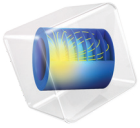


INTRODUCTION TO
COMSOL Multiphysics

5.5

COMSOL

Introduction to COMSOL Multiphysics

© 1998–2019 COMSOL

Protected by patents listed on www.comsol.com/patents, and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,457,932; 8,954,302; 9,098,106; 9,146,652; 9,323,503; 9,372,673; 9,454,625; and 10,019,544. Patents pending.

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

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

Version: COMSOL 5.5

Contact Information

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

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

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

Part number: CM010004

Contents

Introduction 5

COMSOL Desktop® 6

Example 1: Structural Analysis of a Wrench. 33

Model Wizard. 34

Geometry 36

Materials 39

Global Definitions. 40

Physics and Boundary Conditions 42

Mesh 46

Study 47

Results 48

Convergence Analysis 51

Example 2: The Busbar — A Multiphysics Model 59

Model Wizard. 61

Global Definitions. 65

Auto Completion and Find for Parameters and Variables . 66

Geometry 68

Materials 73

Physics and Boundary Conditions 77

Mesh 85

Study 87

Results 87

Building an Application with the Application Builder. 96
Advanced Topics. 110
Parameters, Functions, Variables, and Couplings 110
Material Properties and Material Libraries. 114
Adding Meshes. 116
Adding Physics 119
Parametric Sweeps. 141
Parallel Computing. 152
COMSOL Multiphysics Client-Server 156
Appendix A — Building a Geometry 159
Appendix B — Keyboard and Mouse Shortcuts 176
Appendix C — Language Elements and Reserved Names . 181
Appendix D — File Formats 195
Appendix E — Connecting with LiveLink™ Add-Ons. 202

Introduction

Read this book if you are new to COMSOL Multiphysics®. It provides an overview of the COMSOL® environment with examples that show you how to use the COMSOL Desktop® user interface and the Model Builder. It also provides a quick introduction to creating applications using the Application Builder.

If you have not yet installed the software, install it now according to the instructions at www.comsol.com/product-download.

In addition to this book, an extensive documentation set is available after installation. Tutorials can be found throughout the COMSOL website, including in the Video Gallery at www.comsol.com/videos and on the COMSOL Blog at www.comsol.com/blogs.

COMSOL Desktop®

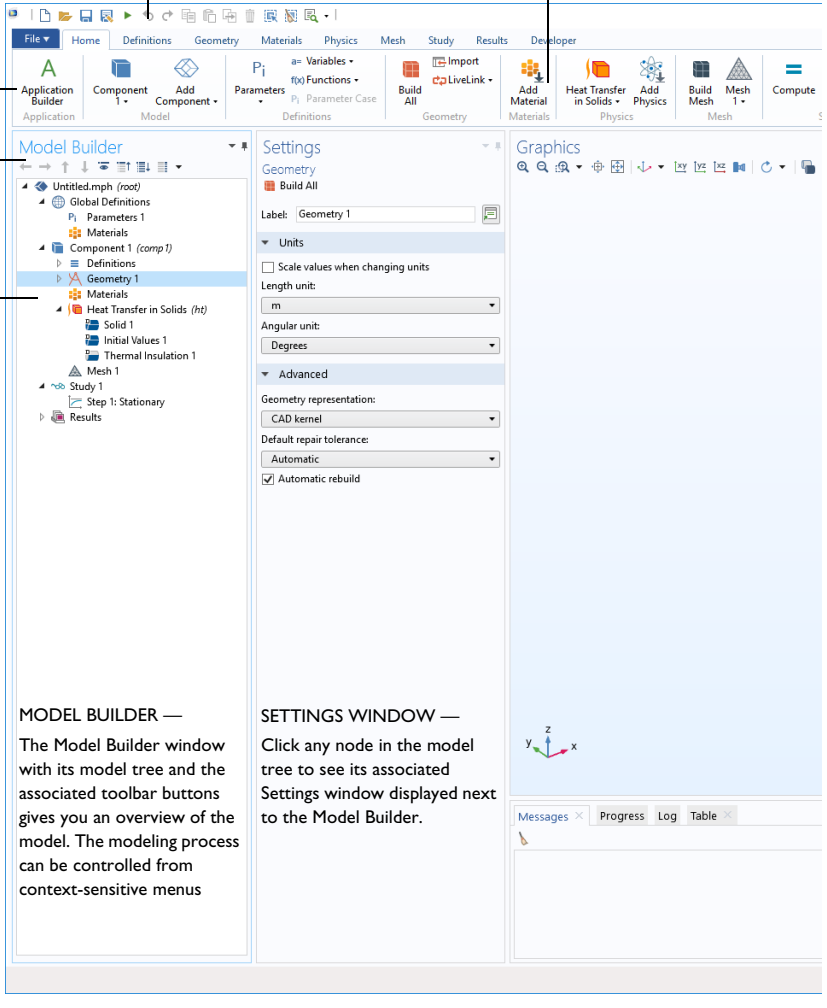
QUICK ACCESS TOOLBAR — Use these buttons for access to functionality such as file open/save, undo/redo, copy/paste, and delete.

RIBBON — The ribbon tabs have buttons and drop-down lists for controlling all steps of the modeling process.

APPLICATION BUILDER — Click this button to switch to the Application Builder and start building an application based on your model.

MODEL BUILDER TOOLBAR

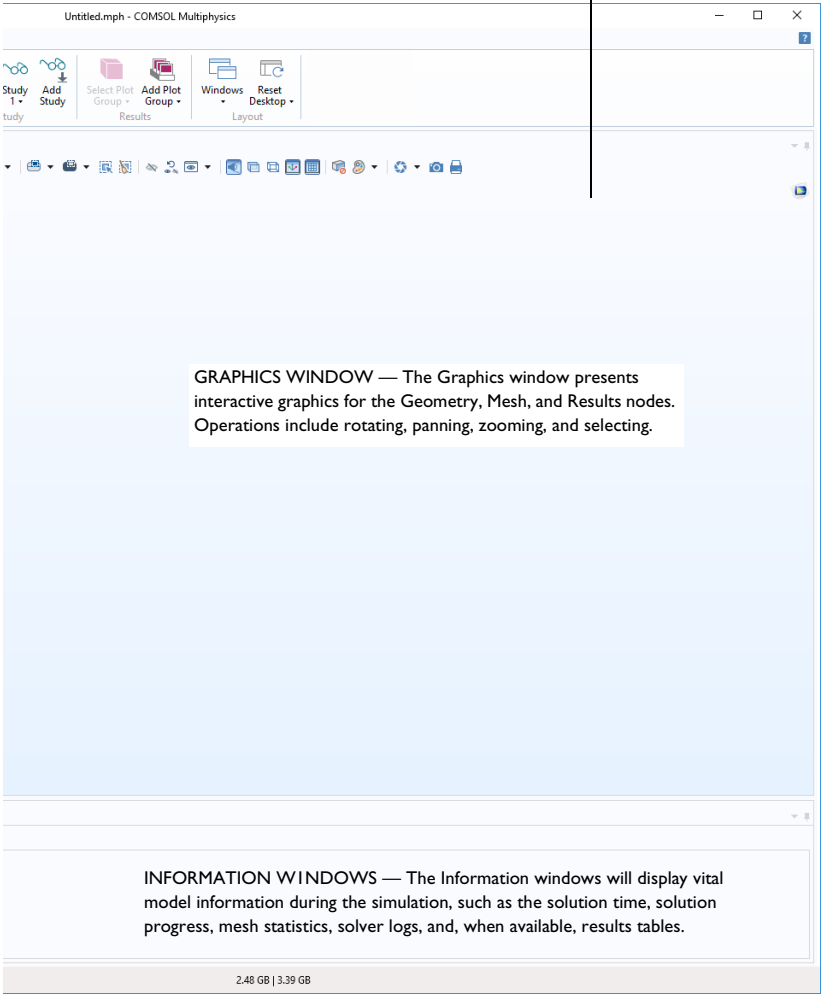
MODEL TREE — The model tree gives an overview of the model and all of the functionality and operations needed for building and solving



MODEL BUILDER — The Model Builder window with its model tree and the associated toolbar buttons gives you an overview of the model. The modeling process can be controlled from context-sensitive menus

SETTINGS WINDOW — Click any node in the model tree to see its associated Settings window displayed next to the Model Builder.

GRAPHICS WINDOW TOOLBAR



The screenshot on the previous pages is what you will see when you first start modeling in COMSOL Multiphysics using the Model Builder. The COMSOL Desktop® user interface provides a complete and integrated environment for physics modeling and simulation as well as application design, providing you with the tools necessary to build a user-friendly interface for your models. You can customize the desktop to your own needs. The windows can be resized, moved, docked, and detached. Any changes you make to the layout will be saved when you close the session and available again the next time you open COMSOL Multiphysics. As you build your model, additional windows and widgets will be added. (See page 26 for an example of a more developed desktop.) Among the available windows and user interface components are the following:

Quick Access Toolbar

The Quick Access Toolbar gives access to functionality such as **Open**, **Save**, **Undo**, **Redo**, **Copy**, **Paste**, and **Delete**. You can customize its content from the **Customize Quick Access Toolbar** list (the downward-facing arrow to the right of the toolbar).

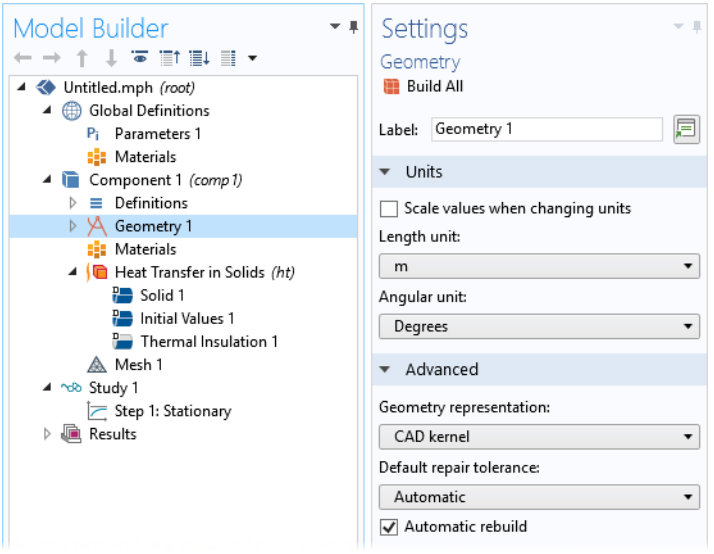
Ribbon

The ribbon at the top of the desktop gives access to commands used to complete most modeling tasks. The ribbon is only available in the Windows® version of the COMSOL Desktop environment and is replaced by menus and toolbars in the OS X and Linux® versions. Click the **Application Builder** button to switch from the Model Builder to the Application Builder and start building an application based on your model.

Settings Window

This is the main window for entering all of the specifications of the model, including the dimensions of the geometry, properties of the materials, boundary conditions and initial conditions, and any other information that the solver will

need to carry out the simulation. The picture below shows the **Settings** window for the **Geometry** node.



Plot Windows

These are the windows for graphical output. In addition to the **Graphics** window, **Plot** windows are used for visualizing results. Several **Plot** windows can be used to show multiple results simultaneously. A special case is the **Convergence Plot** window, an automatically generated **Plot** window that displays a graphical indication of the convergence of the solution process while a model is running.

Information Windows

These are the windows for nongraphical information. They include:

- **Messages**: Various information about the current COMSOL Multiphysics session is displayed in this window.
- **Progress**: Progress information from the solver in addition to stop buttons.
- **Log**: Information from the solver, such as the number of degrees of freedom, solution time, and solver iteration data.
- **Table**: Numerical data in table format as defined in the **Results** node.
- **External Process**: Provides a control panel for cluster, cloud, and batch jobs.

Other Windows

The **Windows** drop-down list in the **Home** tab of the ribbon gives you access to all COMSOL Desktop windows. On OS X and Linux®, you will find this in the **Windows** menu. In addition to the windows described earlier you can access the following windows:

- **Add Physics and Add Multiphysics**: Expand the physics interfaces in a model.
- **Add Study**: Change the type of study.
- **Add Material from Library** and the **Material Browser**: Access the material property libraries.
- **Application Libraries**: Choose among the collections of MPH-files containing tutorial models and runnable applications with accompanying documentation.
- **Part Libraries**: Load one of the predefined parameterized geometry models.
- **Selection List**: Show a list of geometry objects, domains, boundaries, edges, and points that are currently available for selection.
- **Properties**: View model tree node properties.
- **Debug Log**: Display debug information for methods.
- **Recovery Files**: View recovery files.
- **Comparison Results**: Compare the Model Builder and Application Builder settings of two MPH files.

Progress Bar with Cancel Button

The **Progress Bar** with a button for canceling the current computation, if any, is located in the lower right-hand corner of the COMSOL Desktop interface.

Dynamic Help

The **Help** window provides context-dependent help texts about windows and model tree nodes. If you have the **Help** window open in your desktop (by typing F1, for example), you will get dynamic help (in English only) when you click a node or a window. From the **Help** window, you can search for other topics, such as menu items.

The Model Builder and the Application Builder

The two main components of the COMSOL Desktop environment are the Model Builder and the Application Builder.

The Model Builder is the tool where you define the model and its components, such as how to solve it, the analysis of results, and creating reports. You do that by

building a model tree. The model tree reflects the underlying data structure, the model object, which stores the state of the model including settings for geometry, mesh, physics, boundary conditions, studies, solvers, postprocessing, and visualizations.

The Application Builder allows you to quickly create an application with a specialized user interface that is easy to use. An application is typically based on a model created with the Model Builder. The Application Builder provides two important tools for creating applications: The Form editor and the Method editor. In addition, an application can have a menu bar or a ribbon. The Form editor includes drag-and-drop capabilities to easily access and include user interface components, such as input fields, graphics windows, and buttons. The Method editor is a programming environment that, for example, allows you to modify the model represented by the model object data structure. You can also use the Method editor to add user interface logic and auxiliary functionality to an application. The Java® programming language is used to write code in the Method editor, which means that all Java® syntax and Java® libraries can be used.

In the Method editor, program code is stored in methods. A method is another name for what is known in other programming languages as a subroutine, function, or procedure. Methods can be used to modify the model object data structure in the Model Builder and the Application Builder. A method can be used both for modifying the model object of the current session and for modifying the model object in a running application.

This book, *Introduction to COMSOL Multiphysics*, gives a detailed introduction to using the Model Builder and a quick introduction to the Application Builder. For detailed information on how to use the Application Builder, including the Form editor and Method editor, see *Introduction to Application Builder*.

Applications, COMSOL Server, and COMSOL Compiler

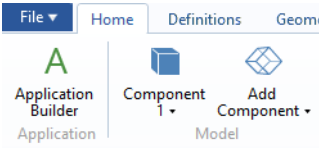
The Application Builder is included in the Windows® version of COMSOL Multiphysics. With a COMSOL Multiphysics license, applications can be run from the COMSOL Desktop environment. Although applications cannot be built with the macOS or Linux® versions of the software, applications can still be run with COMSOL Multiphysics on those platforms.

With a COMSOL Server™ license, applications can be run in major web browsers on a variety of operating systems and hardware platforms. In addition, you can run applications by connecting to COMSOL Server with an easy-to-install COMSOL Client for Windows®.

The COMSOL Client for Windows® allows a user to run applications that require a LiveLink™ product for CAD (this functionality is not available when running applications with a web browser).

Running applications in a web browser does not require any installation and no web browser plug-ins are needed. Running an application in a web browser supports interactive graphics in 1D, 2D, and 3D. In a web browser, graphics rendering in 3D are based on WebGL™ technology, which comes included with all major web browsers.

To create an application based on your model, click the **Application Builder** button, accessible from the **Home** tab in the ribbon.

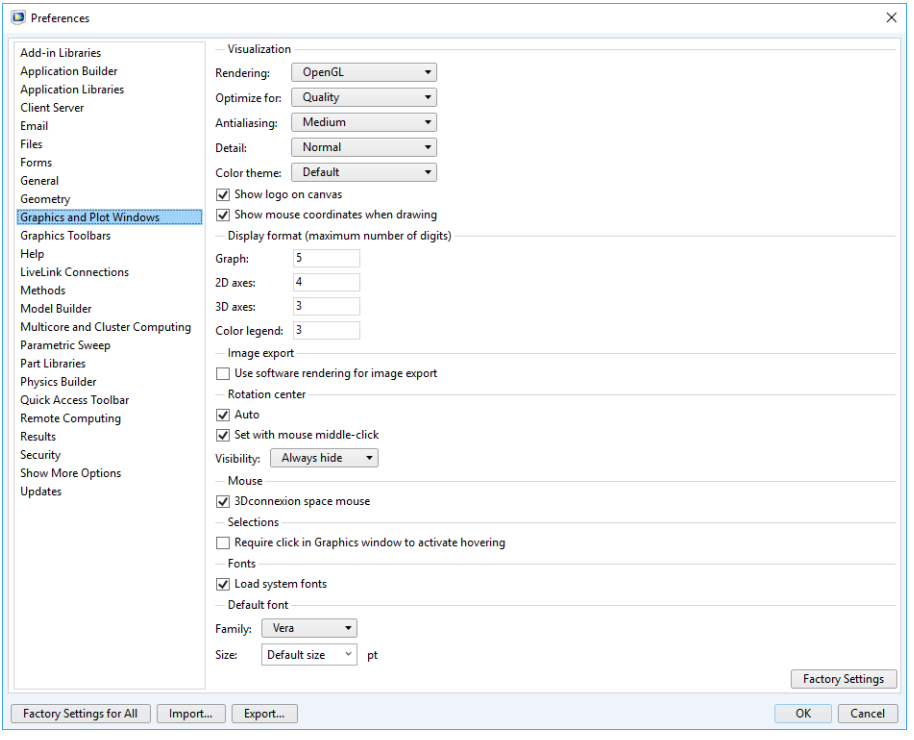


By using the COMSOL Compiler product you can compile your application (MPH-file) to an executable file for Windows®, Linux®, and macOS. You can freely distribute the executable and it can be run without any license file.

For more information on creating COMSOL applications, refer to the section "Building an Application with the Application Builder" on page 96 and the book *Introduction to Application Builder*.

Preferences

Preferences are settings that affect the modeling environment. Most are persistent between modeling sessions, but some are saved with the model. You access the preferences from the **File** menu by selecting **Preferences**.

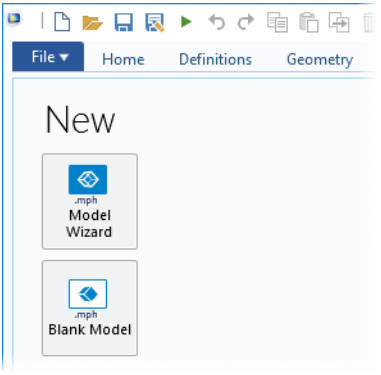


In the **Preferences** window, you can change settings such as graphics rendering, the number of displayed digits for results, the maximum number of CPU cores used for computations, or paths to user-defined application libraries. Take a moment to browse your current settings to familiarize yourself with the different options.

There are three graphics rendering options available: **OpenGL**®, **DirectX**®, and **Software Rendering**. The **DirectX**® option is not available in macOS or Linux®, but is available in Windows® if you choose to install the DirectX® runtime libraries during installation. If your computer does not have a dedicated graphics card, you may have to switch to **Software Rendering** for slower but fully functional graphics. A list of recommended graphics cards can be found at:

www.comsol.com/system-requirements

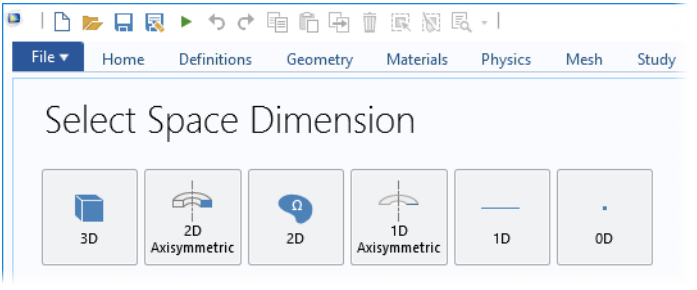
You can set up a model guided by the **Model Wizard** or start from a **Blank Model** as shown in the figure below.



