is the directory where a report was last saved.

If, instead, **Custom** is selected from the **Report filename** list, enter the path and filename in the **Filename** field or click **Browse** to launch the **Specify Report File** dialog box, browse to the desired location, and enter the filename in the **File name** field.

Publishing and Editing an HTML Report

Once an HTML report is created, it can be published or edited in any HTML editor. If you want to publish a report on the Internet or send it to a colleague, also send the HTML file as well as the folder with the images and style sheet.

IMAGES

Select the **Disable image generation** check box if you want to quickly rewrite the report to include changes that only affect the text. If the report includes images that are not

available, COMSOL overrides this setting and generates new images. From the **Size** list, you can select the image size as **Extra small** (240-by-180 pixels for HTML screen output/2.4-by-1.8 inches for print-optimized output), **Small** (320-by-240 pixels/3-by-2.4 inches), **Medium** (480-by-360 pixels/4-by-3 inches), or **Large** (600-by-450 pixels/6-by-4.5 inches). All images have an aspect ratio of 4:3. From the **Type** list, select the image file type: **PNG** (the default), **JPEG**, or **BMP** (not supported for reports in Microsoft Word format; choosing this option gives PNG images).



For the best image quality, use PNG images.



Select **Options>Preferences** from the main menu and then click **Results** to edit the default report settings. See [Preferences Settings](#) for more information.

The Documentation Node

The main **Documentation** node () contains information about the formatting and defaults for a documentation. Click the **Preview Selected** () or **Preview All** () button to show a preview of the document in the **Preview** window. Click the **Write** button () in the toolbar for the **Documentation** settings window to create a document. The **Write** option is also available by right-clicking any node in the documentation. Selecting **Write** from any documentation node's context menu generates the entire document.

FORMAT

You can select to create a document in one of the following formats, which you choose from the **Output format** list:

- **HTML** (the default format), for creating the document as an HTML file for display in a web browser.
- **Help plug-in**, for creating the document as a plug-in of Eclipse Help format that can become an integral part of the documentation and help system.

Settings for Documentation in HTML Format

When generating documentation, you need to specify its name and title where to store the file. Enter the output directory for the HTML files in the **HTML output directory**

field, or click **Browse** to open the **Specify HTML Output Directory** dialog box and browse to the desired location. Also specify the document's name a title in the **Document name** and **Document title** fields, respectively.

Settings for Documentation as Help Plug-in

When creating a help plug-in, you need to specify the following plug-in location details and properties:

- Enter the output directory for the help plug-in in the **Plug-in output directory** field, or click **Browse** to open the **Specify Plug-in Output Directory** dialog box and browse to the desired location.
- Enter a **Plug-in prefix** to use a common name space for your help plug-ins. You can specify a standard plug-in prefix for your organization on the **Builder Tools** page in the **Preferences** dialog box.
- Enter the plug-in name and title in the **Plug-in name** and **Plug-in title** fields, respectively. The complete plug-in name is formed from the plug-in prefix and plug-in name. The plug-in title is the title that appears in the **Contents** tree on the left side of the standalone **Help** window or on the **Contents** page of the **Help** window for context help integrated in the COMSOL Desktop.
- From the **Add to** list, select **None** (the default) to not link the plug-in to the contents of any COMSOL product, or select COMSOL Multiphysics or any of its modules or LiveLink products to add the documentation to one of the products. As an advanced option, you can select **Custom** to link to a custom plug-in that you specify in the **Link to plug-in named** field and the **Link to anchor ID** field. The custom plug-in must extend the org.eclipse.help.toc extension point and contain the specified anchor for your documentation plug-in to link to.
- In the **Vendor** field you can enter the name of your organization. This name appears in the manifest file inside the generated plug-in.
- Enter the version of the plug-in in the **Plug-in version** field. The default version is 1.0.0. Like the vendor name, this number is written to the plug-in's manifest file.

The Title Page

By default, all nonempty reports start with a title page (if it is not needed, right-click the applicable node and choose **Delete**). The **Title Page** node () defines general settings and information about the model. The default name is the model's filename.

FRONT MATTER

In the **Report title** list, choose between linking the report's title to the MPH-file's name (**From model**) or specifying it independently (**Custom**). If you choose **Custom**, use the **Title** field to give the title of the report (the default is based on the MPH-file).

From the **Image** list, select an option for an image in the report title—**None**, **Model thumbnail** (the default), or any of the plot groups' plots in the model.

From the **Layout** list, select **Table** (the default) to present the model settings such as author and date in a table, or select **Headings** to present these settings using headers.

If the **Show logotype** check box is selected (the default) the report include a logotype that you have specified in the **Preferences** dialog box or, if no such logotype is available, the COMSOL logotype.

Use the **Author**, **Date**, **Company**, and **Version** fields if desired to provide that information in the report. Clear the check box in front of each setting to exclude it from the report.

The **Summary** text box contains the comments from the model's **Root** node (the model description) by default. Clear the associated check box to exclude the summary from the report.

The **Acknowledgment** text box is empty by default. Clear the associated check box to exclude the acknowledgment from the report.

The Table of Contents

The **Table of Contents** node () contains the table of content for the report.

LEVELS

The **Section levels in table of contents** list determines how many section levels to include in the table of contents: 1–5 (default: 2).

TABLE OF CONTENTS

This section contains the current table of contents.

Sections in the Report

The **Section** nodes () provide the structure of the report. You can add sections in several levels by right-clicking a **Section** node to add additional **Section** nodes as subsections. The **Section** node's context menu also contains two submenus: **Basic**, for adding basic report component such as paragraphs, images, and tables, and **Model**, for

adding information about the model such as the geometry, mesh, physics interfaces, and plot groups.

SECTION HEADING

From the **Source** list, select the source of the section's heading:

- **Custom** (the default). You then specify the heading in the **Heading** field.
- **From first child node**. The section heading is the name of the first child node under the **Section** node.

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**.

The group of report components provide basic building blocks for a report as described in [Table 20-12](#).

TABLE 20-12: BASIC REPORT AND DOCUMENTATION COMPONENTS

REPORT COMPONENT	ICON	DESCRIPTION
Bibliography		Adds a reference or bibliography to the report or document. Right-click to add Reference www nodes.
Code		Adds a text block for code using a code (monospace) font. You can also make part of the text using an italic or bold variant of the code font.
Equation		Adds an equation to the report or document. You can use LaTeX markup directly or import the equation as an image. Under Equation preview you can see the equation that the LaTeX commands that you enter create.
Heading		Adds a heading to the report or document with a text from the Text field and a layout for the level (Level 1–Level 6) from the Level list. The default is to use the level where the Heading node appears.
Image		Adds an image to the report or document. Select the image source from the Source list: Plot group to select the plot from available plots in the Plot group list or External to use any external image file in PNG, Windows Bitmap (BMP), or JPEG format. Add a Caption if desired.

TABLE 20-12: BASIC REPORT AND DOCUMENTATION COMPONENTS

REPORT COMPONENT	ICON	DESCRIPTION
List		Adds a list. By default, the Numbered check box is selected, giving a numbered list; clear the check box for an unordered (bullet) list. Right-click the List node to add List Item nodes.
List Item		Right-click the List node to add this node with a Text area for the list item's contents. Right-click to add Code , Equation , Image , Table , Text , or other List nodes for inserted texts, equations, images, or tables in the list or for creating nested lists.
Note		Adds a Note node for adding one of the following note types, which you select from the Type list: Note (the default), Caution , Important , Model , See also , or Tip . From the Show list, select Icon (the default) to display the icon only, Description (the type), or Icon and description . Then add the text for the note.
Reference		Right-click a Bibliography node to add references. The reference information that you provide is formatted based on the type of reference that you select from the Type list: Journal article (the default), Book , Conference paper , Thesis , or Web .
Table		Adds a table with a Title and a Number of columns (default: 3 columns). Right-click to add a Table Heading Row and Table Rows .
Table Heading Row		Right-click the Table node to add this node and then define headings for each column.
Table Rows		Right-click the Table node to add this node and then add the contents for each column in a row of a table.
Text		Provides a Text area where text can be added (including HTML tags for formatting and links).

For all **Text**, **List Item**, and **Note** nodes' settings, a set of tools above and beyond the text field provides a quick way to add formatting to the text:

- The formatting tools above the text provide character formats for user-interface labels, emphasis, code (standard, bold, and italic), equation components (bold, variables, and constants), subscript, and superscript. To convert a part of the text to any of these character formats, highlight the text that you want to format and then

click **L**, for example, to mark the text as a user-interface label (a sans-serif boldface font) using the HTML tags <1> and </1> before and after the text.

- From the character tools below the text, click the character that you want to insert, for example, click **Ω** to insert an uppercase omega as \Omega in the text. The character tools include lowercase and uppercase Greek letters and the en-dash (–) and em-dash (—) punctuation symbols.

Click **Preview Selected** () to display a preview of the text, including formatting, in the **Preview** window.

Mathematical Symbols and Special Characters

COMSOL supports a subset of the LaTeX language for creating equations as part of the documentation or in user-developed physics interfaces and other applications. Commands include Greek and other characters, mathematical symbols and operators, arrows, text and font formats, and environments for text and mathematical typesetting. The following tables and lists contain the commands that are available for creating equations and other mathematical text.



If the LaTeX syntax is not correct or not included in COMSOL, the equation preview is empty, but no error appears.

GREEK AND OTHER CHARACTERS

The following table contains the supported lowercase and uppercase Greek letters and the Swedish character Å:

TABLE 20-13: GREEK AND OTHER CHARACTERS

COMMAND (UPPERCASE)	CHARACTER	COMMAND (LOWERCASE)	CHARACTER
		\alpha	α
		\beta	β
\Gamma	Γ	\gamma	γ
\Delta	Δ	\delta	δ
		\varepsilon	ε
		\epsilon	€
		\zeta	ζ
		\eta	η

TABLE 20-13: GREEK AND OTHER CHARACTERS

COMMAND (UPPERCASE)	CHARACTER	COMMAND (LOWERCASE)	CHARACTER
\Theta	Θ	\theta	θ
		\vartheta	ϑ
		\iota	ι
		\kappa	κ
\Lambda	Λ	\lambda	λ
		\mu	μ
		\nu	ν
\Xi	Ξ	\xi	ξ
\Pi	Π	\pi	π
		\varpi	ϖ
		\rho	ρ
\Sigma	Σ	\sigma	σ
		\varsigma	ς
		\tau	τ
\Upsilon	Υ	\upsilon	υ
\Phi	Φ	\phi	ϕ
		\varphi	φ
		\chi	χ
\Psi	Ψ	\psi	ψ
\Omega	Ω	\omega	ω
\AA	Å		

ACCENTS

The following accents are available:

TABLE 20-14: GREEK AND OTHER CHARACTERS

COMMAND	ACCENT	COMMAND	ACCENT
\acute{e}	é	\bar{e}	̄e
\breve{e}	᷇	\check{e}	᷈e
\ddot{e}	᷉	\dot{e}	᷊e

TABLE 20-14: GREEK AND OTHER CHARACTERS

COMMAND	ACCENT	COMMAND	ACCENT
\grave	è	\hat	ê
\tilde	~ e	\vec	→ e

MATHEMATICAL SYMBOLS AND OPERATORS

The following mathematical symbols and operators are available:

TABLE 20-15: GENERAL SYMBOLS AND MATHEMATICAL OPERATORS

COMMAND	SYMBOL	COMMAND	SYMBOL
\dots	...	\nabla	∇
\ldots	...	\bot	⊥
\hbar	□	\diamondsuit	◆
\Re	ℜ	\neg	¬
\forall	∀	\not	¬
\cdots	...	\imath	ı
\Im	ℐ	\exists	Ξ
\prime	'	\triangle	Δ
\top	⊤	\heartsuit	♥
\flat	♭	\vdots	⋮
\Diamond	◇	\aleph	ℵ
\mho	℧	\emptyset	∅
\infty	∞	\angle	∠
\clubsuit	♣	\pounds	£
\ddots	⋱	\Box	◻
\wp	℘	\partial	∂

TABLE 20-15: GENERAL SYMBOLS AND MATHEMATICAL OPERATORS

COMMAND	SYMBOL	COMMAND	SYMBOL
\surd	$\sqrt{}$	\spadesuit	\spadesuit
\dag	\dagger	\ddag	\ddagger
\S	\S	\P	\P
\copyright	\circledcirc	\textregistered	\circledR

The following table lists the available “big” mathematical operator as well as binary mathematical operators and relations:

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\sum	Σ	\prod	Π
\coprod	\coprod	\int	\int
\bigoplus	\bigoplus	\bigcup	\bigcup
\bigcap	\bigcap	\bigsqcup	\bigsqcup
\oint	\oint	\bigotimes	\bigotimes
\bigvee	\bigvee	\bigwedge	\bigwedge
\biguplus	\biguplus	\bigodot	\bigodot
\pm	\pm	\cdot	\cdot
\times	\times	\cup	\cup

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\sqcup	\sqcup	\vee	\vee
\sqcap	\sqcap	\oplus	\oplus
\otimes	\otimes	\lhd	\lhd
\unlhd	\lneq	\mp	\mp
\div	\div	\setminus	\setminus
\cap	\cap	\sqcap	\sqcap
\wedge	\wedge	\land	\wedge
\ominus	\ominus	\oslash	\oslash
\rhd	\rhd	\unrhd	\rhd
\star	\star	\ast	\ast
\circ	\circ	\bullet	\bullet
\uplus	\uplus	\amalg	\amalg
\dagger	\dagger	\ddagger	\ddagger
\wr	\wr	\leq	\leq
\leq	\leq	\ll	\gg
\prec	\prec	\preceq	\preceq
\subset	\subset	\subseteq	\subseteq
\sqsubset	\sqsubset	\sqsubseteq	\sqsubseteq
\in	\in	\vdash	\vdash
\mid	$ $	\geq	\geq
\geq	\geq	\gg	\gg

TABLE 20-16: BIG AND BINARY MATHEMATICAL OPERATORS AND RELATIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
\succ	\succ	\succeq	\succcurlyeq
\supset	\supset	\supseteq	\supseteqq
\sqsupset	\sqsupset	\sqsupseteq	\sqsupseteqq
\ni	\ni	\owns	\ni
\dashv	\dashv	\parallel	\parallel
\notin	\notin	\equiv	\equiv
\doteq	\doteq	\sim	\sim
\simeq	\simeq	\approx	\approx
\cong	\cong	\bowtie or \Join	\bowtie
\proto	\circ	\models	\models
\perp	\perp	\asymp	\asymp
\neq	\neq	\ne	\neq
\bigtriangleup	\bigtriangleup	\bigcirc	\bigcirc
\bigtriangledown	\bigtriangledown	\triangleleft	\triangleleft
\triangleright	\triangleright	\diamond	\diamond
\smile	\smile	\frown	\frown

ARROWS

The following table includes the available types of arrows:

TABLE 20-17: ARROWS

COMMAND	SYMBOL	COMMAND	ARROW
\leftarrow or \gets	\leftarrow	\rightarrow or \to	\rightarrow
\Leftarrow	\Leftarrow	\Rightarrow	\Rightarrow

TABLE 20-17: ARROWS

COMMAND	SYMBOL	COMMAND	ARROW
\Leftrightarrow	\Leftrightarrow	\leftrightarrow	$\leftarrow\rightarrow$
\hookleftarrow	\hookleftarrow	\leftharpoonup	$\leftarrow\swarrow$
\leftharpoondown	\leftharpoondown	\leftrightharpoons	$\leftarrow\rightleftharpoons$
\nearrow	\nearrow	\swarrow	$\swarrow\leftarrow$
\leadsto	\leadsto	\hookrightarrow	$\hookrightarrow\rightarrow$
\rightharpoonup	\rightharpoonup	\rightharpoondown	$\rightarrow\leftharpoonup$
\searrow	\searrow	\nwarrow	$\nwarrow\leftarrow$
\downarrowtobar	\downarrow	\uparrowtobar	$\uparrow\downarrow$
\downtrianglefilled	\blacktriangledown	\uptrianglefilled	$\blacktriangle\uparrow$
\mapsto	\mapsto	\longmapsto	\longrightarrow
\longleftarrow	\longleftarrow	\longrightarrow	\longrightarrow
\longleftrightarrow	\longleftrightarrow	\Longleftarrow	$\Longleftarrow\Longleftrightarrow$
\Longrightarrow	\Longrightarrow	\Longleftrightarrow	$\Longleftrightarrow\Longleftrightarrow$
\iff	\iff		

DELIMITERS AND ENVIRONMENTS

The following tables includes the available delimiter, spaces, and environments:

TABLE 20-18: DELIMITERS, SPACES, ENVIRONMENTS, BOXES

COMMAND	SYMBOL/ EXPLANATION	COMMAND	SYMBOL/EXPLANATION
\uparrow	\uparrow	\updownarrow	\updownarrow
\downarrow	\downarrow	\Uparrow	\Uparrow
\Downarrow	\Downarrow	\Downarrow	\Downarrow
\lbrack	\lbrack	\rbrack	\rbrack

TABLE 20-18: DELIMITERS, SPACES, ENVIRONMENTS, BOXES

COMMAND	SYMBOL/ EXPLANATION	COMMAND	SYMBOL/EXPLANATION
\lbrace	{	\rbrace	}
\vert		\backslash	\
\Vert		\lfloor	[
\lceil	\lceil	\rfloor	\rceil
\rceil	\rceil	\langle	\langle
\rangle	\rangle	\left	Delimiter sizing (see Note below)
\right	Delimiter sizing (see Note below)	\quad	Explicit horizontal spacing
\qquad	Double explicit horizontal spacing	\raisebox	Creates a box containing text; it is used to raise or lower text.
\mbox	Enclose text in a box	\phantom	Adds an invisible component to, for example, balance subscripts
\begin	Invoke the <i>array</i> environment (see Note below)	\end	End the <i>array</i> environment (see Note below)
\unicode	Display Unicode characters as supported by the font: \unicode{ÅÄÖ}, for example	\hspace	Horizontal space



The \left and \right commands must be used in pairs to provide flexible delimiters that fit the formula inside. Put the desired delimiter—(and), for example—immediately after the \left and \right commands. For example, \left(\frac{x}{y} \right) provides x/y as a fraction within parentheses that fit the expression's size.



The `\begin` and `\end` commands must be used in pairs to mark the beginning and end of an environment. The only supported environment is the `array`. For example, `\begin{array}{clcr} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}` creates a matrix with 2 rows and three columns.

MATHEMATICAL FUNCTION NAMES

The following function commands provide the function name using a Roman font:

TABLE 20-19: FUNCTIONS

COMMAND	SYMBOL	COMMAND	SYMBOL
<code>\arccos</code>	<code>arccos</code>	<code>\cos</code>	<code>cos</code>
<code>\csc</code>	<code>csc</code>	<code>\exp</code>	<code>exp</code>
<code>\ker</code>	<code>ker</code>	<code>\limsup</code>	<code>limsup</code>
<code>\arcsin</code>	<code>arcsin</code>	<code>\cosh</code>	<code>cosh</code>
<code>\deg</code>	<code>deg</code>	<code>\gcd</code>	<code>gcd</code>
<code>\lg</code>	<code>lg</code>	<code>\ln</code>	<code>ln</code>
<code>\arctan</code>	<code>arctan</code>	<code>\cot</code>	<code>cot</code>
<code>\det</code>	<code>det</code>	<code>\hom</code>	<code>hom</code>
<code>\lim</code>	<code>lim</code>	<code>\log</code>	<code>log</code>
<code>\arg</code>	<code>arg</code>	<code>\coth</code>	<code>coth</code>
<code>\dim</code>	<code>dim</code>	<code>\inf</code>	<code>inf</code>
<code>\liminf</code>	<code>liminf</code>	<code>\max</code>	<code>max</code>
<code>\sinh</code>	<code>sinh</code>	<code>\sup</code>	<code>sup</code>
<code>\tan</code>	<code>tan</code>	<code>\tanh</code>	<code>tanh</code>
<code>\min</code>	<code>min</code>	<code>\Pr</code>	<code>Pr</code>
<code>\sec</code>	<code>sec</code>	<code>\sin</code>	<code>sin</code>

SPECIAL MATHEMATICAL TYPESETTING

There are two mathematical formula components with a special syntax: `\frac` for fractions and `\sqrt` for roots:

- Use the syntax `\frac{numerator}{denominator}` to create a fraction. For example, the expression `\frac{n!}{k!(n-k)!}` produces the following output:

$$\frac{n!}{k!(n-k)!}$$

- Use the syntax `\sqrt[order]{expression}` to create a root surrounding an expression. The `[order]` argument is optional; without it, the syntax produces a square root. For example, `\sqrt[n]{1+x^2}` produces the following output:

$$\sqrt[n]{1+x^2}$$

TEXT AND FONT ELEMENTS

The following syntax elements are available for creating different text elements and fonts:

TABLE 20-20: VARIOUS TEXT AND FONT OPERATIONS

COMMAND	EXPLANATION	COMMAND	EXPLANATION
<code>\textsuperscript</code>	Superscripts	<code>^</code>	Superscripts
<code>\textsubscript</code>	Subscripts	<code>_</code>	Subscripts
<code>\overline</code>	Overlining	<code>\underline</code>	Underlining
<code>\overleftarrow</code>	Overlining using a left-pointing arrow	<code>\underleftarrow</code>	Underlining using a left-pointing arrow
<code>\overrightarrow</code>	Overlining using a right-pointing arrow	<code>\underrightarrow</code>	Underlining using a right-pointing arrow
<code>\overbrace</code>	Overlining using a brace	<code>\underbrace</code>	Underlining using a brace
<code>\textnormal</code>	Normal text	<code>\textbf</code>	Boldface text
<code>\textit</code>	Text in italics	<code>\textrm</code>	Text in Roman font
<code>\mathnormal</code>	Normal mathematical mode (the default)	<code>\mathbf</code>	Mathematical boldface text
<code>\mathit</code>	Mathematical text in italics	<code>\mathrm</code>	Mathematical text in Roman font
<code>\displaystyle</code>	Size for equations in display mode	<code>\textstyle</code>	Size for equations in text mode
<code>\scriptstyle</code>	Size for first subscript or superscript	<code>\scriptscriptstyle</code>	Size for subsequent subscripts and superscripts
<code>\emph</code>	Emphasize text in normal (Roman) text mode	<code>\tiny</code>	Smallest font size in text mode
<code>\scriptsize</code>	Second smallest fontsize in text mode	<code>\footnotesize</code>	Third smallest font size in text mode
<code>\small</code>	Fourth smallest fontsize in text mode	<code>\normalsize</code>	Normal font size in text mode

TABLE 20-20: VARIOUS TEXT AND FONT OPERATIONS

COMMAND	EXPLANATION	COMMAND	EXPLANATION
\large	Fifth largest font size in text mode	\Large	Fourth largest font size in text mode
\LARGE	Third largest font size in text mode	\huge	Second largest font size in text mode
\Huge	Largest font size in text mode		



The `\textsuperscript` and `^` syntax alternatives are identical for creating superscripts. Likewise, `\textsubscript` and `_` are identical for creating subscripts.

SPECIAL CONTROL SEQUENCES AND CHARACTERS

The following special control sequences and special characters are available:

TABLE 20-21: SPECIAL CONTROL SEQUENCES AND SPECIAL CHARACTERS

SEQUENCE/CHARACTER	DESCRIPTION	SEQUENCE/CHARACTER	DESCRIPTION
\#	# character	\:	Medium space
\\$	\$ character	\; ;	Thick space
\%	% character	\! !	Negative thin space
\&	& character	\\$, or /[to start and /] to end	Start and end mathematical mode in text mode
_	_ character	\%	Insert comments
\{	{ character	\&	Separate items in arrays
\}	} character	\~	Nonbreaking space
\	character	_	Subscript
\<space>	Space	\^	Superscript
\,	Thin space	\, \{ \}, [,]	Command syntax elements

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:

- 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

Definitions Report Nodes

Right-click nodes to select items from the **Model** submenu and to add the nodes described in this section. These report components provide information about nodes added to the model under Definitions and the Global Definitions node (for variables).



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

COORDINATE SYSTEM

Use the **Coordinate System** report node (), selected from the **Model** submenu, to add coordinate system settings information to the report. Select a **Coordinate system** from the list. Select the **Include settings** check box as required.

FUNCTION

Use the **Function** report node ()**,** selected from the **Model** submenu, to add a function image and the function's settings to the report. Under **Referenced Function**, select a **Function** from the list. The **Include image** check box is selected by default. Select the **Include settings** check box as required.

INFINITE ELEMENT DOMAIN

Use the **Infinite Element Domain** report node ()**,** selected from the **Model** submenu, to include the selection and settings for an infinite element domain if it is present in

the model. From the **Source** list, select the Infinite Element Domain node to report. Select the **Include settings** check box to include the infinite element settings.

MODEL COUPLING

Use the **Model Coupling** report node (🔗), selected from the **Model** submenu, to include the settings and a selection image for a model coupling if it is present in the model. Select the **Model coupling** from the list. The **Include settings** check box is selected by default if there is a model coupling. Select the **Include selection image** check box as required. The image shows the selection for the model coupling and is not present if the model coupling has no selection.

PAIR

Use the **Pair** report node (鹣), selected from the **Model** submenu, to include the settings and a selection image for identity pairs and contact pairs if present in the model. Select the **Pair** from the list. The **Include settings** check box is selected by default if there is a pair. Select the **Include selection image** check box as required. The image shows the selection for the pair and is not present if the pair has no selection.

PERFECTLY MATCHED LAYER

Use the **Perfectly Matched Layer** report node (��), selected from the **Model** submenu, to include the selection and settings for a perfectly matched layer (PML) if it is present in the model. From the **Source** list, select the **Perfectly Matched Layer** node to report. Select the **Include settings** check box to include the PML settings. The image shows the selection for the PML and is not present if the PML has no selection.

PROBE

Use the **Probe** report node (🎯), selected from the **Model** submenu, to includes the settings and a selection image for a Probe if it is included in the model. Select the **Probe** from the list. The **Include settings** check box is selected by default if there is a probe. Select the **Include selection image** check box as required. The image shows the selection for the probe and is not present if the probe has no selection.

SELECTION

Use the **Selection** report node (選擢), selected from the **Model** submenu, to includes the settings and a selection image for selections. Select the **Selection** from the list. The **Include settings** check box is selected by default if there is a selection. Select the **Include selection image** check box as required. The image is not present if the selection is empty.

VARIABLES

Use the **Variables** report node (), selected from the **Model** submenu, to include global or local variable settings. Select the **Variables** from the list.

Results Report Nodes

Right-click to select items from the **Model** submenu and to add the nodes described in this section. These report components provide information about nodes added to the model under Results. For any of these click the **Go to Source** button () to move to the source node under the applicable node under **Results**.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

DATA SET

Use the **Data Set** report node (), selected from the **Model** submenu, to include the settings and a selection image for the referenced Data set. Select a **Data set** from the list or click the **Data Sets**. The **Include settings** check box is selected by default. Select the **Include selection image** check box as required.

DERIVED VALUES

Use the **Derived Values** report node (), selected from the **Model** submenu, to add derived values settings information to the report. Select a **Derived values** from the list. The **Include settings** check box is selected by default.

EXPORT

Use the **Export** report node (), selected from the **Model** submenu, to add any of the images or animations added to the model Export branch. From the **Object** list, select from any available image or animation objects to include in the report (or select **None**).



Animations are not supported for reports in Microsoft Word format; if included, they are silently ignored when writing the report.

After selecting the **Object**, select an option from the **Size** list—**Object setting** to use the width and height specified in the image or animation settings, or **Report image type** to use the image format specified in the report's root node settings.

Select an option from the **File format** list—**Object setting** to use the file format for the selected object or **Report image type** to be determined by the image type setting specified in the report’s root node.

In the **Caption** field enter text as required. By default, this field is left empty and no caption is included in the report.

PLOT GROUP

Use the **Plot Group** report node (), selected from the **Model** submenu, to add plots to the report. Select the **Plot group** from the list. From the **Caption source** list, select **From plot group title** (the default) to use the plot’s title as the caption, **Custom** to enter a different **Caption** in the field, or **None** for no caption.

TABLE

Use the **Table** report node (), selected from the **Model** submenu, to add the table settings to the report. Select the **Table** from the list.

Geometry Report Node

Use the **Geometry** report node (), selected from the **Model** submenu, to add the image, units (length and angular), and statistics to the report for specific geometry features.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

REFERENCED GEOMETRY

Select a **Geometry** from the list. The **Include image** and **Include units** check boxes are selected by default. Select the **Include statistics** check box as required.

In the **Features** table under **Name**, all the features used in the geometry sequence are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove feature-specific settings, in the **Settings** column, click to cycle between the check mark icon () to include a feature, and the delete icon () to remove a feature from the report.

Material Report Node

Use the **Material** report node (), selected from the **Model** submenu, to add the image, selection, and settings to the report for the material property groups in the material.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

MATERIAL

Select a **Material** from the list. The **Include image** and **Include selection** check boxes are selected by default. Select the **Include settings** check box as required. The image shows the selection for the material and is not present if the material has no selection or is completely overridden by other materials' selections.

In the **Features** table under **Name**, all the material properties used in the material are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove material properties settings, in the **Settings** and **Functions** columns, click to cycle between the check mark icon () to include a feature, and the delete icon () to remove a feature from the report.

Mesh Report Node

Use the **Mesh** report node (), selected from the **Model** submenu, to add the image and statistics to the report.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

MESH

Select a **Mesh** from the list. The **Include image** check box is selected by default. Select the **Include statistics** check box as required.

In the **Features** table under **Name**, all the mesh features are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove settings, in the **Settings** column, click to cycle between

the check mark icon (✓) to include a feature, and the delete icon (✘) to remove a feature from the report.

Model Report Node

Use the **Model** report node (), selected from the **Model** submenu, to include information from a Model node in the model (selected from the Model list in the Model Settings section).



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

MODEL SETTINGS

Select the **Model** from the list. Select the **Include unit system** and **Include geometry shape order** check boxes as required.

Parameters Report Node

Use the **Parameters** report node (), selected from the **Model** submenu, to include the Global Parameters in the model.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

Physics Interface Report Node

Use the **Physics Interface** report node (), selected from the **Model** submenu, to add an image and table showing the selection, equations, settings, and a table of all included physics features to the report.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

REFERENCED INTERFACE

Select a **Physics interface** from the list. The **Include selection image** and **Include feature table** check boxes are selected by default. Select the **Include selection table**, **Include equations**, and **Include settings** check boxes as required. The image shows the selection for the physics and is not present if the physics has no selection.

In the **Features** table under **Name**, all the physics interface features are listed. By default, all settings and selections are included in **Intermediate** and **Complete** reports, and no settings or selections are included for a **Brief** report.

To add or remove settings and selections, in the **Settings** and **Selection** columns, click to cycle between the check mark icon (✓) to include a feature, and the delete icon (✘) to remove a feature from the report.

Under the **Features** table, you can also select from the check boxes to include variables, shape functions, weak expressions, and constraints contained in the physics node's **Equation View** subnode. By default, a **Complete** report includes all of these items and the **Intermediate** reports include variables. **Brief** reports do not include any items.

Select the **Include variables**, **Include shape functions**, **Include weak expressions**, and **Include constraints** check boxes as required.

Root Report Node

Use the **Root** report node (), selected from the **Model** submenu, to include information from the model's Root node—model name and path, COMSOL program version, the used products, and unit system.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

GLOBAL SETTINGS

Select the **Include name**, **Include path**, **Include program**, **Include unit system**, and **Include used products** check boxes as required.

Solver Report Node

Use the **Solver** report node (), selected from the **Model** submenu, to add settings for the solver nodes in the solver sequence that is referenced in the Sequence list.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

SOLVER

Select a **Sequence** from the list. Select the **Include log** to include the solver log.

In the **Features** table under **Name**, all the solver features are listed. By default, all settings are included in **Intermediate** and **Complete** reports, and no settings are included for a **Brief** report. To add or remove settings, in the **Settings** column, click to cycle between the check mark icon () to include a feature, and the delete icon () to remove a feature from the report.

Study Report Node

Use the **Study** report node (), selected from the **Model** submenu, to add settings for a Study in the model to the report.



Go to [Common Results Node Settings](#) for links to information about this section: **Node Properties**.

STUDY

Select a **Study** from the list.

In the **Features** table under **Name**, all the studies are listed. By default, all settings are included for all report types. To add or remove settings, in the **Settings** column, click to cycle between the check mark icon () to include a feature, and the delete icon () to remove a feature from the report.

Running COMSOL

This chapter provides an overview of the different ways that you can run the COMSOL Multiphysics® software in addition to running the COMSOL Desktop® graphical user interface on a dedicated computer, including client/server and distributed-memory architectures and cloud-based computing.

Running COMSOL

The primary way to access the COMSOL Multiphysics® functionality is through the COMSOL Desktop®. This section describes alternative means of accessing COMSOL’s functionality.

COMSOL Client/Server

The COMSOL Multiphysics client/server architecture lets you access the COMSOL server—COMSOL’s computational engine—as a separate process. The COMSOL server is single user server allowing multiple session of the same user, one session at a time. 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 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 Multiphysics 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.

 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 Guide*.

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 API can be used for developing standalone applications based on COMSOL functionality.

The most basic use of the COMSOL 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 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 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 API Reference Manual*.

See the *COMSOL Installation 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 Multiphysics 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](#).

You can also run COMSOL batch entirely from a command prompt.

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.

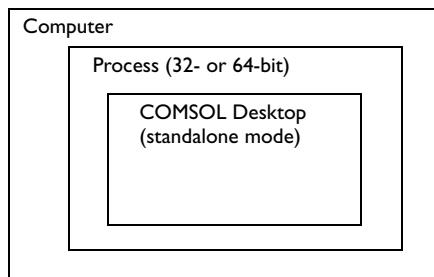
LiveLink for MATLAB

The LiveLink™ for MATLAB® provides access to a COMSOL Multiphysics 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 API and the Java® interface in MATLAB. 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

Standalone COMSOL

The most straightforward way of running COMSOL Multiphysics® is as a standalone application.



Running COMSOL as a Client/Server

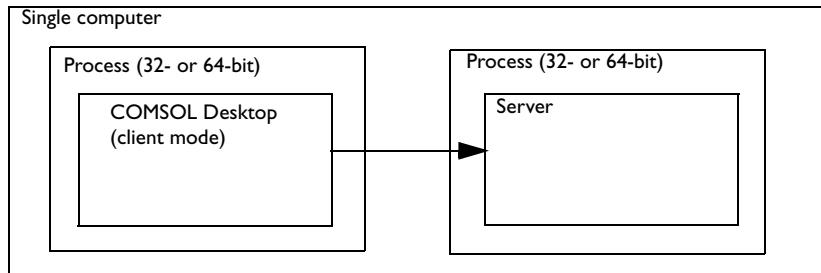
The COMSOL client and server applications are available on all platforms.



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.

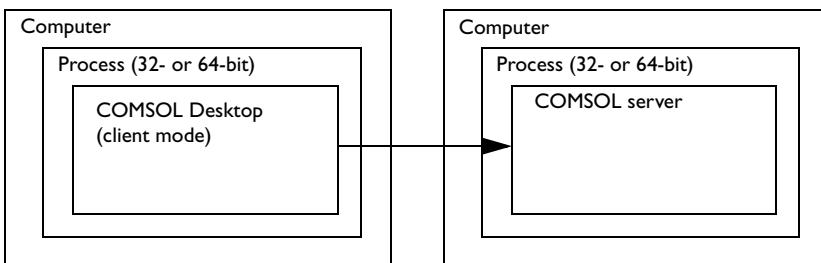
RUNNING COMSOL MULTIPHYSICS CLIENT/SERVER ON THE SAME COMPUTER

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

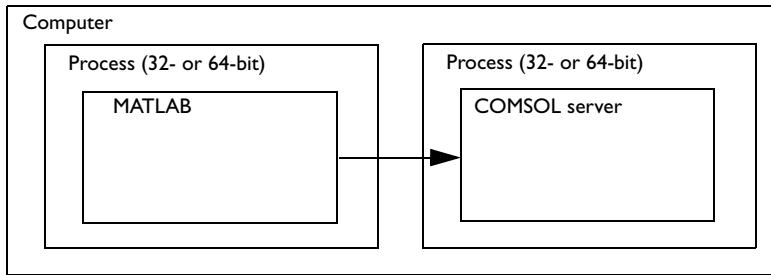
The COMSOL client and COMSOL Multipysics server can also run on different computers, but this configuration requires a floating network license (FNL).



Running COMSOL with MATLAB

COMSOL Multipysics 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 used 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 Multiphysics[®] 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.

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.



Running the COMSOL server and 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 4.3b** and then **Client Server**, and select **COMSOL Multiphysics 4.3b 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).



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.



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 Multiphysics® 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.

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.



These plots run in parallel in 3D: slice plots, isosurface plots, volume plots, line plots, deformed-shape plots, and streamline plots.



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 *COMSOL Installation 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 Multiphysics 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 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 Multiphysics 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.

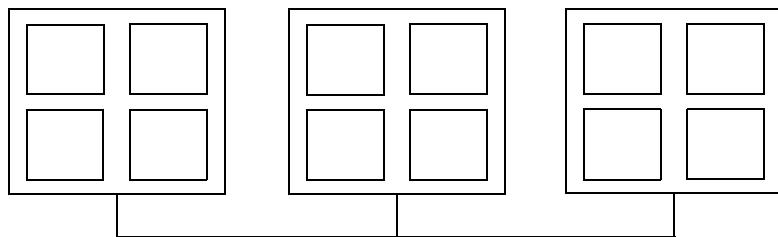


Figure 21-1: Schematic of a cluster with 3 physical nodes (computers) with 4 processors each.

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.



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 21-1](#), 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 Multiphysics 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, SCGS, and Vanka
- Multigrid
- Domain decomposition

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**.

	You must have a floating network license (FNL) to run COMSOL in distributed memory parallel mode. See Cluster License Handling for more information.
	Cluster Computing (Study)
	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.

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.



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 4.3b 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.



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.

The COMSOL Commands

The following sections describe the `comsol` commands on the Windows®, Linux®, and Macintosh platforms.

COMSOL Commands on Windows

Use a COMSOL command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL commands is

```
<command> [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. There are several different commands (See `<command>` in the command syntax) that can be combined with optional targets to achieve various results. The table below lists the major available commands and targets.:.

TABLE 21-1: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
comsol	Run standalone COMSOL Multiphysics	
comsolserver	Start COMSOL Multiphysics server	
comsol client	Run COMSOL Multiphysics client	
comsolbatch	Run a COMSOL MPH-file or class file	
comsolcompile	Compile a Model Java-file	
comsolcluster	Run COMSOL Desktop on a cluster	Requires a floating network license (FNL)
comsolclusterbatch	Run COMSOL cluster version in batch mode	Requires a floating network license (FNL)
comsolclusterserver	Run COMSOL cluster server	Requires a floating network license (FNL)

TABLE 21-1: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
comsolserver matlab	Start MATLAB® and connect to a COMSOL server	Requires a LiveLink™ for MATLAB® license
comsol convertpre35a	Convert 3.0–3.5 models	

The commands are available in 32-bit versions in the subdirectory `bin\win32` in the COMSOL installation directory, and in 64-bit versions in the subdirectory `bin\win64` in the COMSOL installation directory. The COMSOL installer sets up a few of the possible commands on your start menu and your desktop. In Windows® 8, you can click the shortcut **COMSOL Launchers** on the Apps screen. This makes a folder with shortcuts to all COMSOL commands available.

To create additional customized commands, you can create shortcuts including all argument and put them on your desktop. You can also issue COMSOL commands in a command window. To conveniently access the command in a command window, you need to set up the Windows path to include one of the paths `bin\win32` or `bin\win64` in the COMSOL installation directory.

INI FILES

For each launcher file, there is a corresponding `.ini` file in the same directory. It is sometimes recommended that these files are edited. For example, you can add options to any of the above commands by modifying the corresponding ini file. To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsolbatch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the COMSOL command and target. Table 21-2 lists the options (See [`<options>`] in the command syntax) available for all COMSOL commands. Always issue these options between the command and the target (if any).

TABLE 21-2: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
<code>-h</code>	Print general help	
<code><target> -h</code>	Print target-specific help	

TABLE 21-2: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
-3drend ogl dx9 sw	3D renderer: OpenGL, DirectX, or software rendering	
-docroot <path>	Specify custom path to the COMSOL documentation root directory.	See Documentation and Model Library Root Directories
-modelsroot <path>	Specify custom path to the COMSOL Model Library root directory.	See Documentation and Model Library Root Directories
-np <no. of processors>	Number of processors	See Shared-Memory Options
-numasets <no. of sets>	Number of NUMA sets	See Shared-Memory Options
-mpmode throughput turnaround owner	Multiprocessor mode	See Shared-Memory Options
-blas {auto} mkl acml path	BLAS library to use	See BLAS Options
-blaspath <path>	BLAS library path	See BLAS Options
-ipv6	Activate IPv6 support	
-c <path>	License file path	
-prefsdir <path>	Preference directory	
-tmpdir <path>	Temporary file directory	
-version	Print COMSOL version	
-version <target>	Print target version	
-ckl	Use class-kit license	
-autosave {on} off	Control saving of recovery files	
-recoverydir <path>	Path to recovery directories	

For the `-tmpdir` option, COMSOL software uses the specified directory to store temporary files. The `-prefsdir` option specifies the directory where COMSOL should store the preference file.

Documentation and Model Library Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Library root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Library root directories can be useful for administering Model Library Update; see [The Model Library](#) and [Preferences for Updates](#).

Shared-Memory Options

Use the option `-np` to control the number of core and processors used. The default is to use all available cores and processors.

Use the option `-numasets` to control the number of Non-Uniform Memory Access node sets COMSOL should take into account. This is usually the number of processor sockets that the hardware is using.

Depending on how loaded the machine is, you can control how COMSOL uses the available processors with the `-mpmode` option. The following options are available:

TABLE 21-3: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
throughput	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
turnaround	Typically provides the best performance when no other processes than COMSOL are active.
owner	Provides the highest performance in most cases.

You may need to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides for the following BLAS related options:

TABLE 21-4: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL for Intel processors, and ACML for AMD processors with SSE2 support, otherwise MKL. (This is the default option.)
mkl	Use the Intel MKL library.

TABLE 21-4: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
acml	Use the AMD ACML library.
path	Use a BLAS library specified using the option -blaspath or the environment variable COMSOL_BLAS_PATH. The library must support the standard FORTRAN BLAS interface.

Both MKL and ACML are distributed along with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in /lib/ARCH where ARCH is the architecture (win32 or win64) or somewhere in the standard search path. Also provide the path to any sub-libraries needed by the library. Set the search path to point to the directory where the library is installed. To do so, use the environment variable PATH.

COMSOL COMMANDS

In addition to the options in [Table 21-2](#), the standalone COMSOL command supports the following option.

TABLE 21-5: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
-open <file>	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ for MATLAB® also needs to connect to a COMSOL server.

The Windows syntax for the COMSOL server command is

```
comsolserver [<options>] [<target arguments>]
```

The following target arguments are available for a COMSOL server command.

TABLE 21-6: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTION	DESCRIPTION
-user <user>	Specify login name for a user
-port <port>	Specify a TCP/IP port to listen for connect attempts.

TABLE 21-6: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTION	DESCRIPTION
-passwd reset nostore	Specify that you want to provide a new password. To avoid storing the new password on file use nostore
-login {info} force never auto	Ask for login information. info means that only missing information is asked for. force resets the password. never requires that the login information is available. auto automatically creates a new username and password
-multi on {off}	Accept repeated client connections
-silent	Do not listen to standard input
-graphics	Start the server with graphics libraries. This displays plots on the server when you are connected with a client (that is, not with the COMSOL GUI).

Accessing the COMSOL Server Computer

The server computer can be accessed in several ways. If it is dedicated to a single person, you can sit down at that machine and log in on it. You can also connect to the server computer by using Remote Desktop. Start the COMSOL server from the **Start** menu. If several people want to access a single Windows computer to run the COMSOL server, you must use Windows Terminal Server or another tool that allows multiple users to log in on the same Windows server. In some Windows versions, Microsoft® provides a Telnet Server with which you can log in through a terminal window. When using a terminal window to log in on Windows, use the **comsolserver** command to start the COMSOL server.

Login Information

When a COMSOL server is started for the first time, you are asked for a user name and password. Select a user name and a password, which COMSOL Multiphysics then uses in communications between the COMSOL client and the server. You must also specify a matching user name and password in the **Connect to Server** dialog box. The software writes this login information in the subdirectory **.comsol/v43b/login.properties** in your Windows home directory.

Client/Server Security Issues

COMSOL Multiphysics can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.

 Always make sure that untrusted users cannot access the COMSOL login information. Protect the file `.comsol/v43b/login.properties` in your home directory. This is important when running COMSOL Multiphysics in client/server mode. Alternatively, start the COMSOL server with the `-passwd nostore` option, and clear **Remember Password** when connecting to the server. This ensures that your login information is not stored on file.

Once a COMSOL server is started, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command.

TABLE 21-7: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
<code>-port <port></code>	Specify a TCP/IP port to connect to
<code>-server <server name></code>	Specify server to connect to
<code>-open <file></code>	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. Run both Model MPH-files and Model Java-files with the COMSOL batch command. Model Java-files need to be compiled before running.

The Windows syntax for the COMSOL batch command is

```
comsolbatch [<options>] [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-8: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH TARGET ARGUMENTS	DESCRIPTION
-inputfile <file name>	Run a Model MPH-file or class file
-outputfile <file name>	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output
-job <job tag>	The batch job to run
-study <study tag>	The study to compute
-pname <parameter name>	Comma-separated list of parameter names
-plist <parameter value>	Comma-separated list of parameter values
-batchlog <file name>	File to store log in
-client	Run as client
-host	Connect to host
-port	Connect to port
-graphics	Start COMSOL batch with graphics libraries. This displays plots during analysis.
-nosave	Do not save the resulting model

Example

To use the COMSOL Batch mode to solve a model, run the following command:

```
comsolbatch -inputfile in.mph -outputfile out.mph -job b3 -pname v -plist 10
```

This command starts COMSOL Batch, solves the model in the Model MPH-file with the given file name using the active solver settings in the model, and stores the solution in the out.mph.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a Model Java-file for use by the COMSOL batch command or for loading class files into the GUI. The Windows syntax for the COMSOL compile command is

```
comsolcompile [<options>] [<target arguments>] <file>.java
```

The Java file is mandatory. The following optional target arguments are available⁴

TABLE 21-9: COMSOL CLUSTER TARGET ARGUMENTS

COMSOL COMPILE TARGET ARGUMENTS	DESCRIPTION
-jdkroot <path>	Path to the JDK root
-classpathadd <classpath>	Additional classpath
-verbose	Verbose output

COMSOL CLUSTER COMMANDS

All COMSOL cluster commands require a floating network license.

The Windows syntax for the COMSOL cluster command is

```
mpiexec -n <no. of nodes> <command> [<options>] [<target arguments>]
```

The following cluster commands are available:

TABLE 21-10: COMSOL CLUSTER TARGETS

COMSOL CLUSTER COMMANDS	DESCRIPTION
comsolclusterbatch	Run COMSOL on a cluster in batch mode
comsolclusterserver	Run COMSOL server on a cluster
comsolcluster	Run COMSOL Desktop on a cluster

The preferred way of starting COMSOL jobs is from the Job Configurations node in the COMSOL Desktop.



Micromixer—Cluster Version: Model Library path **COMSOL_Multiphysics/Tutorial_Models/micromixer_cluster**

If you need to start COMSOL cluster jobs from the command line, the preferred way is to use the `comsolclusterbatch` command since the `comsolclusterserver` and `comsolcluster` commands require TCP/IP access from your client computer to the cluster node where COMSOL runs.

The Windows Configuration

- Make sure that Windows HPC Server 2008 or Windows Compute Cluster Server 2003 is installed. Running distributed COMSOL on other Windows versions is not supported.
- Make sure that the Windows HPC Server 2008 working directory is set to point to the `comsol` command directory (`<path to COMSOL install directory>\bin\win64`). The install directory must be shared between the nodes on your cluster. In some network configurations the firewall prevents you from starting MPI on a shared executable. To register the executable with the firewall use the `clusrun` command to execute the `hpcfutil` command on all nodes, for instance to register `comsolclusterbatch` use `clusrun /all hpcfutil register comsolclusterbatch.exe <shared path to COMSOL install directory>\bin\win64\comsolclusterbatch.exe`
- Also make sure that the Microsoft Visual Studio 2010 and 2008 Runtimes are installed on all nodes. They are called `vcredist_*.exe`. You can install them from the root directory of the DVD using the `clusrun` command for instance.
- Also make sure that all nodes that you intend to run COMSOL on have access to the license manager and that you can start COMSOL running in nondistributed mode. The nodes require access to the license manager in order to check out licenses.

Example of the COMSOL Batch Command

Schedule a job with the command

```
mpiexec -n -1 comsolclusterbatch.exe -np 2 -inputfile <filename>
```

to run a COMSOL batch on a number of computational nodes given by `mpiexec`. For further information about the `mpiexec` command and Windows HPC Server 2008, consult the documentation that was shipped with the product and the online manuals.

Example of the COMSOL Server Command

When a COMSOL server cluster job is created, a preference directory must be set and be reachable from all nodes to avoid problems with the server login; see [The COMSOL Commands](#) and [Login Information](#). The preferences can be generated by starting COMSOL server once on the head node using the command

```
comsolserver.exe -prefsdir <prefsdir>
```

where `<prefsdir>` is a preference directory *common* to all nodes.

When the COMSOL server is started on the cluster, the port number is written to standard output, so a standard output file and a standard error file must be set for the cluster job. To start a COMSOL server, schedule a job with the following command:

```
mpiexec -n -1 comsolclusterserver.exe -np 2 -prefsdir <prefsdir>
```

The argument **-1** indicates that the number of computational nodes is decided at the **mpiexec** launch. You must be able to access the cluster node where the COMSOL server runs from the COMSOL client computer.

COMSOL MPI Options

The COMSOL cluster target arguments specify what MPI library to use and what Scalapack version to use. There are several implementations of MPI. COMSOL by default uses the Windows HPC Server 2008 or Windows CCS 2003 MPI libraries. COMSOL also supports most MPI implementations based on MPICH2. It is recommended that the default library is used. COMSOL also has a compatibility mode that is activated by adding the option **-mpi mpich2**. When using this option both the variables **PATH** and **LD_LIBRARY_PATH** must include the MPI implementation. It is also possible to use other MPI libraries based on MPICH2 using the option **-mpipath <path to shared library>**. The following target arguments are available for a COMSOL cluster commands.

TABLE 21-11: COMSOL CLUSTER TARGET ARGUMENTS

COMSOL BATCH TARGET ARGUMENTS	DESCRIPTION
-mpi {auto} mpich2 wccs2003 whpc2008 user path	MPI library to use
-mpipath <path>	MPI library path
-scalapack {auto} mpich2 wccs2003 whpc2008 user path	Scalapack library to use
-scalapackpath <path>	Scalapack library path

The Cluster Computing study allows you to set up a batch job for submission to a Windows HPC Server 2008 job scheduler or Windows Compute Cluster Server 2003 job scheduler. There are several settings that you can configure in the **comsol.ini** file to get default settings:

```
-Dcs.scheduler=<IP or network address>
-Dcs.clusteruser=<Username on cluster>
-Dcs.rundir=<Where the model file is located on the cluster>
-Dcs.comsoldir=<Installation path to comsol on the cluster>
```

Additionally you can configure the commands:

```
-Dcs.precmd=<Command line>
```

```
-Dcs.postcmd=<Command line>
```

This adds commands prior to the `comsol` command and after the `comsol` command. You can add {nn} or {perhost} to any of these pre- or postcommands. This configures the Cluster Computing study to use the number of nodes and number of nodes on each host from the corresponding settings for the Cluster Computing study. For more information see the documentation for the Cluster Computing study.

COMSOL MATLAB COMMAND

Use the COMSOL `matlab` command to access the COMSOL API through MATLAB. enter:

```
comsolserver matlab
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsolserver matlab` command:

TABLE 21-12: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
<code>-mlroot <path></code>	MATLAB installation directory
<code>-host <hostname></code>	Connect to host
<code>-port <hostname></code>	Connect to port
<code>-desktop</code>	Start with Desktop
<code>-nodesktop</code>	Start without Desktop
<code>-mlnosplash</code>	Start without MATLAB splash screen
<code>-graphics</code>	Start the server with graphics libraries. This enables plotting on the server. Available only when running <code>comsolserver matlab [<options>]</code> .

COMSOL Commands on Linux

Use the `comsol` command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL command is

```
comsol [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. The `comsol` command can be combined with optional targets to achieve various results. The table below lists the command and targets.:.

TABLE 21-13: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
<code>comsol</code>	Run standalone COMSOL Multiphysics	
<code>comsol server</code>	Start COMSOL Multiphysics server	
<code>comsol client</code>	Run COMSOL Multiphysics client	
<code>comsol batch</code>	Run a COMSOL MPH-file or class file	
<code>comsol compile</code>	Compile a Model Java-file	
<code>comsol server matlab</code>	Start MATLAB® and connect to a COMSOL server	Requires LiveLink™ for MATLAB® license
<code>comsol convertpre35a</code>	Convert 3.0–3.5 models	
<code>comsol mpd</code>	Run the COMSOL multiprocessing daemon	Requires CLUSTERNODE license

The `comsol` command is located in the `bin` folder in the COMSOL installation directory.

INI FILES

There is a number of `.ini` files in the subdirectories `glnx86` and `glnxa64` in the `bin` directory. It is sometimes recommended that you edit these files. For example, you can add options to any of the above commands by modifying the corresponding `ini` file. To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsol batch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the `comsol` command and target. Table 21-14 lists the options (See [*<options>*] in the command syntax) available for all `comsol` commands. Always issue these options between the command and the target (if any).

TABLE 21-14: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
<code>-h</code>	Print general help	
<code><target> -h</code>	Print target-specific help	
<code>-32</code>	Run 32-bit COMSOL	
<code>-64</code>	Run 64-bit COMSOL	
<code>-3drend ogl sw</code>	3D renderer: OpenGL or software rendering	
<code>-comsolinifile</code>	Specify custom path to .ini-file used when starting COMSOL	
<code>-docroot <path></code>	Specify custom path to the COMSOL documentation root directory.	See Documentation and Model Library Root Directories
<code>-modelsroot <path></code>	Specify custom path to the COMSOL Model Library root directory.	See Documentation and Model Library Root Directories
<code>-np <no. of processors></code>	Number of processors	See Shared-Memory Options
<code>-numasets <no. of sets></code>	Number of NUMA sets	See Shared-Memory Options
<code>-mpmode throughput turnaround owner</code>	Multiprocessor mode	See Shared-Memory Options
<code>-blas {auto} mkl acml path</code>	BLAS library to use	See BLAS Options
<code>-blaspath <path></code>	BLAS library path	See BLAS Options
<code>-ipv6</code>	Activate IPv6 support	
<code>-nn <no. of nodes></code>	Number of nodes	See COMSOL Cluster Commands
<code>-nnhost <no. of nodes></code>	Number of nodes on each host	See COMSOL Cluster Commands
<code>-f <path></code>	Path to hostfile	See COMSOL Cluster Commands

TABLE 21-14: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
-mpi {auto} intel mpich2 wccs2003 whpc2008 user path	MPI library to use	See COMSOL Cluster Commands
-mpipath <path>	MPI library path	See COMSOL Cluster Commands
-mpiroot <path>	MPI library root path	See COMSOL Cluster Commands
-mpirsh {rsh} ssh	Use rsh or ssh when booting MPD	See COMSOL Cluster Commands
-mpibootstrap {ssh} rsh fork slurm ll lsf sge jmi	Set bootstrap server for Hydra	See COMSOL Cluster Commands
-mpibootstrapexec <path>	Executable used by bootstrap server	See COMSOL Cluster Commands
-mpidebug <debug level>	Set the MPI output level	See COMSOL Cluster Commands
-mpienablex	Enable Xlib forwarding	See COMSOL Cluster Commands
-mpifabrics fabric1:fabric2	Select network fabrics where fabric1 is one of <shm dapl tcp tmi ofa> and fabric2 is one of <dapl tcp tmi ofa>	See COMSOL Cluster Commands
-mpd	Use MPD instead of Hydra launcher	See COMSOL Cluster Commands
-scalapack {auto} mpich2 wccs2003 whpc2008 user path	Scalapack library to use	See COMSOL Cluster Commands
-scalapackpath <path>	Scalapack library path	See COMSOL Cluster Commands
-clustersimple	Simple startup of cluster	See COMSOL Cluster Commands
-c <path>	License file path	
-prefsdir <path>	Preference directory	
-tmpdir <path>	Temporary file directory	
-version	Print COMSOL version	
-version <target>	Print target version	

TABLE 21-14: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
-ck1	Use class-kit license	
-forcegcc	Force load of GCC libraries	
-forcecomsolgcc	Force load of GCC libraries shipped with COMSOL	
-autosave <{on} off>	Control saving of recovery files	
-recoverydir <path>	Path to recovery directories	

For the `-tmpdir` option, COMSOL software uses the specified directory to store temporary files. The `-prefsdir` option specifies the directory where COMSOL should store the preference file.