Creating a Model Guided by the Model Wizard

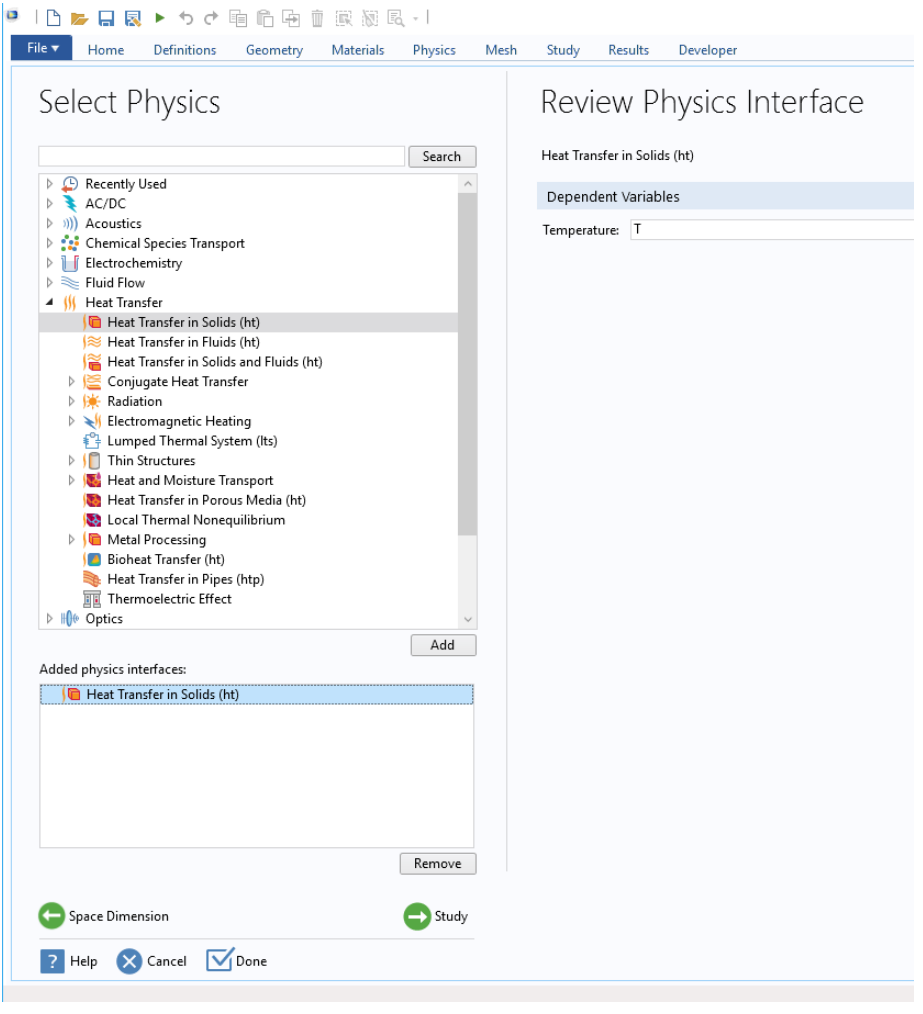
The Model Wizard will guide you in setting up the space dimension, physics, and study type in a few steps:

1 Start by selecting the space dimension for your model component: **3D**, **2D Axisymmetric**, **2D**, **1D Axisymmetric**, or **0D**.

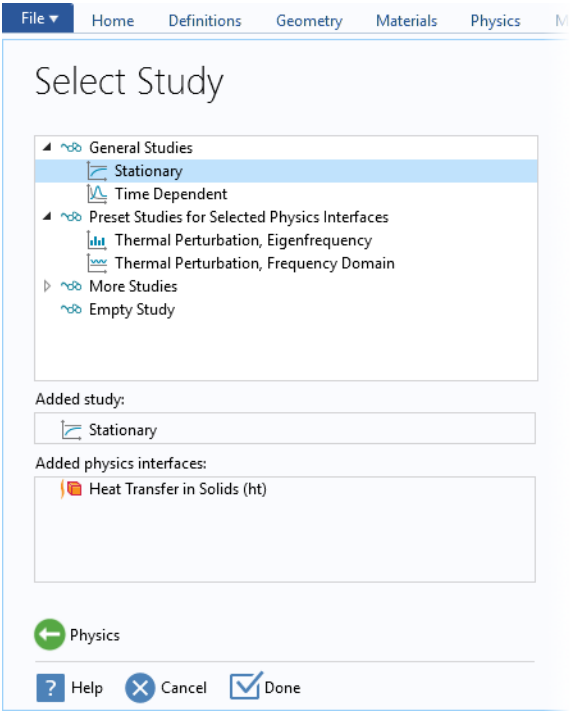


2 Now, add one or more physics interfaces. These are organized in a number of physics branches in order to make them easy to locate. These branches do not directly correspond to products. When products are added to your COMSOL

Multiphysics installation, one or more branches will be populated with additional physics interfaces.



3 Select the Study type that represents the solver or set of solvers that will be used for the computation.



Finally, click **Done**. The desktop is now displayed with the model tree configured according to the choices you made in the Model Wizard.

Creating a Blank Model

The **Blank Model** option will open the COMSOL Desktop interface without any **Component** or **Study**. You can right-click the model tree to add a **Component** of a certain space dimension, physics interface, or **Study**.

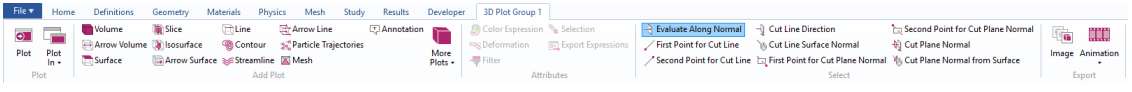
The Ribbon and Quick Access Toolbar

The ribbon tabs in the COMSOL Desktop environment reflect the modeling workflow and give an overview of the functionality available for each modeling step, including building simulation applications from your models.

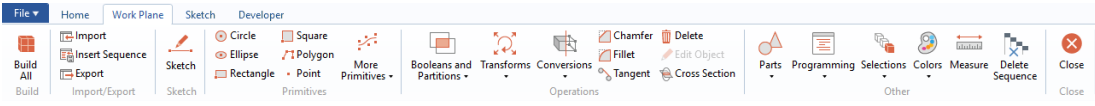
The **Home** tab contains buttons for the most common operations for making changes to a model, running simulations, and for building and testing applications. Examples include changing model parameters for a parameterized geometry, reviewing material properties and physics, building the mesh, running a study, and visualizing the simulation results.

There are standard tabs for each of the main steps in the modeling process. These are ordered from left to right according to the workflow: **Definitions**, **Geometry**, **Materials**, **Physics**, **Mesh**, **Study**, **Results**, and **Developer**.

Contextual tabs are shown only if and when they are needed, such as the **3D Plot Group** tab, which is shown when the corresponding plot group is added or when the node is selected in the model tree.



Modal tabs are used for very specific operations, when other operations in the ribbon may become temporarily irrelevant. An example is the **Work Plane** modal tab. When working with work planes, other tabs are not shown, since they do not present relevant operations.



The Ribbon vs. The Model Builder

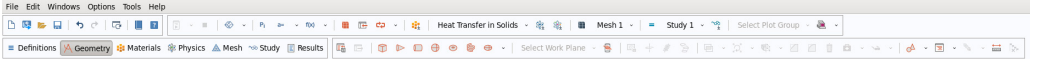
The ribbon gives quick access to available commands and complements the model tree in the **Model Builder** window. Most of the functionality accessed from the ribbon is also accessible from contextual menus by right-clicking nodes in the model tree. Certain operations are only available from the ribbon, such as selecting which desktop window to display. In the COMSOL Desktop interface for macOS and Linux®, this functionality is available from toolbars, which replace the ribbon on these platforms. There are also operations that are only available from the model tree, such as reordering and disabling nodes.

The Quick Access Toolbar

The **Quick Access Toolbar** contains a set of commands that are independent of the ribbon tab that is currently displayed. You can customize the **Quick Access Toolbar** and add most commands available in the **File** menu, including commands for undoing and redoing recent actions, as well as for copying, pasting, duplicating, and deleting nodes in the model tree. You can also choose to position the **Quick Access Toolbar** above or below the ribbon.

macOS and Linux®

In the COMSOL Desktop environment for macOS and Linux®, the ribbon is replaced by a set of menus and toolbars.



The instructions in this book are based on the Windows® version of the COMSOL Desktop environment. However, running COMSOL Multiphysics and the COMSOL Desktop environment in macOS and Linux® is very similar, keeping in mind that the ribbon user interface components can instead be found in the corresponding menus and toolbars.

The Model Builder and the Model Tree

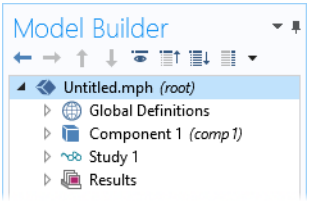
Using the Model Builder, you build a model by starting with the default model tree, adding nodes, and editing the node settings.

All of the nodes in the default model tree are top-level parent nodes. You can right-click on them to see a list of child nodes, or subnodes, that you can add beneath them. This is the means by which nodes are added to the tree.

When you click on a child node, you will see its node settings in the **Settings** window. It is here that you can edit node settings.

It is worth noting that if you have the **Help** window open, which is achieved either by selecting **Help** from the **File** menu or by pressing the function key F1, then you will also get dynamic help (in English only) when you click on a node.

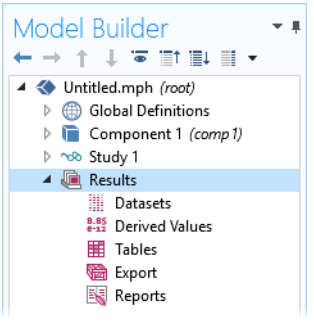
The Root, Global Definitions, and Results Nodes



A model tree always has a root node (initially labeled `Untitled.mph`), a **Global Definitions** node, and a **Results** node. The label on the root node is the name of the multiphysics model file, or MPH file, to which the model is saved. The root node has settings for author name, default unit system, and more.

The **Global Definitions** node has a **Parameters** and **Materials** subnode by default. The **Global Definitions** node is where you, among other things, define parameters, variables, functions, and couplings that can be used throughout the model tree. They can be used, for example, to define the values and functional dependencies of material properties, forces, geometry, and other relevant features. The **Global Definitions** node itself has no settings, but its child nodes have plenty of them. The **Materials** subnode stores material properties that can be referenced in the **Component** nodes of a model.

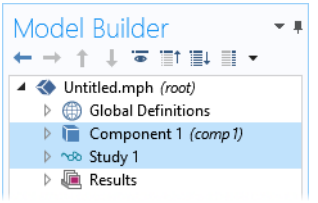
The **Results** node is where you access the solution after performing a simulation and where you find tools for processing the data. The **Results** node initially has five subnodes:



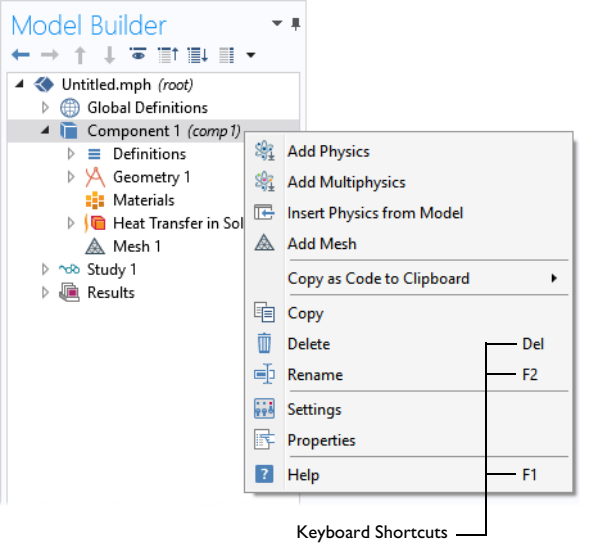
- **Datasets**, which contains a list of solutions you can work with.
- **Derived Values**, which defines values to be derived from the solution using a number of postprocessing tools.
- **Tables**, which is a convenient destination for the **Derived Values** or for **Results** generated by probes that monitor the solution in real-time while the simulation is running.
- **Export**, which defines numerical data, images, and animations to be exported to files.
- **Reports**, which contains automatically generated or custom reports about the model in HTML or Microsoft® Word format.

To these five default subnodes, you may also add more **Plot Group** subnodes that define graphs to be displayed in the **Graphics** window or in **Plot** windows. Some of these may be created automatically, depending on the type of simulations you are performing, but you may include additional figures by right-clicking on the **Results** node and choosing from the list of plot types.

The Component and Study Nodes



In addition to the three nodes just described, there are two additional top-level node types: **Component** nodes and **Study** nodes. These are usually created by the Model Wizard when you create a new model. After using the Model Wizard to specify what type of physics you are modeling and what type of **Study** (for example, steady-state, time-dependent, frequency-domain, or eigenfrequency analysis) you will carry out, the Model Wizard automatically creates one node of each type and shows you their contents.



It is also possible to add more **Component** and **Study** nodes as you develop the model. A model can contain multiple **Component** and **Study** nodes and it would be confusing if they all had the same name. Therefore, these types of nodes can be renamed to be descriptive of their individual purposes.

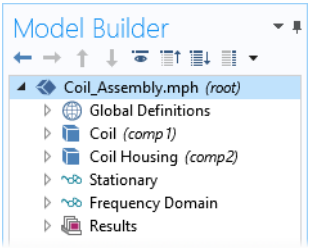
If a model has multiple **Component** nodes, they can be coupled to form a more sophisticated sequence of simulation steps.

Note that each Study node may carry out a different type of computation, so each one has a separate **Compute** button = .

To be more specific, suppose that you build a model that simulates a coil assembly that is made up of two parts, a coil and a coil housing. You can create two **Component** nodes, one that models the coil and the other the coil housing. You can then rename each of the nodes with the name of the object. Similarly, you can also create two **Study** nodes, the first simulating the stationary or steady-state behavior of the assembly, and the second simulating the frequency response. You can rename these two nodes to be **Stationary** and **Frequency Domain**.

When the model is complete, save it to a file named `Coil Assembly.mph`. At that point, the model tree in the Model Builder looks like the figure below.

In this figure, the root node is named `Coil Assembly.mph`, indicating the file in which the model is saved. The **Global Definitions** node and the **Results** node each have their default name. Additionally, there are two **Component** nodes and two **Study** nodes with the names chosen in the previous paragraph.



Parameters, Variables, and Scope

Global Parameters

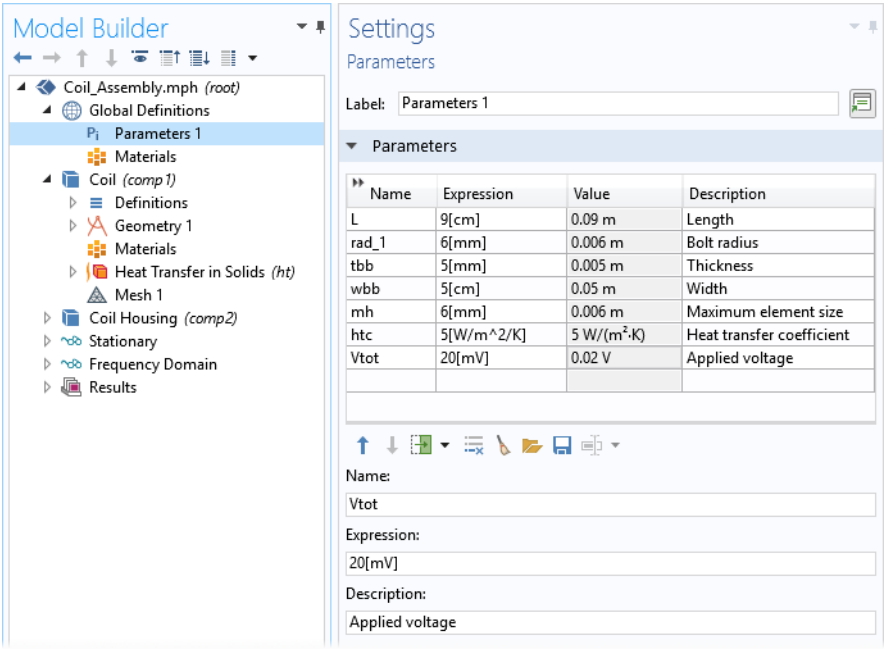
Global parameters are user-defined constant scalars that are usable throughout the model. That is to say, they are "global" in nature. Important uses are:

- Parameterizing geometric dimensions.
- Specifying mesh element sizes.
- Defining parametric sweeps (simulations that are repeated for a variety of different values of a parameter such as a frequency or load).

A global parameter expression can contain numbers, global parameters, built-in constants, built-in and user-defined functions with global parameter expressions as arguments, and unary and binary operators. For a list of available operators, see "Appendix C — Language Elements and Reserved Names" on page 181. Because these expressions are evaluated before a simulation begins, global parameters may not depend on the time variable t. Likewise, they may not depend on spatial variables like x, y, or z, nor on the dependent variables for which your equations are solving.

It is important to know that the names of parameters are case sensitive.

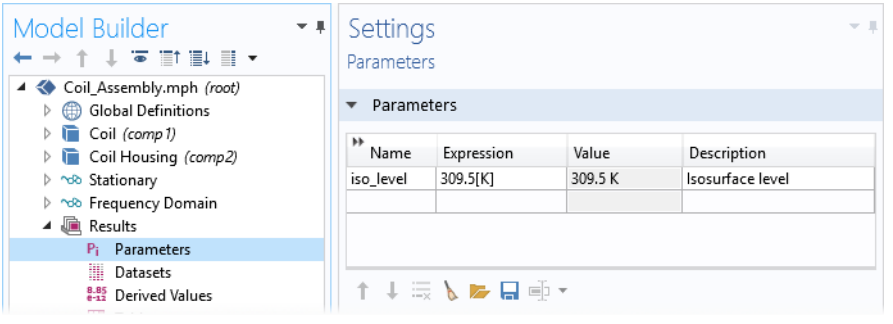
You define global parameters in the **Parameters** node in the model tree under **Global Definitions**.



Note that you can add multiple **Parameters** nodes and they will be named **Parameters 1**, **Parameters 2**, etc. In addition, **Parameters** can be organized into several cases and then swept over, for more information see “Parametric Sweeps” on page 141.

Results Parameters

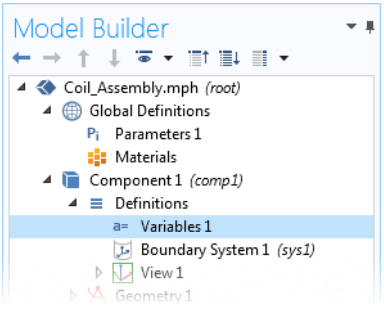
For greater flexibility, it is possible to define parameters that are only used in the **Results** node. Using these parameters does not require resolving the model.



Result parameters may depend on other parameters and built-in functions.

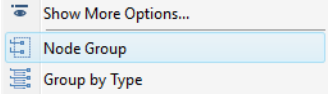
Variables

Variables have associated **Variables** nodes in the model tree and can be defined either in the **Global Definitions** node or in the **Definitions** subnode of any **Component** node.



Naturally, the choice of where to define the variable depends on whether you want it to be global (that is, usable throughout the model tree) or locally defined within a single **Component** node. Like a parameter expression, a variable expression may contain numbers, parameters, built-in constants, and unary and binary operators. However, it may also contain variables like `t`, `x`, `y`, or `z`; functions with variable expressions as arguments; and dependent variables that you are solving for in addition to their space and time derivatives.

If you have a model with many **Variables** nodes it may be useful to group them together by using the **Node Group** feature, as shown below.



Node groups can be used to group nodes in **Global Definitions**, **Definitions** under a **Component**, **Materials**, physics interfaces, and results.

Variables Used in Applications

Model parameters and variables can be used in applications. For example, you can let the user of an application change the value of a parameter. In addition, variables to be used in applications can be defined in the Application Builder, in the application tree under the **Declarations** node. Such variables can also be used in model methods.