Documentation and Model Library Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Library root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Library root directories can be useful for administering Model Library Update; see [The Model Library](#) and [Preferences for Updates](#).

Shared-Memory Options

Use the option `-np` to control the number of cores and processors used. The default is to use all available cores and processors.

Use the option `-numasets` to control the number of Non-Uniform Memory Access node sets COMSOL should take into account. This is usually the number of processor sockets that the hardware is using.

Depending on how loaded your machine is, you can control how COMSOL uses the available processors. The following options are available:

TABLE 21-15: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
throughput	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
turnaround	Typically provides the best performance when no other processes than COMSOL are active.
owner	Provides the highest performance in most cases.

You may need to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides for the following BLAS related options:

TABLE 21-16: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL for Intel processors, and ACML for AMD processors with SSE2 support, otherwise MKL. (This is the default option.)
mkl	Use the Intel MKL library
acml	Use the AMD ACML library
path	Use a BLAS library specified using the option <code>-blaspath</code> or the environment variable <code>COMSOL_BLAS_PATH</code> . The library must support the standard FORTRAN BLAS interface.

Both MKL and ACML are distributed along with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in `/lib/ARCH` where `ARCH` is the architecture (`glnx86` or `glnxa64`) or somewhere in the standard search path. You must also provide the path to any sublibraries needed by the library. You can also set the search path to point to the directory where the library is installed. To do so, use the environment variable `LD_LIBRARY_PATH`.

GCC Options

By default COMSOL uses the GCC libraries installed on the system. If COMSOL is unable to start, COMSOL attempts to use the GCC libraries shipped with COMSOL. To force COMSOL to use the shipped GCC libraries, use the `-forcecomsolgcc`

option. The option `-forcegcc` is mainly intended for use together with the LiveLink™ interface for MATLAB; use it if you are unable to make function call backs to MATLAB.

COMSOL COMMANDS

In addition to the options in [Table 21-14](#), the standalone COMSOL command supports the following option.

TABLE 21-17: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
<code>-open <file></code>	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ *for* MATLAB® also needs to connect to a COMSOL server.

The syntax for the COMSOL server command is

```
comsol [<options>] server [<target arguments>]
```

The following target arguments are available for a COMSOL server command.

TABLE 21-18: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTIONS	DESCRIPTION
<code>-user <user></code>	Specify login name for a user
<code>-port <port></code>	Specify a TCP/IP port to listen for connect attempts.
<code>-passwd reset nostore</code>	Specify that you want to provide a new password. To avoid storing the new password on file use <code><nostore></code>
<code>-login {info} force never auto</code>	Ask for login information. <code>info</code> means that only missing information is asked for. <code>force</code> resets the password. <code>never</code> requires that the login information is available. <code>auto</code> automatically creates a new username and password
<code>-multi on {off}</code>	Accept repeated client connections

TABLE 21-18: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTIONS	DESCRIPTION
-silent	Do not listen to standard input
-graphics	Start the server with graphics libraries. This displays plots on the server when you are connected with a client (that is, not with the COMSOL GUI).

Accessing the COMSOL Server Computer

To access the computer running the COMSOL server simply log in on the server computer by using `ssh` or a similar command, then enter the `comsol server` command.

Login Information

When you start a COMSOL server for the first time, you are asked for a user name and password. Select a user name and a password, which COMSOL then uses in communications between the COMSOL client and the server. You must also specify a matching user name and password in the **Connect to Server** dialog box. The software writes this login information in the subdirectory `.comsol/v43b/login.properties` in your home directory.

Client/Server Security Issues

COMSOL Multipysics can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.



Always make sure that untrusted users cannot access the COMSOL login information. Protect the file `.comsol/v43b/login.properties` in your home directory. This is important when using COMSOL Multipysics' client/server feature. Alternatively, start the COMSOL server with the `-passwd nostore` option, and clear **Remember Password** when connecting to the server. This ensures that your login information is not stored on file.

Once you start a COMSOL server, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with a the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command.

TABLE 21-19: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
-port <port>	Specify a TCP/IP port to connect to
-server <server name>	Specify server to connect to
-open <file>	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. You can run both Model MPH files and Model Java-files with the COMSOL batch command. Model Java-files need to be compiled before running.

The syntax for the COMSOL batch command is

```
comsol [<options>] batch [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-20: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
-inputfile <file name>	Run a Model MPH-file or class file
-outputfile <file name>	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output
-job <job tag>	The batch job to run
-study <study tag>	The study to compute

TABLE 21-20: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
-pname <parameter name>	Comma separated list of parameter names
-plist <parameter value>	Comma separated list of parameter values
-batchlog <file name>	File to store log in
-client	Run as client
-host	Connect to host
-port	Connect to port
-graphics	Start COMSOL batch with graphics libraries. This displays plots during postprocessing.
-nosave	Do not save the resulting model

Example

To use the COMSOL Batch mode to solve a model, run the following command:

```
comsol batch -inputfile in.mph -outputfile out.mph
```

This command starts COMSOL Batch, solves the model in the Model MPH-file with the given file name using the active solver settings in the model, and stores the solution in the `out.mph`.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a Model Java-file for use by the COMSOL batch command or for loading class files into the GUI. The syntax for the COMSOL compile command is

```
comsol [<options>] compile [<target arguments>] <file>.java
```

The Java® file is mandatory. The following optional target arguments are available⁴:

TABLE 21-21: COMSOL COMPILE OPTIONS

COMSOL COMPILE OPTIONS	DESCRIPTION
-jdkroot <path>	Path to the JDK root
-classpathadd <classpath>	Additional classpath
-verbose	Verbose output

COMSOL CLUSTER COMMANDS

Use the COMSOL command with the option `-nn <no. of nodes>` to run COMSOL on clusters.

The syntax for the COMSOL cluster command is

```
comsol -nn <no. of nodes> [<options>] [<target>] [<target arguments>]
```

The following cluster commands are available:

TABLE 21-22: COMSOL CLUSTER TARGETS

COMSOL CLUSTER COMMANDS	DESCRIPTION
comsol -nn <nn> batch	Run COMSOL on a cluster in batch mode
comsol -nn <nn> server	Run COMSOL server on a cluster
comsol -nn <nn>	Run COMSOL Desktop on a cluster

The preferred way of starting COMSOL cluster jobs is from the Study node in the COMSOL Desktop. If you need to start COMSOL cluster jobs from the command line, the preferred way is to use the `comsol -nn <nn> batch` command since the `comsol -nn <nn> server` and `comsol -nn <nn>` commands require TCP/IP access from your client computer to the cluster node where COMSOL runs.

Running on Linux

COMSOL uses Hydra by default to initialize the MPI environment. Hydra is more scalable than MPD and it does not require any additional commands to launch.



Troubleshooting Distributed COMSOL and MPI

To launch COMSOL with Hydra use the command line

```
comsol -nn <number of compute nodes> -f <filename>
```

The file `<filename>` should contain the host names of the compute nodes that you intend to use. You can find out the hostname of each node from the Linux command `hostname`. Each node should be listed on a separate line in the file. You can also list the IP address of each node. The file may contain more compute nodes than you actually intend to use. You can set the remote node access mechanism that is used for connecting using the switch `-mpibootstrap`. The valid options are `ssh`, `rsh`, `fork`, `slurm`, `ll`, `lsf`, `sge` and `jmi`. This is important if the cluster only supports a different remote node access mechanism than `ssh` because `ssh` is the default protocol used. Use the switch `-mpibootstrapexec` to set the path to the remote node access mechanism such as `/usr/bin/ssh`. The option `-mpidebug` sets the output level from MPI. The default is level 4. You can control the network fabrics used for communication with the option `-mpifabrics fabric1:fabric2` where `fabric1` is one of `shm`, `dapl`, `tcp`, `tmi`, or `ofa`, and `fabric2` is one of `dapl`, `tcp`, `tmi`, or `ofa`. Use this option if you are

having trouble with the default fabrics used. Use `-mpienablex` to enable Xlib forwarding. Xlib forwarding is off by default.

Previously there was a shorthand for performing the COMSOL MPI environment initialization and starting COMSOL. The `-clustersimple` option is still supported but is equivalent to the Hydra command by default, for example

```
comsol -nn 4 -clustersimple
```

You should usually run COMSOL in batch mode. Use the command

```
comsol -clustersimple batch -inputfile input.mph -outputfile  
output.mph
```

It allows the Intel MPI library to automatically detect the number of nodes that were scheduled to the program. Restricting the number of processes with the `-nn` switch allows COMSOL to combine MPI with multithreading. This is the most efficient way to run COMSOL.

Using the MPD launcher

In previous versions of COMSOL, the MPI environment was launched by MPD. You can still use MPD if you use the switch `-mpd` but it is recommended that you use Hydra.

In order to start MPI, have a file named `.mpd.conf` in your home directory to which you alone have access. This file should contain the single line

```
secretword = <your secret word here>
```

On Intel MPI shipped with COMSOL, the `.mpd.conf` file is optional.

Below, the details of the individual cluster commands are described.

Before you start COMSOL, you must initialize the MPI environment. A so called multiprocessing daemon (MPD) must run on each computer node that you intend to use. To start MPD on several computer nodes, enter

```
comsol -nn <number of compute nodes> mpd boot -f <filename>
```

The file `<filename>` should contain the host names of the compute nodes that you intend to use. You can find out the hostname of each node from the Linux command `hostname`. Each node should be listed on a separate line in the file. You can also list the IP address of each node. The file may contain more compute nodes than you actually intend to use. As an alternative to using the `-f <filename>` option, you can put the list of compute nodes in a file named `mpd.hosts` in your home directory. You can set the protocol that is used for connecting using the switch `-mpirsh`. The valid

options are `rsh` and `ssh`. This is important if the cluster only supports `ssh` because `rsh` is the default protocol used. Make sure that all nodes were booted by listing them with the command

```
comsol mpd trace
```

Start distributed COMSOL with the `-nn` option. For example, enter

```
comsol -nn <number of computational nodes> -mpd server
```

to start a COMSOL server running on a specific number of computational nodes. The number of computational nodes can exceed the number of compute nodes. Use the `-nnhost` option if you want to force several computational nodes on a compute node. Avoid starting more computational nodes than the total number of cores that you have available on a compute node. When you have finished using distributed COMSOL, you should take down all the MPDs. Enter the command

```
comsol mpd allexit
```

to stop all MPDs. To obtain more information about the `comsol mpd` commands, add the `-h` option to the commands, for example, `comsol mpd boot -h`.

Start MPD on a single computer with the command

```
comsol mpd mpd &
```

This is useful when running all computational nodes on a single multiprocessor computer or when you have difficulties attaching computational nodes because of firewalls. In the second case you can start an MPD on each node and attach them by specifying the main port and host. Use

```
comsol mpd trace -1
```

and

```
comsol mpd mpd --port <the port number reported> --host <the  
hostname reported>
```

You can also start COMSOL with the `-clustersimple` option. This option automatically starts and terminates the MPD daemon. It uses the `mpd.hosts` file in your home directory to determine what computational nodes to use.

Starting Distributed COMSOL—Linux Examples

Make sure that COMSOL is able to start on all nodes where you intend to run COMSOL.



Each node requires access to the license manager. If the node is unable to check out a license, it aborts the startup process.

A simplified version is used when the `-clustersimple` switch is set or the Hydra launcher is used. An example follows. Start four computational nodes on hosts listed in the file `hosts` using distributed COMSOL and simplified start:

```
comsol -nn 4 -clustersimple -f hosts alternatively
comsol -nn 4 -clustersimple -f hosts server alternatively
comsol -nn 4 -clustersimple -f hosts batch -inputfile in.mph -outputfile
out.mph
```

where `-clustersimple` is optional for Hydra.

If you use MPD. Start an MPD on a single computer. Then start distributed COMSOL on two computational nodes (on the same host) each using three processors, and finally stop the MPD:

```
comsol mpd mpd &
comsol -nn 2 -np 3 -mpd alternatively
comsol -nn 2 -np 3 -mpd server alternatively
comsol -nn 2 -np 3 -mpd batch -inputfile in.mph -outputfile out.mph
comsol mpd allexit
```

The example above could be used if you have a very small model with a very large amount of parametric steps, where using `mpd` on a single computer might be beneficial.

Start three MPDs on the compute nodes with hostnames defined in the file `myhosts`. Each line in the file should specify the host address or IP-address of a node. Make sure the MPDs were correctly booted. Then start a distributed COMSOL server on three computational nodes, and finally stop the MPDs. First make sure that you can connect to all the computers with `ssh` without having to use your password (see the man pages for `ssh`). Also make sure that all computers have access to the same COMSOL installation and that they are using the same Linux version. There are two options for starting a session. One more detailed and one shorthand version. An example of the detailed version:

```
comsol -nn 4 mpd boot -f myhosts
comsol mpd trace
comsol -nn 4 -mpd alternatively
```

```

comsol -nn 4 -mpd server alternatively
comsol -nn 4 -mpd batch -inputfile in.mph -outputfile out.mph
comsol mpd allexit

```

MPI Options

There are several implementations of MPI. COMSOL is shipped with the Intel MPI library but should also support most MPI implementations based on MPICH2. It is recommended that you use the default Intel MPI library. For running COMSOL on a computer that has MPICH2 installed, COMSOL also has a compatibility mode that you can activate by adding the option `-mpi mpich2`. When using this option both the variables `PATH` and `LD_LIBRARY_PATH` must include your MPI implementation. It is also possible to use other MPI libraries based on MPICH2 using the option `-mpipath <path to shared library>` and `-mpiroot <path to root of mpi library installation>`. [Table 21-14](#) lists the MPI related options, `-mpi`, `-mpipath`, `-scalapack`, and `-scalapackpath`. Additionally the comsol MPI arguments are configurable inside the comsol start script. To configure comsol to work with a job scheduler through the Cluster Computing study you can set the options

```

-Dcs.precmd=<Command line>
-Dcs.postcmd=<Command line>

```

in the `comsol.ini` file. This adds commands prior to the `comsol` command and after the `comsol` command. You can add `{nn}` or `{perhost}` to any of these pre- or postcommands. This configures the Cluster Computing study to use the number of nodes and number of nodes on each host from the corresponding settings for the Cluster Computing study. For more information see the documentation for the Cluster Computing study.

Troubleshooting Distributed COMSOL and MPI

The Hydra launcher is the main MPI environment. The syntax for Hydra commands is

```
comsol [<options>] hydra [<Hydra command>] [<target arguments>]
```

TABLE 21-23: COMSOL HYDRA COMMANDS

COMSOL MPD COMMANDS	DESCRIPTION
<code>cleanup</code>	Run mpicleanup command
<code>mpitest</code>	Run a distributed test program
<code>tune</code>	Run mpitune command

Use the `-h` switch for more information about each command.

COMSOL ships with the Intel MPI library but should be compatible with most MPICH2 compatible MPI libraries. To download the latest version of Intel MPI

library runtime visit <http://software.intel.com/en-us/intel-mpi-library>. To run comsol with another version of Intel MPI or other MPI library set `-mpiroot` to the root path of the MPI library and. In case the downloaded library is not compatible with the version Comsol uses, this should usually not be the case, also set `-mpipath` to the dynamically loaded library that should be used. The default of the Intel MPI library is to use `ssh` as communication protocol. If you require another communication protocol use the option `-mpibootstrap <protocol>`. If you are using a scheduler the Intel MPI library is often able to detect the environments it is running from using the `-clustersimple` switch and you do not need to set up a hosts file. If you are using a PBS or Torque scheduler add `-mpiarg -rmk -mpiarg pbs` to the command line in order for Intel MPI to interpret the environment correctly. The Intel MPI library automatically tries to detect the best option for communication and uses InfiniBand if it detects it. To verify that COMSOL is using infiniband check the output from the startup of COMSOL, it should not mention tcp transfer mode. If you have problems running on a Myrinet network add the options `-mpiarg -mx` to the command line. If you have problems running on a Qlogic network, add the options `-mpiarg -psm` to the command line. In some cases it helps if you combine the option with the environment variable `PSM_SHAREDCONTEXTS_MAX` set to 1. You can control the fabrics used for communication with the option `-mpifabrics fabric1:fabric2` where `fabric1` is equal to `fabric2` or `fabric1` is `shm`.

If comsol aborts during start make sure that all nodes are able to access the license manager and that comsol can be started on each node when not running distributed. Sometimes there is additional information in the log files located in `$HOME/.comsol/v43b/configuration/comsol/*.log`. If this does not help start the mpitest program to make sure that the MPI library is working as it should using the command

```
comsol -nn <number of nodes> -f <host file> hydra mpitest
```

For more verbose information about the startup process when using Hydra, use `-mpiarg -verbose` or set `-mpidebug` to a value greater than the default 4.

The MPD daemon can be used in several ways to troubleshoot problems with the comsol MPI environment. The syntax for MPD commands is

```
comsol [<options>] mpd [<MPD command>] [<target arguments>]
```

TABLE 21-24: COMSOL MPD COMMANDS

COMSOL MPD COMMANDS	DESCRIPTION
boot	Run mpdboot command
mpd	Run mpd command

TABLE 21-24: COMSOL MPD COMMANDS

COMSOL MPD COMMANDS	DESCRIPTION
exit	Run mpdexit command
allexit	Run mpdallexit command
cleanup	Run mpdcleanup command
trace	Run mpdtrace command
check	Run mpdcheck command
ringtest	Run mpdringtest command
listjobs	Run mpdlistjobs command
sigjob	Run mpdsigjob command
killjobs	Run mpdkilljobs command
mpitest	Run a distributed test program
tune	Run mpdtune command
help	Run mpdhelp command

Use -h switch for more information about each command.

When using MPD, use `comsol mpd check` command to display important information. For more verbose information about the startup process from the MPD daemon use the -v and/or -d switches or set the environment variable or set `-mpidebug` to a value greater than the default 4. If the MPD is booted and comsol is not starting make sure that the MPI environment is working for instance by running the `comsol mpd mpitest` command.

COMSOL MATLAB COMMAND

Use the COMSOL `matlab` command to access the COMSOL API through MATLAB®. Enter:

```
comsol server matlab [<options>]
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsol server matlab` command:

TABLE 21-25: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
<code>-mlroot <path></code>	MATLAB installation directory
<code>-host <hostname></code>	Connect to host
<code>-port <hostname></code>	Connect to port

TABLE 21-25: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
-desktop	Start with Desktop
-nodesktop	Start without Desktop
-minosplash	Start without MATLAB splash screen
-graphics	Start the server with graphics libraries. This enables plotting on the server. Available only when running <code>comsol server matlab [<options>]</code> .

COMSOL Commands on Macintosh

Use the COMSOL command to start COMSOL products with detailed start-up options.

The general syntax of the COMSOL command is

```
comsol [<target>] [<options>] [<target arguments>]
```

where square brackets indicate optional arguments. The `comsol` command can be combined with optional targets to achieve various results. The table below lists the command and targets.

TABLE 21-26: COMSOL COMMANDS TARGETS

COMMAND AND TARGET	DESCRIPTION	AVAILABILITY
comsol	Run standalone COMSOL Multiphysics	
comsol server	Start COMSOL Multiphysics server	
comsol client	Run COMSOL Multiphysics client	
comsol batch	Run a COMSOL MPH file or class file	
comsol compile	Compile a Model Java file	
comsol server matlab	Start MATLAB and connect to a COMSOL server	Requires LiveLink™ for MATLAB® license
comsol convertpre35a	Convert 3.0–3.5 models	

The `comsol` command is located in the `bin` folder in the COMSOL installation directory.

INI FILES

There is a number of .ini files in the subdirectories `maci32` and `maci64` in the `bin` directory. It is sometimes recommended that you edit these files. For example, you can add options to any of the above commands by modifying the corresponding ini file. To change the option `opt` to value `val`, add the line

```
-Dopt=val
```

to the file `comsol.ini`. Change the file `comsolbatch.ini` for `comsol batch`, and similarly for the other COMSOL targets.

OPTIONS

You can enter various options after the COMSOL command and target. [Table 21-14](#) lists the options (See [`<options>`] in the command syntax) available for all comsol commands. Always issue these options between the command and the target (if any).

TABLE 21-27: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
<code>-h</code>	Print general help	
<code><target> -h</code>	Print target-specific help	
<code>-32</code>	Run 32-bit COMSOL	
<code>-64</code>	Run 64-bit COMSOL	
<code>-3drend ogl sw</code>	3D renderer: OpenGL or software rendering	
<code>-docroot <path></code>	Specify custom path to the COMSOL documentation root directory.	See Documentation and Model Library Root Directories
<code>-modelsroot <path></code>	Specify custom path to the COMSOL Model Library root directory.	See Documentation and Model Library Root Directories
<code>-np <no. of processors></code>	Number of processors	See Shared-Memory Options
<code>-mpmode throughput turnaround owner</code>	Multiprocessor mode	See Shared-Memory Options
<code>-blas {auto} mkl acml path</code>	BLAS library to use	See BLAS Options
<code>-blaspath <path></code>	BLAS library path	See BLAS Options
<code>-ipv6</code>	Activate IPv6 support	
<code>-c <path></code>	License file path	

TABLE 21-27: COMSOL OPTIONS (CURLY BRACKETS INDICATE DEFAULT VALUES)

COMSOL OPTION	DESCRIPTION	REFERENCE
-prefsdir <path>	Preference directory	
-tmpdir <path>	Temporary file directory	
-version	Print COMSOL version	
-version <target>	Print target version	
-ckl	Use classkit license	
-autosave <{on} off>	Control saving of recovery files	
-recoverydir <path>	Path to recovery directories	

For the `-tmpdir` option, COMSOL software uses the specified directory to store temporary files. The `-prefsdir` option specifies the directory where COMSOL should store the preference file.

Documentation and Model Library Root Directories

In a default COMSOL installation, the documentation files are located in the directory `doc` under the installation root directory. You can use the `-docroot` option if you want to move the documentation directory to a different location. Similarly, use the `-modelsroot` option if you want to move the Model Library root directory `models` from its default location under the COMSOL installation root. Relocating the documentation and Model Library root directories can be useful for administering Model Library Update; see [The Model Library](#) and [Preferences for Updates](#).

Shared-Memory Options

Use the option `-np` to control the number of core and processors used. The default is to use all available cores and processors.

Depending on how loaded your machine is, you can control how COMSOL uses the available processors. The following options are available:

TABLE 21-28: COMSOL MULTIPROCESSOR MODE OPTIONS

MPMODE OPTION	DESCRIPTION
throughput	Is expected to give the best performance when several different processes are running actively at the same time as COMSOL.
turnaround	Typically provides the best performance when no other processes than COMSOL are active.
owner	Provides the highest performance in most cases.

You may need to experiment to find the options that work best for your configuration.

BLAS Options

BLAS is a set of functions for basic linear algebra operations. A large portion of the computational engine in COMSOL relies on BLAS. COMSOL provides the following BLAS-related options:

TABLE 21-29: COMSOL BLAS OPTIONS

BLAS OPTION	DESCRIPTION
auto	Determine BLAS library automatically: MKL.
mkl	Use the Intel MKL library.
path	Use a BLAS library specified using the option -blaspath or the environment variable COMSOL_BLAS_PATH. The library must support the standard FORTRAN BLAS interface.

MKL is distributed along with COMSOL.

If you want to use a different BLAS library than the default, make sure that COMSOL can find the library. The simplest way for COMSOL to find a library is to put it in /lib/ARCH where ARCH is the architecture (maci32 or maci64) or somewhere in the standard search path. You must also provide the path to any sublibraries needed by the library. You can also set the search path to point to the directory where the library is installed. To do so, use the environment variable DYLD_LIBRARY_PATH.

COMSOL COMMANDS

In addition to the options in [Table 21-14](#), the standalone COMSOL command supports the following option.

TABLE 21-30: COMSOL COMMAND-LINE ARGUMENTS

COMSOL OPTIONS	DESCRIPTION
-open <file>	Open file

COMSOL SERVER COMMANDS

Use a COMSOL server command to start a COMSOL process ready to process computational requests. A COMSOL server listens for TCP/IP connections from COMSOL clients. A COMSOL Desktop can become a COMSOL client by connecting to a COMSOL server. The LiveLink™ interface for MATLAB also needs to connect to a COMSOL server.

The syntax for the COMSOL server command is

```
comsol [<options>] server [<target arguments>]
```

The following target arguments are available for a COMSOL server command.

TABLE 21-31: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL SERVER OPTIONS	DESCRIPTION
-user <user>	Specify login name for a user
-port <port>	Specify a TCP/IP port to listen for connect attempts.
-passwd reset nostore	Specify that you want to provide a new password. To avoid storing the new password on file use <nostore>
-login {info} force never auto	Ask for login information. info means that only missing information is asked for. force resets the password. never requires that the login information is available. auto automatically creates a new username and password
-multi on {off}	Accept repeated client connections
-silent	Do not listen to standard input

Accessing the COMSOL Server Computer

To access the computer running the COMSOL server simply log in on the server computer by using `ssh` or a similar command, then enter the `comsol server` command.

Login Information

When you start a COMSOL server for the first time, you are asked for a user name and password. Select a user name and a password, which COMSOL then uses in communications between the COMSOL client and the server. You must also specify a matching user name and password on the settings page in the **Model Navigator**, which opens when you start the COMSOL client. The software writes this login information in the file `login.properties`. The login information is located in `Library/Preferences/COMSOL/v43b/login.properties` in your home directory.

Client/Server Security Issues

COMSOL can operate in a client/server mode where COMSOL runs as a separate client and a server. COMSOL uses a TCP/IP connection to send data between the server and the client.

	Always make sure that untrusted users cannot access the COMSOL login information. Protect the file <code>Library/Preferences/COMSOL/v43b/login.properties</code> . This is important when using COMSOL's client/server feature. Alternatively, start the COMSOL server with the <code>-passwd nostore</code> option, and clear Remember Password when connecting to the server. This ensures that your login information is not stored on file.
---	--

Once you start a COMSOL server, a person with access to your login information could potentially connect to your COMSOL server. When a COMSOL client connects or disconnects from a remote computer, the COMSOL server displays a message. The connection from the client to the server is made with the TCP protocol.

The server and client are mutually authenticated using a challenge handshake authentication protocol, which means that login information cannot be easily obtained by someone eavesdropping on the network communication. The TCP connection between the client and the server is otherwise not encrypted. If you require encryption of the TCP connection, you can use third-party software based on protocols such as SSH or IPSEC.

COMSOL CLIENT COMMANDS

Use a COMSOL client command to start a COMSOL Desktop with a the Connect to Server dialog box open.

The syntax for the COMSOL client command is

```
comsol [<options>] client [<target arguments>]
```

The following target arguments are available for a COMSOL client command.

TABLE 21-32: COMSOL TARGET COMMAND-LINE ARGUMENTS

COMSOL CLIENT OPTIONS	DESCRIPTION
<code>-port <port></code>	Specify a TCP/IP port to connect to
<code>-server <server name></code>	Specify server to connect to
<code>-open <file></code>	Open file

COMSOL BATCH COMMANDS

Use the COMSOL batch command to run COMSOL jobs without a GUI. You can run both Model MPH files and Model Java files with the COMSOL batch command. Model Java files need to be compiled before running.

The syntax for the COMSOL batch command is

```
comsol [<options>] batch [<target arguments>]
```

Its detailed target arguments are:

TABLE 21-33: COMSOL BATCH-SPECIFIC ARGUMENTS

COMSOL BATCH OPTIONS	DESCRIPTION
-inputfile <file name>	Run a Model MPH-file or class file
-outputfile <file name>	Save a Model MPH-file using the given file name. If output is not given, the input file is overwritten with the output
-job <job tag>	The batch job to run
-study <study tag>	The study to compute
-pname <parameter name>	Comma-separated list of parameter names
-plist <parameter value>	Comma-separated list of parameter values
-batchlog <file name>	File to store log in
-client	Run as client
-host	Connect to host
-port	Connect to port
-nosave	Do not save the resulting model

Example

To use the COMSOL Batch mode to solve a model, run the following command:

```
comsol batch -inputfile in.mph -outputfile out.mph -job b3 -pname  
v -plist 10
```

This command starts COMSOL Batch, solves the model in the Model MPH-file with the given file name using the active solver settings in the model, and stores the solution in the *out.mph*.

THE COMSOL COMPILE COMMAND

The COMSOL compile command compiles a model Java file for use by the COMSOL batch command or for loading class files into the GUI. The syntax for the COMSOL compile command is

```
comsol [<options>] compile [<target arguments>] <file>.java
```

The Java file is mandatory. The following optional target arguments are available^{*}

TABLE 21-34: COMSOL COMPILE OPTIONS

COMSOL COMPILE OPTIONS	DESCRIPTION
-jdkroot <path>	Path to the JDK root
-classpathadd <classpath>	Additional classpath
-verbose	Verbose output

COMSOL MATLAB COMMAND

Use the COMSOL MATLAB command to access the COMSOL API through MATLAB. Type:

```
comsol server matlab [<options>]
```

which launches a COMSOL server in a console window, starts MATLAB, and connects MATLAB to the COMSOL server.

The following options are available for the `comsol server matlab` command:

TABLE 21-35: COMSOL MATLAB OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
-mlroot <path>	MATLAB installation directory
-host <hostname>	Connect to host
-port <hostname>	Connect to port
-desktop	Start with Desktop
-nodesktop	Start without Desktop
-mlnosplash	Start without MATLAB splash screen

The COMSOL Convertpre35a Command

Use the `comsol convertpre35a` command to convert a directory with models made in COMSOL 3.0–3.5 to COMSOL 3.5a. To use the command, enter

```
comsol [<options>] <input directory> <output directory> [<logfile>]
```

where *<input directory>* is the input directory, *<output directory>* is the output directory, and [*<logfile>*] is an optional log file. If you do not provide the third argument, the log is printed on standard output.

TABLE 21-36: COMSOL CONVERTPRE35A OPTIONS

COMSOL MATLAB OPTIONS	DESCRIPTION
-c35aroot <path>	Installation path of COMSOL 3.5a

Running COMSOL on the Amazon™ Cloud (Amazon EC2™)

The Amazon Elastic Compute Cloud™, also known as the Amazon EC2™, allows you to dynamically allocate virtual computer resources at a small hourly fee. If you have access to a COMSOL *Floating Network License* (FNL), you can run one or several computations in the cloud. For this to work, the remote virtual computer in the cloud needs access to, and be able to check out, an FNL license key from an on-premise COMSOL license manager. This license manager should be running on a computer within your organization's network. In this way, the COMSOL session in the cloud is dialing back to your on-premise license manager.



The Floating Network License (FNL) option for COMSOL is required to run jobs on a remote computer or use cloud resources.

This tutorial explains how to run COMSOL on the Amazon Elastic Compute Cloud (EC2™). The step-by-step instruction assume that you work from a Windows environment. Linux and Macintosh instructions, where needed, are available separately at the end of this guide.

In this section:

- [Nomenclature](#)
- [Requirements](#)
- [Introduction](#)
- [Signing up for an EC2 Account and Logging In](#)
- [Launching and Using the Remote Instance](#)
- [Preparing the COMSOL License Management](#)
- [Installing COMSOL on the AMI](#)
- [Terminating the Instance](#)
- [Connecting a COMSOL Desktop GUI to a Remote Instance](#)
- [Connecting to an EC2 Instance from Linux and Macintosh without Java](#)
- [Copying a Custom COMSOL Version to the EC2 or other cloud services](#)

- [Creating a Cloud Cluster Using the StarCluster Utility on Linux](#)
- [Using the Remote Access Functionality in COMSOL using Linux](#)

Nomenclature

These terms are used in this guide.

- *EC2*, Amazon Elastic Compute Cloud.
- *AWSTM*, Amazon Web Services.
- *AMI*, An Amazon Machine Image is a special type of pre-configured operating system and virtual application software, which is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2). It serves as the basic unit of deployment for services delivered using EC2.

Requirements

The following requirements should be fulfilled for running COMSOL on the cloud.

- Access to a Floating Network License (FNL) of COMSOL Multiphysics 4.3a or later.
- Make sure your IT-department has configured your organization's firewall not to deny outbound traffic on port 22. This is required to access your Amazon virtual machine instance.
- For added security, the firewall in the cloud should be configured to only allow traffic to and from your on-site public ip address. You will do this yourself while setting up the could service, see [Figure 21-3](#).

Introduction

The following steps are needed to run COMSOL on EC2, and are detailed in this guide.

- 1 Sign up for an EC2 account.
- 2 Launch a remote instance (AMI).
- 3 Install COMSOL on the remote instance.
- 4 Run job.
- 5 Terminate instance.

The last step is important since there is a fee charged for active instances.



Step 3 is only needed the first time you run on EC2, since you can save the COMSOL installation between sessions, see further the section [Terminating the Instance](#) below.

[Signing up for an EC2 Account and Logging In](#)

Now create an account on Amazon.

- 1 Navigate to the Amazon web page <http://aws.amazon.com> and create an account. Then return to the account page to continue these steps.
- 2 Select the menu **My Account>AWS Management Console**. Then click **EC2** to get the EC2 Dashboard. you may also get there by selecting the **Services>EC2** menu. This is where you will control most of your cloud activity.

Figure 21-2: EC2 Console Dashboard.

The best way to get updated information about launching the AMI is to follow the guide located at <http://docs.amazonaws.com/AWSEC2/latest/GettingStartedGuide>. Also, a full set of PDF guides are located at <https://aws.amazon.com/documentation/ec2/>.



Only some of the key steps from the online guide are detailed in this guide.

 For a new user, Amazon provides micro instances free of charge (Free Usage Tier). These are useful for experimenting and verifying that everything works as expected.

SETUP AND PROVISIONING

Search for COMSOL on the AWS marketplace (aws.amazon.com/marketplace). Go to the COMSOL page, which describes the base operating system and AWS products required to use COMSOL Multiphysics on EC2. Prices are also shown for the various instance types available to use. It is also required that you read the COMSOL Multiphysics End User License Agreement (EULA) at this stage. This can be accessed at the bottom of the page.

LAUNCHING

- 1** Click **Continue**. The EC2 launcher page shows the versions of COMSOL Multiphysics included in this AMI along with the region, instance type, firewall settings and key pair. The default instance type is High-Memory 4XL (m2.4xlarge). This instance type provides a good environment to run large simulations and/or cluster sweeps. Other instance types can be chosen if they are more suited to your application. An estimate of the charges are shown assuming this instance will run 24x7 for 1 month.

 You can also start the COMSOL AMI directly by clicking Launch Instance from the EC2 Dashboard as shown in [Figure 21-2](#). Then select the AWS Marketplace Option and search for COMSOL.

- 2** An autogenerated security group is provided in the **Firewall Settings (Create new based on seller settings)**. This will open port 22 required to connect with SSH to

your instance once running. However, you can select any security group you have defined on the EC2 Dashboard under **Network&Security>Security Groups**.

- 3 You can select the key pair you would like to use to securely connect to your instance from the **Key Pair** section. If you do not have any key pairs defined in the current region, you can create them from within your EC2 Dashboard, under **Network&Security>Key Pairs**.

 Name, download and store the key in a safe place, for example in a protected working directory where it is named for example `johns_key.pem`. The key can then be reused for future instances. Both the Security Groups and Key Pairs are unique per region selected.

- 4 Click **Launch with 1-click**. In a few minutes your EC2 instance will be available for use.
- 5 For increased security, restrict access to only your public IP address. Find the **Security Group** under **Network & Security** in the EC2 Dashboard. Click your autogenerated security group and click the **Inbound** tab at the bottom. Select **Create a New Rule: SSH**. Enter your public IP address and click **Add Rule** and **Apply Rule Changes**.
-

 To determine your public IP address, search Google for *What is my IP-address?*

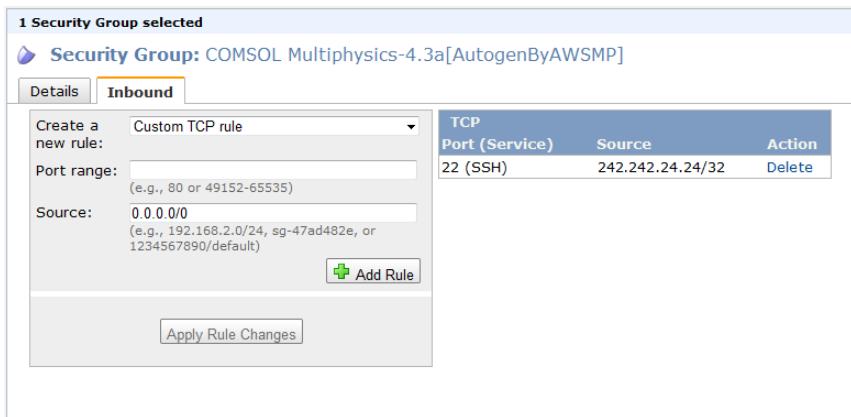


Figure 21-3: Security group after successfully restricting the IP address to 242.242.24.24.

To obtain the hostname of your new instance, open your *EC2 Console Dashboard*, select the new running instance and in the description section, the public hostname is given on the *Public DNS* line, for example

`ec2-75-101-201-244.compute-1.amazonaws.com`

The username for this instance will be `ec2-user`.

CONNECTING TO THE COMSOL EC2 INSTANCE

You can connect to your COMSOL EC2 instance via SSH, and transfer files via SCP. This section demonstrates how to do this in your web browser using the Java SSH Client. If you do not have Java® installed, you can, however, use any SSH and SCP client you like, for example WinSCP, PuTTY, or Linux built-in SSH clients (see further [Connecting to an EC2 Instance from Linux and Macintosh without Java](#)).

- | In the **Navigation** pane on EC2 Dashboard, click **Instances**.

2 Right-click your instance and click **Connect**.

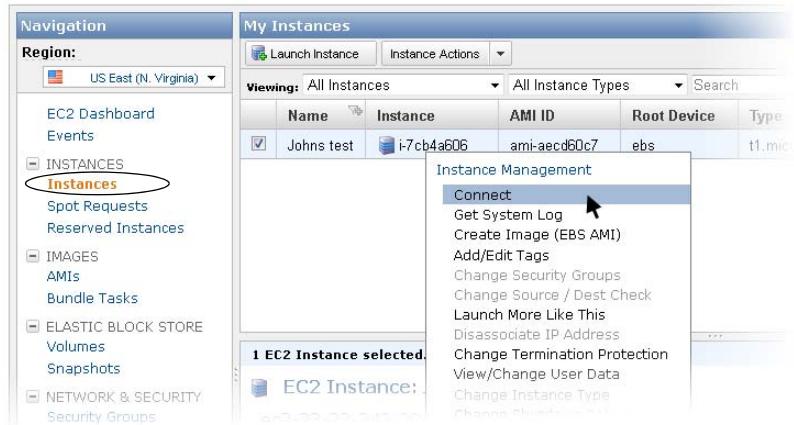
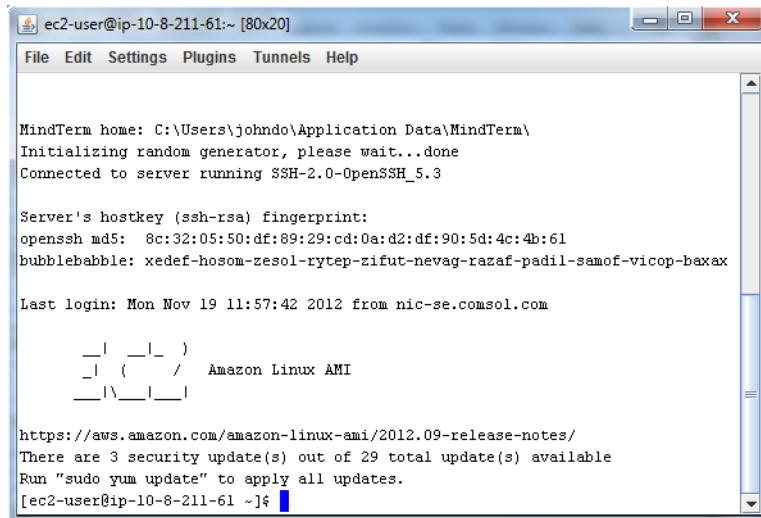


Figure 21-4: Find the instance in the dashboard. Right-click the instance and select “Connect”. A new window opens and you then select “Connect from your browser using the Java SSH Client (Java Required).”

3 Select **Connect from your browser using the Java SSH Client (Java Required)**.

- Fill in **User name:** ec2-user
- Fill in the private key path you defined in the [Launching](#) step.
- Click [Launch SSH client](#).
- Enter yes to any installation and security fingerprint questions.

You should now have a terminal (remote shell) up and running with access to the command prompt of the AMI.



The screenshot shows a Windows MindTerm terminal window titled "ec2-user@ip-10-8-211-61:~ [80x20]". The window contains the following text:

```
MindTerm home: C:\Users\johndo\Application Data\MindTerm\  
Initializing random generator, please wait...done  
Connected to server running SSH-2.0-OpenSSH_5.3  
  
Server's hostkey (ssh-rsa) fingerprint:  
openssh md5: 8c:32:05:50:df:89:29:cd:0a:d2:df:90:5d:4c:4b:61  
bubblebabble: xedef-hosom-zesol-rytep-zifut-nevag-razaf-padil-samof-vicop-baxax  
  
Last login: Mon Nov 19 11:57:42 2012 from nic-se.comsol.com  
  
_ _| _ _ )  
_ | ( _ / Amazon Linux AMI  
_ \_ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2012.09-release-notes/  
There are 3 security update(s) out of 29 total update(s) available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-8-211-61 ~]$
```

Figure 21-5: Remote shell.

Now that you are logged in to the remote shell, you will want to work privately. Create a private directory by entering the commands:

```
mkdir Private  
chmod 700 Private
```

You can also confirm the resources you have using this command:

```
df -h
```

TRANSFERRING FILES

- I Transfer a small .mph file for later use. In the remote shell window, select **Plugins>SCP file transfer**. In your local COMSOL installation directory, find the file `models/COMSOL_Multiphysics/Equation-Based_Models/point_source.mph` and transfer it to your remote `Private` directory by clicking the --> button.



Don't have a local COMSOL installation? Download CSOMOL from <http://www.comsol.com/support/download>.

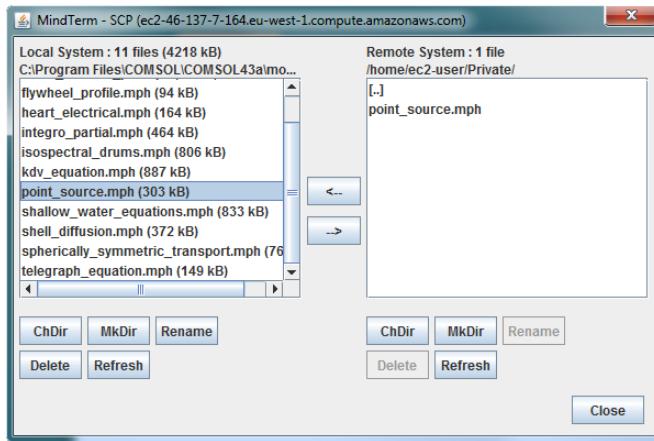


Figure 21-6: File transfer dialog box.

Preparing the COMSOL License Management

The COMSOL Multiphysics AMI is a “Bring Your Own License (BYOL)”. You need a valid COMSOL Multiphysics floating network license (FNL) to run the software. This license needs to be managed by a license manager running locally (on-premise). You will not be able use a license server on the EC2 instance.

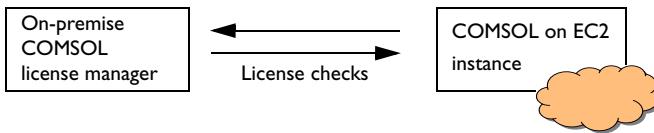


Figure 21-7: Principle for COMSOL License Management.

To maintain high security, you need to forward TCP ports over encrypted tunnels to your on-premise license manager. This procedure is explained in the sections below.

VERIFYING THE ON-PREMISE LICENSE SERVER

You need to make sure that your on-premise COMSOL license manager works before continuing.

Start the local COMSOL Desktop GUI. It should start without error messages. Check the menu **Options>Licenses** to verify that the license manager delivers all the add-on products you are licensed. If you have problems with this step, please contact your local COMSOL administrator.



Don't have a local COMSOL installation? Download COMSOL from <http://www.comsol.com/support/download>.

OPENING ENCRYPTED TUNNELS TO THE LICENSE MANAGER

You have already tested SSH and SCP to communicate with the remote instance. However, for the remote COMSOL installation to operate you need to also open up the appropriate encrypted tunnel ports back to your on-premise license manager from the EC2 instance.

- 1 Contact your local COMSOL license administrator to get the license file that was provided by your COMSOL sales representative.
- 2 The first few lines of the original license file should be similar to

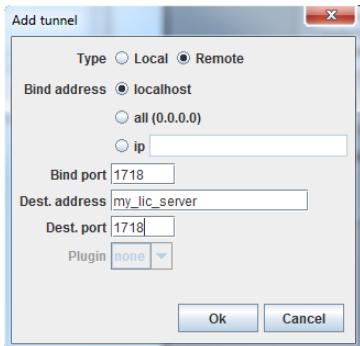
```
#-----
# COMSOL 4.3b FNL License No: 12345678
# Visit www.comsol.com/install for latest installation instructions
#-----
SERVER my_lic_server 012345678901 1718
USE_SERVER
VENDOR LMCOMSOL port=1719
FEATURE SERIAL...
```

Here, `my_lic_server` is the name of the on-premise COMSOL license server. You will need this name when opening the ports. The ports 1718 and 1719 are default port numbers used in all license files provided by COMSOL.

Now open the ports for the license management.

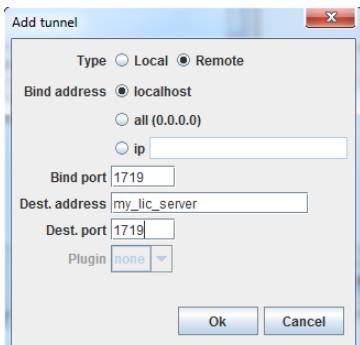
- 1 In the remote shell window, select the menu **Tunnels>Setup**, then **Add**
- 2 Enter:
 - Type: `Remote`
 - Bind address: `localhost`
 - Bind port: `1718`
 - Destination: `my_lic_server`
 - Dest. port: `1718`

where `my_lic_server` is the name or ip-address of your license server in the original `license.dat` file, see the section [Opening Encrypted Tunnels to the License Manager](#).



3 Click **OK**.

4 Repeat the steps above to add a second tunnel for port 1719, with all other settings equal. Then click **OK** and **Dismiss**.



Installing COMSOL on the AMI

Now you can proceed with the installation of COMSOL Multiphysics in the remote terminal. Enter:

```
cd ~/Private  
/comsol/COMSOL43b_dvd/setup ami
```

| Answer the following questions:

- Enter the installation directory: (press enter for default:
`/home/ec2-user/Private/`)
- Accept the license agreement (press the space bar to flip the pages until the last one, and answer **yes** to the question if you agree).
- Enter the path to the license file: (press enter for default: `1718@localhost`)

Continue with the installation by answering yes.

If you have problems with this step, go to the [Troubleshooting](#) section.

The installer installs the modules specified by your license, excluding the Model Library and documentation. A new directory is created in your `Private` directory, containing the COMSOL Multiphysics installation.

VERIFYING THE EC2 COMSOL INSTALLATION

- Still standing in the `Private` directory, enter

```
comsol43b/bin/comsol batch -inputfile point_source.mph -outputfile out.mph
```

There should be a brief log printed in the terminal, and the prompt should be returned with no error messages.

If you get error messages starting with

```
Exception:  
com.comsol.nativejni.F1NativeException: License error -15 Cannot connect to  
license server system
```

please refer to the [Troubleshooting](#) section.

You are now ready to run larger COMSOL jobs. Remember to transfer any results files back to your local machine before terminating the EC2 instance.

Terminating the Instance

When you are done with your session, it is recommended that you save your COMSOL installation and session configuration for later use. To do this, find the instance in the **EC2 Dashboard>Instances**, right click it and select **Create Image (EBS AMI)**. When you start your next Amazon EC2 instance you will find the COMSOL installation under the Images section.

Terminate the instance by right-clicking it in the list and selecting **Terminate**.



Remember, if you launched an instance in the Free Usage Tier, there are no charges. However, if you launched a regular instance, as soon as your instance starts to boot, you are billed for each hour or partial hour that the instance is kept running, even if the instance is idle. You will stop incurring charges for a regular instance as soon as the instance status changes to shutting down or terminated.

When you have decided that you no longer need the instance, you can terminate it. Terminating an instance effectively deletes it. You cannot reconnect to the instance after it is terminated. This differs from stopping the instance; you are still charged for a stopped instance, and you can restart a stopped instance.

Connecting a COMSOL Desktop GUI to a Remote Instance

You can run COMSOL interactively on the cloud using a local COMSOL Desktop GUI.

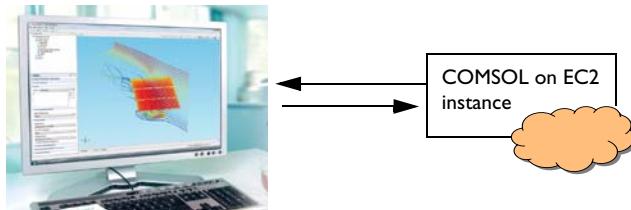


Figure 21-8: Working with the COMSOL Desktop using cloud computational resources seamlessly

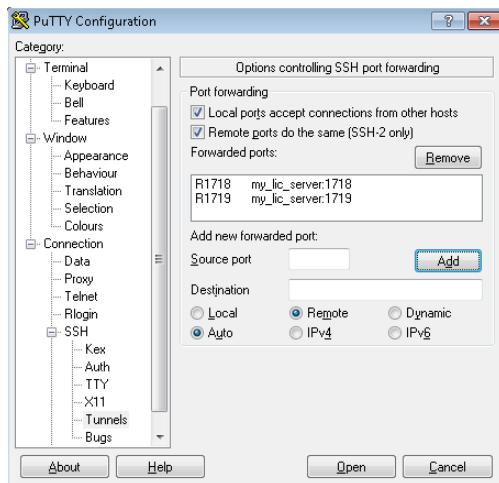
This can be done in two ways:

- Use the COMSOL Remote Access method. This method is based on file transfers between the local GUI and the remote instance. This is the recommended method since the COMSOL Desktop GUI performance is reliable and fast and independently of the internet connection data throughput.
- Use the client-server architecture. This is the less recommended method since even for fairly fast internet connections, the GUI tends to be slow.

CONNECTING USING THE REMOTE ACCESS METHOD

The COMSOL Desktop GUI needs secure connections for logging in and transferring files. On Windows, PuTTY is one tool that provides ssh and scp for this purpose. The procedure is described below. For Linux and Macintosh see [Using the Remote Access Functionality in COMSOL using Linux](#).

- 1 Install and configure PuTTY and generate keys to connect to the remote instance as described here: <http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/putty.html>.
- 2 Shut down the connection. Restart PuTTY, and extend the settings above with specifying the tunnel ports for the COMSOL license manager. Click **Connections**, **SSH** and then **Tunnels**.
- 3 Specify **Source port: 1718, Destination: my_lic_server:1718**, and click the **Remote** radio button. Click **Add**. Here, **my_lic_server** is the name or ip address of your on-premise COMSOL license manager.
- 4 Repeat the same procedure for port 1719 and click **Add**. The result is shown below



- 5 Go to **Session** and click **Save**.
- 6 Click **Open** to open the connection to the remote instance. In this new PuTTY terminal you can do file management and administration while working on COMSOL jobs.

Now start COMSOL locally and use the remote access functionality in the Cluster Computing and Cluster Sweep study node.



- [Cluster Computing \(Study\)](#) and [Cluster Sweep](#)
- [Remote Computing Preferences](#)

To do the following, first click the **Show** button () in the Model Builder and select **Show Advanced Study Options** from the menu. Then right-click a **Study** node and add a **Cluster Computing** or **Cluster Sweep** node. In the **Remote and Cloud Access** section of the **Cluster** settings window, enter or select these options:

- Select the **Run remote** check box and choose **SSH** from the **Remote invoke command** list.
- Set **SSH command** to PuTTY.
- Set the **SSH Directory** to your local PuTTY installation, for example
C:\Program Files (x86)\PuTTY
- The **SSH key file** is the path to your key .ppk file that you created with PuTTYgen.
- Use **Forward ports** to list the ports to forward for the license manager. Use comma separation, for example 1718, 1719. The **Port host** is the name of the local license server, for example my_lic_server.
- Set **SSH User:** ec2-user
- From the **File transfer command** list select **SCP** and from the **SCP command** list select **PuTTY**. Set **SCP directory** to the same as **SSH Directory** above. The **SCP key file** is the same as **SSH key file** above. **SCP user** is the same as **SSH user** above.

- In **Remote hosts**, enter the address to the Amazon computer, for example `ec2-23-22-5-238.compute-1.amazonaws.com`. Note that you can have several hosts listed, which is useful if you are running a Cluster Sweep, for example.
- Under **Remote OS** select **Linux** from the list.

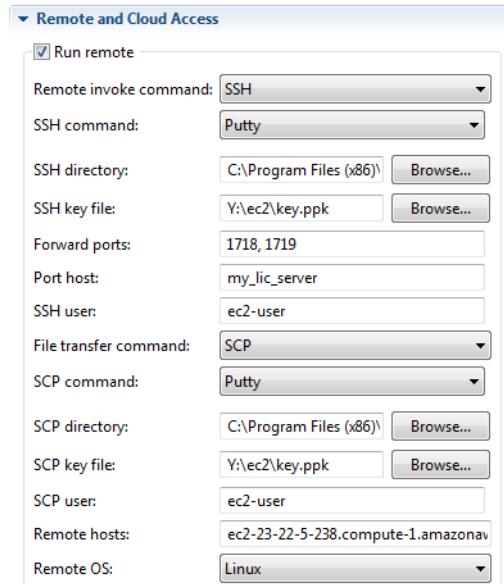


Figure 21-9: An example of Remote and Cloud Access settings section.

In the **Batch Settings** section set the following properties:

- **Cluster type:** Not distributed (if not).
- **Directory:** Set a local directory where you have write access.

- Select the **Specify external COMSOL batch directory path** check box and specify /home/ec2-user/Private
- Select the **Specify external COMSOL installation directory path** check box and specify /home/ec2-user/Private/

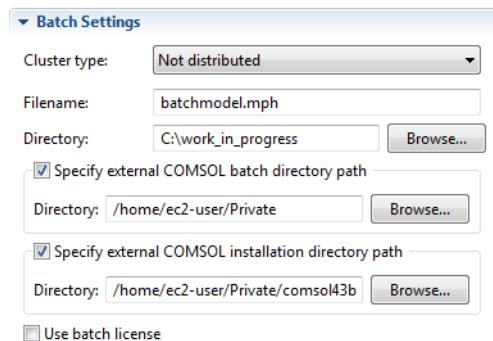


Figure 21-10: An example of the Batch Settings section.

To save this as a default configuration, click the **Save** button at the top of the **Cluster Computing** settings window. Click the **Compute** button () to start the computation.

CONNECTING USING THE CLIENT-SERVER METHOD

This method is only described for reference. The previously described method, [Connecting Using the Remote Access Method](#) is recommended before the client-server method, due to the relative slowness of the GUI for low- to mid-range Internet connections.

Set up the COMSOL AMI with the Java SSH as described in [Installing COMSOL on the AMI](#). Open port 2036 for the client-server connection:

- 1 Select menu **Tunnels>Setup**, then **Add...**
- 2 Enter:
 - Type: Local
 - Bind address: localhost
 - Bind port: 2036
 - Destination: localhost
 - Dest. port: 2036

3 Click **OK** and **Dismiss**.

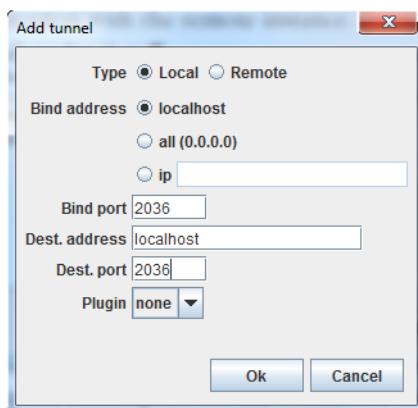


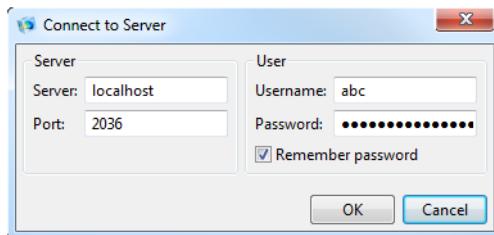
Figure 21-11: Port forwarding for the client-server connection.

4 In the remote shell, start a COMSOL server:

```
comsol43b/bin/comsol server
```

Enter a username and a password for the client-server connection.

5 Now open a COMSOL Desktop locally and select **File>Client-Server>Connect to server**. Specify **Server:** localhost, **Port:** 2036. Enter the username and password just specified when the COMSOL server was started.



Now you are ready to run a job.