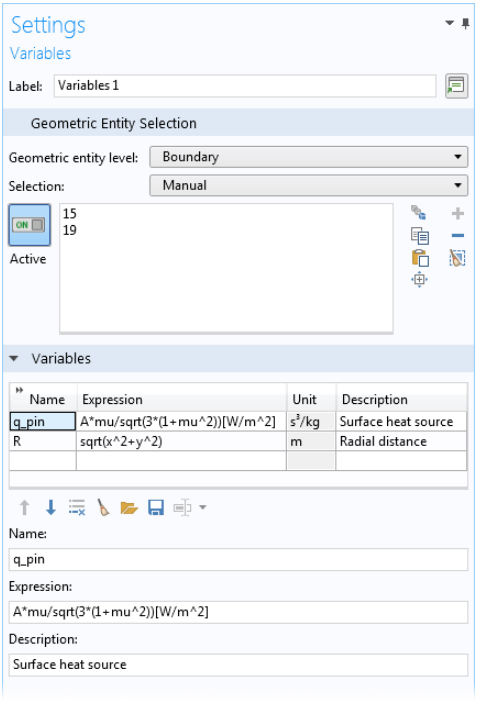
Scope

The "scope" of a parameter or variable is a statement about where it may be used in an expression. All global parameters are defined in the **Global Definitions** node of the model tree as a **Parameters** subnode. This means that they are global in scope and can be used throughout the model tree.

A variable may also be defined in the Global Definitions node, as a **Variables** subnode, and have global scope, but they are subject to other limitations. For example, variables may not be used in **Geometry**, **Mesh**, or **Study** nodes (with the one exception that a variable may be used in an expression that determines when the simulation should stop).

A variable that is instead defined under the **Definitions** subnode of a **Component** node has local scope and is intended for use in that particular **Component** (but, again, not in the **Geometry** or **Mesh** nodes). They may be used, for example, to specify material properties in the **Materials** subnode of a **Component** or to specify boundary conditions or interactions. It is sometimes valuable to limit the scope of the variable to only a certain part of the geometry, such as certain boundaries. For that purpose, provisions are available in the settings for a variable to select whether to apply the definition either to the entire geometry of the **Component** or only to a **Domain**, **Boundary**, **Edge**, or **Point**.

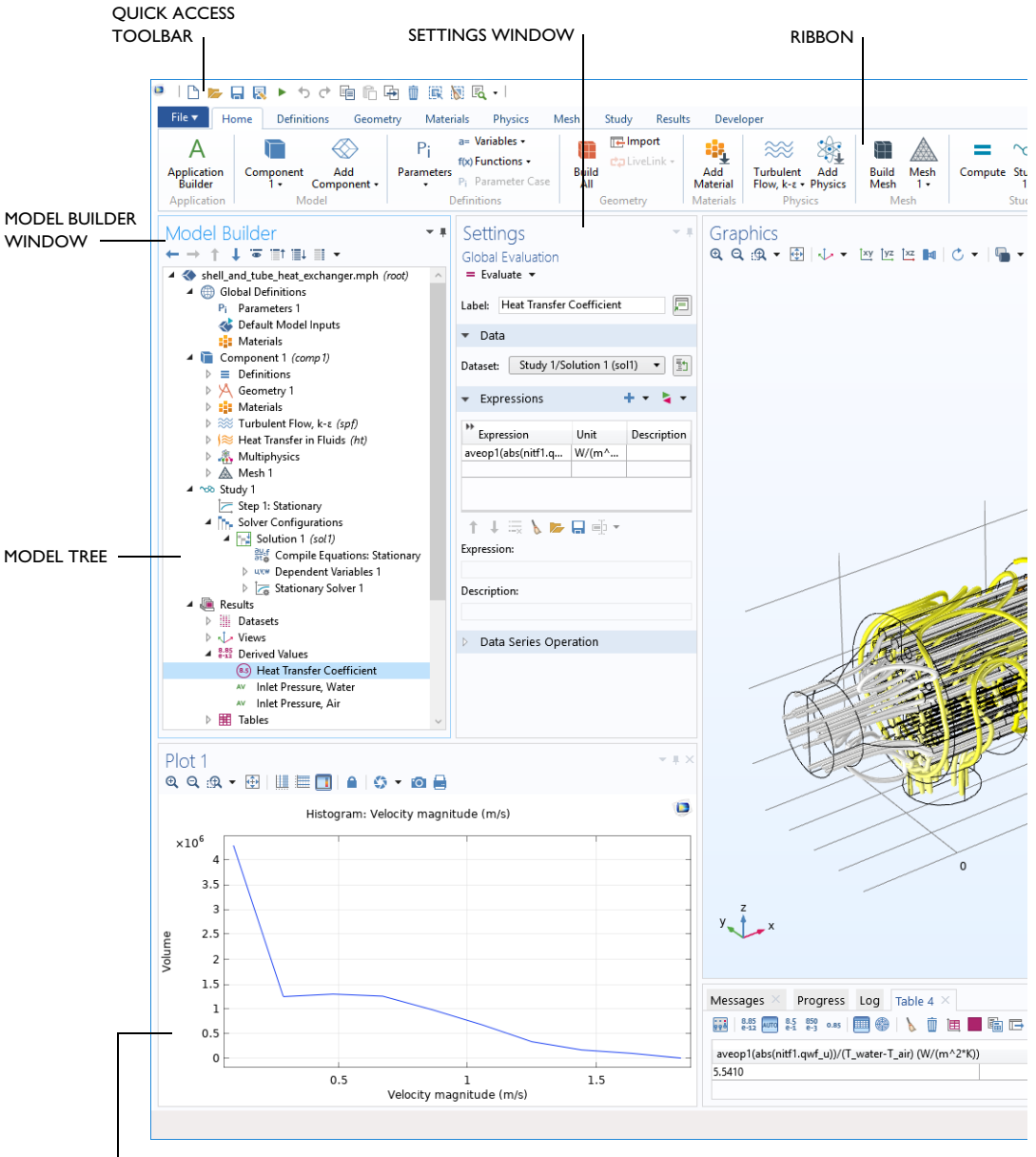
The figure at right shows the definition of two variables, `q_pin` and `R`, for which the scope is limited to just two boundaries identified by numbers 15 and 19.



Such a group of numbers is called a **Selection** and can be named and then referenced elsewhere in a model. This can be useful, for example, when defining material properties or boundary conditions that will use the variable at certain boundaries but not elsewhere. To give a name to the **Selection**, click the **Create Selection** button ( ) to the right of the **Selection** list.

Although the variables defined in the **Variables** node under the **Component > Definitions** subnode are intended to have local scope, they can still be accessed outside of the **Component** node in the model tree by being sufficiently specific about their identity. This is done by using a "dot-notation" where the variable name is preceded by the name of the **Component** node in which it is defined and they are joined by a "dot". In other words, if a variable named `foo` is defined in a **Component** node named `MyModel`, then this variable may be accessed outside of the Component node by using `MyModel.foo`. This can be useful, for example, when you want to use the variable to make plots in the **Results** node. Variables defined under the **Declarations** node in the Application Builder are available globally in form objects and methods but cannot be used in the Model Builder.

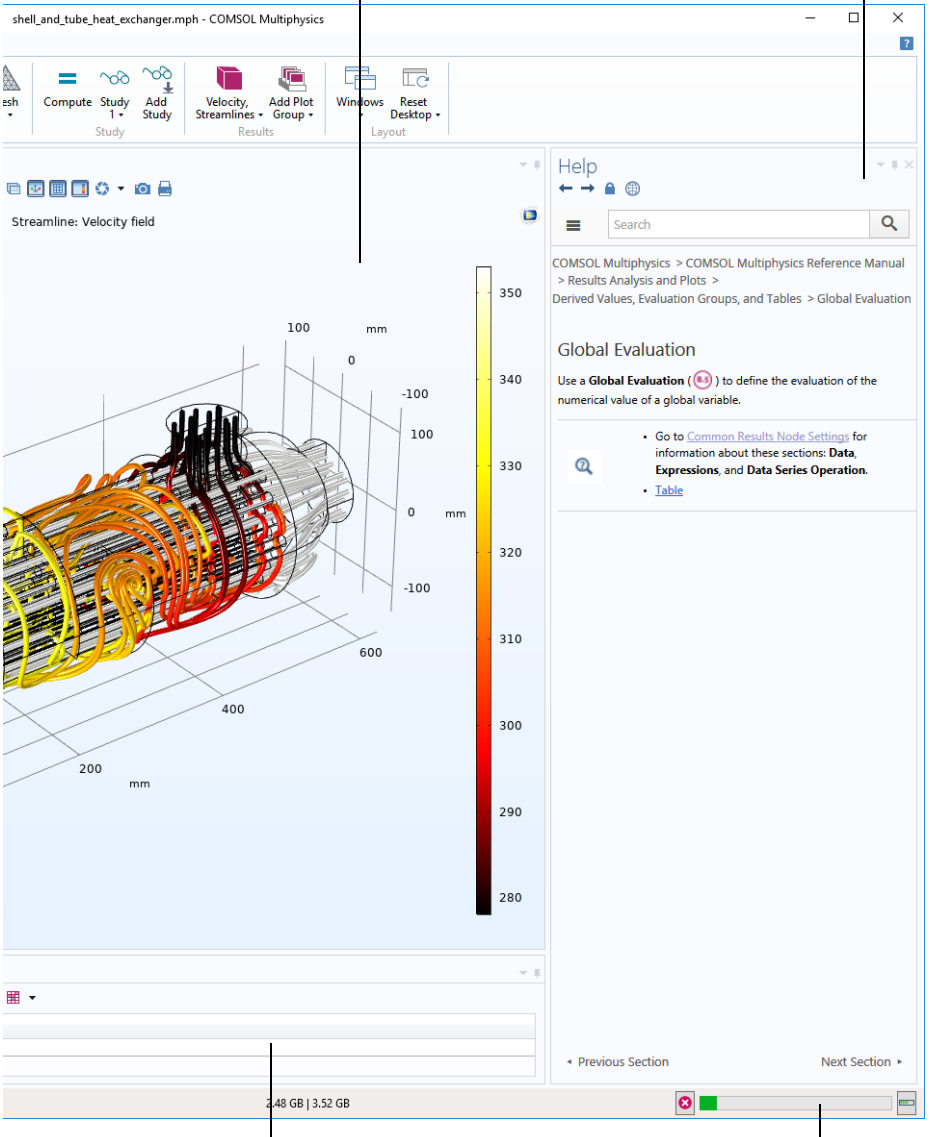
The following spread shows an example of a customized desktop with additional windows.



PLOT WINDOW — The Plot window is used to visualize Results quantities, probes, and convergence plots. Several Plot windows can be used to show multiple results simultaneously.

GRAPHICS WINDOW

DYNAMIC HELP — Continuously updated with online access to the Knowledge Base and Application Gallery. The Help window enables easy browsing with extended search functionality.



INFORMATION WINDOWS

PROGRESS BAR WITH CANCEL BUTTON

Built-In Constants, Variables, and Functions

COMSOL Multiphysics comes with many built-in constants, variables, and functions. They have reserved names that cannot be redefined by the user. If you use a reserved name for a user-defined variable, parameter, or function, the text you enter will turn orange (a warning) or red (an error) and you will get a tooltip message if you select the text string.

Some important examples are:

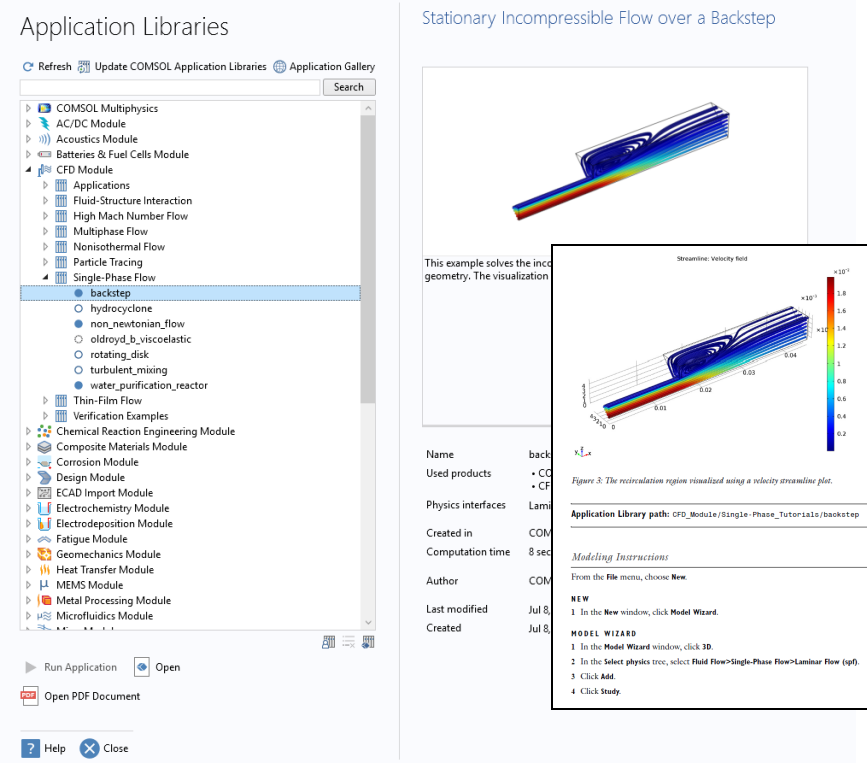
- Mathematical constants such as `pi` (3.14...) or the imaginary unit `i` or `j`
- Physical constants such as `g_const` (acceleration of gravity), `c_const` (speed of light), or `R_const` (universal gas constant)
- The time variable, `t`
- First- and second-order derivatives of the dependent variables (the solution) whose names are derived from the spatial coordinate names and dependent variable names (which are user-defined variables)
- Mathematical functions such as `cos`, `sin`, `exp`, `log`, `log10`, and `sqrt`

See "Appendix C — Language Elements and Reserved Names" on page 181 for more information.

The Application Libraries

The Application Libraries are collections of MPH files containing tutorial models and runnable applications with accompanying documentation. Tutorial models show how to use the Model Builder and have documentation that includes theoretical background and step-by-step instructions. Runnable applications have instructions on how to use the application. You can readily inspect and edit all tutorial models and applications to make them your own. Each physics-based add-on module comes with its own application library with examples specific to its applications and physics area. You can use the step-by-step instructions and the MPH files as a template for your own modeling.

To open the **Application Libraries** window, select **Application Libraries** from the **Windows** menu on the **Home** toolbar or from the **File** menu . Then, search by application name or browse under a module folder name.



Click **Open** , **Run Application** , or **Open PDF Document** . Alternatively, select **Help > Documentation** from the **File** menu to search by application name or browse by module. Note that the **Run Application** option is only available for MPH files that are runnable applications.

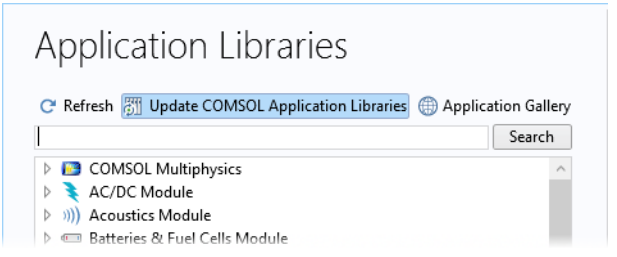
The MPH files in the Application Libraries can have three formats — Solved MPH files, Compact MPH files, or Preview MPH files:

- Solved MPH files, including all meshes and solutions. In the Application Libraries window, these appear with the icon. If the MPH file size exceeds 25 MB, a tip with the text “Large file” and the file size appears when you position the cursor at the model’s node in the Application Libraries tree.
- Compact MPH files, with all settings for the model but without built meshes and solution data to save space in the installation. Some MPH files have no solutions for other reasons such that the computation time is short and the

results are easily regenerated. You can open these to study the settings and to mesh and re-solve. It is also possible to download the full versions — with meshes and solutions — of most of these when you update your Application Libraries. These appear in the Application Libraries window with the icon. If you position the cursor at a compact file in the Application Libraries window, a **No solutions stored** message appears. A solved version is available for download via the Application Library Update (except in a few cases where the file is a template related to another, regular, model).

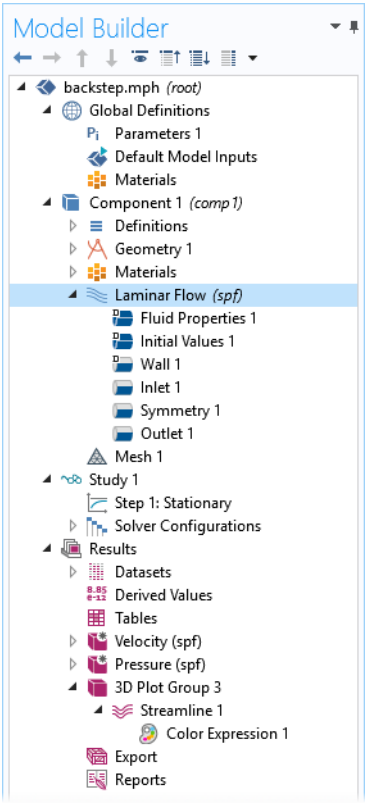
- Preview MPH files, with only a model description and thumbnail image available. These appear in the Application Libraries window with the icon. The file is a preview of an application or a model file, including the model description and information about used products and physics interfaces and the computation time. Solved and compact versions are available for download. Model documentation will be available if you have specified the use of online help (see the *Reference Manual* for details on how to do this).

The Application Libraries are updated on a regular basis by COMSOL. To check all available updates, click **Update COMSOL Application Libraries** at the top of the **Application Libraries** window. You can also find this option from the **File > Help** menu (Windows® users) or from the **Help** menu (macOS and Linux® users). This connects you to the COMSOL website, where you can access new applications and the latest updates.



If your computer has an internet connection, then you can click the **Application Gallery** button to access a wide range of additional examples from the COMSOL website.

In the **Model Builder** window, every step of the modeling process, from defining global variables to the final report of results, is displayed in the model tree.



From top to bottom, the model tree defines an orderly sequence of operations.

In the following branches of the model tree, the node order makes a difference and you can change the sequence of operations by moving the subnodes up or down the model tree:

- **Geometry**
- **Materials**
- **Physics**
- **Mesh**
- **Study**
- **Plot Groups**

Nodes may be reordered by these methods:

- Drag-and-drop
- Right-clicking the node and selecting **Move Up** or **Move Down**
- Pressing Ctrl + Up arrow or Ctrl + Down arrow

In other branches, the ordering of nodes is not significant with respect to the sequence of operations, but some nodes can be reordered for readability. Child nodes to **Global Definitions** is one such example.

You can view the sequence of operations presented as program code statements by saving the model as a **Model File for MATLAB**® or as a **Model File for Java**® after having selected **Compact History** in the **File** menu. Note that the model history keeps a complete record of the changes you make to a model as you build it. As such, it includes all of your corrections, including changes to parameters and boundary conditions and modifications of solver methods. Compacting this history removes all of the overridden changes and leaves a clean copy of the most recent form of the model steps. In the Application Builder, you can use the **Record Method** option to view and edit program code statements in the Method editor.

As you work with the COMSOL Desktop interface and the Model Builder, you will grow to appreciate the organized and streamlined approach. However, any description of a user interface is inadequate until you try it for yourself. In the next chapters, you are invited to work through two examples to familiarize yourself with the software.

Example 1: Structural Analysis of a Wrench

This simple example requires none of the add-on products to COMSOL Multiphysics®. For more fully-featured structural mechanics models, see the Structural Mechanics Module application library.

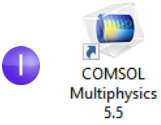
At some point in your life, it is likely that you have tightened a bolt using a wrench. This exercise takes you through a structural mechanics model that analyzes this basic task from the perspective of the structural integrity of the wrench subjected to a worst-case loading.

The wrench is, of course, made from steel, a ductile material. If the applied torque is too high, the tool will be permanently deformed due to the steel's elastoplastic behavior when pushed beyond its yield stress level. To analyze whether the wrench handle is appropriately dimensioned, you will check if the mechanical stress level is within the yield stress limit.

This tutorial gives a quick introduction to the Model Builder workflow. It starts with opening the Model Wizard and adding a physics option for solid mechanics. Then a geometry is imported and steel is selected as the material. You then explore the other key steps in creating a model by defining a parameter and boundary condition for the load; selecting geometric entities in the Graphics window; defining the Mesh and Study; and finally, examining the results numerically and through visualization.

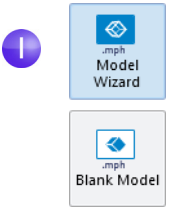
If you prefer to practice with a more advanced model, read this section to familiarize yourself with some of the key features, and then go to the tutorial "Example 2: The Busbar — A Multiphysics Model" on page 59.

1 To start the software, double-click the COMSOL Multiphysics icon on the desktop, which will take you to the New window with two options for creating a new model: **Model Wizard** or **Blank Model**.



If you select **Blank Model**, you can right-click the root node in the model tree to manually add a **Component** and a **Study**. For this tutorial, click the **Model Wizard** button.

If the COMSOL Desktop user interface is already open, you can start the Model Wizard by selecting **New** from the **File** menu. Choose the **Model Wizard**.



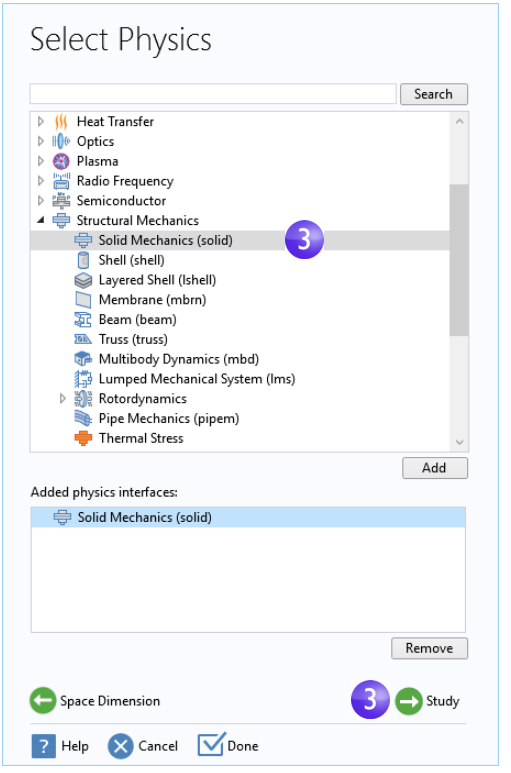
The Model Wizard will guide you through the first steps of setting up a model. The next window lets you select the dimension of the modeling space.

2 In the **Select Space Dimension** window, select 3D.

3 In **Select Physics**, select **Structural Mechanics > Solid Mechanics (solid)** . Click **Add**.

Without add-on modules, **Solid Mechanics** is the only physics interface available in the **Structural Mechanics** folder. In the picture to the right, part of the **Structural Mechanics** folder is shown as it appears when all add-on modules are available.

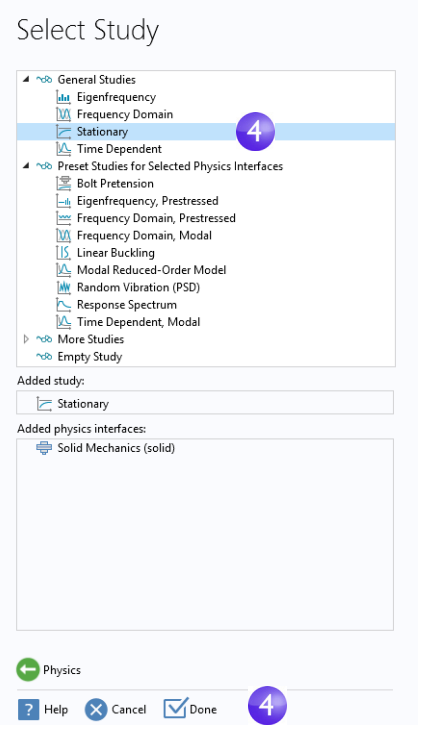
Click **Study** to continue.



4 Click **Stationary** under **Preset Studies**. Click **Done** once you have finished.

Preset studies have solver and equation settings adapted to the selected physics, which is, in this example, **Solid Mechanics**. A **Stationary** study is used in this case — there are no time-varying loads or material properties.

Any selection from the **Custom Studies** branch requires manual settings.



Geometry

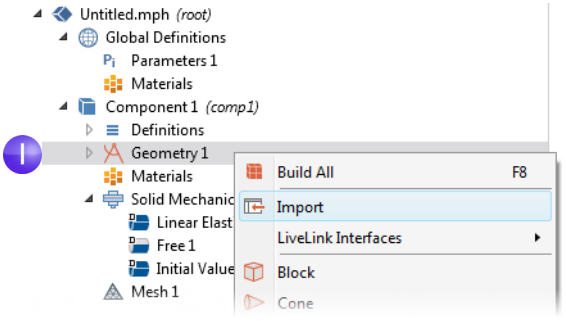
This tutorial uses a geometry that was previously created and stored in the COMSOL native CAD format, `.mphbin`. To learn how to build your own geometry, see "Appendix A — Building a Geometry" on page 159.

File Locations

The location of the application library that contains the file used in this exercise varies based on the software installation and operating system. In Windows®, the file path will be similar to:

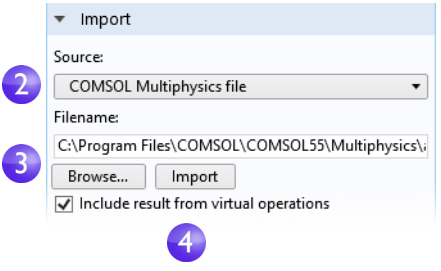
`C:\Program Files\COMSOL\COMSOL55\Multiphysics\applications`.

1 In the Model Builder window, under **Component 1**, right-click **Geometry 1** and select **Import**.



As an alternative, you can use the ribbon and click **Import** from the **Geometry** tab.

2 In the **Settings** window for **Import**, from the **Source** list, select **COMSOL Multiphysics file**.



3 Click **Browse** and locate the file `wrench.mphbin` in the application library folder of the COMSOL installation folder. Its default location in Windows® is

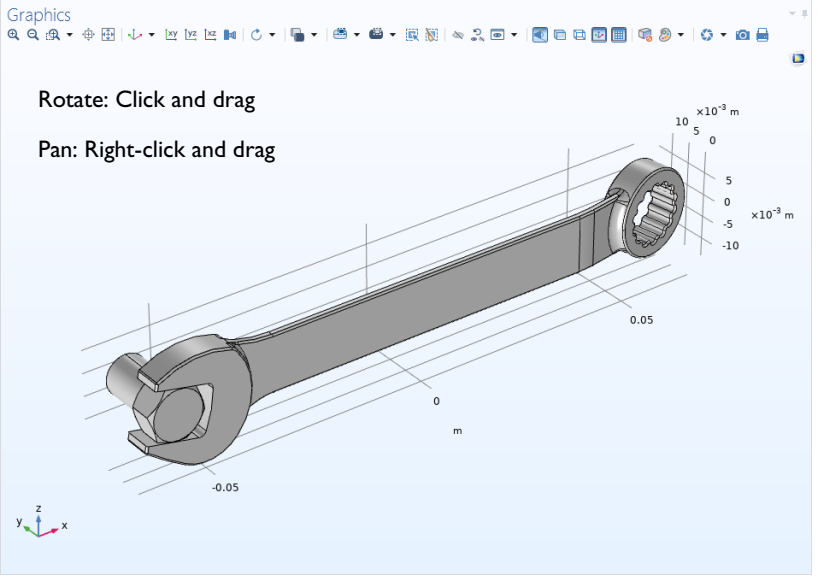
`C:\Program Files\COMSOL\COMSOL55\Multiphysics\applications\COMSOL_Multiphysics\ Structural_Mechanics\wrench.mphbin`

Double-click to add or click **Open**.

4 Click **Import** to display the geometry in the **Graphics** window.

5 Click the wrench geometry in the **Graphics** window and then experiment with moving it around. As you point to or click the geometry, it changes color. Click the **Zoom In**, **Zoom Out**, **Go to Default 3D View**, **Zoom Extents**, and **Transparency** buttons on the **Graphics** window toolbar to see what happens to the geometry:

- To rotate, click and drag anywhere in the **Graphics** window.
- To move, right-click and drag.
- To zoom in and out, click the mouse scroll wheel, continue holding it, and drag.
- To get back to the original position, click the **Go to Default 3D View** button on the toolbar.



Also see “Appendix B — Keyboard and Mouse Shortcuts” on page 176 for additional information.

The imported model has two parts, or domains, corresponding to the bolt and the wrench. In this exercise, the focus will be on analyzing the stress in the wrench.

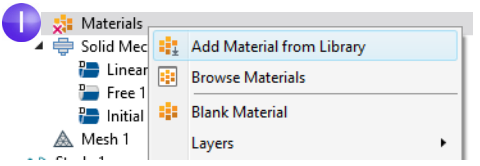
Materials

The **Materials** node stores the material properties for all physics and all domains in a **Component** node. Use the same generic steel material for both the bolt and tool. Here is how to choose it in the Model Builder.

1 Open the **Add Materials window**.

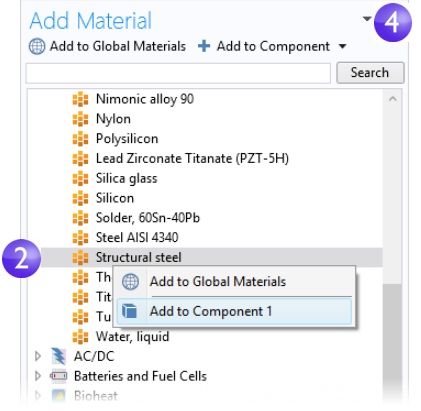
You can open the Add Materials window in either of these two ways:

- Right-click **Component 1 > Materials** in the **Model Builder** and select **Add Material from Library**
- From the ribbon, select the **Home** tab and then click **Add Material.**

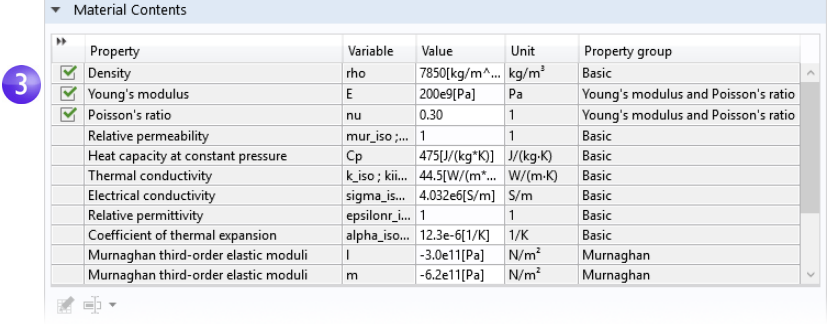


2 In the Add Material window, click to expand the **Built-In** folder. Scroll down to find **Structural steel**, right-click, and select **Add to Component 1**.

3 Examine the **Material Contents** section in the **Settings** window for **Material** to see the properties that are available. Properties with green check marks are used by the physics in the simulation.



4 Close the **Add Material** window.



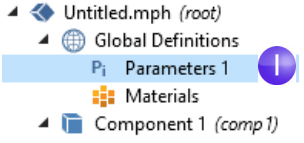
Also see the busbar tutorial sections “Materials” on page 73 and “Customizing Materials” on page 115 to learn more about working with materials.

Global Definitions

You will now define a global parameter specifying the load applied to the wrench.

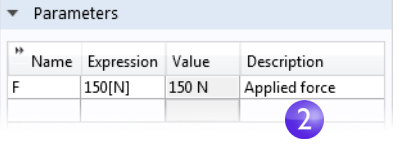
Parameters

1 In the **Model Builder**, under **Global Definitions**, choose **Parameters 1**.



2 Go to the **Settings** window for **Parameters 1**. In the **Parameters** table, enter these settings:

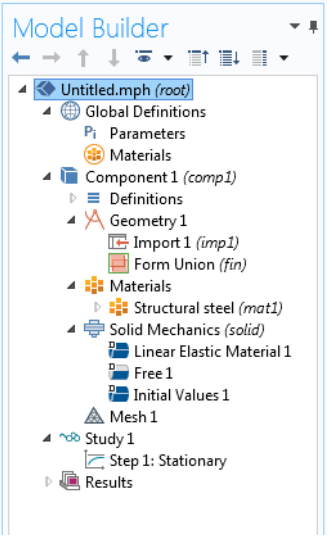
- In the **Name** column or field, enter `F`.
- In the **Expression** column or field, enter `150[N]`. The square-bracket notation is used to associate a physical unit to a numerical value; in this case, the unit of force in Newtons. The **Value** column is automatically updated based on the expression entered once you leave the field or press Return.
- In the **Description** column or field, enter `Applied force`.



If you have a **Parameters** table with more than one entry, you can sort the table with respect to a particular column by clicking on the corresponding header.

The sections “Global Definitions” on page 65 and “Parameters, Functions, Variables, and Couplings” on page 110 show you more about working with parameters.

So far, you have added the physics and study, imported a geometry, added the material, and defined one parameter. The **Model Builder** node sequence should now match the figure to the right. The default feature nodes under **Solid Mechanics** are indicated by a **D** in the upper-left corner of the node icon .



The default nodes for **Solid Mechanics** are: **Linear Elastic Material**, **Free**, and **Initial Values**.

The node **Linear Elastic Material** is the default material model for the **Solid Mechanics** interface. The node **Free** is a boundary condition that allows all boundaries to move freely without a constraint or load. The node **Initial Values** is used for specifying initial displacement and velocity values for a nonlinear or transient analysis (not applicable in this case).

The default physics nodes cannot be removed. Instead, you specify physics settings that deviate from that of the default nodes by adding additional nodes. These additional nodes can override or contribute to the settings of the default nodes as well as other nodes. For more information, see “Override and Contribution: Exclusive and Contributing Nodes” on page 126.

At any time, you can save your model and then open it later in exactly the state in which it was saved.

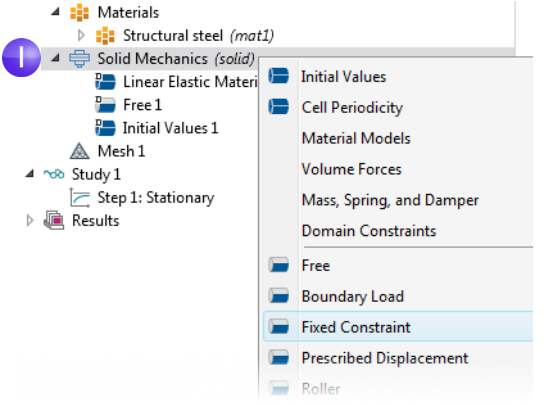
3 From the File Menu, select **File > Save As**. Browse to a folder where you have write permissions, and save the file as `wrench.mph`.

With the geometry and materials defined, you are now ready to set the boundary conditions.

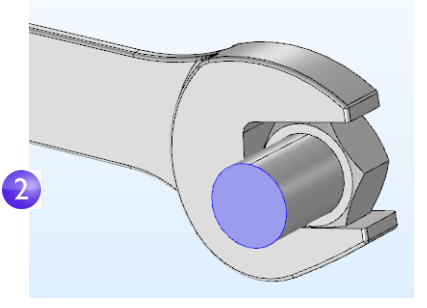
1 In the **Model Builder**, right-click **Solid Mechanics (solid)** and select **Fixed Constraint** .

This boundary condition constrains the displacement of each point on a boundary surface to be zero in all directions.

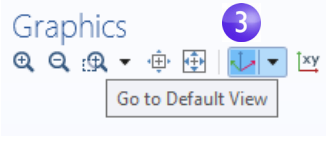
You can also use the ribbon and select, from the **Physics** tab, **Boundaries > Fixed Constraint**.



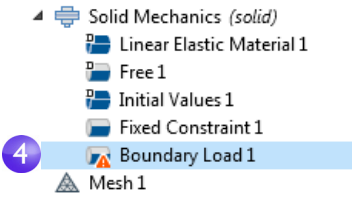
2 In the **Graphics** window, rotate the geometry by clicking anywhere in the window and then dragging the wrench into the position shown. Click on the exposed front surface of the partially modeled bolt. The boundary turns blue indicating that it has been selected. The **Boundary number** in the **Selection** list should be **35**.



3 Click the **Go to Default View** button on the **Graphics** toolbar to restore the geometry to the default view.

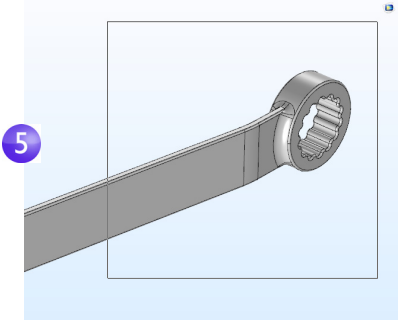


4 In the **Model Builder**, right-click **Solid Mechanics (solid)** and select **Boundary Load**. A **Boundary Load** node is added to the **Model Builder** sequence.

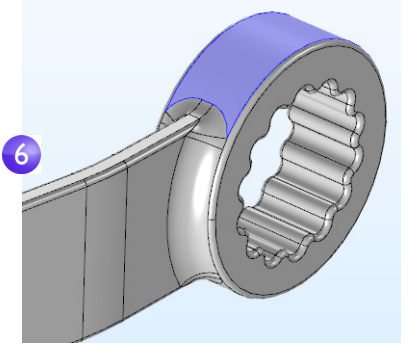


The "warning" attribute displayed on the Boundary Load node indicates that its boundary selection is empty. The next step will be to select a boundary.

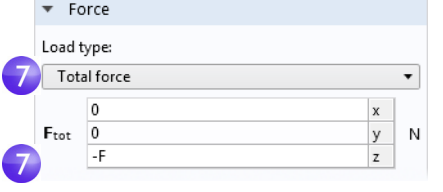
5 In the **Graphics** window, click the **Zoom Box** button on the toolbar and drag the mouse to select the square region shown in the figure to the right. Release the mouse button to zoom in on the selected region.



6 Select the top socket face (boundary **111**) by clicking the boundary to highlight it in blue and add it to the **Selection** list.



7 In the **Settings** window for **Boundary Load**, under **Force**, select **Total force** as the **Load type** and enter `-F` in the text field for the *z* component. The negative sign indicates the negative *z* direction (downward). With these settings, the load of 150 N will be distributed uniformly across the selected surface.

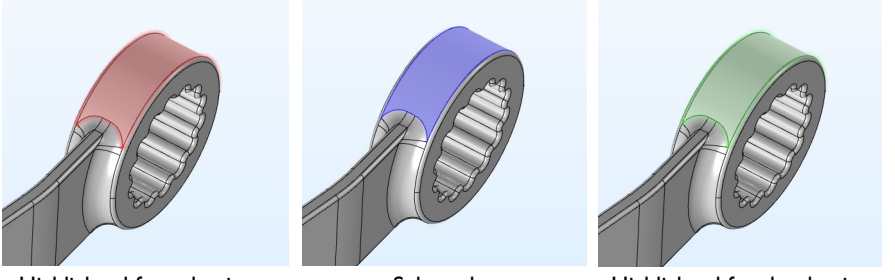


Note that to simplify the modeling process, the mechanical contact between the bolt and the wrench is approximated with a material interface boundary condition. Such an internal boundary condition is automatically defined and guarantees continuity in normal stress and displacement across a material interface. A more detailed analysis including mechanical contact can be done with the Structural Mechanics Module.

Selecting Boundaries and other Geometric Entities

When a boundary is unselected, its color is typically gray. The exception is when you use the material **Appearance** setting available in **Materials**; see page 76. To select a boundary, first hover over it. This highlights the boundary in red, assuming the boundary was previously unselected. Now, click to select the boundary by using the left mouse button. The boundary now turns blue. Its boundary number will appear in the **Selection** list in the **Settings** window of the corresponding boundary condition. Once a boundary is selected and you hover over it again, the boundary turns green. If you click a boundary highlighted in green, the boundary is deselected and now turns gray again. The same technique for selecting and deselecting is applicable to geometry objects, domains, boundaries, edges, and points.