Running a Parametric Sweep on Simple Cluster on the Cloud

You can easily run a parametric sweep in a cluster, without the need to set up a cluster with StarCluster. This is practical if you have a limited number of parametric steps. If you have large number of steps, more than 10 or so you should set up a StarCluster that administrates the cluster nodes.

- 1 Use the Cluster Sweep feature in COMSOL as described in the section [Connecting Using the Remote Access Method](#).
- 2 Start several Amazon Instances, preferably by right-clicking your prepared AMI (see [Terminating the Instance](#)).
- 3 Open a PuTTY session to each of the started instances, see [Connecting Using the Remote Access Method](#).
- 4 Set up the parameter steps in the Cluster Sweep settings.
- 5 In the Remote Hosts section of the Cluster Sweep settings (see [Figure 21-9](#)), add the IP addresses of all used instances, separated by commas.

Connecting to an EC2 Instance from Linux and Macintosh without Java



Most Linux and Macintosh machines include an SSH client by default. If you do not have one, refer to the OpenSSH project, which provides a free implementation of the full suite of SSH tools. For more information, go to <http://www.openssh.org>.

- 1 On the local Linux or Macintosh computer, use the `chmod` command to make sure the private key file is not publicly viewable. You cannot use the key if it is not protected. For example, if the file key is `johns_key.pem`, then enter:
`chmod 400 johns_key.pem`
- 2 In the **Navigation** area on the EC2 Dashboard, click **Instances**.
- 3 Right-click the instance and click **Connect**.
- 4 Click **Connect using a standalone SSH client**. AWS™ automatically detects the public DNS address of the instance and the key pair name you launched the instance with.

- 5 Copy the example command provided in the EC2 console.

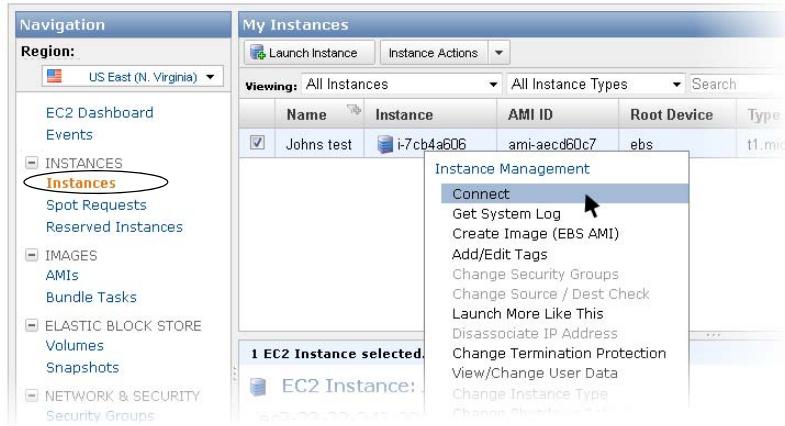


Figure 21-12: Find the instance in the dashboard. Right-click the instance and select “Connect>Connect using a standalone SSH client” to get the command line needed to connect to the instance.

Something similar to the following displays:

```
ssh -i johns_key.pem ec2-user@ec2-23-22-243-20.compute-1.amazonaws.com
```

DEFINE VARIABLES

To make these procedures easier, define some variables.

- Set the user:

```
export EC2USER=ec2-user
```

The default user name for a Linux AMI is always `ec2-user`.

- Set the address (example):

```
export EC2COMPUTER=ec2-23-22-243-20.compute-1.amazonaws.com
```

Depending on the local shell, you may need to use `setenv` instead of `export` in these commands.

LOGGING IN TO THE INSTANCE AND CREATING A PRIVATE DIRECTORY

- I Log in to the instance using the variable names:

```
ssh -i johns_key.pem $EC2USER@$EC2COMPUTER
```

- 2 Answer yes to the security question.

- 3** When you are logged in to the remote shell, it is likely you will want to work privately, so it is good practice to create a Private directory:

```
mkdir Private  
chmod 700 Private
```

TRANSFERRING FILES TO THE REMOTE INSTANCE FROM LINUX AND MACINTOSH

- I** Upload a small test model for later use, found in `models/COMSOL_Multiphysics/Equation-Based_Models/point_source.mph` available in your local COMSOL installation.



Don't have a local COMSOL installation? Download COMSOL from <http://www.comsol.com/support/download>.

Enter on your local computer:

```
scp -i johns_key.pem \  
<root>/models/COMSOL_Multiphysics/Equation-Based_Models/point_source.mph \  
$EC2USER@${EC2COMPUTER}:~/Private/
```

Where `<root>` is your local COMSOL installation path, for example `/usr/local/`.

- 2** On the remote shell, make sure the file was transferred:

```
ls ~/Private
```

VERIFYING A LOCAL COMSOL INSTALLATION ON LINUX

You need to test your local COMSOL license server before proceeding.

If you have access to a graphical display, start the COMSOL Desktop. It should start without error messages. From the menu **Options>Licenses**, verify that the license manager delivers all the add-on products expected based on your COMSOL configuration.



Don't have a local COMSOL installation? Download COMSOL from <http://www.comsol.com/support/download>.

If you only have access to a Linux terminal without graphics, then:

- I** Log in to a local Linux computer.

2 Copy the `point_source.mph` file to a local Linux working directory.

`point_source.mph` is a small mph file included in the COMSOL Model Library. It is in the COMSOL installation directory under

`models/COMSOL_Multiphysics/Equation-Based_Models/point_source.mph`

3 After copying the file, enter:

```
comsol batch -inputfile point_source.mph -outputfile point_out.mph
```

A log message on standard output should report the solving and saving of the file without error messages.

INSTALLING AND TESTING COMSOL ON THE EC2 INSTANCE

Now you can proceed with the installation of COMSOL Multiphysics in the remote instance. You have already tested SSH and SCP to communicate with the remote instance. However, for the remote COMSOL installation to operate you need to also open up the appropriate encrypted tunnel ports back to your on-premise license manager from the EC2 instance. Before continuing, make sure you have tested the on-premise license manager as described in the section [Connecting to an EC2 Instance from Linux and Macintosh without Java](#).

- I** Use SSH to forward the ports for SERVER and VENDOR from the on-premise COMSOL license server to the remote server. Therefore, exit the remote connection and reconnect:

```
exit  
ssh -i johns_key.pem -R 1718:my_lic_server:1718 \  
-R 1719:my_lic_server:1719 $EC2USER@$EC2COMPUTER
```

Here, `my_lic_server` is the name of your on-premise license server.

2 Install COMSOL:

```
cd ~/Private  
/comsol/COMSOL43b_dvd/setup ami
```

Answer the following questions:

- Enter the installation directory: (press enter for default: `/home/ec2-user/Private/`)
- Accept the license agreement (press the space bar to flip the pages until the last one, and answer `yes` to the question if you agree).
- Enter the path to the license file: (press enter for default: `1718@localhost`)

Continue with the installation by answering yes.

If you have problems with this step, go to the [Troubleshooting](#) section.

The installer installs the modules specified by your license, excluding the Model Library and documentation. A new directory is created in your `Private` directory, containing the COMSOL Multiphysics installation.

- 3 Run the previously uploaded `point_source.mph` model in the remote shell:

```
cd ~/Private  
comsol43b/bin/comsol batch -inputfile point_source.mph -outputfile out.mph
```

If everything works correctly, a log message on standard output displays a brief report of the solving and saving of a model without any error messages.

If you have problems with this step, go to the [Troubleshooting](#) section.

You are now ready to run COMSOL on the cloud.

Creating a Cloud Cluster Using the StarCluster Utility on Linux



StarCluster is an open source cluster-computing toolkit for Amazon's Elastic Compute Cloud (EC2). StarCluster has been designed to automate and simplify the process of building, configuring, and managing clusters of virtual machines on Amazon's EC2 cloud. StarCluster allows anyone to easily create a cluster computing environment in the cloud suited for distributed and parallel computing applications and systems. It is maintained by the MIT star project, and can be found at <http://star.mit.edu/cluster>.

The StarCluster tool simplifies the creation of a cluster on Amazon and can be used to run COMSOL in cluster mode. First, follow these instructions to create a volume containing the COMSOL installer that can be used by StarCluster:

- 1 Launch an AMI instance as described in the section [Launching and Using the Remote Instance](#).
- 2 In the Instances view on the EC2 Dashboard, right-click the instance and select Create Image. Type in an Image Name and click Yes, Create.
- 3 Go to the Snapshots view and right-click the 2 GB snapshot that was created. Select Create Volume from Snapshot. In the Volumes section you can find the created volume and its Volume ID (e.g., vol-12345abc). This ID will be used when configuring StarCluster.

Follow the instruction on the Starcluster site to install Starcluster on your local computer.

- 1 Install StarCluster according to the web page <http://star.mit.edu/cluster>.

Note: you can install StarCluster on-premise, but also on the cloud side. The preferred way is to install on-premise. If you cannot do that, install on the cloud side, use these commands instead of the instruction on the web page above.

```
sudo yum install python-devel gcc make  
sudo easy_install StarCluster
```

- 2 Set up the StarCluster config file by giving the command `starcluster help` and choosing the *Create config file* option.

- 3 When editing the `config` file, make sure that the `NODE_IMAGE_ID=` setting corresponds to the `x86_64` 64-bit Amazon machine image.

- 4 For added security it is recommended that you uncomment `PERMISSIONS` and that `PERMISSIONS=ssh` is set. Also, uncomment the `[permission ssh]` section setting and set `CIDR_IP` to your IP-address.

- 5 Fill out the fields in the AWS Credentials and Connection Settings section. The values for `WS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` can be found in the **User>Security Credentials** section of the EC2 dashboard.

Note: If you installed StarCluster on the cloud instance, make sure to create separate keys for use on the cloud instance.

- 6 To make the COMSOL installation accessible to StarCluster, create a new volume section in the config file

```
[volume comsol]  
VOLUME_ID = vol-12345abc  
MOUNT_PATH = /comsol
```

- 7 Add `comsol` to the keyword `VOLUMES` in the config file

```
VOLUMES = comsol
```

- 8 Use the command `starcluster createkey` to create a key with the same name and filename as in the config file.

Continue by installing COMSOL on the preconfigured `smallcluster` example cluster using the `starcluster` command. Either use spot instances or pay the full computer price. Note that you may lose your nodes after some time if you bid too low.

- 1 To find out the spot history enter:

```
starcluster spothistory m1.small -d 5
```

- 2 Start the cluster with:

```
starcluster start -c smallcluster smallcomsol \
```

```
{optional --bid {your bid here}}
```

For example:

```
starcluster start -c smallcluster smallcomsol --bid 0.20
```

This command also lists the address of the master node. You should now see the two instances on the EC2 dashboard, under Instances. Make sure to select the right **Region**.

Also, when the cluster is started you can list it with:

```
starcluster lc
```

This lists information including, for example:

```
Cluster nodes:  
master running i-e7789c9a ec2-184-72-201-244.compute-1.amazonaws.com  
node001 running i-5549ad28 ec2-54-242-55-117.compute-1.amazonaws.com (spot  
sir-e098c414)  
Total nodes: 2
```

- 3 Now set the following:

```
export EC2USER=sgeadmin  
export EC2COMPUTER=ec2-184-72-201-244.compute-1.amazonaws.c
```

INSTALL COMSOL ON THE CLUSTER AND TEST IT

- 1 Create a Private directory

```
ssh -i ~/.ssh/mykey.rsa $EC2USER@$EC2COMPUTER
```

Answer yes to the security question. When you are logged in to the remote shell:

```
mkdir Private
```

```
chmod 700 Private
```

- 2 Install COMSOL on the cluster, following the instructions in the section [Installing COMSOL on the AMI](#), with the exception that you should specify `1718@master` when asked for the license file path.

- 3 In order to test the cluster you need to have access to the license server from all nodes when you log into the master.

On your local computer, enter:

```
starcluster sshmaster smallcomsol \  
"echo GatewayPorts yes >> /etc/ssh/sshd_config"  
starcluster sshmaster smallcomsol "service ssh restart"
```

This tells the master that it should forward the ports from the remote cluster nodes.

- 4 Log in to the master node (see also [Copying a Custom COMSOL Version to the EC2 or other cloud services](#)):

```
ssh -i ~/.ssh/mykey.rsa -R 1718:my_lic_server:1718 \
-R 1719:my_lic_server:1719 $EC2USER@$EC2COMPUTER
```

Prepare the license file for the cluster, which is the same procedure as [Copying a Custom COMSOL Version to the EC2 or other cloud services](#). Then change `localhost` to `master` in the `license.dat` file on the remote master. See example below:

```
SERVER master ANY 1718
USE_SERVER
VENDOR LMCOMSOL port=1719
```

- 5 COMSOL is ready to be tested on the master node. First upload a small test model, for example, the `point_source.mph` file (see [Copying a Custom COMSOL Version to the EC2 or other cloud services](#)).
- 6 From the remote COMSOL installation enter the following:

```
cd ~/Private
comsol43b/comsol batch -inputfile point_source.mph
```

COMSOL should now open normally. Solving and saving the model should work without any error messages.

RUNNING COMSOL IN DISTRIBUTED MODE

You can now run COMSOL in distributed mode on the cluster.

- 1 Enter the names of the hosts in a hostfile:

```
cat > ~/hostfile
master
node001
```

- 2 Press the return key and press Ctrl+D.
- 3 Since the cluster is not backed by Infiniband, the following command tells COMSOL to use tcp instead.

```
comsol -nn 2 -f ~/hostfile -mpifabrics tcp batch -inputfile \
~/Private/point_source.mph
```

COMSOL should now start up on both master and node001 and be able to perform the simulation.

You are now prepared to start a larger cluster and run COMSOL on it using the same technique.



Remember to terminate the cluster once you are done, since you are billed until it is stopped or terminated.

Using the Remote Access Functionality in COMSOL using Linux



The remote access functionality requires COMSOL 4.3a or later.

Once a remote instance or cluster is created, the local COMSOL Desktop can be used to initiate simulations on Amazon. This is done through the remote access functionality in the Cluster Computing and Cluster Sweep study node.



- [Cluster Computing \(Study\)](#) and [Cluster Sweep](#)
- [Remote Computing Preferences](#)

To do the following, first click the **Show** button () in the Model Builder and select **Show Advanced Study Options** from the menu. Then right-click a **Study** node and add a **Cluster Computing** or **Cluster Sweep** node. In the **Remote and Cloud Access** section of the **Cluster** settings window, enter or select these options as shown:

- Check boxes to **Run remote** and set **Remote invoke command** to **SSH**.
- Set **SSH command** to **SSH** (or PuTTY or User) and possibly set **SSH directory** if needed.
- The **SSH key file** is the path to your key file.
- You can use **Forward ports** to list the ports to forward and the name of the **Port host** in order to forward the license request. As an alternative you can have an SSH connection running that forwards the ports. See [Copying a Custom COMSOL Version to the EC2 or other cloud services](#).
- Set the **File transfer command** to **SCP** and set **SCP command** to **SCP** (or PuTTY or User), and possibly **SCP directory** if needed. The SCP key file is the path to your key file.

- In **Remote hosts**, name the address to the Amazon computer. Note that you can have several hosts listed, which is useful if you are running a Cluster Sweep for example.
- Under **Remote OS** set Linux or possibly Native if you are running Linux.

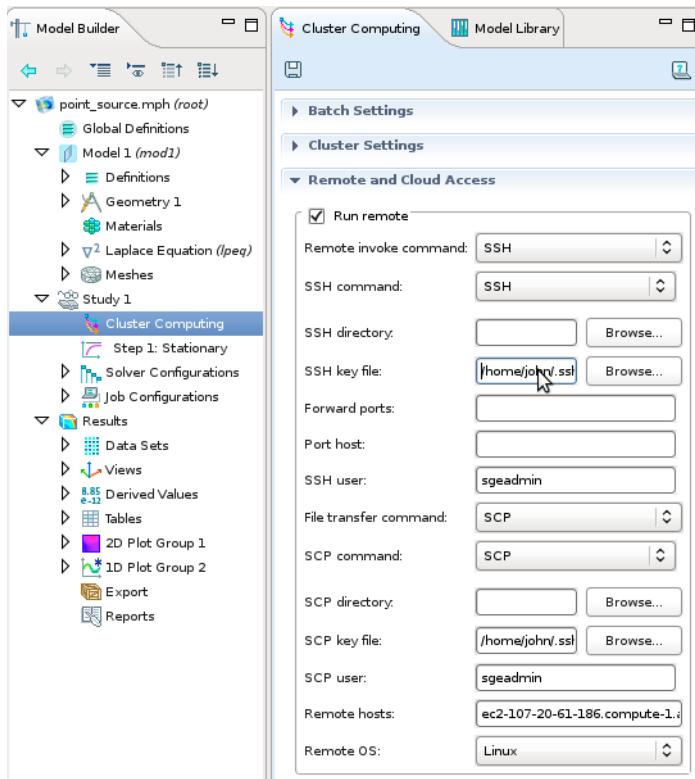


Figure 21-13: Cluster settings in the user interface, Remote and Cloud Access section.

- Under **Batch settings** enable the **Specify external COMSOL batch directory path** and **Specify external COMSOL installation directory path** settings and set the paths to the remote COMSOL installation directory and the directory where you want to save your work in (for example, Private). Depending on whether you are running a cluster simulation or a simple batch job, set **Cluster type** to **General** or **Not distributed**. For a cluster you must also point to the **Host** file.
- For the Starcluster functionality, open **Options>Preferences>Cluster Computing** and enter the following in **Postpend command:** and click Save.

```
-mpifabrics tcp
```

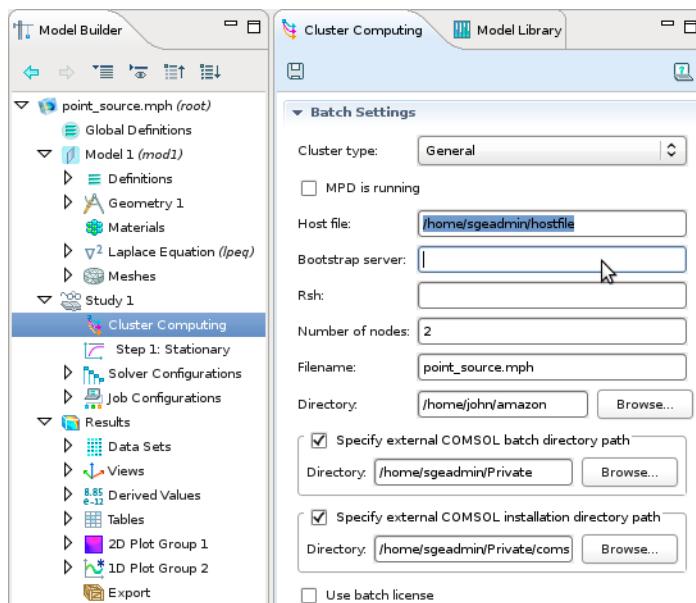


Figure 21-14: Cluster settings in the user interface, Batch Settings section.



Click the **Save as default** button to save these settings for future use.



Cluster Computing Preferences

You are now ready to compute the solution by clicking the **Compute** () button.

Finish your session by entering `starcluster terminate smallcomsol` in your local Linux shell. Alternatively, go to the EC2 console and right click each node and select Terminate.

Troubleshooting

- Problem 1: The installation of COMSOL on the remote instance stalls or gives error messages after specifying license manager 1718@localhost.
 - Solution 1: Your COMSOL system administrator has changed the default ports 1718 and 1719 in the COMSOL license server. Change the port forwarding to the new port numbers, see further [Preparing the COMSOL License Management](#).
 - Solution 2: You have set the incorrect license server name in the port forwarding step [Preparing the COMSOL License Management](#). Double check the server name with your COMSOL license administrator.
 - Solution 3: Try creating a license file manually on the remote instance:
Providing you use default ports, the license.dat file for the EC2 instance should contain only the three lines:

```
SERVER localhost ANY 1718
USE_SERVER
VENDOR LMCOMSOL port=1719
```

You can create this file in the remote shell (see [Figure 21-5](#)) with the following commands

```
cd ~/Private
cat > license.dat
SERVER localhost ANY 1718
USE_SERVER
VENDOR LMCOMSOL port=1719
```

then press Ctrl+D. You can also use the nano editor in the remote shell to create the file, or create it locally and transfer it to the remote instance as described in [Transferring Files](#). Now reinstall COMSOL in the remote instance, and point to the license.dat file when prompted for license information.

- Problem 2: When verifying the remote COMSOL installation, error messages are shown.
 - The ports forwarding to the license server is incorrect. See solution to Problem 1 above.

Copying a Custom COMSOL Version to the EC2 or other cloud services



If you want to use another COMSOL version than those available in the EC2 Marketplace (see [Setup and Provisioning](#)), or if you want to use COMSOL on a different could service than Amazon, you can upload the entire COMSOL installation to the could. To save bandwidth, create a .zip file of the local COMSOL installation and transfer it to the remote instance. To make the zip file as small as possible, skip the `models` and `doc` directories, which are located in the COMSOL installation root directory. There may be transfer costs involved in copying data to and from the instance.

If you work on Windows locally, you have to upload the entire COMSOL DVD to the EC2 or other instance.

This is, however, both time consuming and costly due to possible transfer charges. Instead, it is recommended that a local Linux installation is made and only the parts that are needed are uploaded for the computations:

- 1 Install COMSOL locally, then find the installation directory and make a zipped copy. If the COMSOL root is `/usr/local/`, enter:

```
cd /usr/local  
zip -r ~/comsol.zip -x /doc/* /models/*
```

- 2 Copy the zipped COMSOL installation from the local computer to the remote computer:

```
scp -i johns_key.pem ~/comsol.zip \  
$EC2USER@${EC2COMPUTER}:~/Private/
```

- 3 Also copy the small example mph file:

```
scp -i johns_key.pem point_source.mph \  
$EC2USER@${EC2COMPUTER}:~/Private/
```

- 4 On the remote computer, unzip the file:

```
cd ~/Private  
unzip -q comsol.zip
```

- 5** Add the line: "export MKL_DYNAMIC=FALSE" to the file .bashrc in the remote shell by the command:

```
echo export MKL_DYNAMIC=FALSE >> ~/.bashrc
```

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 program 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 where an Eulerian equation is transformed into an equation written with respect to a mesh, which may be moving in relation to both the Eulerian frame and the Lagrangian frame. The COMSOL Multiphysics solvers have built-in support for the necessary transformation of derivatives.

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 n th-grade polynomial in terms of backward differences.

basis function A function φ_i in the *finite element space* such that the i th 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 a *Bézier curve*. A *Bézier patch* is a function of two variables with an array of control points.

bidirectional constraint A constraint enforced by reaction terms affecting both equations in a constraint of the type $u_1 = u_2$. *Symmetric constraints* are an important special case. See also *reaction terms* and *constraint*.

Boolean operations Boolean operations are used to construct a *geometry object* from other geometry objects. 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 dimension one less than the space dimension for the geometry (a *face* in a 3D geometry, an *edge* in a 2D geometry, and a *vertex* in a 1D geometry). In a mathematical context, the symbol $\partial\Omega$ 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*.

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 operation 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

$$\left\{ \begin{array}{ll} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f & \text{in } \Omega \\ \mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) + q u = g - h^T \boldsymbol{\mu} & \text{on } \partial\Omega \\ h u = r & \text{on } \partial\Omega \end{array} \right.$$

composite geometry 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 products with a number of windows such as the Model Builder window, the Graphics window, and each model tree node's settings window.

COMSOL binary file A binary data file with the extension `.mphbin` that contains geometry objects or mesh objects.

COMSOL text file A text data file with the extension `.mphtxt` that contains geometry objects or mesh objects.

COMSOL server The COMSOL server is single user server allowing multiple session of the same user, one session at a time.

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. The built-in constants in COMSOL includes *mathematical and numerical constants* and *physical constants*.

constraint Restriction imposed upon the dependent variables on the form $R(u_1, u_2, \dots) = 0$. A *Dirichlet boundary condition* is a special case. *Neumann boundary conditions* are not regarded as constraints. When a constraint is added, the finite element algorithm adds corresponding *reaction terms* to the system of equations. These generalized *reaction forces* modify the *flux conditions* so that the resulting model becomes solvable.

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*.

contributing node A boundary condition or source is *contributing* when it adds to other boundary conditions or sources defined on the same geometric entity. Examples of contributing boundary conditions are loads in structural mechanics and heat flux components in heat transfer. See also *exclusive nodes*.

control point *Bézier* and *NURBS* curves and surfaces are defined by a set of points known as *control points*. The locations of 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.

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* (where no vertex is isolated), 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 is 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.

deformation gradient In solid mechanics, it contains the complete information about the local straining and rotation of the material. It is a nonsingular matrix with positive determinant, as long as material cannot be annihilated.

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*. 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 with properties suitable for representing certain electromagnetic vector fields. The degrees of freedom on the boundary of a mesh element correspond to normal components of the field.

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:

$$\lambda^2 e_a u - \lambda d_a u + \nabla \cdot (-c \nabla u - \alpha u) + \beta \cdot \nabla u + \alpha u = 0$$

elliptic PDE A linear stationary 2nd-order elliptic PDE has the form

$$\nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

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*.

exclusive node A boundary condition or material model in a domain is *exclusive* when there can only be one such node defined for a given geometric entity. Adding another exclusive boundary condition to the same boundary, for example, the last added boundary condition (last in the Model Builder tree) overrides any other similar boundary condition defined on the same boundary. Examples of exclusive boundary conditions are prescribed displacements in structural mechanics and specified temperature in heat transfer. See also *contributing node*.

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 computational domain. See also *boundary*.

extrude To create a 3D geometry object from a 2D geometry object in a *work plane* or a planar face in 3D by translating (extruding) it in the normal direction.

extrusion model coupling A coupling defined 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 A *geometric entity* describing a bounded part of a *surface* in a 3D geometry. A *face* is a *boundary* in a 3D geometry. See also *domain*.

fallback feature Used with the pair node to enable pairs to have the option to add additional subnodes with conditions for nonoverlapping parts of the pair.

FEM See *finite element method*.

Fick's law The first law relates the concentration gradients to the diffusive flux of a solute infinitely diluted in a solvent. The second law introduces the first law into a differential material balance for the solute.

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.

finalized geometry The resulting geometry used for assigning materials and physics. COMSOL creates the finalized geometry by forming a union of all geometry objects or by forming an assembly where the geometry objects are individual parts. The finalized geometry consists of *geometric entities*.

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 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 condition A boundary condition which specifies the value of the *normal flux* across a boundary, also known as a *natural boundary condition*. A (generalized) *Neumann boundary condition* is a special case.

flux vector The general flux vector is as below, with three terms: the first term describes diffusion, the second term describes convection with a velocity $-\alpha$, and the third term γ is a source term. 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 or 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 tet mesher The mesh generator creating unstructured tetrahedral meshes.

free triangle mesher The mesh generator creating unstructured triangular meshes.

function COMSOL supports *user-defined functions*, which can be analytic, piecewise, and interpolation functions and special types of common functions that implements, for example, steps, ramps, and other wave forms. There are also common built-in *mathematical functions* such as trigonometric functions, logarithms, and special functions.

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

$$\begin{cases} e_a \frac{\partial^2 u}{\partial t^2} + d_a \frac{\partial u}{\partial t} + \nabla \cdot \Gamma = F & \text{in } \Omega \\ -\mathbf{n} \cdot \Gamma = G + \left(\frac{\partial R}{\partial u}\right)^T \mu & \text{on } \partial\Omega \\ 0 = R & \text{on } \partial\Omega \end{cases}$$

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

$$\mathbf{n} \cdot (c \nabla u + \alpha u - \gamma) + q u = g - h^T \mu$$

The generalized Neumann condition is often called just *Neumann condition* in the documentation.

generalized reaction force see *reaction term*.

geometric entities The basic parts that constitute the finalized geometry: In 3D they are divided in the following four types or *geometric entity levels*: *domains*, *boundaries* (*faces*), *edges*, and *points* (*vertices*). In 2D, there are no *faces*, and the *edges* are the *boundaries*. In 1D there are only *domains* and *points*, which are also the *boundaries*.

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. No physics is formulated directly in the geometry frame—only the *material frame* and *spatial frame* have physical significance. The geometry frame is used only as a reference for the Deformed Geometry interface and for postprocessing. When there is no Deformed Geometry interface present, the geometry frame is identical to the material frame.

geometric entity level The *geometry entity 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 *geometric entities*.

geometry model A collection of *geometric entities* that form a complete geometric description of the model.

geometry object An object generated by a geometry feature. See also *point object*, *curve object*, *surface object*, *primitive geometry object*, *solid object*, and *mixed 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*

$$e_a \frac{\partial^2 u}{\partial t^2} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

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 files can be imported into COMSOL 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 usually 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 edges 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*.

initial condition The starting values for the dependent variables in a time-dependent analysis and for *nonlinear iterations* or other iterative solvers.

integration model coupling A coupling that evaluates integrals of expressions over the source and returns a single scalar value when used in the destination, which for this type of model coupling is the entire model. 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 (points) 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 *flux conditions* when a constraint is added. The Lagrange multiplier often has a physical meaning and an interpretation as a (generalized) *reaction force*. 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*.