The figure below shows the different selection states for a boundary.



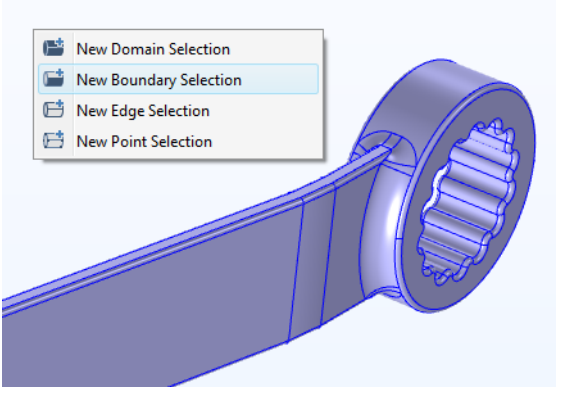
Highlighted for selection | Selected | Highlighted for deselection

To select internal surfaces that are hidden behind the one closest to you, use one of the following methods to cycle through the boundaries at the mouse pointer: roll the mouse scroll wheel, use the keyboard + and - buttons, use the keyboard arrow-up and arrow-down buttons, or use two-finger drag on a touchpad.

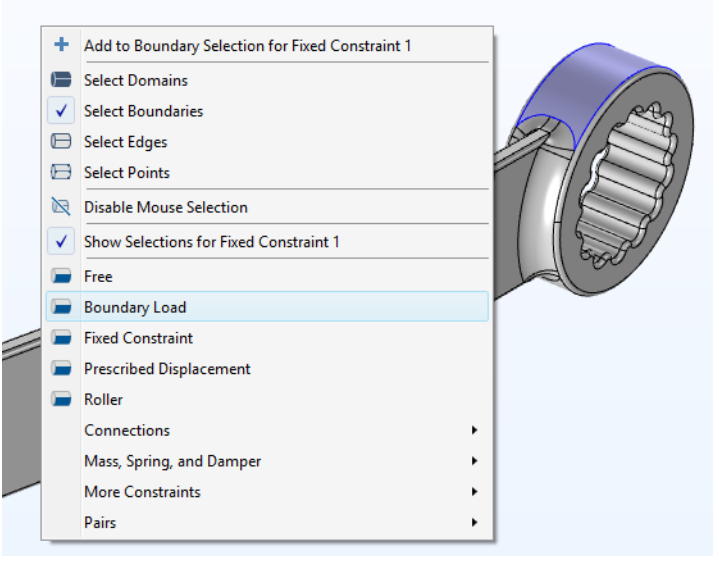
Selecting Using the Graphics Window Context Menus

As an alternative selection method, as of version 5.5, you can right-click in the **Graphics** window and select one of the options for domain, boundary, edge, or point selection.

For example, to assign a **Boundary Load** boundary condition, first right-click and select **New Boundary Selection**, as shown in the figure below.



Click to select the boundaries to which you would like to apply the boundary condition. (In this case, the top socket face; boundary **111**.) Then, right-click again and select **Boundary Load**, as shown in the figure below.



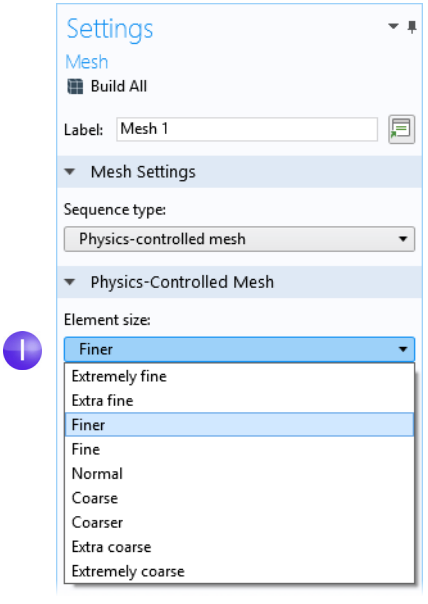
You can also use this method to add additional boundaries to the selections of an already created boundary condition. In the example above, you could, for example, select the menu option **Add to Boundary Selection for Fixed Constraint 1** (although it wouldn't be applicable in this example).

Mesh

The mesh settings determine the resolution of the finite element mesh used to discretize the model. The finite element method divides the model into small elements of geometrically simple shapes, in this case tetrahedra. In each tetrahedron, a set of polynomial functions is used to approximate the structural displacement field — how much the object deforms in each of the three coordinate directions. The stress is then computed by differentiating the displacement field.

In this example, because the geometry contains small edges and faces, you will define a slightly finer mesh than the default setting suggests. This will better resolve the variations of the stress field and give a more accurate result. Refining the mesh size to improve computational accuracy always involves some sacrifice in speed and typically requires increased memory usage.

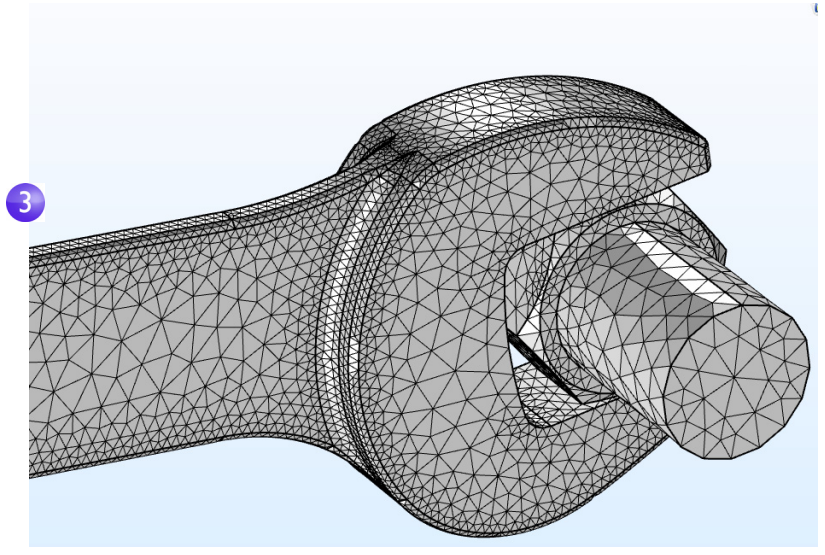
1 In the **Model Builder**, under **Component 1**, click **Mesh 1**. In the **Settings** window for **Mesh**, under **Mesh Settings**, select **Finer** from the **Element size** list.



2 Click the **Build All** button in the **Settings** window or on the **Mesh** toolbar.

The resulting mesh will require about 5GB to solve. If you have a computer with less than 5GB of RAM you can select **Fine** from the **Element size** list (instead of **Finer**.)

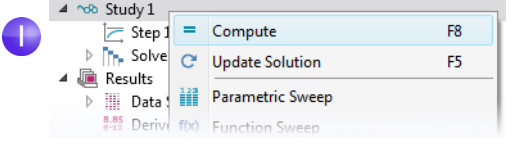
3 After a few seconds, the mesh is displayed in the **Graphics** window. Rotate the wrench to take a look at the element size distribution (this may vary a bit depending on the version you are running).



Study

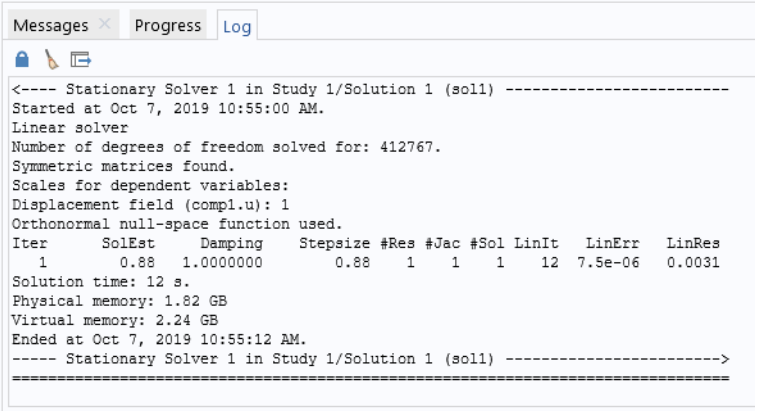
In the beginning of setting up the model, you selected a **Stationary** study, which implies that a stationary solver will be used. For this to be applicable, the assumption is that the load, deformation, and stress do not vary in time. To start the solver:

1 Right-click **Study 1** and select **Compute** (or press F8).



After a few seconds of computation time, the default plot is displayed in the **Graphics** window. During computation, a **Progress** window shows progress information from the solvers. After the computation you can find other useful information about the computation in the **Messages** and **Log** windows; Click the **Messages** and **Log** tabs under the **Graphics**

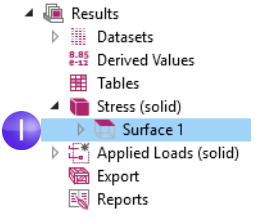
window to see the kind of information available to you. The **Messages** window can also be opened from the **Windows** drop-down list in the **Home** tab of the ribbon.



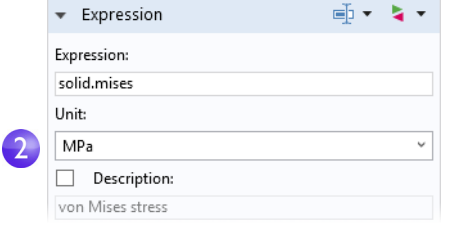
Results

The von Mises stress is displayed in the **Graphics** window in a default Surface plot with the displacement visualized using a Deformation subnode. Change the default unit (N/m^2) to the more suitable MPa as shown in the following steps.

1 In the **Model Builder**, expand the **Results > Stress (solid)** node, then click **Surface 1**.

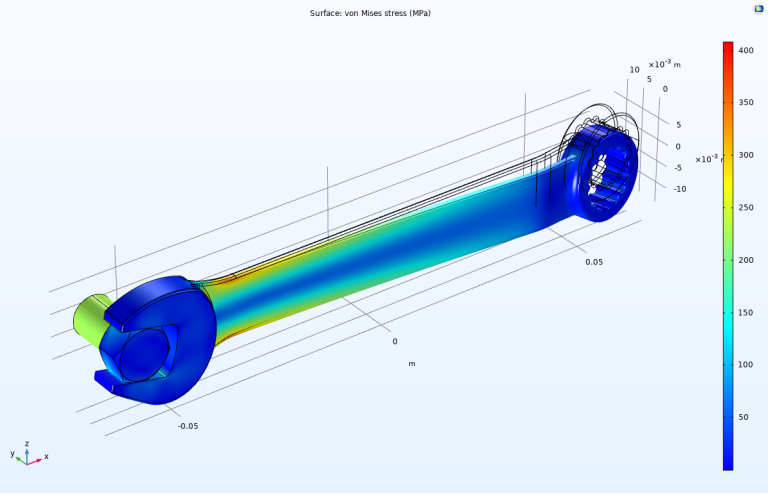


2 In the **Settings** window under **Expression**, from the **Unit** list, select **MPa** (or type `MPa` in the field).



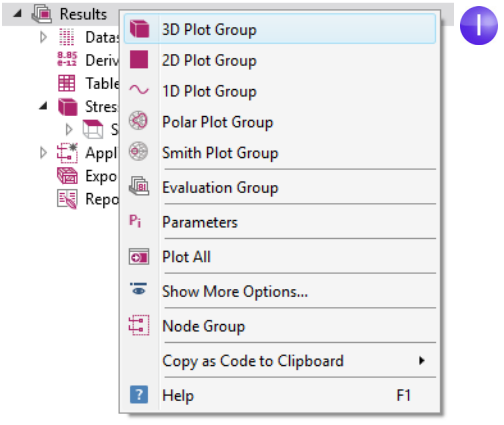
3 Click the **Plot** button on the toolbar of the **Settings** window for the **Surface** plot and then click the **Go to Default 3D View** button on the **Graphics window** toolbar.

The von Mises stress distribution is shown in the bolt and wrench under an applied vertical load.

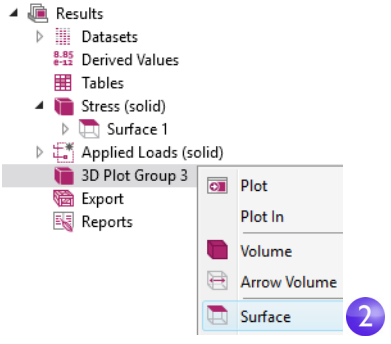


For a typical steel used for tools like a wrench, the yield stress is about 600 MPa, which means that we are getting close to plastic deformation for our 150 N load (which corresponds to about 34 pounds force). You may also be interested in a safety margin of, say, a factor of three. To quickly assess which parts of the wrench are at risk of plastic deformation, you can plot an inequality expression such as `solid.mises>200[MPa]`.

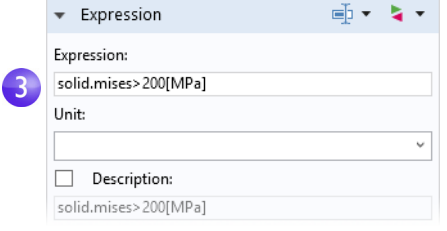
1 Right-click the **Results** node and add a **3D Plot Group**.



2 Right-click the **3D Plot Group 3**node and select **Surface**.



3 In the **Settings** window for **Surface,** click the **Replace Expression** button and select **Model > Component 1 > Solid Mechanics > Stress > solid.mises-von Mises stress - N/m^2** by double-clicking. When you know the variable name beforehand, you can also directly enter `solid.mises` in the **Expression** field. Now edit this expression to: `solid.mises>200[MPa]`.

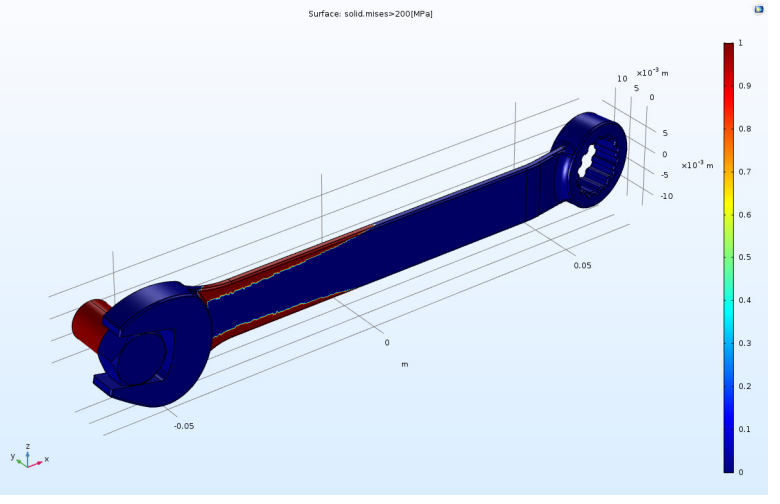


This is a boolean expression that evaluates to either 1 for true or 0 for false. In areas where the expression evaluates to 1, the safety margin is exceeded.

4 Click the **Plot** button.

5 In the **Model Builder**, click **3D Plot Group 3**. Press F2 and, in the **Rename 3D Plot Group** dialog box, enter `Safety Margin`. Click **OK**. Alternatively, edit the label at the top of the **Settings** window.

The resulting plot shows that the stress in the bolt is high, but the focus of this exercise is on the wrench. If you wish to comfortably certify the wrench for a 150 N load with a factor-of-three safety margin, you would need to change the handle design somewhat, such as by making it wider.



5

You may have noticed that the manufacturer, for various reasons, has chosen an asymmetric design for the wrench. Because of that, the stress field may be different if the wrench is flipped around. Try now, on your own, to apply the same force in the other direction and visualize the maximum von Mises stress to see if there is any difference.

Convergence Analysis

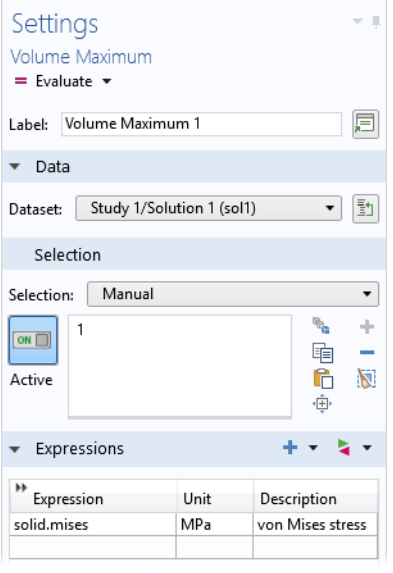
The results from a simulation software such as COMSOL Multiphysics can shorten design times dramatically by, for example, reducing the number of experiments or product tests. However, simulation software is not a substitute for real-world testing. This is especially important if there are risks for physical or environmental damage. One way to reduce (but not eliminate) the risk of drawing the wrong conclusions from simulation results is to perform a convergence analysis.

To check the accuracy of the computed maximum von Mises stress in the wrench, you can now continue with a mesh convergence analysis. Do that by using a finer mesh and therefore a higher number of degrees of freedom (DOFs).

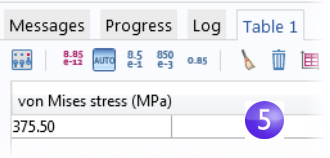
This section illustrates some more in-depth functionality and the steps below could be skipped at a first reading. In order to run the convergence analysis below, a computer with at least 4GB of memory (RAM) is recommended.

Evaluating the Maximum von Mises Stress

1. To study the maximum von Mises stress in the wrench, in the **Results** section of the model tree, right-click the **Derived Values** node and select **Maximum > Volume Maximum**.
2. In the **Settings** window for **Volume Maximum**, under **Selection**, choose **Manual** and select the wrench, domain 1, by clicking on the wrench in the **Graphics** window. We will only consider values in the wrench domain and neglect those in the bolt.
3. In the first row of the **Expressions** table, in the **Expression** column, type `solid.mises`. To find the expression for the von Mises stress, you can also click the **Replace Expression** button and browse to this variable.
4. In the **Expressions table**, type `MPa` as the **Unit**.



5 In the **Settings** window for **Volume Maximum**, click **Evaluate** to evaluate the maximum stress. The result will be displayed in a Table window and will be approximately 375 MPa.



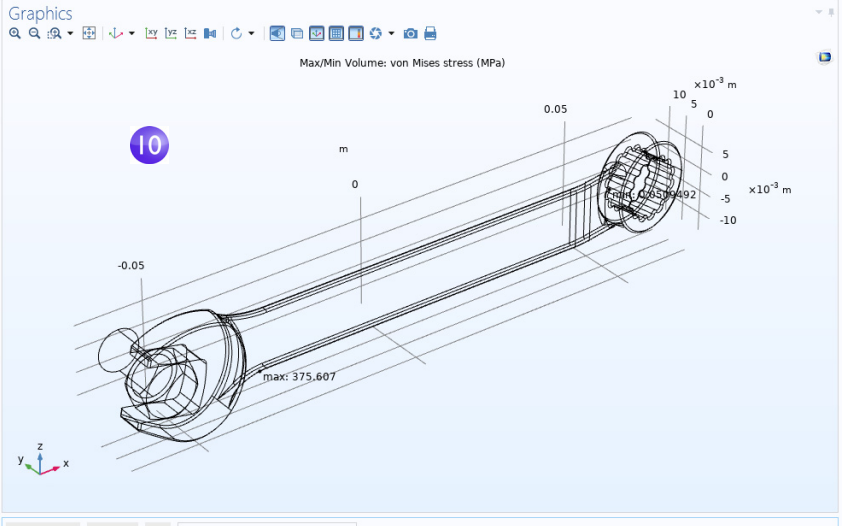
6 To see where the maximum value is attained, you can use a **Max/Min Volume** plot. Right-click the **Results** node and add a **3D Plot Group**. Next, right-click the **3D Plot Group 4** node and select **More Plots > Max/Min Volume**.

7 To plot values in the wrench and not in the bolt, right-click the **Max/Min Volume** node and select **Selection** and select the wrench domain number 1.

8 In the **Settings** window for **Max/Min Volume**, in the **Expression** text field, type `solid.mises`.

9 In the **Settings** window under **Expression**, from the **Unit** list, select **MPa** (or enter `MPa` in the field).

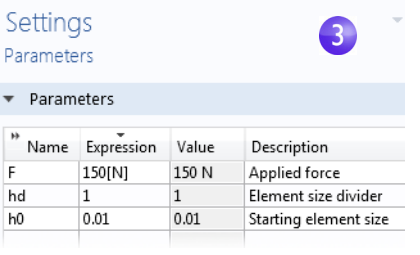
10 Click the **Plot** button. This type of plot simultaneously shows the location of the maximum and minimum values and also their coordinate location in the table below (the exact values may differ depending on the version you are running).



Parameterizing the Mesh

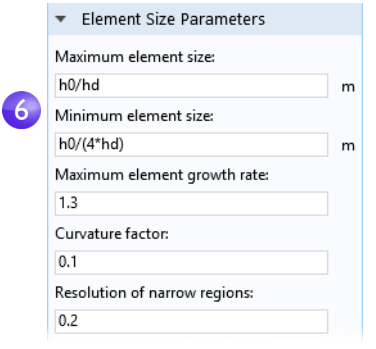
We will now define a parametric sweep for successively refining the mesh size while solving and then finally plot the maximum von Mises stress vs. mesh size. First, we define the parameters that will be used for controlling the mesh density.

1. In the **Model Builder**, click **Parameters 1** under **Global Definitions**.
2. Go to the **Settings** window for **Parameters**. In the **Parameters** table (or under the table in the fields), enter these settings:
   - In the **Name** column or field, enter `hd`. This parameter will be used in the parametric sweep to control the element size.
   - In the **Expression** column or field, enter `1`.
   - In the **Description** column or field, enter `Element size divider`.
3. Now, enter another parameter with **Name** `h0`, **Expression** `0.01`, and **Description** `Starting element size`. This parameter will be used to define the element size at the start of the parametric sweep.



4. In the **Model Builder**, under **Component 1,** click **Mesh 1**. In the **Settings** window for **Mesh**, select **User-controlled mesh** from the **Sequence type** list.
5. Under **Mesh 1**, click the **Size** node.
6. In the **Settings** window for **Size,** under **Element Size**, click the **Custom** button.

   Under **Element Size Parameters**, enter:
   - `h0/hd` in the **Maximum element size** field.
   - `h0/(4*hd)` in the **Minimum element size** field.
   - `1.3` in the Maximum element growth rate field.
   - `0.1` in the Curvature factor field.
   - `0.2` in the Resolution of narrow regions field.

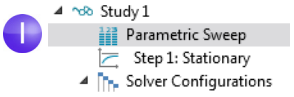


See page 85 for more information on the element size parameters.

Parametric Sweep and Solver Settings

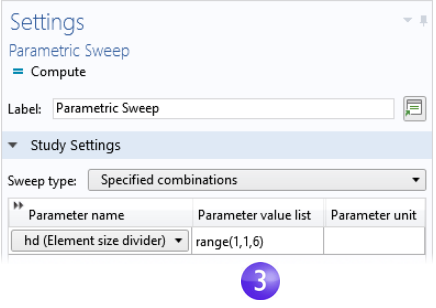
As a next step, add a parametric sweep for the parameter `hd`.

1 In the **Model Builder**, right-click **Study 1** and select **Parametric Sweep**. A **Parametric Sweep** node is added to the **Model Builder** sequence.



2 In the **Settings** window for **Parametric Sweep**, under the table in the **Study Settings** section, click the **Add** button +. From the **Parameter name** list in the table, select `hd`.

3 Enter a range of **Parameter values** to sweep for. Click the **Range** button and enter the values in the **Range** dialog box. In the **Start** field, enter `1`. In the **Step** field, enter `1`, and in the **Stop** field, enter `6`. Click **Replace**. The **Parameter value list** will now display `range(1,1,6)`.



The settings above make sure that as the sweep progresses, the value of the parameter `hd` increases and the maximum and minimum element sizes decrease.

See page 141 for more information on defining parametric sweeps.

For the highest value of `hd`, the number of DOFs will exceed one million. Therefore, we will switch to a more memory-efficient iterative solver.

4 Under **Study 1 > Solver Configurations > Solution 1**, expand the **Stationary Solver 1** node, and right-click **Suggested Iterative Solver** and select **Enable**. Selecting an iterative solver option typically reduces memory usage but can require physics-specific tailoring of the solver settings for efficient computations.

5 Under **General** in the **Settings** window for **Suggested Iterative Solver**, change the **Preconditioning** to **Right**. (This is a low-level solver option, which in this case will suppress a warning message that would otherwise appear. However, this setting does not affect the resulting solution. Preconditioning is a mathematical transformation used to prepare the finite element equation system for using the iterative solver.)

6 Click the **Study 1** node and select **Compute** =, either in the **Settings** window or by right-clicking the node. You can also click **Compute** in the ribbon **Home** or **Study** tab. The computation time will be a few minutes (depending on the computer hardware) and memory usage will be about 4GB.

Results Analysis

As a final step, analyze the results from the parametric sweep by displaying the maximum von Mises stress in a table.

1 In the Model Builder under **Results > Derived Values**, select the **Volume Maximum 1** node MAX.

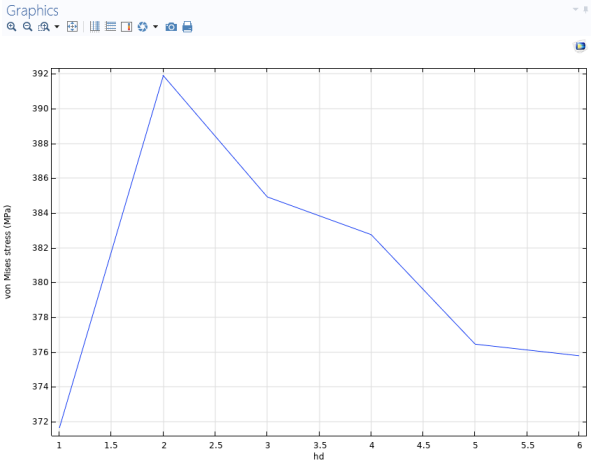
The solutions from the parametric sweep are stored in a new **Dataset** named **Study 1/Parametric Solutions 1**. Now, change the **Volume Maximum** settings accordingly:

2 In the **Settings** window for **Volume Maximum**, change the **Dataset** to **Study 1/Parametric Solutions 1**.

3 Click the arrow next to the **Evaluate** button at the top of the **Settings** window for **Volume Maximum** and select to evaluate in a **New Table**. This evaluation may take 20 seconds or so.

4 To plot the results in the **Table**, click the **Table Graph** button at the top of the **Table** window (the first few values will vary depending on the version due to slight mesh variations for coarser meshes before a converged value has been reached).

hd	von Mises stress (MPa)
1.0000	371.63
2.0000	391.90
3.0000	384.92
4.0000	382.75
5.0000	376.45
6.0000	375.79



It is more interesting to plot the maximum value vs. the number of DOFs. This is possible by using the built-in variable `numberofdofs`.

5 Right-click the **Derived Values** node and select **Global Evaluation**.

6 In the **Settings** window for **Global Evaluation**, change the **Dataset** to **Study 1/Parametric Solutions 1**.

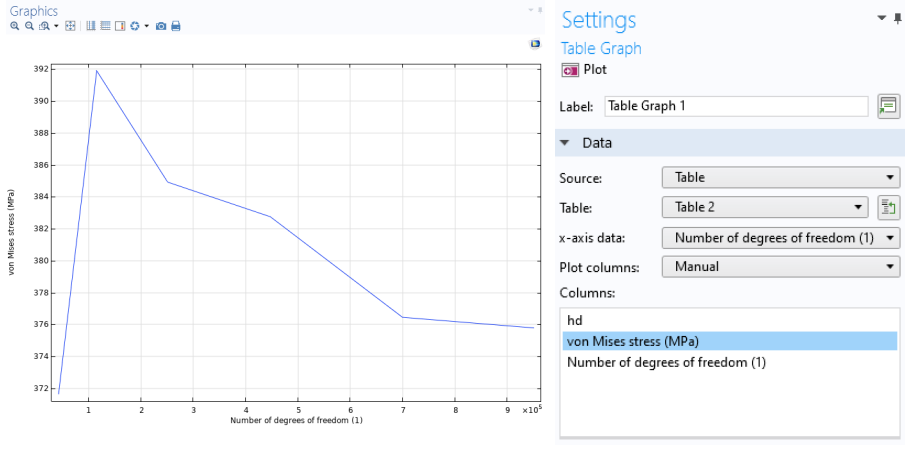
7 In the **Expressions** field, enter `numberofdofs`.

You can type the first few characters of the variable name and then use the keyboard shortcut Ctrl+Space to get a variable suggestion from the list of available variables.

8 In the **Settings** window for **Global Evaluation**, click the arrow next to the **Evaluate** button and select the option to evaluate in **Table 2**. This displays the DOF values for each parameter next to the previously evaluated data.

This convergence analysis shows that the computed value of the maximum von Mises stress in the wrench handle will decrease from the high original value, for a mesh with about 40,000 DOFs, to 375 MPa for a mesh with about 1,000,000 DOFs. It also shows that above 500,000 DOFs the results have essentially the same accuracy as for 1,000,000 DOFs.

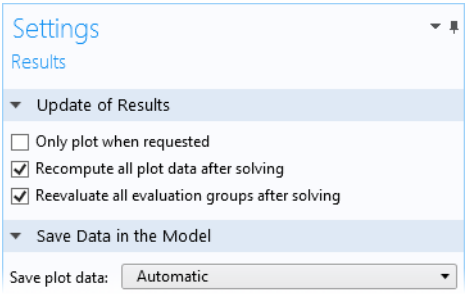
The figure below shows the associated **Table Graph** and **Table Graph Settings** window.



Note that depending on the version of COMSOL Multiphysics that you are running, the above values may vary slightly due to variations in the number of elements generated by the meshing algorithm.

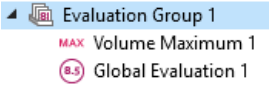
As an alternative to **Derived Values**, you can use **Evaluation Groups**. An **Evaluation Group** shares some of the properties of **Plot Groups** as well as **Derived Values**. An **Evaluation Group** comes with an integrated table whereas **Derived Values** outputs to a table of your choice. In addition, **Evaluation Groups** allows you to automatically

re-evaluate after solving. To enable this, select the corresponding check box in the **Settings** window of the **Results** node as shown in the figure below.

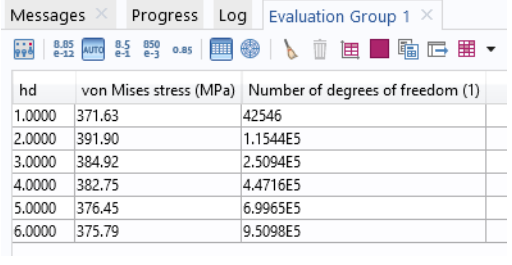


In this **Settings** window you can also request to regenerate all plots after solving.

You add an **Evaluation Group** in the same way as a **Plot Group** by right-clicking the **Results** node. The figure below shows an **Evaluation Group** that performs the same task as the **Derived Values** described earlier.



The figure below shows the corresponding **Evaluation Group** table.



hd	von Mises stress (MPa)	Number of degrees of freedom (1)
1.0000	371.63	42546
2.0000	391.90	1.1544E5
3.0000	384.92	2.5094E5
4.0000	382.75	4.4716E5
5.0000	376.45	6.9965E5
6.0000	375.79	9.5098E5

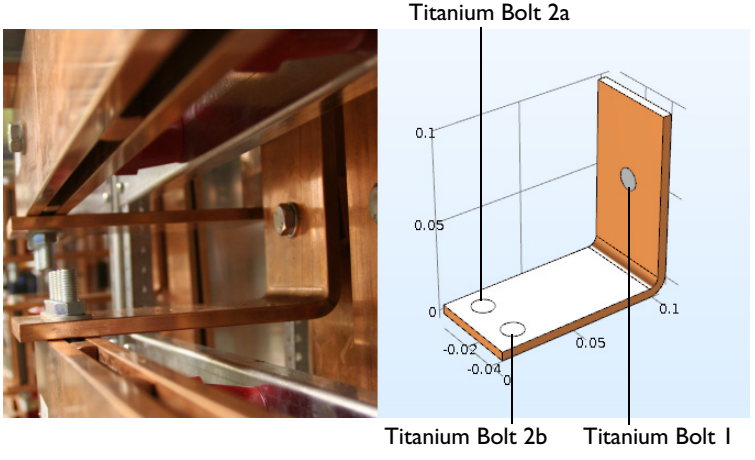
This concludes the wrench tutorial.

Electrical Heating in a Busbar

This tutorial demonstrates the concept of multiphysics modeling in COMSOL Multiphysics using the Model Builder. You will learn how to use one of the many preconfigured multiphysics combinations as well as subsequently add the effects of physics phenomena not initially included in the analysis. At the end, you will have built a truly multiphysics model.

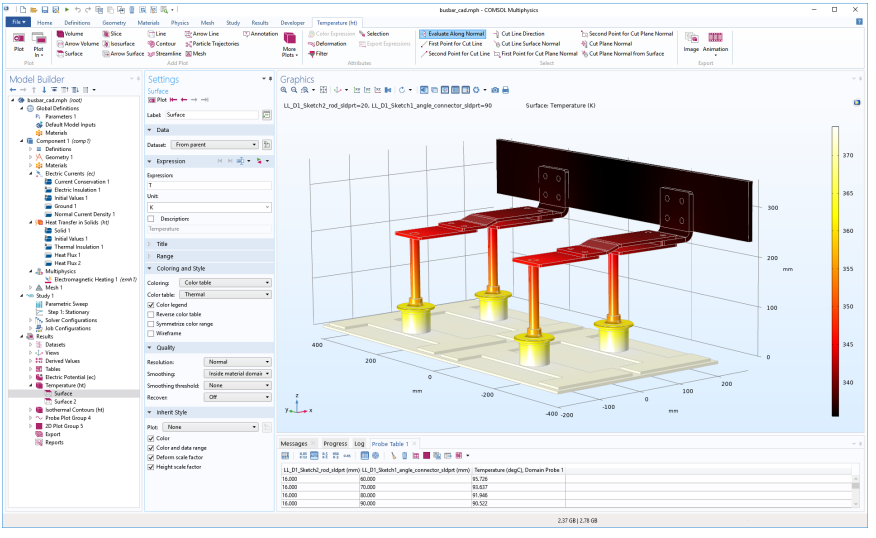
The model that you are about to create analyzes a busbar designed to conduct a high amount of direct current in an industrial setting (see picture below). The current conducted in the busbar, from bolt 1 to bolts 2a and 2b, produces heat due to the resistive losses, a phenomenon referred to as Joule heating. The busbar is made of copper, while the bolts are made of a titanium alloy. Busbar bolts are usually made of steel, but in this example, we will assume a highly corrosive environment. Therefore, a titanium alloy was chosen.

Under normal operational conditions, the currents are predominantly conducted through the copper. This example, however, illustrates the effects of an unwanted electrical loading of the busbar through the bolts. The fact that there are different materials is important because titanium has a lower electrical conductivity than copper and will be subjected to a higher current density.



The goal of your simulation is to precisely calculate how much the busbar heats up. Once you have captured the basic multiphysics phenomena, you will have the chance to investigate thermal expansion that generates structural stresses and strains in the busbar and the effects of cooling by an air stream.

The Joule heating effect is described by conservation laws for electric current and energy. Once solved for, the two conservation laws give the temperature and electric field, respectively. All surfaces, except the bolt contact surfaces, are cooled by natural convection in the air surrounding the busbar. You can assume that the exposed parts of the bolt do not contribute to the cooling or heating of the device. The electric potential at the upper-right vertical bolt surface is 20 mV and the potential at the two horizontal surfaces of the lower bolts is 0 V. This corresponds to a relatively high and potentially unsafe loading of this type of busbar. More advanced boundary conditions for electromagnetics analysis are available with the AC/DC Module, such as the capability to give the total current on a boundary. A more advanced model that includes multiple busbars in an assembly, as shown in the figure below, can be found in some of the add-on products, including the AC/DC Module and a few of the LiveLink™ products for CAD.



Busbar Model Overview

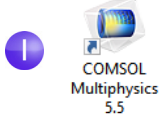
More in-depth and advanced topics included in this tutorial are used to show you some of the many options available in COMSOL Multiphysics. The following topics are covered:

- “Parameters, Functions, Variables, and Couplings” on page 110, where you learn how to define functions and nonlocal couplings.
- “Material Properties and Material Libraries” on page 114 shows you how to customize a material and add it to your own material library.

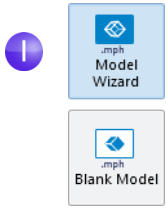
- “Adding Meshes” on page 116 gives you the opportunity to add and define two different meshes and compare them in the Graphics window.
- “Adding Physics” on page 119 explores the multiphysics capabilities by adding solid mechanics and laminar flow to the busbar model.
- “Parametric Sweeps” on page 141 shows you how to vary the width of the busbar using a parameter and then solve for a range of parameter values. The result is a plot of the average temperature as a function of the width.
- “Parallel Computing” on page 152, gives you an overview of how to solve on clusters.
- “COMSOL Multiphysics Client-Server” on page 156, gives you an introduction to using the COMSOL Multiphysics client-server mode of operation.

Model Wizard

1 To open the software, double-click the COMSOL Multiphysics icon on the desktop.



When the software opens, click the **Model Wizard** button. You can also start the Model Wizard at any time by selecting **New** from the **File** menu. Then, choose **Model Wizard**.



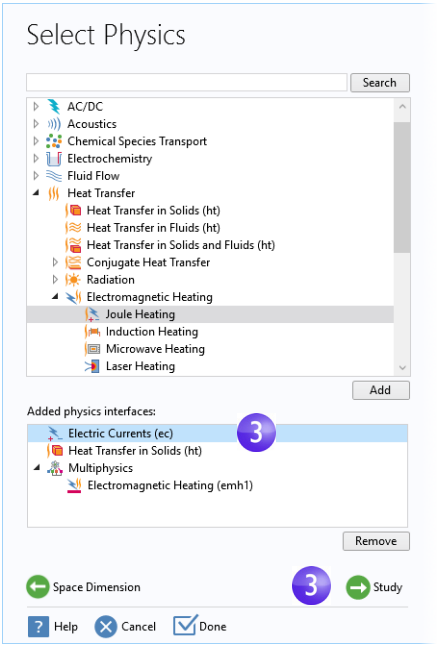
2 In the **Select Space Dimension** window, click **3D**.

3 In the **Select Physics** window, expand **Heat Transfer > Electromagnetic Heating**. Then, right-click **Joule Heating** and choose **Add Physics**. Click the **Study** button.

You can also double-click or click the **Add** button to add physics.

(Another way to add physics is to open the **Add Physics** window by right-clicking the **Component** node in the Model Builder and selecting **Add Physics**.)

Note that you may have fewer items in your physics list depending on the add-on modules installed. The figure on the right is shown for the case where all add-on modules are installed.



4 In the **Select Study** window, click to select the **Stationary** study type.

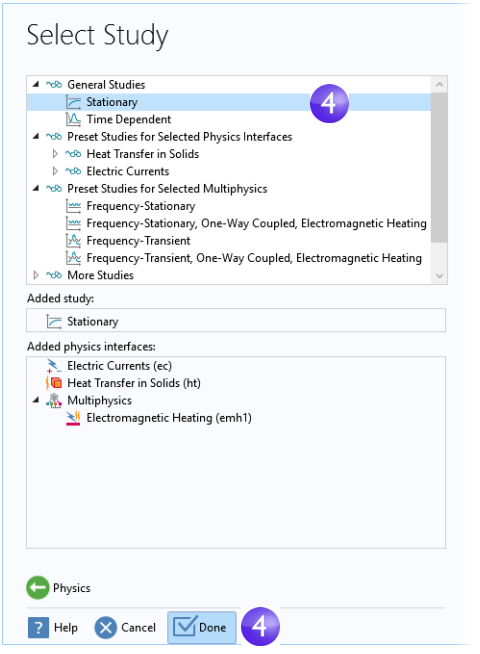
Click the **Done** button.

Choosing a stationary study means that we are solving for the steady-state temperature of the busbar when it has reached thermal equilibrium with its surroundings.