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*.

mathematical and numerical constants Built-in common mathematical constants such as π and i and numerical constants such as the machine precision or machine epsilon.

mesh A subdivision of the entities of a geometric model into, for example, triangles (2D) or tetrahedrons (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. When remeshing is not used, the mesh frame is identical to the *geometry frame*.

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*.

mixed object A nonempty *geometry object* that is not a *solid object*, *surface object*, *curve object*, or *point object*. For example, the union of a solid object and a curve object is a mixed object.

mode reduction A model-reduction technique for reducing systems with many degrees of freedom, such as large finite element models, to a form with fewer degrees of freedom for dynamic system simulations and analysis. See also *state-space model*.

model coupling User-defined model couplings are used to couple data within a model (geometry) or between different models (geometries). See also *extrusion model coupling*, *projection model coupling*, and *integration model coupling*. Model couplings can be reused with different arguments (for example, for integrating different quantities over the same domain).

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 calling on the COMSOL API. Use a text editor to extend and modify the Model Java-file. Compiling and running a Model Java-file creates the COMSOL model.

Model MPH-file A binary data file with the extension .mph that contains a COMSOL model. Often also just called model file.

Model M-file An M-file containing commands that create a COMSOL 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 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 model.

Model Wizard Part of the COMSOL Desktop that is used to start building a model. It contains the Select Space Dimension, Select Physics, and Select Study Type pages.

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.

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 possibly 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 numerical 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 curve and surface representation scheme. A NURBS representation can be divided into a number of *rational Bézier curves* or surfaces.

operator, operator function A user-defined *operator function*, or just *operator*, are similar to a *function* but behaves differently. For example, COMSOL includes differentiation operators that take expressions as input arguments to define a derivative of an expression with respect to a variable. There are also built-in arithmetic, relational, and logical operators.

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*

$$d_a \frac{\partial u}{\partial t} + \nabla \cdot (-c \nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + \alpha u = f$$

where d_a and c are positive.

parameter A constant that can take 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.

physical quantity A quantity (quantifiable property) that can be used in the mathematical equations of science and technology.

physics user interfaces Sets of physics nodes for different types of physics in the COMSOL Desktop environment. The physics user interfaces (sometimes referred to as the *physics*) contain predefined equations and boundary conditions and a set of nodes for setting up models for that type of physics.

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 element when it is used to eliminate values below it is called the *pivot value*.

point A location in space. Often used in a narrower sense with the same meaning as *vertex*.

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 model coupling A coupling 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*.

reaction force see *reaction term*.

reaction term Terms that are automatically added to the system of equations in order to enforce a *constraint*. Reaction terms from boundary constraints appear as a *flux condition* and share the same physical meaning. Using an analogy from structural mechanics, reaction terms are sometimes referred to as (generalized) reaction forces.

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 planar face by rotating it about an axis.

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 See *solid object*.

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 object*.

solid object A geometry object whose *vertices*, *edges*, and *faces* all have an adjacent *domain*.

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*.

solution vector A vector with all the *degrees of freedom* (values of the *dependent variables*) as its components. See also *solution matrix*.

solver sequence A 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 (also called the Eulerian frame) is used in connection with an *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 smooth mathematical function 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*.

surface object A geometry object without domains, isolated edges, or isolated vertices. Typically a trimmed surface is represented as a surface object.

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 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*.

symmetric constraint A *constraint* which is enforced by *reaction terms* chosen so as to preserve the symmetry of symmetric unconstrained systems. This choice of reaction terms is unique and leads to a *bidirectional constraint* which modifies the equations corresponding to all dependent variables appearing in the constrained expression.

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 A constraint enforced by *reaction terms* that only affect one of the dependent variables in a constraint of type $u_1 = u_2$. The other dependent variables are treated as independent with respect to the unidirectional constraint. Compare to *symmetric constraint*. See also *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*.

user-defined variable A user-defined variable can be defined on a global level or on any geometric entity in terms of *dependent variables*, *independent variables*, *parameters*, *constants*, and other *variables*.

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 an edge 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 *constraint* 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. Physics user interfaces in COMSOL 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*. From a work plane with a 2D geometry, 3D geometry objects can be created using *extrude* or *revolve* operations.

I n d e x

- 0D spatial dimension 41
- 1D models, solving 442
- 1D plot group (node) 1308
- 1D, 2D, and 3D graphics toolbar buttons
 - 355
- 2D axisymmetric models 1331
- 2D axisymmetric models, solving 442
- 2D axisymmetry
 - laminar flow and 734
- 2D models, solving 442
- 2D plot group (node) 1310
- 2D view (node) 386
- 3D models
 - guidelines for solving 444
 - infinite elements and 346
- 3D plot group (node) 1310
- A**
 - About COMSOL Multiphysics box 72
 - absolute IPS systems 180
 - absolute pressure 791
 - absolute tolerances, solvers and 1115
 - AC impedance stationary (node) 1063
 - AC impedance studies 1215
 - AC impedance time dependent (node)
 - 1063
 - AC/DC Module 808
 - accumulated probe table 1362
 - accurate fluxes
 - creating variables for 135
 - ACML (AMD Core Math Library) 1428,
 - 1440
 - Acoustics Module 687
 - activating adaptive mesh refinement 1034
 - Adams-Basforth method 912, 1117
 - adaptive mesh refinement 1034, 1036
 - adaptive mesh refinement (node) 1126,
 - 1209
- adaptive solvers 1126
- add Material Library (button) 557
- add to selection (button) 353
- adding
 - arrays, example 463
 - extrusions, example 458
 - geometry sequences 413
 - geometry, example 448, 450
 - mesh features 469
 - mesh sequences 469–470
 - model geometry 407
 - multiphysics couplings 615
 - nodes 48, 85
 - objects to Boolean operations 509
 - physics interfaces 42, 408
 - polygon segments 475
 - rotations, example 452, 459
 - solvers to solve nodes 1079
 - source terms 902
 - study nodes 1004
 - study steps 1010
 - user model library 121
 - variable expressions to plots 1242
 - variables 212
- adj operator 225
- adjacent (node) 376
- adjacent geometry objects 349
- adjacent selection (node) 383
- adjoint methods 1188
- adjoint sensitivity 964, 966
- advanced (node), solvers 1132
- advanced physics options, expanding 133
- advanced settings 607
- affine transformations 424
- ALE method 977–978
 - limitations of 985

- algebraic equation (node) 943
- algebraic multigrid preconditioners and solvers 1171
- align with work plane button 420
- alpha blending 389
- ambient light 392
- Ampère’s law (node) 674
- AMS (node) 1137
- analysis, sensitivity 1187
- analytic function (node) 263
- analytic functions, materials and 586
- angle, for complex-valued solution 1285
- angular unit, for geometry 434
- animation (node) 1365
- anisotropic (material node) 577
- anisotropic material properties 570
- anisotropic materials 622
 - ALE and 977
 - coordinate systems and 305
 - modeling 154
- anisotropic matrix, materials 155
- anisotropic, Voigt notation (material node) 577
- antialiasing
 - plot images 1374
 - screenshot images 38, 402
- appearance, of icons 78
- applying reaction terms 888
- arbitrary Lagrangian-Eulerian formulation. See ALE
- Arnoldi method 1090
- ARPACK 1090
- array (node) 501
- arrays and ranges 254
- arrow line (node) 1312
- arrow surface (node) 1313
- arrow volume (node) 1313
- Arruda-Boyce (material node) 578
- artificial diffusion 126, 200
- artificial viscosity 200
- aspect ratio 390
- assemble (node) 1209
- assembly object 436
- assembly, geometry 435
- associative geometry 441
- asymptotic waveform evaluation. See AWE
- at operator, spatial 236
- attribute features, solvers 1081
- attribute nodes, solvers 1126
- author name, resetting 95
- author, include in history 101
- automatic rebuild, of geometry 435
- automatic remeshing 987, 1036
- automatic remeshing (node) 1138
- auxiliary dependent variable (node) 926
- auxiliary space Maxwell solver 1137
- average (node)
 - data set 1267
 - model coupling 300
- average coupling operators 285
- average values, on boundaries 253
- average, data series operation 1261
- average, time-dependent expression 237
- AVI file formats 1365
- AWE solver 1041
- AWE solver (node) 1087
- axial symmetry
 - boundary conditions 149, 160
 - cylindrical coordinates 443
- axial symmetry (node) 710
- axis (node) 386
- axis limits, locking 386
- axisymmetric geometries
 - PDE formulation in 865
- spatial coordinates for 246

- axisymmetric models
 - cylindrical coordinate systems, and 444
 - particle tracing with mass theory 1331
 - physics interfaces, and 602
 - revolution data sets and 1281
 - selecting 41
 - solid mechanics interface 826
- azimuth axis
 - of spherical coordinate system 316
- B**
 - Babuska-Brezzi condition 738
 - backward differentiation formulas 1106
 - backward Euler method 233
 - ball (node)
 - meshes 506
 - selections 373
 - ball operators 226
 - ball selection (node) 380
 - ball, sphere geometry node 497
 - band-pass filtering 278
 - base vector system (node) 307
 - Basquin (material node) 580
 - Basset history term 748
 - batch (node)
 - job configurations 1223
 - study types 1046–1047
 - batch command 1431, 1443, 1458
 - batch commands 1408
 - batch data (node) 1225
 - batch jobs, synchronizing with GUI 1049
 - batch settings 1049–1050
 - batteries and fuel cells materials 595–596
 - bdf operator 226
 - BDF solvers, generalized alpha and 1107
 - Bernstein basis 476
 - Bézier polygon (node) 475
 - BH curve (material node) 574
 - BiCGStab iterative method 1158
 - BLAS 911
 - BLAS libraries 1417, 1427, 1440, 1455
 - Blatz-Ko (material node) 579
 - Blinn-Phong lighting model 567
 - block (node) 477
 - bndenv operator 229
 - body load (node) 838
 - Boolean expressions 230
 - Boolean operations
 - compose 502
 - compose and 502
 - difference 509
 - features 423
 - intersection 516
 - overview 406
 - union 528
 - Boolean selections 375, 383
 - borrowing licenses 96
 - boundary
 - definition 349
 - elements, meshes 468
 - similarity coupling operators 293
 - boundary conditions
 - axial symmetry 160
 - definition 156
 - electromagnetics theory 623
 - for non-overlapping pair parts 318
 - heat equation, and 776
 - identity pairs 318
 - inlet and outlet, theory 744
 - scattering and port 335
 - wall distance interface 949
 - boundary coordinate system (node) 309
 - boundary current source (node) 663
 - boundary electromagnetic heat source (node) 815
 - boundary element refinement 1342
 - boundary fluxes
 - variables for 135

- boundary heat source (node) 801
- boundary heat source variable 775
- boundary layer properties (node) 535
- boundary layers (node) 519
- boundary layers, meshes 487
- boundary load (node) 839
- boundary load, flux condition 157
- boundary loads theory 854
- boundary mode analysis (node) 1071
- boundary nodes
 - deformed geometry 998
 - electric currents interface 656
 - electrostatics interface 639
 - heat transfer 786
 - magnetic fields interface 673
 - moving mesh 992
 - pressure acoustics, frequency domain interface 700
 - single-phase flow 753
 - solid mechanics interface 833
 - transport of diluted species 721
- boundary point probe (node) 327
- boundary point probes 324
- boundary probe (node) 325
- boundary selection 608
- boundary stress (node) 762
- boundary-value problems 938
- Boussinesq approximation 741
- box (node)
 - meshes 507
 - selections 373
- box selection (node) 380
- branches, in the Model Builder 46
- British engineering unit systems 180
 - special units in 189
- British thermal units 188
- browse update (button) 12
- buckling analysis 1064
- build all
 - geometry 410
 - meshes 470
- build preceding
 - geometry 414
- build selected
 - geometry 408
 - meshes 470
- building expressions 209
- building models 43
- built-in materials database 551
- bulk modulus
 - elastic moduli 849
 - pressure acoustics 690
- bulk modulus (material node) 577
- C**
 - Cam-Clay material model (node) 581
 - camera (node) 390
 - camera settings 390
 - camera, locking 389
 - canceling solvers 1024
 - canonical systems 846
 - Cartesian coordinate systems 246, 305, 442
 - cell Reynolds number 760
 - center of rotation 390
 - centroid operator 227
 - CFD Module 787
 - high Mach number flow 736
 - non-Newtonian flows 733
 - slip condition, and 743
 - swirl flow 734
 - CFL number
 - pseudo time stepping, and 739, 752, 1105
 - wave PDE 911
 - CGSA unit systems 179
 - special units in 190
 - chamfer (node) 502

change cross-section (node) 653
change of state, analysis cases 80
change thickness (node) 836
change thickness (out-of-plane) (node)
 654
changing. see editing
charge conservation (node) 641
charge relaxation theory 625–627
check mark definition, materials 565
Chemical Reaction Engineering Module
 627
chirality 486
circle (node) 478
circle operators 226
circular cone 479
circumcenter operator 227
circumferential wave number 693
classical PDEs 872
clear selection (button) 353
clearing meshes 472
client/server
 security issues with 1430, 1442, 1457
client/server architecture 1406
clipboard
 copying images 402
 copying table data to 1353
cluster computing 1226
cluster computing study (node) 1049
cluster jobs 1050
 on Linux 1421
 on Windows 1420
cluster settings 1177
cluster sweep, study (node) 1054
clusters
 license handling for 1422
 parallel mode 1407
coarse grid correction 1172
coarse solver (node) 1139
codecs 109
coefficient form PDE (node) 899
coefficient form PDE interface 893
coefficient form, solving 1086
coefficient of thermal expansion
 for non-SI units 573
Coffin-Manson (material node) 580
collapse edges (node) 537
collapse faces (node) 537
collapsing all nodes 52
color (button) 38
color expression (node) 1347
color legends, position 1311
color tables 1249
colors
 of materials 566
 of plots and plot groups 1293
comet tail plots 1256
comments, adding to models 94
compact MPH-files 88
compact notation, classical PDEs 872
comparing solutions 1274
compile equations (node) 1210
compile equations, global equations and
 941
complement (node) 375
complement selection (node) 383
complex variables, splitting 136, 1211
complex-valued output 1134
complex-valued problems 1031
component variables 248
compose (node) 502
composite edges and faces 533
composite object (node) 439
composite solid objects 406
composition 561
compressible flow 736
compute differential check box 1262

compute nodes 1407
COMSOL Batch 1431, 1443, 1458
COMSOL batch feature 1408
comsol commands
 on Linux 1435
 on Macintosh 1452
 on Windows 1424
COMSOL Desktop
 client/server architecture and 1406
 connecting to a server 1414
 help window 64
 languages for 37
 layout 34
 Model Library 68
COMSOL Java API 1408
COMSOL MATLAB command 1435,
 1451, 1459
COMSOL Multiphysics files 510
COMSOL Multiphysics server
 accessing 1429, 1456
concentration (node) 725
conductive heat flux variable 774
cone (node) 479
cone, geometry 482
conic sections 477
conical frustum 479
conjugate gradients solver 1163
conjugate gradients, iterative method
 1158
conjugate gradients, iterative solver 163
connecting
 COMSOL to a server 92, 1414
 MATLAB to a server 1415
conservation of energy 768
consistent stabilization methods 137, 201
consistent stabilization settings 608
constants
 as predefined variables 221
mathematical 218
numerical 218
constitutive relations theory 620–621
constraint 868
constraint (node) 905
constraint elimination 1136
constraint force Jacobian 883
constraint group (node) 176
constraint groups 80, 173
constraint settings 164, 608, 888
constraint-handling method 1136
constraints
 defined 156
 editing 150
 equation views node and 152
 on time derivatives 889
 structural mechanics 128
contact impedance (node) 665
contact pair (node) 322
contact pairs 318
context help 62, 64
context menus 48, 142
continuation solver 1032
continue solving 1004
continuity (node)
 acpr interface 710
 heat transfer 803
 interior boundaries 159
 pressure acoustics, frequency domain
 interface 710
continuous tangent, when selecting 371,
 373, 380
contour (node)
 data set 1268
 plots 1313
contours, level labels and 1314
contributing nodes 145
contribution

global 965
integral 965
listing 146
probe 965
control field (node) 1139
control points 476
control state (node) 1140
control variable field 971
convection and diffusion (node) 722
convection-diffusion equation (node) 903
convective cooling, flux condition 157
convective terms 715
convergence data, copying 1027
convergence plots 1027
generating 1005
convergence tests 125
convergence, discontinuous functions
and 281
conversions menu
convert to curve 503
convert to point 504
convert to solid 504
convert to surface 505
convert (node), meshes 520
convert to curve (node) 503
convert to point (node) 504
convert to solid (node) 504
convert to surface (node) 505
converting units 193
convolution integrals 228
Cook-Torrance lighting model 567
coordinate system line (node) 1314
coordinate system selection 608
coordinate system settings (node) 959
coordinate system surface (node) 1314
coordinate system volume (node) 1314
coordinate systems
absolute 305
base vector 307
boundary 309
Cartesian 305
cylindrical 310
infinite elements and 343
mapped 312, 318
physics symbols and 828
relative 305
rotated 313
scaling 316
solid mechanics theory 846
spherical 315
user defined 305
copy (node), geometry 506
copy domain (node) 521
copy edge (node) 521
copy face (node) 521
copying
COMSOL models 90
convergence data 1027
geometric entities 354
geometry objects 432
mesh solutions 988
nodes 51
screenshots to clipboard 402
selections 352
tables in results 1353
user-defined material libraries 558
corner refinement (node) 536
Coulomb gauge 633
coupling operators
average 285
boundary similarity 293
general extrusion 291
integration 285
linear extrusion 292
maximum 285
minimum 285

couplings, adding 615
create plot button 262, 1286
create selection (button) 359
create vertex (node), meshes 508
creating
 1D geometry models 446
 2D geometry models 446
 3D geometries 419
 3D geometry models 455
composite geometry objects 423
cross-section plots 1294
default plots 1287
deformed meshes 988
geometries 417, 440
materials libraries 558
memory efficient geometry 123
meshes 468
multiphysics models 614
selections 352
smooth functions 281
user-defined selections 371, 377, 379
user-defined views 387
variables 212
work planes, example 460

Creo Parametric custom file folder 117

cross section (node), geometry 507

cross-section plots
 creating 1294, 1301
 cut plane data sets 1305
 cut point data sets for 1298
 data sets and plot groups, deleting 1297
 data sets for 1264
 line 2D 1300
 line 3D 1302
 line plots 1300
 point 1298
 surface 1305

toolbar 1295

crosswind diffusion
 consistent stabilization method 137
 definition 204, 781
 fluid flow 737
 heat transfer and 781

cubic Bézier curve 477

cubic Hermite spline, for interpolation 1035

cubic splines 273

curl 865

curl-curl operators
 preconditioners for 1161
 SOR vector and 1195

current conservation (node) 658

current distribution initialization (study step node) 1059

current node
 geometry 413

current node, geometry 413

current source (node) 661

current sources, theory 627

curve
 inflection point for 477
 quadratic 477
 third-degree 477

curvilinear coordinate systems 953

custom color tables 1251

customizing plot titles 1244

cut line 2D (node) 1269

cut line 3D (node) 1269

cut line data sets 1300

cut plane (node) 1270

cut plane data sets 1305

cut planes 1294

cut point 1D (node) 1271

cut point 2D (node) 1271

cut point 3D (node) 1271

cut point data sets 1298
cyclic color tables 1250
cyclic voltammetry (node) 1075
cylinder (node)
 geometry 481
 meshes 508
 selections 373
cylinder selection (node) 380
cylindrical coordinate systems 826
cylindrical system (node) 310
cylindrical system, PDE for 865
cylindrical wave radiation (node) 708

D Damköhler number 203
damped Newton method 1101
damping (node) 837
damping factors 1102
damping models 858
damping ratios 1096
data (node) 1370
data export, duplicates and 1365
data sets
 average 1267
 contour 1268
 cut line 1269
 cut plane 1270
 cut point 1271
 defining 1264
 edge 2D and 3D 1273
 function 1273
 integral 1267
 isosurface 1274
 join 1274
 maximum and minimum evaluation
 1276
 mesh 1276
 mirror 1276
 parameterized curve 1277
 parameterized surface 1278

parametric extrusion 1279
particle 1280
revolution 1281
selections, adding 1265
solution 1284
surface 1285
data sets (node) 1264
default
 BLAS library 1417
 color tables, changing 1253
 coordinate system variables 444
 desktop settings 36
 fonts 36
 help searches 63
 initial values 155
 Model Builder nodes 83
 model identifiers 84
 physics interface nodes 148
 plots, creating 1287
 plots, generating 1020
 quality settings 1258
 resetting expand sections 103
 resetting sections to display 132
 resetting show settings 102
 solvers, showing 1421
 study types in equation views 611
 unit systems 179
 user name and password 1414
 views 365

default solver 1079
defining
 cross-section plots 1294
 data sets 1264
 derived values and tables 1350
 functions 262
 global parameters 211
 isotropic materials 849
 parameters 211

plots 1286
definitions (node), local and global 208
deformation (node) 1347
deformed configuration (node) 507
deformed configurations, remeshing 988
deformed geometry
 vs. moving mesh 976
deformed geometry interface 997
degeneracies, in geometry 441
degree, Bézier curve and 476
degrees of freedom, analysis case 80
Delaunay algorithm 527
delete entities (node), meshes 509
deleting
 geometric entities 431, 508
 geometry sequences 415
 interior boundaries, example 453
 mesh features 472
 nodes 86
 physics 616
 recovery files 91
 restoring a material library 559
 solvers 1019, 1079
 tables 1352
 undoing 53
DEM data 264
dependent variables
 entering initial values 155
 reviewing 42
 shape functions 151
dependent variables (node) 1119
depends operator 229
derivative recovery 1258–1259
derived value (node) 1222
derived values
 global evaluations 1357
 global matrix evaluations 1357
 point evaluation 1357
volume, surface, and line averages 1354, 1356
volume, surface, and line integration 1355
descriptions, for physics interface variables 150
desktop layouts, setting 36
dest operator 228
destination maps 286
destination selection (node) 907
destination, for model couplings 283
detaching windows 35
diagonal components, materials 154
dielectric losses (material node) 574
dielectrics and perfect conductors 623
difference (node)
 geometry 509
 selections 375, 383
differentiation operators 227
diffuse light 392
diffusion (node) 722
diffusion method, for curvilinear coordinates 955
DIN number 561
dipole source (node) 702
dipole sources 690
direct (node) 1140
direct solvers
 MUMPS 1143
 PARDISO 1144
 SPOOLES 1145
directional light (node) 393
Dirichlet boundary condition 156
Dirichlet boundary condition (node) 904
Dirichlet conditions 163
disabling
 geometric entities 508
 geometry sequences 415

mesh features 472
solvers 1019

disco color tables 1251

disconnecting
from MATLAB 1414
from servers 1415

disconnecting from servers 92

discontinuous functions 281

discontinuous Galerkin (DG) 909

discontinuous Galerkin FEM 865

discontinuous Galerkin formulation 747

discontinuous Galerkin method 1117

discrete states (node) 945

discretization 607

discretization (node) 934

discretization section, expanding 133

disk geometry 478

disk operators 226

dispersive materials 622

displacement gradients 847

displaying
equations 149, 610
precision settings 100
sensitivity analysis 966
user defined views 365

distance equation (node) 949

distributed impedance (node) 664

distributed loads, theory 853

distributed memory mode
cluster license handling 1422
floating network license and 1420
Linux and Windows 1417

distributed ODE (node) 943

distributed ODEs and DAEs 937

distributed parameter sweeps 1419

distribution (node) 537

divergence 865

documentation 10

documentation root directory
specifying on Linux 1439
specifying on Mac OS X 1454
specifying on Windows 1427

documentation window 65

domain decomposition (node) 1146

domain heat source variable 775

domain material 554

domain nodes
deformed geometry 998
electric currents interface 656
electrostatics interface 639
heat transfer 786
magnetic fields interface 673
moving mesh 992
pressure acoustics, frequency domain
interface 700
single-phase flow 753
solid mechanics interface 833
transport of diluted species 721

domain point probe (node) 326

domain point probes 324

domain probe (node) 325

domain selection 608

domain solver (node) 1149

domain type 565

domains
associative geometry 441
coordinate systems 305
definition 349
hiding 369
splitting 427
variables, and 212

dotted line, under parts of equations 610

double dogleg solver 1025, 1150, 1185

double dot product 733

down operator 238

downloading

- model meshes and solutions 68
- d**rag force 748
- drag-and-drop
 - to copy nodes 51
 - to move nodes 50
- drop rules 1156
- drop tolerance 1156
- Drucker-Prager (material node) 581
- Duhamel-Hooke's law 847
- duplicates, and data export 1365
- duplicating. *see* copying.
- DXF files 514
- E**ccentric cone (node) 482
- Eclipse Help plug-in, for documentation 1382
- edge (node), meshes 524
- edge 2D (node) (data set) 1273
- edge 3D (node) (data set) 1273
- edge groups (node) 539
- edge load (node) 840
- edge map (node)
 - meshes 540
 - model couplings 295
- edge nodes
 - electric currents interface 656
 - electrostatics interface 639
 - heat transfer 786
 - pressure acoustics, frequency domain interface 700
 - solid mechanics interface 833
- edge probe (node) 325
- edge selection 608
- edge source (node) 902
- edge tolerance 1330
- edges, definition 349
- edit object (node) 510
- editing
 - analysis cases 80
- constraints 150
- default fonts 36
- Desktop language 37
- Desktop layout 35
- equations 149
- geometry objects 510
- language on GUI 37
- material properties 570
- Model Library root folder 119
- node properties 93
- nodes 410
- reports 1380
- results tables 1352
- solver configurations 1019
- variables 150
- weak-form expressions 150
- effective volumetric heat capacity 789
- eigenfrequency study
 - pressure acoustics 694
 - solid mechanics 857
- eigenfrequency study (node) 1037
- eigenvalue
 - initial values 155
 - preset studies 166
 - problem solving 877
 - variables 243
 - with operator and 239
- eigenvalue linearization point 878
- eigenvalue shift 1142
- eigenvalue solver (node) 1090
- eigenvalue study (node) 1039
- eigenvalues 1086
- eigenvectors, scaling of 1091
- Eikonal equation 948, 951
- elastic material properties 835
- elastic moduli 849
- elasticity matrix 849
- elasticity method, for curvilinear coordi-

nates 956
elastoplastic material model (material node) 578
elastoresistance form (material node) 576
electric currents interface 655
 theory 630
electric displacement field (node) 647
electric fields theory 625
electric insulation (node) 662
electric potential (node) 645
 electric potential, constraint 157
electrode potential (material node) 574
electrolyte conductivity (material node) 574
electromagnetic heat source (node) 814
electromagnetic units 179
electrostatic units 180
electrostatics interface 638
 theory 628
element filters 1257
element order 134
element orders and types 892
element Péclet number 197
element quality 515–516
element size 515
element volume, relative 990
elements, shrinking in plots 1257
elevation function (node) 264
eliminated stiffness matrix 1136
elimination constraint handling 1136
elimination of constraints 1136
ellipse 477
ellipse (node) 483
ellipsoid (node) 484
elliptic PDEs 1142
elliptic problems 1142
emailing COMSOL 11
EMU unit systems 179
 special units in 190
enabling
 geometry sequences 415
 mesh features 472
encryption, for TCP connection 1457
end caps for helix 486
entering
 initial values for dependent variables 155
 model names 119
 parameters 211
 ranges and vector-valued expressions 253
 shape function variables 250
 values for anisotropic materials 154
env operator 229
equation residual 137
equation section, expanding 133
equation settings 153
equation view 149, 607
equation-based modeling 862
equations
 displaying 149, 610
 editing 149–150
 viewing 149, 610
error (node)
 geometry 415
 solvers 1080
error estimate
 absolute tolerances and 1109
 functional-based 1130
 using L2 norm 1130
error indicators 1126
errors and warnings 71
estimating memory use 123
ESU unit systems 180
 special units in 191