A preconfigured **Study** has solver and equation settings adapted to the selected combination of physics interface; in this example, **Joule Heating**.

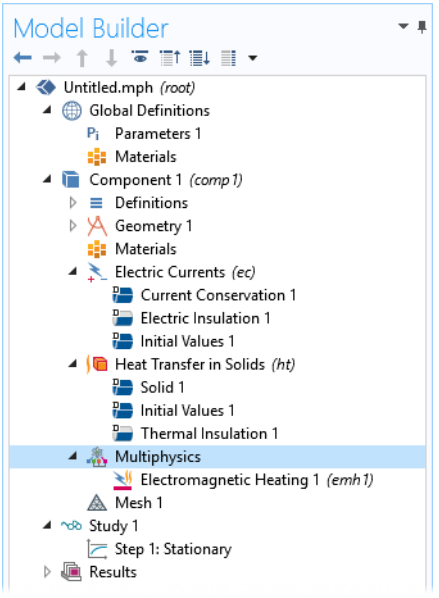
Any selection from the **Custom Studies** branch may require manual tuning.

Note that you may have fewer study types in your study list depending on the installed add-on modules.



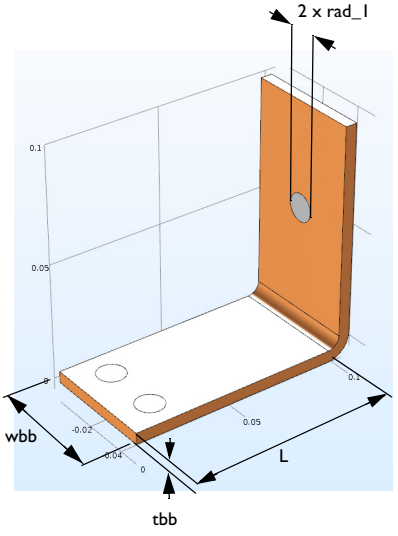
A physics interface is a user interface for a specific physics area that defines the equations together with settings for mesh generation, solvers, visualization, and results. A multiphysics interface is a combination of physics interfaces together with multiphysics couplings. The **Joule Heating** multiphysics interface consists of two physics interfaces, **Electric Currents** and **Heat Transfer in Solids**, together with the multiphysics coupling **Electromagnetic Heating** that appear in the Multiphysics branch. This method of mixing physics is very flexible and

makes it possible to use all the capabilities of the participating single-physics interfaces for multiphysics purposes.



To save time, it is recommended that you load the geometry from a file. In that case, you can skip to "Geometry" on page 68.

If, on the other hand, you want to draw the geometry yourself, the **Global Definitions** node is where you define the parameters. First, complete steps 1 through 3 below to define the parameter list for the model. Then, follow step 4 and skip to the section "Appendix A — Building a Geometry" on page 159.



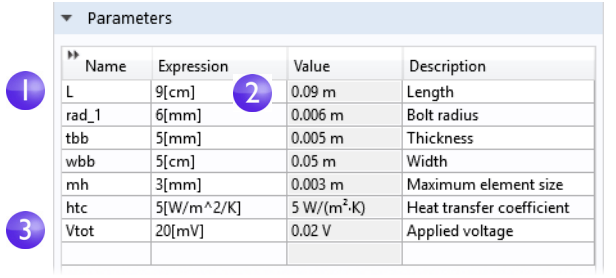
The **Global Definitions** node in the Model Builder stores **Parameters**, **Variables**, and **Functions** with a global scope. The model tree can hold several model components simultaneously, and the definitions with a global scope are made available for all components. In this particular example, there is only one **Component** node in which the parameters are used. If you wish to limit the scope to this single component, you could define, for example, **Variables** and **Functions** in the **Definitions** subnode, available directly under the corresponding **Component** node. However, no **Parameters** can be defined here because **Parameters** that are defined under **Global Definitions** are always global.

Since you will run a parametric study of the geometry later in this example, define the geometry using parameters from the start. In this step, enter parameters for the length of the lower part of the busbar, `L`; the radius of the titanium bolts, `rad_1`; the thickness of the busbar, `tbb`; and the width of the device, `wbb`.

You will also add the parameters that control the mesh, `mh`; a heat transfer coefficient for cooling by natural convection, `htc`; and a value for the voltage across the busbar, `Vtot`.

1. Under **Global Definitions**, choose **Parameters**. In the **Parameters** table, click the first row under **Name** and enter `L`.
2. Click the first row under **Expression** and enter the value of `L`, `9[cm]`. You can enter the unit inside the square brackets.
3. Continue adding the other parameters: `rad_1`, `tbb`, `wbb`, `mh`, `htc`, and `Vtot` according to the **Parameters** list below. It is a good idea to enter descriptions for

variables in case you want to share the model with others and for your own future reference.



Click the **Save** button on the Quick Access Toolbar and name the model `busbar.mph` or use the corresponding **File** menu option. Then, go to "Appendix A — Building a Geometry" on page 159.

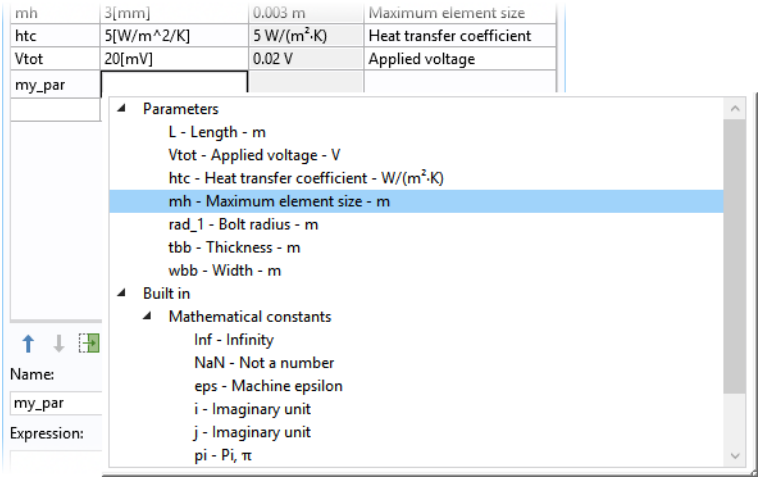
Auto Completion and Find for Parameters and Variables

The model tree contains a large number of parameters and variables. To make it easier to find them, you can use auto completion and find tools as described below.

Auto Completion

You can use Ctrl+Space for auto completion of parameter and variable names. For example, in the **Parameters** table, click an empty **Expression** field and then press

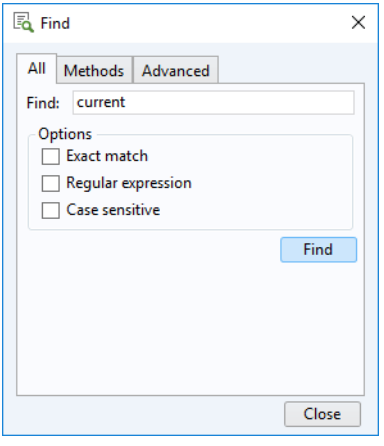
Ctrl+Space, you will get a list of parameters and variables, as shown in the figure below.



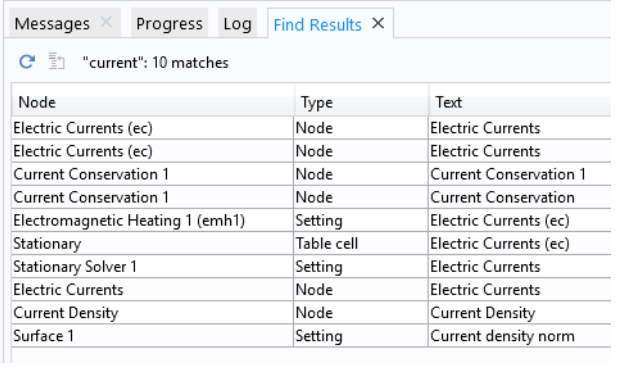
This functionality is also available for other edit fields in the Model Builder.

FIND

To find parameters and variables in the model tree, you can click the **Find** button in the **Quick Access Toolbar** or use the keyboard shortcut Ctrl+F. This opens the **Find** window.



The results of the search are displayed in the **Find Results** window, as shown in the figure below, coming from the completed busbar model.



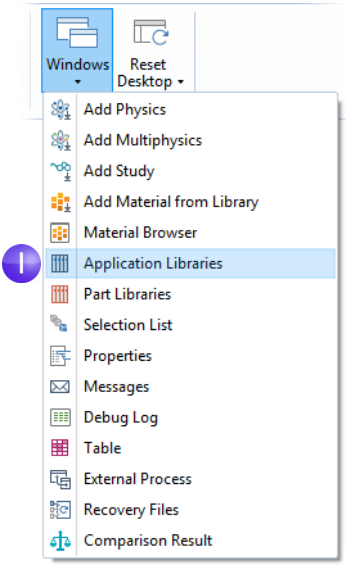
Node	Type	Text
Electric Currents (ec)	Node	Electric Currents
Electric Currents (ec)	Node	Electric Currents
Current Conservation 1	Node	Current Conservation 1
Current Conservation 1	Node	Current Conservation
Electromagnetic Heating 1 (emh1)	Setting	Electric Currents (ec)
Stationary	Table cell	Electric Currents (ec)
Stationary Solver 1	Setting	Electric Currents
Electric Currents	Node	Electric Currents
Current Density	Node	Current Density
Surface 1	Setting	Current density norm

You can double-click any of the rows in this window to open the corresponding **Settings** window.

Geometry

This section describes how the geometry can be opened from the Application Libraries. The physics, study, parameters, and geometry are included with the model file you are about to open.

1 Select **Application Libraries** from the **Windows** drop down menu in the **Home** tab.



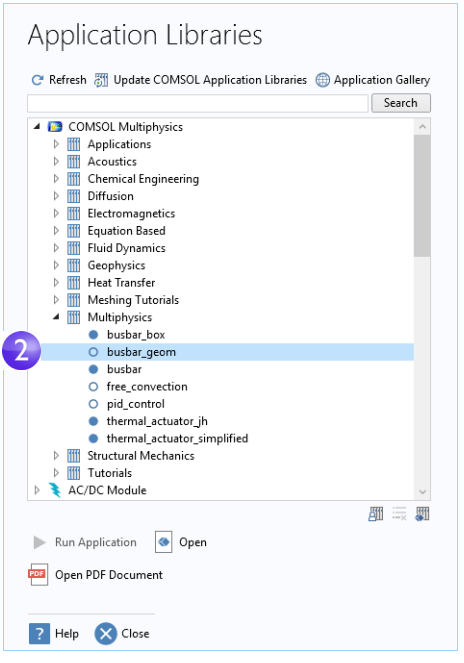
2 In the **Application Libraries** tree under **COMSOL Multiphysics > Multiphysics**, select **busbar geom.**

To open the file, you can:

- Double-click the name
- Right-click and select an option from the menu
- Click one of the buttons under the tree

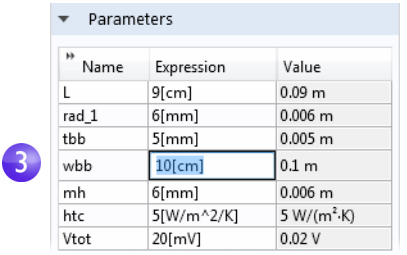
You can select **No** if prompted to save `Untitled.mph`.

The geometry in this file is parameterized. In the next few steps, we will experiment with different values for the width parameter, `wbb`.

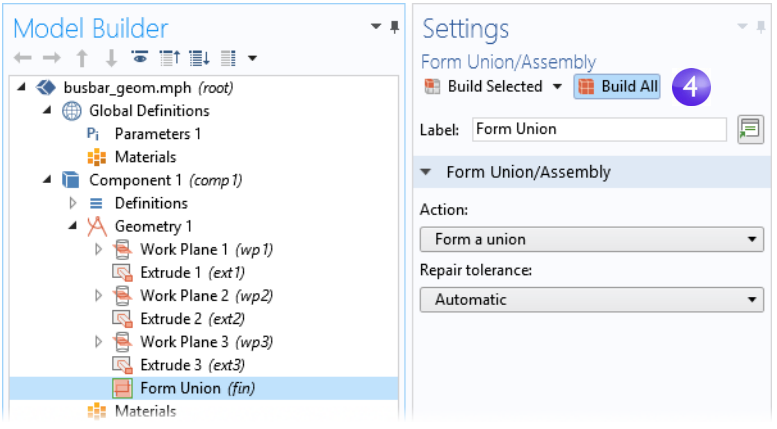


3 Under **Global Definitions**, click the **Parameters** node.

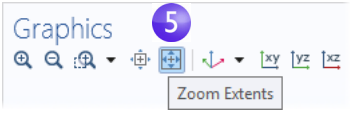
In the Settings window for **Parameters**, click in the **Expression** column for the `wbb` parameter and enter `10[cm]` to change the value of the busbar width.

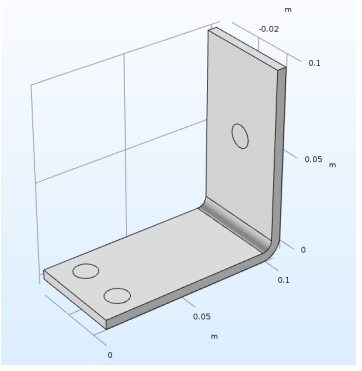


4 In the **Model Builder**, under **Component 1 > Geometry 1**, click the **Form Union** node and then the **Build All** button in the **Settings** window to rerun the geometry sequence. You can also use the ribbon and click **Build All** from the **Geometry** group in the **Home** tab.

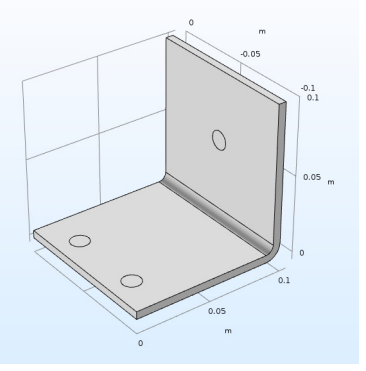


5 In the **Graphics** toolbar, click the **Zoom Extents** button to see the wider busbar in the **Graphics** window.





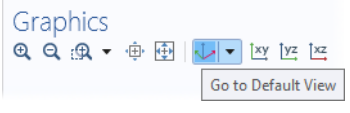
wbb = 5cm



wbb = 10cm

6 Experiment with the geometry in the **Graphics** window:

- To rotate the busbar, click and drag the pointer anywhere in the **Graphics** window.
- To move it, right-click and drag.
- To zoom in and out, click the scroll wheel, continue holding it, and drag.
- To get back to the original position, click the **Go to Default 3D View** button on the toolbar.



7 Return to the **Parameters** table and change the value of `wbb` back to `5[cm]`.

8 In the **Model Builder**, click the **Form Union** node and then click the **Build All** button to rerun the geometry sequence.

9 On the **Graphics** toolbar, click the **Zoom Extents** button .

7

Parameters

Name	Expression	Value
L	9[cm]	0.09 m
rad_1	6[mm]	0.006 m
tbb	5[mm]	0.005 m
wbb	5[cm]	0.05 m
mh	6[mm]	0.006 m
htc	5[W/m^2/K]	5 W/(m²·K)
Vtot	20[mV]	0.02 V

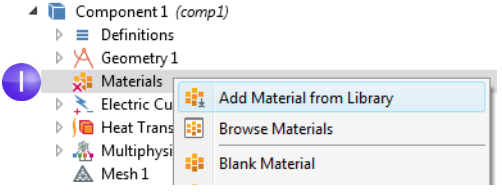
10 If you built the geometry yourself, you are already using the `busbar.mph` file, but if you opened the file from the Application Libraries, select **Save As from the File menu** and rename the model `busbar.mph`.

After creating or importing the geometry, it is time to define the materials.

Materials

The **Materials** node stores the material properties for all physics and geometrical domains in a **Component** node. The busbar is made of copper and the bolts are made of a titanium alloy. Both of these materials are available from the **Built-In** material database.

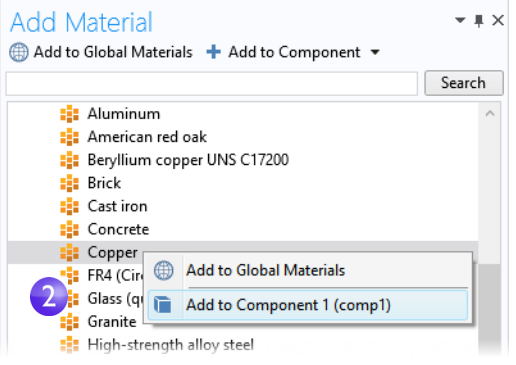
1 In the **Model Builder**, right-click **Component 1 > Materials** and select **Add Material from Library**. By default, the window will open at the right-hand side of the desktop. You can move the window by clicking on the window title and then dragging it to a new location. While dragging the window, you will be presented with several options for docking.



The **Materials** node will show a red × in the lower-left corner if you try to solve without first defining a material, which we will go over in the next few steps.

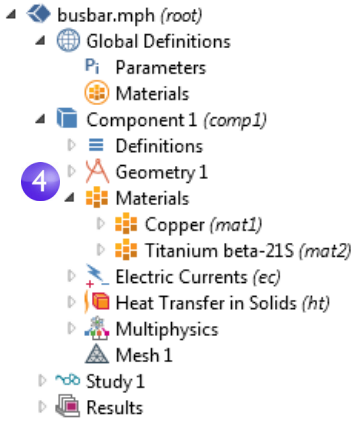
2 In the **Add Material** window, expand the **Built-In** materials folder and locate **Copper**. Right-click **Copper** and select **Add to Component 1**. Alternatively, you can double-click.

A **Copper** node is added to the **Model Builder**.

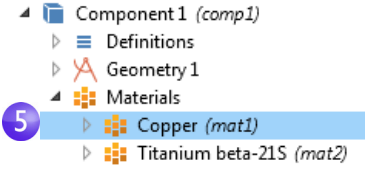


3 In the **Add Material window**, scroll to **Titanium beta-21S** in the **Built-In** material folder list. Right-click and select **Add to Component 1**.

4 In the **Model Builder**, collapse the **Geometry 1** node to get an overview of the model.



5 Under the **Materials** node, click **Copper** .



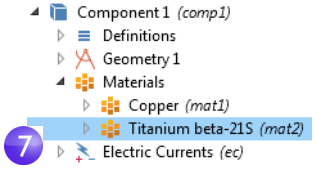
6 In the **Settings** window for **Material**, examine the **Material Contents** section.

Property	Variable	Value	Unit	Property group
Electrical conductivity	sigma_iso...	5.998e7[S/m]	S/m	Basic
Heat capacity at constant pressure	Cp	385[J/(kg*K)]	J/(kg·K)	Basic
Relative permittivity	epsilonr_i...	1	1	Basic
Density	rho	8960[kg/m^3]	kg/m³	Basic
Thermal conductivity	k_iso ; kii...	400[W/(m*K)]	W/(m·K)	Basic
Relative permeability	mur_iso ;...	1	1	Basic
Coefficient of thermal expansion	alpha_iso...	17e-6[1/K]	1/K	Basic
Young's modulus	E	110e9[Pa]	Pa	Young's modulus and Poisson's ratio
Poisson's ratio	nu	0.35	1	Young's modulus and Poisson's ratio
Reference resistivity	rho0	1.72e-8[ohm...	Ω·m	Linearized resistivity
Resistivity temperature coefficient	alpha	0.0039[1/K]	1/K	Linearized resistivity
Reference temperature	Tref	298[K]	K	Linearized resistivity

The **Material Contents** section has useful information about the material property usage of a model. Properties that are both required by the physics and available from the material are marked with a green check mark . Properties required by the physics but missing in the material are marked with a warning sign . A property that is available but not used in the model is unmarked.

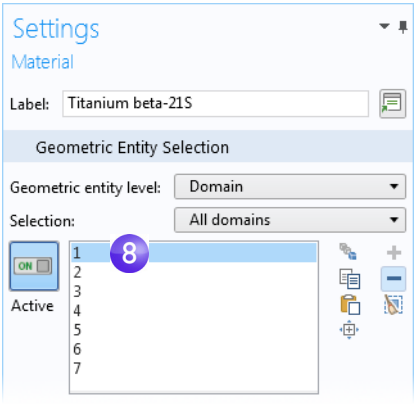
The **Coefficient of thermal expansion**, **Young's modulus**, and **Poisson's ratio** in the table above are not used, but will be needed later when heat-induced stresses and strains are added to the model.