- Euclidean norm
for the relative tolerance 1102–1103
in the threshold drop rule 1156
join data set, and 1275
Euler angles 313–314
Eulerian formulation 977
evaluating
 data in Lagrange or Gauss points 1372
 results with expressions 1242
 results with special operators 222
 tables 1361
events interface 945
exclusive nodes 145
expand (node) 530
expand sections button 132
expanding
 all nodes 52
 sections 133
expanding sections 607
explicit (node) 371
explicit event (node) 946
explicit events 945
explicit selection (node) 379
exponential filter, for wave problems 918
export to file (node) 1222
exporting
 animations 1365
 data from data sets 1370
 images 1373
 job configurations 1222
 mesh data 1372
models 1406
models from the server 1415
models to servers 92
players 1375
plot data 1376
series of images 1365
table data 1373
- expressions
building 209
for physics interface variables 150
harmonic perturbation, and 1262
results, and 1242
exterior boundaries 159
external class (node) 1221
external current density (node)
 electric currents interface 661
 magnetic fields interface 677
external function (node) 265
external process (node) 1225
external process window 70
external surface charge accumulation
 (node) 646
extrude (node), work planes 511
extrusion coupling operators 284
- F** faces, definition 349
fallback conditions 318
far field (node), plots 1316
far field variables 1316
fast Fourier transform 1318
Fatemi-Socie (material node) 580
F-cycles 1172
FFT 1308, 1318
FGMRES iterative method 1158
FGMRES solver 1162
field (node) 1149
file formats 89
 NASTRAN bulk data 510
 STL files 515
 VRML files 515
file locking 89
filenames of generated files 1380
fillet (node) 513
fill-ratio drop rule 1156
filter (node), plots 1348
filtering elements 1257

finalization methods, for geometry 436
finalize (node) 436
finalized geometry 435, 441
Findley (material node) 580
finite element method 865
firewall security warnings 65
first law of thermodynamics 768
fixed constraint (node) 841
fixed mesh (node)
 deformed geometry 998
 moving mesh 992
Flash file formats 1365
flow continuity (node) 765
flow method, for curvilinear coordinates
 956
fluid flow
 Mach number 735
 particle tracing 747
 single-phase theory 732
fluid models, pressure acoustics 701
fluid properties (node) 754
flux (node) 725
flux conditions 156
flux discontinuity (node) 727
flux vectors 868, 914
flux/source (node)
 mathematics interfaces 906
 wave form pde interface 920
fluxes, computing 168
fonts, default for plots 36
foot-pound-second unit systems 180
form assembly (node) 436
form composite edges (node) 535
form composite faces (node) 535–536
form union (node) 436
forward methods 1188
forward sensitivity 964, 966
Fourier's law 769
FPS unit systems 180
 special units in 192
frame time derivative 983
frames
 deformed meshes and 978
 for derived values 1350
 moving, heat transfer 816
 overview of 978
free (node) 843
free deformation (node)
 deformed geometry 1000
 moving mesh 993
free meshing 480
 troubleshooting 485
free quad (node) 524
free tetrahedral (node) 525
free triangular (node) 526
frequency domain (node) 1040
frequency domain modal (node) 1041
frequency domain study
 pressure acoustics 692
 solid mechanics 856
frequency domain, perturbation (node)
 1059
frequency spectrum 1308, 1318
frequency variables 242
frozen rotor (node) 1078
frustrum, cone 482
fsens operator 229
full MPH-files 88
fully coupled (node) 1150
function 1D, 2D, and 3D (nodes)
 data sets 1273
function names, reserved 262
functional-based error estimate 1130
functions
 adding to materials 586
 defining 262

dimensions of inputs/outputs 184
global and local 262
input and output dimensions 184
units for inputs/outputs 263

G Galerkin constraints 797
Galerkin formulation 747
Galerkin least-squares (GLS) 203, 737
Galerkin method 197
gallons 188
Gao (material node) 579
gauge transformation and fixing 633
Gauss' law and charge relaxation theory
 625
Gauss' law equation 628
Gaussian pulse function (node) 266
Gauss-Seidel method 1189
GCC libraries 1440
general extrusion coupling operators 291
general form PDE (node) 901
general form PDE user interface 895
general projection (node) 297
general stress (boundary stress condition) 762
generalized alpha solver, BDF solver and
 1107
generalized constraint force 888
generalized minimum residual solver
 1162
generalized-alpha solver 1106
generating arrays 254
generating default plots 1020
Gent (material node) 579
geometric entities
 buttons 352
 context menu layout 142
 coordinate systems and 305
 definition 349
 hiding and showing 368
selecting 350, 361
selection lists, pasting into 354
variables and 243
visualization cues 353

geometric entity selection 608
geometric multigrid solver 1170
geometric primitives 406, 417, 475
geometric scope
 definitions branch and 47
 geometric entities and 358
 materials and 563
 variables and 241
geometric variables 243
geometrical sensitivity 967
geometry
 adding nodes 408
 automatic rebuild of 435
 building nodes 408
 current node 413
 frames 978
geometry (node) 433
geometry objects
 copying and pasting 432
 editing 412
 importing 513
geometry representation 434
geometry sequence
 adding nodes 413
 definition 406
 inserting from file 437
 warnings and errors 415
geometry sequence (node) 1221
geometry shape order
 using first order 547
geometry toolbar 427–428
geometry, simplifying 442
GIF file formats 1365
global

- contribution 965
coordinate systems 444
parameters 211
variable probes 324
global (node), plots 1317
global constraint (node) 941
global control variable (node) 972
global coordinate systems 846
global definitions (node) 208
global equations (node) 939
global equations, complex variables and 940
global evaluation (node) 1357
global matrix evaluation (node) 1357
global objective (node) 972
global variable probe (node) 328
global variables 212
globally symmetric constraint 887
GMRES iterative method 1158
GMRES solver 1162
GMS. See geometric multigrid solver.
go to default view (button) 365
go to source (button) 51
gradient 865
gradient displacements 847
graphics toolbar buttons 355
graphics window 58
gravitational IPS unit systems 180
 special units 193
gray-scale color tables 1251
Greek characters 1246
Green strains 851
Green-Lagrange strains 851
Gregorian year 188
grid data formats 272, 1371
ground (node) 644
guess variables 338
guidelines, for modeling 122
- H**armonic loads 856
harmonic perturbation plots 1262
HB curve (material node) 575
headlight (node) 399
heat equation (node) 903
heat flux (node) 798
heat flux, computing 170
heat flux, flux condition 157
heat flux, theory 771
heat source (node) 793
heat sources
 defining as total power 794, 802, 805
 line and point 805
heat transfer in fluids (node) 790
heat transfer in solids (node) 786
heat transfer interfaces 783
 theory 768
Heat Transfer Module 787
 additional settings 808
 Newtonian fluids, and 733
 slip condition, and 743
Heaviside functions, smoothed 281
height expression (node) 1349
helix (node) 485
Helmholtz equation (node) 903
Helmholtz's theorem 633
help
 buttons 67
 displaying in browser 101
 files, preloading 65, 101
 searching 63
 topics, searching 66
help (button) 64
help mode 101
help window 62, 65
Hermitian matrices 1133
hexahedral elements, meshes 468
hexahedron (node) 488

- hidden objects, showing in plots 1311
- hide button 607
- hide geometry objects (button) 369
- hide geometry objects (node) 400
- hide selected (button) 369
- hiding geometric entities 368
- hiding the logo 109
- highlighting, of geometry objects 409
- histogram (node) 1319
- histogram, mesh quality 516
- Hoek-Brown (material node) 581
- host computer 1407
- HTML editors 1380
- HTML format
 - for documentation 1381
 - tags 1246
- HTTP secure proxy settings 115
- hybrid modeling 406
- hyperbola 477
- hyperelastic smoothing 984
- I**
 - icons, key 78
 - IDA 1106
 - ideal gas (material node) 575
 - identifiers, for nodes 97
 - identity mapping (node) 296
 - identity pair (node) 320
 - identity pairs 318
 - if operator 229
 - ignore edges (node) 534
 - ignore faces (node) 534
 - ignore vertices (node) 533
 - image (node) 1373
 - image function (node) 267
 - image snapshot (button) 38, 402
 - images
 - model thumbnail 94
 - of models, saving 91
 - plot animations 1365
 - screenshots, capturing 402
- imaginary parts, complex variables 940
- impedance (node) 705
- implicit event (node) 947
- implicit events 945
- implicitly restarted Arnoldi method 1090
- import (node) 513
- importing
 - 2D geometries from DXF files 407
 - 3D geometries from STL and VRML files 407
 - CAD files 407
 - data to tables 1362
 - file data sources, example 270
 - function data 269
 - geometric operations 419
 - geometry objects 513
 - global parameters 211
 - interpolation curves to tables 489, 495
 - material libraries 557
 - mesh operations 502
 - meshes 501, 510
 - models from a server 92
 - models, from the server 1415
 - models, to the server 1406
 - NASTRAN files with pyramid elements 469
 - variables 212
 - virtual geometry 533
- imprints, of parts in assembly 436
- inch-pound-second unit systems 180
- incident pressure field (node) 708
- incomplete LU (node) 1154
- incompressible flow theory 736
- inconsistent stabilization methods 201
- inconsistent stabilization settings 608
- inconsistent units 196
- independent variables 444

indicator states (node) 946
infinite element domain (node) 344
inflection point, for cubic curves 477
inflow (node) 726
information (node), solvers 1080
inheritance and variables 252
inhomogeneous materials 622
ini file
 Linux 1436
 Macintosh 1453
 Windows 1425
initial values
 for nonlinear solvers 1102
 problem types 938
 specifying 155
initial values (node)
 acpr interface 702
 electric currents interface 660
 electrostatics interface 642
 heat transfer 793
 joule heating 814
 magnetic fields interface 676
 PDE interfaces 899
 pressure acoustics, frequency domain
 interface 702
 single-phase, laminar flow 755
 solid mechanics interface 837
 transport of diluted species 723
 wall distance interface 950
initializing a COMSOL Multiphysics server 1414
inlet (node) 758, 957
inlet boundary condition, theory 744
input properties 584
inputs, materials 569
insert expression (button) 1242
inserting
 boundary layer splits 488
element splits, meshes 521
expressions and predefined quantities 1242
geometric entities into selection lists 354
geometry sequences 437
variables into expressions 1242
installation folder, LiveLink 116
integer range 255
integral
 contribution 965
integral (node) (data set) 1267
integral objective 970
integral transforms 228
integral, data series and 1261
integration 1268
integration (node) 299
integration coupling operators 285
integration order 1268
Intel MKL 1144
Intel MPI library 1449
interactive plotting
 cross-section plots 1300, 1302
 isosurfaces 1321
 slice plots 1338
interior boundaries
 definition 159
 removing 423
interior flux (node) 921
interior sound hard boundary (wall)
 (node) 709
interior source (node) 921
intermediate meshes 286
internal degrees of freedom 1024
internally symmetric physics 888
Internet resources 10
interpolated solutions 1035
interpolation 1254

- interpolation curve (node) 488
 - interpolation function (node) 268
 - interpolation functions, materials and 586
 - interpolation, splines 273
 - intersecting geometry, example 449
 - intersection (node)
 - geometry 516
 - selections 375
 - intersection selection (node) 383
 - interval (node) 490
 - interval geometry for 1D models 490
 - inverse functions 269
 - inverted mesh elements 244, 546
 - IPS unit systems 180
 - special units in 192
 - IPv6 1426, 1437, 1453
 - isdefined variable 223
 - isinf operator 230
 - islinear operator 230
 - isnan operator 230
 - isosurface (node)
 - data sets 1274
 - plots 1321
 - isotropic diffusion, in fluid flow 737
 - isotropic diffusion, inconsistent stabilization methods 137, 782
 - isotropic diffusion, stabilization methods 201
 - isotropic materials
 - defining 849
 - elastic properties 835
 - modeling 154
 - iterative (node) 1157
 - iterative algorithms, adaptive solver and 1126
 - iterative solvers
 - conjugate gradients 1163
 - FGMRES 1162
 - GMRES 1162
 - modeling, and 163
 - preconditioners for 1159
 - Vanka 1203
- J**
- jacdepends operator 230
 - Jacobi (node) 1164
 - Jacobian 1030
 - matrix 163, 1104
 - symmetric 1133
 - Java-files 89
 - job (node) 1220
 - job configurations 1018
 - job configurations (node) 1218
 - job scheduler 1051, 1228
 - join (node) 1274
 - join entities (node) 512
 - Joule heating 807
 - Joule heating interface 807
 - Joule heating model (node) 812
 - jump (node) 958
- K**
- key to icons 78
 - keyboard shortcuts 36, 74
 - Khan and Richardson drag force 748
 - Khan-Richardson model 747
 - knot points 491, 493, 526
 - knowledge base, COMSOL 12
 - Krylov preconditioner (node) 1165
- L**
- L2 norm error estimate 1130
 - Lade-Duncan (material node) 581
 - Lagrange element variables 250
 - Lagrange multipliers
 - boundary conditions, and 883
 - interpretations of 169
 - variables for 252
 - weak constraints, and 162
 - Lagrangian formulations 847, 977

Lamé parameters 849
Lamé parameters (material node) 577
laminar flow
 Reynolds number, and 736
laminar flow interface 750
 theory 732
languages, for COMSOL Desktop 37
LAPACK 911
Laplace smoothing 984
Laplace's equation (node) 903
large deformations 851
LaTeX symbols 1388
Lax-Friedrichs flux 910, 917
leaking wall, wall boundary condition 757
legends, position 1310
length unit, for geometry 434
level labels 1314
licenses
 borrowing 96
 display and use of 96
 products list 73
light sources and attributes 391
light sources, hiding 389
lighting models 567
lindev operator 231
line (node), plots 1322
line average (node) 1354
line charge (node) 649
line charge (on axis) (node) 650
line charge (out-of-plane) (node) 651
line current (out-of-plane) (node) 683
line current source (node) 668
line current source (on axis) (node) 668
line graph (node) 1321
line heat source (node) 805
line heat source variable 775
line integration (node) 1355
line maximum and minimum (nodes) 1356
line plots, maximum and minimum 1323
linear buckling (node) 1064
linear elastic material (node) 834
linear elastic materials 847
linear extrusion coupling operators 292
linear projection (node) 298
linear system solvers 1141
 selecting 1141
linear viscoelastic material (node) 578
linearization point
 AWE solver 1088
 model solver 1095
 solutions, and 1243
 stationary solver 1101
linearized resistivity (material node) 574
linper operator 231
linpoint operator 231
linsol operator 231
lintotal operator 231
lintotalavg operator 231
lintotalpeak operator 232
lintotalrms operator 232
Linux
 ini file 1436
Linux clusters 1407, 1417
linzero operator 232
liquids and gases materials 591
LiveLink for MATLAB 1409
load cases
 creating groups for 175
 defining 1034
 definition 173
load group (node) 175
load groups 80, 173
loads
 solid mechanics theory 853
local

- CFL number 739
- definitions 208
- mesh size 515
- properties, materials 570
- variables 212
- local CFL number 752
- local coordinate systems 846
- location
 - max and min, coupling operators 301
 - maximum and minimum plots 1323
- lock (button) 1294
- locked MPH-files 89
- locking plot windows 1294
- log scale, for graphs 1308
- log window
 - definition 60
 - size, setting 101
 - solvers 1022
- logarithmic variables 131
- logical expression
 - for splitting geometry 513
- logical expression (node), meshes 513
- login information 1429, 1442, 1456
- logo, showing and hiding 109
- logs, from solver 1024
- look and feel 102
- loss tangent (material node) 575
- Lotka-Volterra equations 936
- lower gradient limits, crosswind diffusion
 - and 137
- lower limit (node) 1167
- lumped parameters, transformation of
 - 1358
- lumped step (node) 1167
- M**
 - Mach number
 - definition, single-phase flow 735
 - Macintosh
 - ini file 1453
- magnetic field (node) 680
- magnetic fields interface 671
- theory 636
- magnetic insulation (node) 679
- magnetic losses (material node) 575
- magnetic potential (node) 681
- magnitude controlled 1343
- main menu and toolbar 39
- mapped (node) 528
- mapped coordinate systems 318
- mapped infinite elements 342
- mapped meshing 487
- mapped system (node) 312
- mass participation factors 1092
- master meshing sequence 472, 501
- Matake (material node) 580
- Material Browser 555
 - overview 561
- material coordinates 245, 845
- material frame
 - heat transfer 820
- material frames 978, 980
- Material Library 551, 557
- material properties 153, 550, 562
 - referencing 240
- material property groups 550
- materials 622
 - adding functions 586
 - batteries and fuel cells 595–596
 - databases 551
 - definitions 550
 - domain, default 554
 - geometric scope and 563
 - linear elastic 847
 - liquids and gases 591
 - local properties 570
 - MEMS 592
 - output properties 569, 571

piezoelectric 593
piezoresistivity 594
properties, evaluating and plotting 554
searching 561
showing 109
status 565
used in model 560
math symbols 1247, 1388
mathematical constants 218
mathematical functions 219
mathematics, user interfaces for 862
MATLAB 1409, 1415
MATLAB function (node) 275
MATLAB installation folder 116
matrix histogram (node) 1322
matrix symmetry 1133
Matsuoka-Nakai (material node) 581
MatWeb, importing material data from 558
max/min volume, surface, and line (nodes) 1323
maximize (button) 35
maximum (node), data sets 1276
maximum and minimum derived values 1356 location of 301 plot values 1323
maximum coupling operators 285
maximum model coupling (node) 301
maximum number of groups, for clusters 1177
maximum, of data series 1261
Maxwell's equations dielectrics 623 quasi-static approximation 632 theory 619
mean effective thermal conductivity 788
mean effective thermal diffusivity 788
mean operator 232, 238
measurements (node) 431
mechanisms of heat transfer 768
memory use batch mode 1409 estimating 123 example 97
memory, settings for 114
MEMS materials 592
MEMS Module 808
merge edges (node) 538
merge vertices (node) 538
merged vertex 537
mesh (node) data sets 1276 exports feature 1372 plots 1324
mesh control domains (node) 541
mesh control edges (node) 539
mesh control entity edge 534 face 535 vertex 533
mesh control faces (node) 540
mesh control operations 539
mesh control vertices (node) 539
mesh element quality 515
mesh element size 515
mesh elements 468
mesh frames 978, 980
mesh movements 981
mesh object (node) 514
mesh time derivative 983
mesh vertices 468
meshelement variable and duplicates 1365
meshes boundary layers 487

clearing 472
creating 468
distribution, specifying 484
filter elements in plots 1257
importing 501
operations on imported meshes 502
running COMSOL in parallel 1416
size, specifying 482
smoothing methods 984
stop conditions for quality 987
studies, selecting 1033
techniques 477
troubleshooting 485
meshing sequence (node) 1221
message passing interface 1434, 1449
messages window 59
method of lines 1106
M-files 89
Microfluidics Module 733
slip condition, and 743
Microsoft Visual Studio 2010 1433
minimize (button) 35
minimum (node), data sets 1276
minimum coupling operators 285
minimum model coupling (node) 301
minimum, of data series 1261
mirror (node), geometry 517
mirror 2D (node) data set 1276
mirror 3D (node) data set 1276
MKL (Math Kernel Library) 1417, 1427,
 1440, 1455
modal reduced matrices 1096
modal solver (node) 1092
mode analysis (node) 1065
Model Builder
 COMSOL Desktop 34
 context help and 64
 context menus and 48
data set nodes, adding 1264
definitions nodes 208
keyboard shortcuts 75
layout 45
model nodes and 83
moving nodes 50
nodes 77
nodes, adding 48
physics interfaces, adding 614
sequencing and 85
study nodes, adding 1004
viewing names and tags 97
model couplings 283
model documentation
 opening 119
 searching 62
model equation settings 153
model history, resetting 91
model identifier 93
model inputs 615
 for material properties 550
 multiphysics couplings 615
 multiphysics user interfaces 606
Model Java-files 89
model Java-files 89, 1408
Model Library 11
 adding customized folder 121
 models with no stored solution 88
 MPH-file types in 88
 opening 68
 root folder, editing 119
 saving models 90
 toolbar 69
 updating 120
 using 118
Model Library examples
 acpr interface 700
 analytic functions 264

average model couplings 300
boundary and domain probes 326
boundary coordinate systems 310
boundary layer mesh 488
contact impedance 666
contact pairs 323
cut line 3D data set 1270
cut plane data sets 1271
cut point data sets 1272
cylindrical coordinate systems 312
deformed geometry 998
domain point probes and point probes
 327
electric currents interface 656
electrostatics interface 639
event interface 945
free triangular mesh 482
function data sets 1274
Gaussian pulse functions 266
general extrusion model couplings 292
global variable probes 329
global variables 213
harmonic perturbation 1215
heat transfer in fluids 790
heat transfer in solids 789
identity mapping model coupling 296
identity pairs 322
integral and average evaluation data
 sets 1268
integration model couplings 300
interpolation functions 274
Joule heating 810
laminar flow 753
line average derived values 1354
line integration 1356
linear extrusion model coupling 293
local variables 213
magnetic fields interface 673
mapped mesh 487
maximum and minimum model cou-
 plings 301
mirror data sets 1277
multiphysics 613
parameterized curve data sets 1278
parametric extrusion data sets 1279
PDEs 891
perfectly matched layers 341
periodic boundary condition 165
piecewise functions 277
plane wave radiation 707
PMLs 341
pressure acoustics, frequency domain
 interface 700
rotated coordinate systems 315
selecting physics and variables in study
 steps 1018
sensitivity 970
solid mechanics interface 833
step functions 279
surface integration 1355
swept mesh 489, 493
thermodynamics 789
transport of diluted species 721
volume integration 1355
volume maximum derived values 1357
waveform functions 280
Model Library root directory
 specifying on Linux 1439
 specifying on Mac OS X 1454
 specifying on Windows 1427
model M-files 89
model reduced order model (node) 1066
model reduction 1092
model thumbnails
 displaying 95
root node 94

saving 91

model tree 45

- branches 46
- expanding and collapsing 52

Model Wizard

- coordinate systems and 444
- layout 41
- mathematics user interfaces 862
- models, building 43
- physics user interfaces 601
- study types, adding 1004
- using 82

models

- guidelines for creating 122
- images, saving 91
- importing and exporting 92
- saving 90
- thumbnail image of 95

modules, COMSOL Multiphysics 4

Mohr-Coulomb (material node) 581

monopole source (node) 702

monopole sources 690

Mooney-Rivlin (material node) 579

move (node) 518

moving

- between windows 36
- nodes 50
- windows 35

moving frames 816

moving mesh

- vs. deformed geometry 976

moving mesh interface 990

moving wall, wall boundary condition 757

MPa unit systems 180

- special units in 193

MPD 1446

MPH-files 11, 88

- locked 89

MPI 1407

MPI library 1434

multigrid (node) 1168

multigrid cycles

- F-, V-, and W-cycles 1172

multigrid level (node) 1042

multigrid levels, solvers and 1168

multigrid solver 1170

multiparameter sweeps 1043, 1047, 1055

multiphysics couplings 615

multiphysics modeling 606, 613

multiprocessing daemon 1446

multisampling 108

multislice plot (node) 1325

MUMPS solver 1141, 1143, 1416

Murnaghan (material node) 579

N

nabla 865

naming

- models 119
- nodes 97
- physics interface variables 150
- spatial coordinates 84
- variables 240

NASTRAN files 510

- creating domain information from 511

natural boundary condition 880

nautical mile 188

Navier-Stokes equations 732

navigating the COMSOL Desktop 36

nested parametric sweeps 1043, 1047, 1055

Neumann boundary condition 156

new from physics builder 112

new material library (button) 558

Newton iterations

- termination of 1153

Newton method 1101, 1150

niterCMP variable 739

- no flux (node) 724
- no slip, wall boundary condition 756
- no sorting, nodes 54
- no viscous stress (outlet boundary condition) 760
- nodal discontinuous Lagrange shape functions 909
- nodes
 - contributing type 145
 - copying and duplicating 51
 - deleting 86
 - enable and disable 148
 - equation view 149
 - equations, displaying 610
 - exclusive type 145
 - identifiers 97
 - key to the Model Builder 77
 - Model Builder and 45
 - Model Builder, adding 48, 85
 - models, defined 83
 - moving 50
 - names and tags, viewing 97
 - overridden selections 148
 - physics interface defaults 141, 148
 - properties 93
 - properties, editing 93
 - reaction forces and 304
 - referenced source, going to 51
 - renaming 93
 - status 147
 - study types 1004
- nojac operator 232
- non-conservative formulations 715
- non-Latin characters, in plots 36
- nonlinear materials 622
- nonlinear problems, settings 1185
- nonlinear solver 1104
- nonlinear solver algorithms 1101
- nonlinear stationary solver 1086
- non-Newtonian fluids 733
- non-overlapping parts, pairs 318
- nonstandard shape PMLs 338
- norm of difference 1275
- normal acceleration (node) 703
- normal conductive heat flux variable 775
- normal convective heat flux variable 775
- normal current density (node) 663
- normal current density, flux condition 157
- normal distribution, random functions 278
- normal stress 851
- normal stress (boundary condition) 759
- normal stress (material node) 581
- normal stress, normal flow (boundary stress condition) 763
- normal total energy flux variable 775
- normal translational heat flux variable 775
- normal variables 248
- not-a-number, evaluating 1243
- notational conventions, for PDEs 865
- number of degrees of freedom 243, 1212
- numerical constants 218
- numerical diffusion 200
- numerical results, displaying directly 1350
- numerical stabilization techniques 197
- numerical viscosity 201
- NURBS 526
- Nyquist criterion 1173
- Nyquist plot (node) 1326
- objective function 962, 969
 - complex-valued 968
 - specification of 965
- oblique cone 482

obliquely incident waves 337
ODE solver settings 1328
ODEs and DAEs user interface 938
ODEs, distributed 937
of particle trajectories 1254
OGS/GE 1051, 1228
Ohm's law and charge relaxation theory
 625
ohmic heating 807
one-point map (node)
 meshes 540
 model couplings 295
open boundary (node)
 single-phase flow 762
Open Grid Scheduler/Grid Engine 1228
open grid scheduler/grid engine 1051
open PDF documentation (button) 69
opening
 Model Library 118
 recovery files 91
operation features, solvers 1081, 1086
operations on imported meshes 502
operations, undoing and redoing 53
operators
 adj 225
 bdf 226
 centroid 227
 circle, disk, sphere, ball 226
 circumcenter 227
 d 227
 depends 229
 dest 228
 down 238
 dtang 227
 env and bndenv 229
 fsens 229
 if 229
 isinf 230
 islinear 230
 isnan 230
 jacdepends 230
 lindev 231
 linper 231
 linpoint 231
 linsol 231
 lintotal 231
 lintotalavg 231
 lintotalpeak 232
 lintotalrms 232
 linzero 232
 mean 232, 238
 nojac 232
 pd 227
 ppr 233
 pprint 233
 prev 236
 reacf 234
 realdot 234
 sens 236
 shapeorder 236
 side 236
 spatial at 236
 subst 236
 test 233
 timeavg 230, 237
 timeint 237
 try_catch 238
 up 238
 var 239
 with 239
operators, applying to data series 1350
optimization solver (node) 1099
optimizing the display 108
order of elements 892
orthographic projection 390
orthographic scale 391

- orthotropic (material node) 577
- orthotropic materials, coordinate systems and 305
- orthotropic, Voigt notation (material node) 577
- oscillations in the solution 197
- Ottosen (material node) 581
- outflow (node)
 - heat transfer 797
 - transport of diluted species 727
- outlet (node) 759, 958
- outlet boundary condition, theory 744
- out-of-core MUMPS solver 1143
- out-of-core PARDISO solver 1145
- out-of-plane wave number 693
- output properties, materials 569, 571
- overlapping Schwartz method 1146
- overridden nodes 148
- override and contribution 607–608
- override and contribution section, expanding 133
- overrides, listings of 146
- owner mode, for multiprocessor usage 1427, 1440, 1454
- P**
 - Padé expansions 1088
 - pair boundary heat source (node) 801
 - pair contact impedance (node) 665
 - pair nodes
 - electric currents interface 656
 - electrostatics interface 639
 - heat transfer 786
 - magnetic fields interface 673
 - pressure acoustics, frequency domain interface 700
 - single-phase flow 753
 - solid mechanics interface 833
 - transport of diluted species 721
 - pair selection 609
- pair thin thermally resistive layer (node) 803
- pairs 318
- parabola 477
- parabolic PDEs 1142
- parabolic problems 1142
- parallel operations 1407
- parallel solver 1144
- parallel speedup, BLAS libraries 1417
- parallel, running COMSOL in 1416
- parameter sweeps, distributed 1419
- parameter values, for solvers 1033
- parameterization variables 246
- parameterized curve 2D (node) 1277
- parameterized curve 3D (node) 1277
- parameterized surface (node) 1278
- parameters (node) 211
- parameters, infinite elements 344
- parametric (node)
 - job configurations 1219
 - solver attribute feature 1174
- parametric continuation solver 1033
- parametric curve (node) 490
- parametric extrusion 1D (node) 1279
- parametric extrusion 2D (node) 1279
- parametric results, storing 1229
- parametric solver 1032
 - list of parameter values for 1033
- parametric surface (node) 492
- parametric surfaces
 - creating 492
 - from elevation functions 265
- parametric sweep study (node) 1043
- parametric sweeps 1011
 - distributing 1054
- Parasolid 434
- PARDISO solver 1141, 1144, 1416
- partial differential equations 864

particle data set (node) 1280
particle evaluation (node) 1360
particle tracing (node) 1326
particle tracing in fluid flow 747
Particle Tracing Module (operators) 229
particle tracing with mass 747
particle tracing with mass (node) 1330
particle trajectories (node) 1333
particle trajectories, interpolation of
 1254
particle types, in plots 1334
particles, in plots 1334
partition (node) 519
paste selection (button) 359
pasting nodes 51
pasting selections 352
pathlines 1326
PDE 864
PDEs
 coefficient form 893
 elliptic 1142
 general form 895
 parabolic 1142
 stationary 1086
 time-dependent 1086
Péclet number
 local 201
 stabilization and 197
perfect conductors and dielectrics 623
perfect magnetic conductor (node) 682
perfectly matched layer (node) 340
perfectly matched layers. See PML.
performance, optimizing graphics 108
periodic boundary conditions 164, 844
periodic condition (node) 648, 709
 mathematics interfaces 907
 solid mechanics interface 844
 transport of diluted species 728
periodic flow condition (node) 763
periodic heat condition (node) 801
perspective projection 390
phase factors, pressure acoustics 693
phase portrait (node) 1334
phi 1282
physical constants 221
physical memory, size of 97
physical nodes 1407
physics
 adding sequentially 614
 selecting for study 1014
physics builder 112
physics symbols
 coordinate directions for 828
 showing 109, 826
physics user interfaces 41
 default nodes 148
 nodes, by space dimension 142
 studies, and 1014
physics-controlled meshing 477
physics-induced meshes 477
piecewise function (node) 275
piecewise functions, materials and 586
piezoelectric materials 593
piezoresistance form (material node) 576
piezoresistivity materials 594
pivot perturbation strategy in PARDISO
 1144
plane stress and strain 825, 831
plane wave radiation (node) 706
Plasma Module 627, 808
player (node) 1375
plot (node) 1376
plot group (node) 1222
plots
 arrow line 1312
 arrow surface 1313

arrow volume 1313
color expressions 1347
contour 1313
convergence 1027
coordinate systems 1314
cross-section 1298, 1305
cross-section, 2D 1300
cross-section, 3D 1302
derivative recovery 1258
functions and 262
global 1317
height expressions 1349
histograms 1319
isosurface 1321
line 1322
line graphs 1321
locking 1294
maximum and minimum 1323
mesh 1324
multislice 1325
Nyquist 1326
particle tracing 1326
particle tracing with mass 1330
particle trajectories 1333
phase portrait 1334
point graph 1336
principal stress volume and surface
 1336–1337
scatter surface and scatter volume
 1337
slice 1338
streamline 1339
surface 1345
table 1345
titles, customizing 1244
update of 1235
updating 110
volume 1347

plotting
 expressions in results 1242
 material properties 554
 probes, while solving 324
 solutions side by side 1348
 plugflow solver (node) 1099
 PMC. see perfect magnetic conductor.
 PML 335
 Poincaré map (node) 1335
 point (node), geometry 494
 point charge (node) 651
 point charge (on axis) (node) 652
 point current source (node) 669
 point current source (on axis) (node)
 669
 point evaluation (node) 1357
 point graph (node) 1336
 point heat source (node) 806
 point heat source variable 776
 point light (node) 395
 point load (node) 840
 point nodes
 electric currents interface 656
 electrostatics interface 639
 heat transfer 786
 magnetic fields interface 673
 pressure acoustics, frequency domain
 interface 700
 single-phase flow 753
 solid mechanics interface 833
 point probe expression (node) 326, 328
 point selection 608
 point source (node) 902
 points, definition 349
 pointwise constraint (node) 929
 pointwise constraints 163
 Poisson’s equation (node) 903
 Poisson’s ratio 849

Poisson's ratio (material node) 577
polar plot group (node) 1308
polygon (node) 494
polynomial coordinate stretching 337
polynomial-preserving recovery 233,
 1259
poroelastic material (node) 578
port boundary conditions 335
positive definite matrices 1142
postsmoother (node) 1177
postsmothers 1172
potentials, scalar and magnetic 621
ppr and pprint operators 233
precedence, for operators 218
precision, for grids and legends 109
preconditioners
 algebraic multigrid 1171
 geometric multigrid 1170
 selecting 1159
 selection guidelines for 1160
 solving without 1160
 Vanka 1205
predefined
 quantities and expressions 1242
 variables 240
preferences 100
prefixes, for units 181
preordering algorithms 1143
prescribed deformation (node)
 deformed geometry 1000
 moving mesh 993
prescribed displacement (node) 841
prescribed displacement, constraint 157
prescribed displacement, example 161
prescribed mesh displacement (node)
 deformed geometry 999
 moving mesh 992
prescribed mesh velocity (node)
 deformed geometry 1000
 moving mesh 994
prescribed normal mesh velocity (node)
 deformed geometry 1001
 moving mesh 994
presmoother (node) 1177
presmothers 1172
pressure (node) 705
pressure (outlet boundary condition)
 760
pressure acoustics model (node) 701
pressure acoustics, frequency domain
 (acpr) interface 696
 theory 692
pressure loads 855
pressure point constraint (node) 765
pressure, no viscous stress (inlet and
 outlet boundary conditions) 759
pressure-correction method 740
pressure-wave speed (material node)
 577
pressure-wave speeds 850
prestressed analysis 1215
prestressed analysis plots 1262
prestressed analysis, eigenfrequency
 (node) 1061
prestressed analysis, frequency domain
 (node) 1062
prev operator 236
preview
 of documents 1381
 of reports 1379
previous solution (node) 1177
principal stress surface (node) 1337
principal stress volume (node) 1336
principle of virtual work 855, 968
print (button) 37
printing plots 37

prism elements, meshes 468
 Pro/ENGINEER installation folder 116
p
 probe
 contribution 965
 probe objective 965, 971
 processors, number of 114
 product updates
 preference settings 115
 progress (button) 60
 Progress window 60, 1022
 projection coupling operators 284
 projection method for Navier-Stokes
 equations 740
 projection, perspective and orthographic
 390
 prolongation matrices 1171
 property groups 550, 568
 proxy server settings
 for product and Model Library updates
 115
 pseudo time stepping
 advanced settings 752
 laminar flow theory 738
 solvers, and 1105
 publishing a report 1380
 pyramid (node) 495
 pyramid elements, meshes 468

Q quadratic mean 1261
 quadric curves 477
 quadrilateral elements, meshes 468
 quality, optimizing graphics 108
 quasi-static approximation 632

R radiative heat, theory 779
 rainbow color tables 1249
 ramp function (node) 277
 random function (node) 278
 range (button) 254
 ranges
 format options 253
 using arrays 254
 vector-valued expressions and 255
 rank-2 tensor densities 154
 ratio of specific heats 790
 rational Bézier curves 476
 rational coordinate stretching 337
 Rayleigh damping 858
 reacf operator 234, 1268
 reaction forces
 computing 168
 constraints, and 160
 data sets and 1268
 reaction terms
 applying 888
 constraints, and 160
 reactions (node) 724
 real parts, complex variables 940
 realdot operator 234
 reassembly, manual 1134
 recovery files 91, 114
 rectangle (node) 496
 rectangle function (node) 278
 rectangular pyramid 495
 reduced electric fields (node) 1074
 reference (node) 529
 reference coordinates 245, 845
 refine (node) 530
 reflections, in sector data sets 1282
 refpnt variable 832
 refractive index (material node) 575
 regular screen displays 36
 re-initialization on boundaries (node)
 947
 re-initialization on domains (node) 947
 re-initialization on edges (node) 947
 re-initialization on points (node) 947
 relative element volume 990

relative tolerance 1328
remeshing
 a deformed mesh 986
 automatically 987
remeshing, conditions for 1138
remote shell 1051, 1227, 1229
remove from selection (button) 353
removing
 interior boundaries 423
 user model library 121
renaming nodes 93
rendering, detail 108, 367
repair tolerance
 default 435
 for DXF import 515
 for geometry 437
replace expression (button) 1242
report generator 1378
reports
 animation 1365
 data 1370
 player 1375
 plot data 1376
 plot images 1373
reset hiding (button) 369
reset selected (button) 150
resetting
 author name 95
 desktop layouts 36
 model history 91
residual vector 1135
residuals, consistent stabilization and 137
resistive heating 807
resistivity temperature coefficient
 for non-SI units 573
resizing windows 35
response surfaces 1048, 1346
restoring default preferences 100
results
 displaying while solving 1022
 tables, editing 1352
reverting to last saved file 90
reviewing dependent variables 42
revolution ID (node), data set 1281
revolution 2D (node), data set 1281
revolve (node), geometry 520
revolving axisymmetric models 1281
Reynolds number definition 736
Reynolds particle number 748
ribbons, as streamline type 1254
RMS 232, 1261
Robin boundary conditions 906
roller (node) 843
root mean square 1261
rotate (node), geometry 522
rotated system (node) 313
rotations, in sector data sets 1282
row index, in table graphs 1345
Runge-Kutta methods 911, 1117, 1331
Runge-Kutta time stepping 909, 1117
running COMSOL with MATLAB as client/server 1413

S saddle-point problems 1203
sandwich, layers and 478
save model to file (node) 1221
saving
 global parameters 211
 models 90
 recovery files 91, 114
scalar coupling operators 284
scalar density variables, frames and 819
scale (node) 541
scale (node), geometry 523
scale factor, for solution 1285
scaling system (node) 316
scatter surface and scatter volume

(nodes) 1337
scattering boundary condition, modeling and 335
scene light 391
scene light (button) 367
SCGS (node) 1178
SCGS preconditioner 1160
scheduler script 1422
Schwarz methods 1146
scope, of search 66
scoping mechanisms, for variables 240
screenshots 402
scroll lock (button) 61
search options, in Model Library 118
searching
 help 62–63, 66
 materials 561
 Model Library 68, 118
 model names 119
sectionwise data formats 271, 1364, 1371
sector 2D (node), data sets 1282
sector 3D (node), data sets 1282
sector symmetries 1282
sector symmetry (node) 667
secure shell 1051, 1227, 1229
security issues in client/server mode 1430, 1442, 1457
segregated (node) 1181
segregated iterations, substeps of 1184
segregated solution 1181
segregated step (node) 1184
select boundaries (button) 360
select box (button) 362
select objects (button) 363
selecting
 adjacent entities 376, 383
 drop rules for solvers 1156
 geometric entities 361
heat transfer interfaces 783
meshes for studies 1033
physics interfaces 42
space dimensions 41, 442, 445
study types 43
unit systems 181
selection (node), creating 377
selections
 adding to data sets 1265
 Boolean 375, 383
 copying 352
 creating 352, 371, 377, 379
 creating from geometry objects 378
 entities in enclosing ball 373, 380
 entities in enclosing box 373, 380
 entities in enclosing cylinder 373, 380
 explicit 371, 379
 not applicable 148
 pasting 352
 pasting from files or clipboard 354
self intersections, in parametric surfaces 492
sens operator 236
sensitivity (node) 1187
sensitivity analysis 962
sensitivity interface 969
sensitivity study (node) 1075
sequence of operations 85
sequences, in COMSOL 85, 144
servers
 connecting 92
 disconnecting 92, 1414–1415
 importing and exporting 1415
 initializing 1414
 saving models 92
set formula, for Boolean operations 502
set model library root (button) 69
settings window

definition 48
equation view, displaying 149
example 360
keyboard shortcut 75
layout 57
properties 94
selection lists and 359
units 183
settings windows 607
settings, cluster jobs 1050
shape function 880
shape function types 892
shape functions 909
 equations, and 151
 order of 134
variables 249
 weak constraints, and 162
shapeorder operator 236
shared memory mode 1416
shared memory parallel mode 1407
shear modulus (material node) 577
shear modulus expression 849
shear rate magnitude variable 754
shear strain 850–851
shear stress 851
shear-wave speed (material node) 577
shear-wave speeds 850
shift 1142
show button 103, 132, 607
showing
 advanced physics options 138
 advanced results options 140
 advanced study options 139
batch feature 1408
cluster computing 1419
discretization 134
equation sections 133
equation view 134
geometric entities 368
help topics 63
job configurations 1018
labels in the Model Builder 98
legends on plots 1245
logo on canvas 109
logotype on reports 1383
material texture and color 109
override and contribution 134
physics symbols 109, 826
preference settings 102
product licenses used 96
stabilization 136
weak constraints 931
weak contribution on mesh bounda-
 ries 925
work planes 461
shrinking elements 1257
side operator 236
sign functions 281
single-phase flow interface
 laminar flow 750
 theory 732
singularities in geometry 441
SIPG method 747
size (node) 542
slice plot (node) 1338
sliding wall, wall boundary condition 756
 theory 743
slip, wall boundary condition 756
 theory 742
small-signal analysis 1060, 1215
small-signal analysis plots 1262
small-signal analysis, frequency domain
 (node) 1061
smoothers
 efficiency of 1173
Vanka 1205

smoothing discontinuous functions 281
smoothing functions 262
software updates 12
solid mechanics
 prescribed displacement 841
solid mechanics interface 831
 theory 845
solution (node), data sets 1284
solution process, stop buttons 1023
solution times, for models 118
solve for, analysis case 79
solver (node) 1220
solver configurations (node) 1079
solvers
 algebraic multigrid 1171
 canceling 1024
 configurations 1004, 1011
 conjugate gradients 1163
 continue operation 1004
 deleting 1019
 disabling 1019
 events 945
 FGMRES 1162
 geometric multigrid 1170
 GMRES 1162
 hierarchical structure of 1022
 logs from 1024
 overview of 1081
 PARDISO 1144
 running configurations 1079
 sequences 1004
 SPOOLES 1145
solving
 getting results while 1022
 plotting while 324
SOR
 gauge iterative method 1191
 iterative method 1189
line iterative method 1193
vector iterative method 1195
SOR (node) 1189
SOR gauge (node) 1191
SOR line (node) 1193
SOR vector (node) 1195
sort nodes
 by space dimension 54
 by type 54
sound hard boundary (wall) (node) 703
sound soft boundary (node) 704
source maps 286
source nodes, going to 51
source term (node) 902
source terms, adding 902
source, for model couplings 283
space charge density (node) 643
space dimensions
 geometry 442
 geometry, selecting 445
 physics interface nodes 142
 selecting 41
toolbar buttons 355
units and 196
space-separated list 254
space-time derivatives 251
spatial at operators 236
spatial coordinate variables 246
spatial coordinates 845
 editing names of 84
 variables, and 245
spatial frame
 heat transfer and 820
spatial frames 978, 980
special operators, evaluating results 222
species properties (material node) 574
specific heat capacity, definition 769
specifying

mesh distribution 484
mesh sizes 482
specular exponent 567
specular highlight 567
specular light 392
spf.cellRe variable 737
spf.sr variable 754
sphere (node) 497
sphere operators 226
spherical system (node) 315
spherical wave radiation (node) 707
split (button) 427
split (node), geometry objects 524
split complex variables 940
SPOOLES solver 1141, 1416
 definition 1145
 direct node and 1145, 1156
 Incomplete LU node and 1155
spotlight (node) 397
spreadsheet data formats 271, 1371
square (node) 498
SSOR update 1206
stabilization methods 137
stabilization section, expanding 133
stabilization settings 608
stabilization techniques
 crosswind diffusion 204, 781
 Galerkin least-squares (GLS) 203
 streamline diffusion 202
 SUPG 202
standard deviation 1261
standard notation, classical PDEs 872
state (node) 1198
state space (node) 1213
stationary acceleration (node) 1199
stationary PDE problems 1086
stationary plug flow (node) 1059
stationary segregated solver algorithms 1181
stationary solver (node) 1100
stationary study 1008
stationary study (node) 1032
stationary value 1086
stationary, fluid (node) 1067
stationary, one-way coupled (node) 1067
stationary, solid (node) 1067
statistics, solvers 1211
step function (node) 279
step functions 281
sticky help 64
stiffness matrix
 correct versus incorrect 1030
 symmetric 1133
STL files, format 515
STL/VRML files 514, 510
Stokes equations 1104
stop condition 987
stop condition (node) 1200
stop sign definition, materials 565
store solution (node) 1212
storing parametric results 1229
strain 850–851
 axial symmetry 851
 engineering form 851
 shear 851
 tensor form 851
strain tensor 851
strain-charge form (material node) 576
strain-displacement relation 850
 large displacement 851
 small displacement 850
stream ribbons 1254
streamline (node) 1339
streamline diffusion
 consistent stabilization method 137
 definition 202

fluid flow 737
streamline diffusion, consistent stabilization methods 781
streamline length 1342
streamline upwind Petrov-Galerkin (SUPG) 202
stress 851
 normal 851
 shear 851
stress tensor 1336–1337
stress-charge form (material node) 576
stress-strain relation 851
studies
 including physics interfaces 1014
study (node) 1005
study steps
 adding 1010
 Model Builder, and 1004
 selecting physics and variables for 1014
study types
 eigenfrequency 857
 frequency domain, solid mechanics interface 856
 list of 1006
 selecting 43
 stationary, solid mechanics interface 856
subst operator 236
subtracting
 geometry, example 448
 objects from boolean operations 509
successive over-relaxation methods 1189
suffixes for report files 1380
summation 1268
SUPG. See streamline upwind Petrov-Galerkin
surface (node)
data sets 1285
plots 1345
surface average (node) 1354
surface charge density (node) 645
surface current (node) 681
surface integration (node) 1355
surface maximum and minimum (nodes) 1356
surface plots, maximum and minimum 1323
surface-to-ambient radiation (node) 800
sweep (node), geometry 525
swept (node) 531
swept meshes 488
swirl flow 734
symbols for physics 826
symmetric interior penalty method (SIPG) 747
symmetric Jacobian matrix 1133
symmetric matrix, materials 155
symmetric reaction terms 160
symmetric stiffness matrix 1133
symmetric successive over-relaxation method 1189
symmetrically coupled Gauss-Seidel solver 1178
symmetries, and Sector data sets 1282
symmetry (node) 761
 heat transfer 798
 transport of diluted species 727
symmetry, using when modeling 122
synchronizing, batch jobs and GUI 1049, 1057
syntax errors 196, 211
system information, showing 73
system matrix (node) 1359

T table (node), exports 1373
table data, copying 1353

table graph (node), plots 1345
table surface (node), plots 1346
tables
 buffer size 110
 clearing and deleting 1362
 displaying results directly 1350
 evaluating for derived values 1361
 settings window 61
tags, for nodes 97
tags, HTML 1246
tangent (node), geometry 527
tangent variables 247
tangential derivative variables 252
tangential direction, derivatives in 228
Taylor expansion 1088
technical support, COMSOL 11
temperature (node) 796
temperature, constraint 157
temporal averages 1350
temporary files 113
terminating distance 1342
termination criterion 1181
test functions 237
test operator 233
tetrahedral elements, meshes 468
tetrahedron (node) 499
text formatting and symbols 1246
text symbols 1247
theory
 constitutive relations 620–621
 electric currents 630
 electric fields 625
 electromagnetics 619
 electrostatics interface 628
 heat equation definition 768
 heat transfer 768
 inlet and outlet boundary conditions

744

laminar flow 732
magnetic and electric fields 632
magnetic fields interface 636
pressure acoustics, frequency domain
 (acpr) interface 692
single-phase flow 732
solid mechanics interface 845
transport of diluted species interface
 714
wall distance interface 951
wave form PDE 909
thermal color tables 1250
thermal conductivity, frames and 820
thermal conductivity, mean effective 788
thermal diffusivity 788
thermal insulation (node) 796
thermoacoustics model (material node)
 573
thermodynamics package (node) 279
thin client mode 1412
thin diffusion barrier (node) 729
thin impermeable barrier (node) 729
thin low permittivity gap (node) 648
thin thermally resistive layer (node) 803
third-degree curve 477
threshold drop rule 1156
throughput mode, for multiprocessor
 usage 1427, 1440, 1454
thumbnails, model images 91
time dependent, fluid (node) 1068
time dependent, one-way coupled
 (node) 1068
time dependent, solid (node) 1068
time derivatives
 constraints for 889
 variables and 251
time discrete (node) 1037
time discrete solver (node) 1115

time explicit solver (node) 1117
time parametric (node) 1202
time stepping, tolerances 1109
time variables 242
timeavg operator 230, 237
time-dependent adaption 1126
time-dependent modal (node) 1037
time-dependent PDEs 1086
time-dependent solver (node) 1106
time-dependent solver algorithms 1106
time-dependent study 1009
time-dependent study (node) 1034
time-dependent with initialization, fixed
 geometry (node) 1057
time-dependent, fixed geometry (node)
 1058
time-discrete solver 1037
time-harmonic study
 pressure acoustics 692
timeint operator 237
titles for plots 1244
tolerance
 absolute and relative 1109
 edge 1330
 relative 1328
toolbar
 cross-section plots 1295
 external process window 70
 geometric entities 352
 geometry 427–428
 graphics window 355
 help window 67
 main menu 39
 Model Library 69
 results tables 1352
 style 102
top hat functions 278
topics, searching for help 66
torus (node) 500
total energy flux variable 774
total force loads 854
total heat flux 799
total heat flux variable 773
total normal heat flux variable 774
total power 794, 802, 805
traction boundary conditions 762
traffic color tables 1250
transformation matrix 299
transformations, of global matrices 1358
transforms menu
 arrays 501
 copy 506
 mirror 517
 move 518
 rotate 522
 scale 523
transient with initialization (node) 1070
transient, one-way coupled (node) 1069
translation vector 299
translational heat flux variable 774
translational motion (node) 789
transparency 389
transparency (button) 367
transport mechanisms 722
transport of diluted species interface 718
 theory 714
triangle function (node) 280
triangular elements, meshes 468
try_catch operator 238
tuning parameter 137
turnaround mode, for multiprocessor
 usage 1427, 1440, 1454
twist compensation 526
two-point map (node)
 meshes 545
model couplings 295

- types, for nodes 97
- typical wave speed 336
- typographical conventions 12
- U**
 - unbounded domains, modeling 342
 - undefined quantities, evaluating 1243
 - undo last operation 53
 - unexpected units 196
 - unicode-based characters 1246
 - unidirectional constraints 164
 - uniform distribution, random functions 278
 - union (node)
 - geometry 528
 - selections 375, 383
 - union, geometry 435
 - unit circle in Nyquist plots 1326
 - unit systems
 - British engineering units 180
 - CGSA 179
 - EMU 179
 - ESU 180
 - FPS 180
 - gravitational IPS 180
 - IPS 180
 - MPa 180
 - none 181
 - selecting 181
 - SI 179
 - switching 193
 - units
 - and space dimensions 196
 - for function inputs/outputs 263
 - for PDE interfaces 893
 - of variables 150
 - scaling values when changing 434
 - space dimension and 196
 - standard prefixes for 181
 - unexpected or unknown 196
 - unphysical values 128
 - UNS number 561
 - up operator 238
 - updates
 - preference settings 115
 - updating
 - COMSOL software 12, 115
 - Model Library 116
 - plots 110
 - the Model Library 120
 - updating plots 1235
 - upside, of normals 248
 - Use a System Matrix derived values node
 - feature to evaluate a system matrix from an Assemble or Modal feature to a table 1359
 - use weak constraints check box 164
 - user community, COMSOL 12
 - user defined
 - materials 558
 - user model library 121
 - user-controlled meshes 478
 - user-defined material databases 552
 - user-defined method, for curvilinear coordinates 957
 - using
 - coordinate systems 846
 - ranges 254
 - spatial and material coordinates 845
 - the Model Wizard 82
 - utility features, solvers 1081, 1209
 - V**
 - value types
 - complex variables 136
 - global equations 940
 - values, of dependent variables 1029
 - Vanka (node) 1203
 - Vanka algorithm 1205
 - Vanka preconditioner/smooth 1160

var operator 239
Varga (material node) 579
variables
 absolute tolerance 1108
 creating 212
 editing 150
 editing definition 149
 eigenvalue 243
 expressions, inserting into 1242
 field node, solvers 1139, 1149
 for boundary fluxes 135
 for far fields 1316
 for Lagrange multipliers 252
 for material properties 554
 for spatial coordinates 246
 geometric 243
 geometric scope 241
 Lagrange element 250
 material and spatial coordinates 846
 mathematical functions 219
 meshelement 1365
 naming conventions 240
 niterCMP 739
 parameterization 246
 PDE user interfaces 867
 predefined 240
 refpnt 832
 reserved names 210
 scoping 240
 selecting for study 1014
 sensitivity solvers 1187
 shape functions 249
 solver node 1119
 spf.cellRe 737
 study types and 1029
 summary of built-in 259
 tangent and normal 247
 tangential derivative 252
time, frequency, and eigenvalues 242
vector 249
view on statistics page 1211
variance 1261
V-cycles 1172
vector elements
 preconditioners for 1161, 1195
vector-valued expressions and ranges
 253, 255
velocity (inlet and outlet boundary conditions) 758
velocity (Lorentz term) (node) 678
version numbers, adding to models 94
vertex 349
vertices, definition 349
view (node) 385
view 3D (node) 387
view all (button) 369
View menu
 equations, displaying 149
view menu 99
viewing
 deformed meshes 987
 equations 149, 610
 geometry sequences, example 454
 licenses 96
 node names and tags 97
 node properties 93
views
 2D user-defined 386
 camera position 390
 default 365
 geometry, hiding 369
 Graphics window and 365
 lighting, transparency, and wireframe
 rendering 367
 user-defined, creating 387
 zooming 365

- virtual geometry 532–533
- virtual geometry operations 441, 532
- virtual memory, size of 97
- virtual operations menu
 - collapse edges 537
 - collapse faces 537
 - form composite edges 535
 - form composite faces 535–536
 - ignore edges 534
 - ignore faces 534
 - ignore vertices 533
 - merge edges 538
 - merge vertices 538
- viscous model (material node) 573
- visual feedback icons 78
- volume (node), plots 1347
- volume average (node) 1354
- volume force (node) 754
- volume integration (node) 1355
- volume maximum and minimum (nodes)
 - 1356
- volume plots
 - filtering elements in 1257
 - maximum and minimum 1323
 - principal stress 1336–1337
- volumetric heat capacity 789
- VRML files format 515
- W**
 - wall (node) 958
 - single-phase flow 755
 - wall distance interface 950
 - wall distance initialization (node) 1071
 - wall distance interface 948
 - theory 951
 - wall, constraint 157
 - Wang-Brown (material node) 580
 - warning (node)
 - geometry 416
 - solvers 1080
- warning sign definition, materials 565
- warnings 71
 - stored in log file 1101, 1114, 1214
- wave color tables 1250
- wave equation (node) 903
- wave form PDE (node) 916
- wave form PDE interface 914
 - theory 909
- wave numbers 693
- wave speeds 836
- waveform function (node) 280
- W-cycles 1172
- weak constraint (node) 931
- weak constraint settings 608
- weak constraints 162
- weak contribution (node)
 - ODEs and DAEs 941
 - PDEs 923
- weak contribution on mesh boundaries
 - (node) 925
- weak expressions 151
- weak form PDE (node) 901
- weak form user interfaces 896
- weak-form expressions 150
- web sites, COMSOL 12
- widescreen displays 36
- window titles, customizing for plots 1310
- Windows
 - ini file 1425
- Windows Compute Cluster Server 2003
 - 1051, 1228
- Windows HPC cluster 1407
- Windows HPC edition 2008 1051, 1227
- windows, moving between 36
- Winslow smoothing 984
- wireframe rendering 108
- wireframe rendering (button) 367
- with operator 239

- work plane (node) 529
 - work plane clipping (button) 420
 - work planes 419, 456
- Y**
- year, units 188
 - Yeoh (material node) 579
 - yield stress parameters (material node)
 - 582
 - Young's modulus (material node) 577–
 - 578
 - Young's modulus expression 849
- Z**
- zenith axis, spherical coordinate systems
 - 315
 - zero charge (node) 643
 - zero flux (node)
 - mathematics interfaces 906
 - wave form PDE 920
 - zero normal mesh displacement (node)
 - deformed geometry 1002
 - moving mesh 996
 - zero normal mesh velocity (node)
 - deformed geometry 1002
 - moving mesh 995
 - zoom box (button) 365
 - zoom extents (button) 365
 - zoom selected (button) 352
 - zooming 365