Because the copper material is added first, by default, all parts have copper material assigned. In the next step, you will assign titanium properties to the bolts, which override the copper material assignment for those parts.



7 In the **Model Builder**, click **Titanium beta-21S** .

8 Select **All Domains** from the **Selection** list and then click domain 1 in the list. Now remove domain 1 from the selection list.

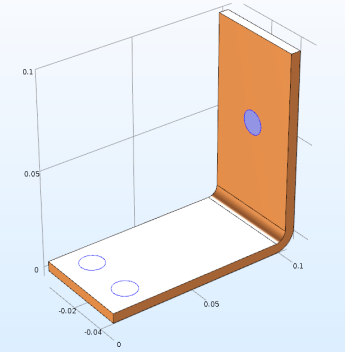


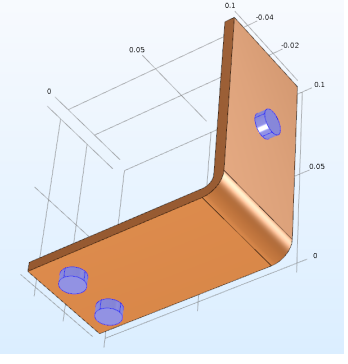
To remove a domain from the selection list (or any geometric entity, such as boundaries, edges, or points), you can use either of these two methods:

-Click domain 1 in the selection list found in the **Settings** window for **Material**. Then, click the **Remove from Selection** button — or press Delete on your keyboard.

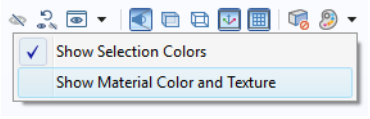
-Alternatively, in the **Graphics** window, click domain 1 to remove it from the selection list.

8 The domains 2, 3, 4, 5, 6, and 7 are highlighted in blue.

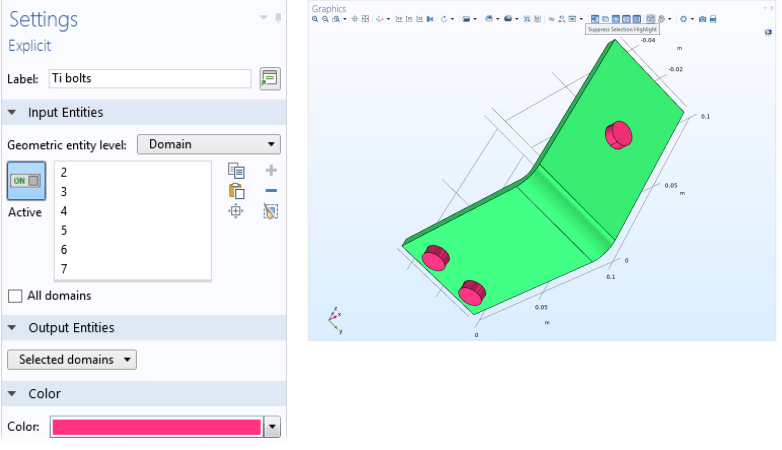




You can enable and disable material color and texture from the **Color** toolbar button in the **Graphics** window. To render the copper components using a realistic color, select **Show Material Color and Texture**. This will also enable material-true rendering of other materials. Here, you can also enable coloring of selections by selecting **Show Selection Colors**.

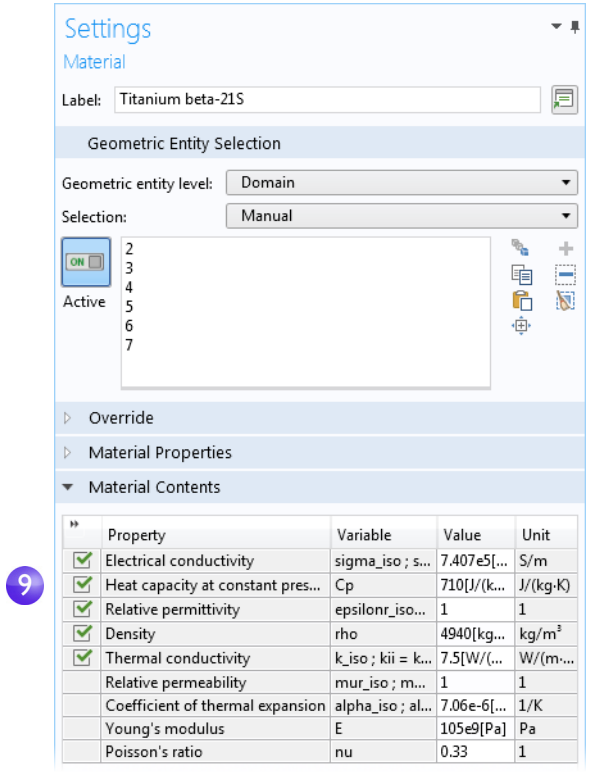


The figures below show the **Settings** and **Graphics** windows corresponding to a selection where a color (pink) other than the default has been used. There is also a second selection (not shown) with another color (green).

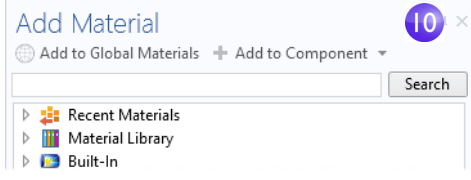


Note that to see the color while creating a selection, you need to click the **Suppress Selection Highlight** button in the **Graphics** toolbar. For more information on **Selections**, see page 122.

9 In the **Settings** window for **Material**, be sure to inspect the **Material Contents** section for the titanium material. All of the properties used by the physics should have a green check mark ☑.



10 Close the **Add Material window** either by clicking the icon in the upper-right corner or by clicking the **Add Material** toggle button in the **Materials** group of the ribbon **Home** tab.

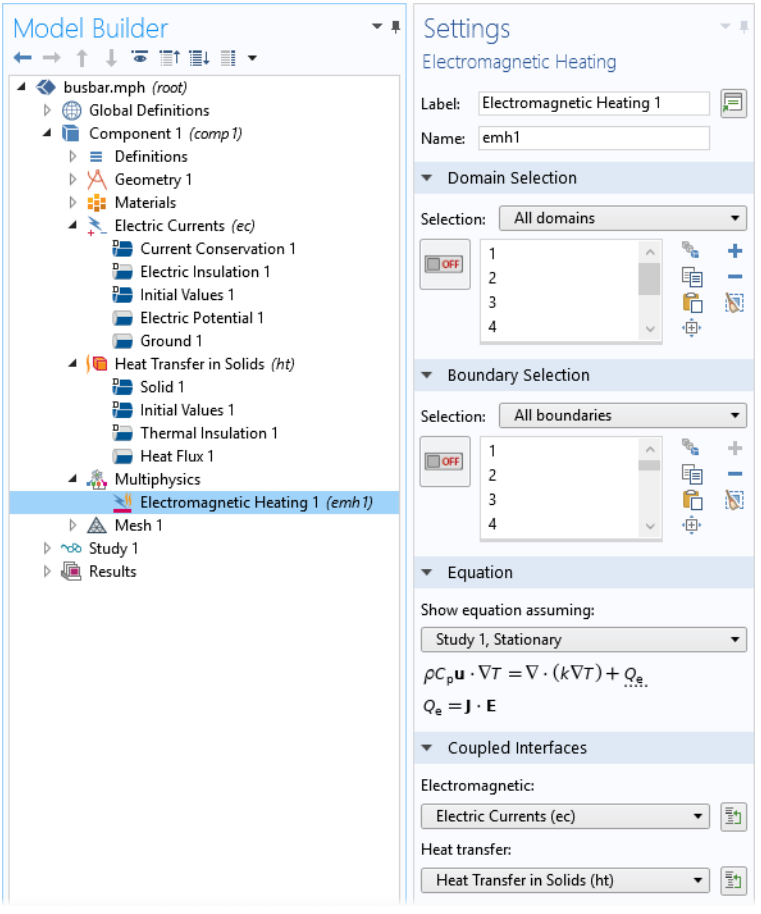


Physics and Boundary Conditions

Next, you will inspect the physics domain settings and set the boundary conditions for the heat transfer problem and the conduction of the electric current.

In the **Model Builder window**, examine the default physics nodes of the multiphysics interface for **Joule Heating**. First, collapse the **Materials** node. Then, click the arrows

next to the **Electric Currents**, **Heat Transfer in Solids**, and **Multiphysics** nodes to expand them.



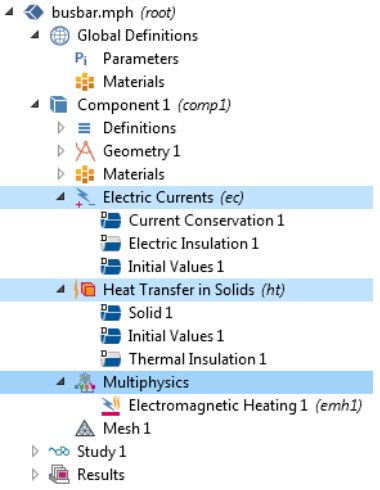
The **D** in the upper-left corner of a node's icon means it is a default node.

The equations that are being solved are displayed in the Equation section of the **Settings** windows of the respective physics nodes.

The default equation form is inherited from the study added in the Model Wizard. For the **Joule Heating** multiphysics interface, the equations are for the temperature and electric potential.

To always display the equations in the **Settings** windows, click the **Show More Options** button on the **Model Builder** toolbar and select **Equation Sections** so that a check mark appears next to it.

The **Heat Transfer in Solids** and **Electric Currents** nodes have the settings for heat conduction and current conduction, respectively.

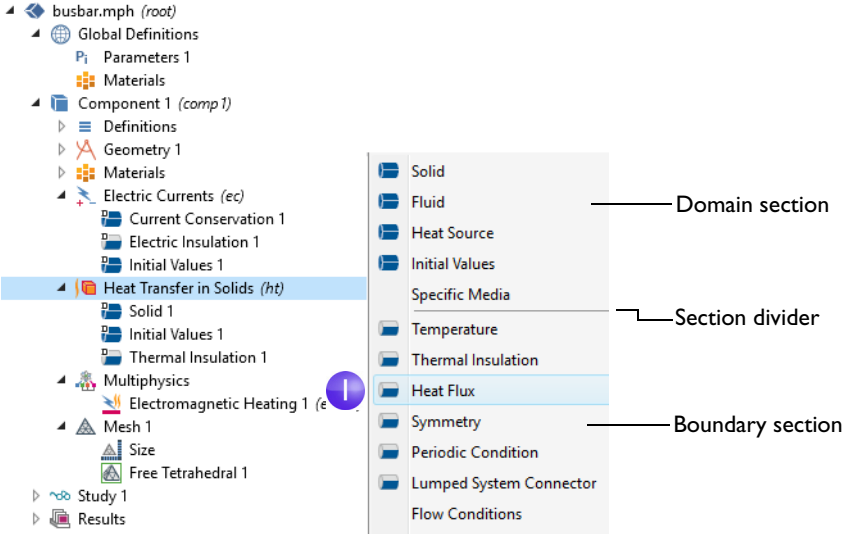


Under the **Electric Currents** node, the **Current Conservation** node represents the conservation of electric current at the domain level and the **Electric Insulation** node contains the default boundary condition for **Electric Currents**.

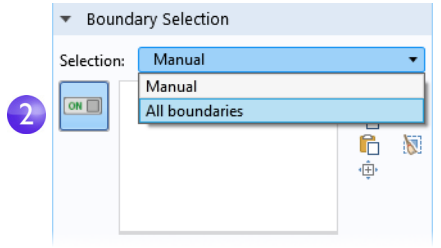
Under the **Heat Transfer in Solids** node, the domain level **Heat Transfer in Solids** node represents the conservation of heat and the **Thermal Insulation** node contains the default boundary condition for **Heat Transfer in Solids**. The heat source for the Joule heating effect is set in the **Electromagnetic Heating** node under the **Multiphysics** node. The **Initial Values** node, found in both the **Electric Currents** and **Heat Transfer in Solids** interfaces, contains initial guesses for the nonlinear solver for stationary problems and initial conditions for time-dependent problems.

Now, define the boundary conditions.

1. Right-click the **Heat Transfer in Solids** node . In the second section of the context menu — the boundary section — select **Heat Flux**.



2 In the **Settings** window for **Heat Flux**, select **All boundaries** from the **Selection** list.

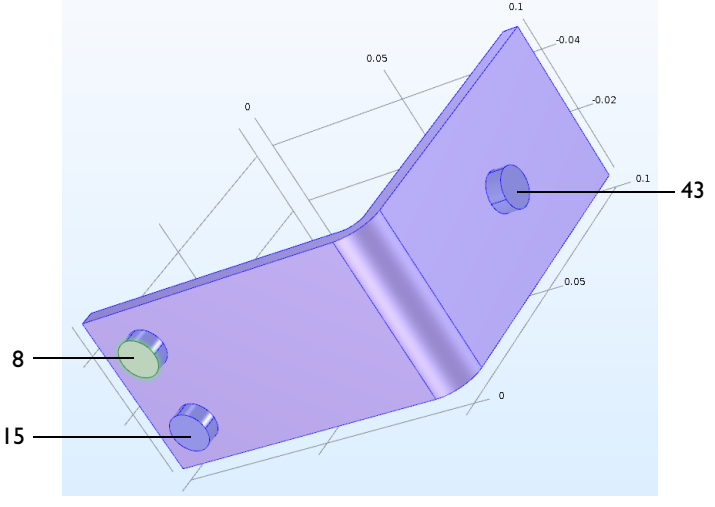


Assume that the circular bolt boundaries are neither heated nor cooled by the surroundings.

In the next step, you will remove the selection of these boundaries from the heat flux selection list, which leaves them with the default **Thermal Insulation** boundary condition for the **Heat Transfer in Solids** interface.

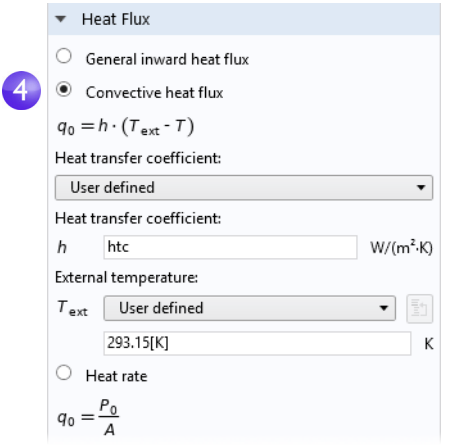
3 Rotate the busbar to view the back. Move the mouse pointer over one of the circular titanium bolt surfaces to highlight it in green. Click the bolt surface to remove this boundary selection from the **Selection** list. Repeat this step to remove the other two circular bolt surfaces from the selection list. Boundaries 8, 15, and 43 are removed.

3 Cross-check: Boundaries 8, 15, and 43 are removed from the Selection list.



4 In the **Settings** window for **Heat Flux** under **Heat Flux**, click the **Convective heat flux** button. Enter `htc` in the **Heat transfer coefficient** field, h.

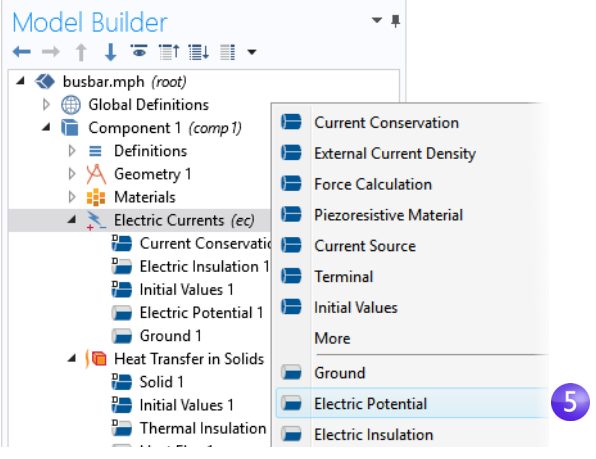
This parameter was either entered in the **Parameter** table in "Global Definitions" on page 65 or imported with the geometry.



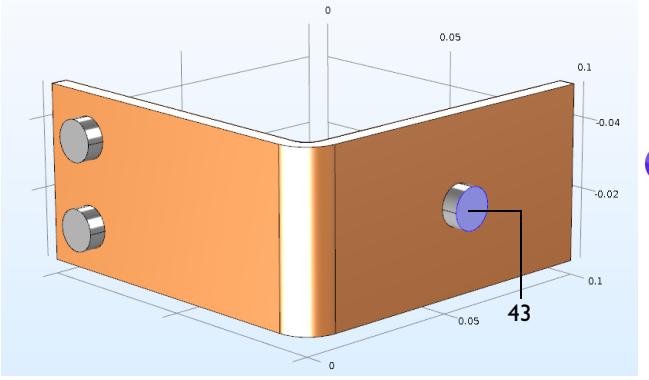
More advanced boundary conditions for heat transfer analysis are available with the Heat Transfer Module. Examples include the capability to give the total deposited power or to set boundary conditions for common natural convection scenarios such as for a horizontal or vertical wall.

Continue by setting the boundary conditions for the electric current according to the following steps:

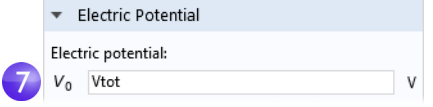
5 In the **Model Builder**, right-click the **Electric Currents** node. In the second section of the context menu — the boundary section — select **Electric Potential**. An **Electric Potential** node is added to the model tree.



6 Move the mouse pointer over the circular face of the single titanium bolt to highlight it and then click to add it (boundary 43) to the **Selection** list.

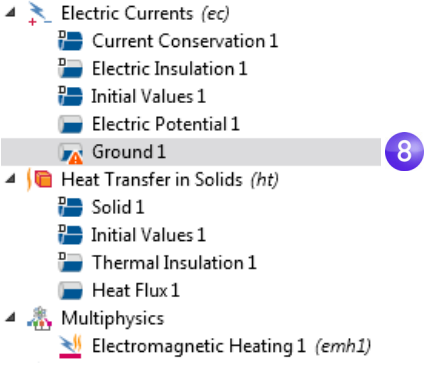


7 In the **Settings** window for **Electric Potential**, enter `Vtot` in the **Electric potential** field.



The last step is to set the surfaces of the two remaining bolts to ground.

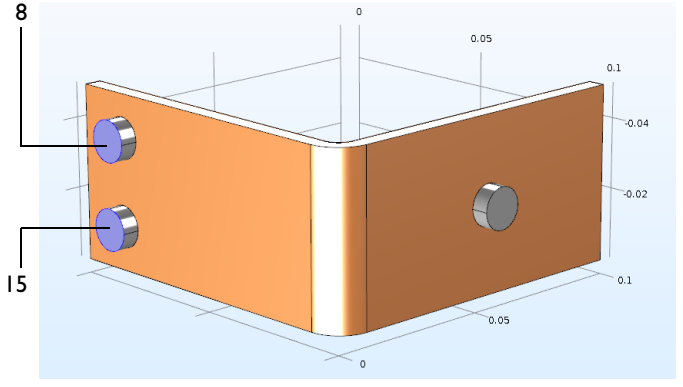
8 In the **Model Builder**, right-click the **Electric Currents** node. In the boundary section of the context menu, select **Ground**. A **Ground** node is added to the model tree. The model tree node sequence should now match this figure.



The "warning" attribute displayed on the **Ground** node indicates that its boundary selection is empty. The next step will be to select boundaries.

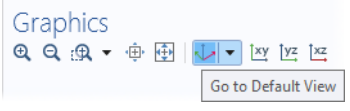
9 In the **Graphics** window, click one of the remaining bolts to add it to the **Selection** list.

9 Cross-check: Boundaries 8 and 15.



Repeat this step to add the last bolt. Boundaries 8 and 15 are added to the selection list for the **Ground** boundary condition.

10 On the **Graphics** toolbar, click the **Go to Default View** button .

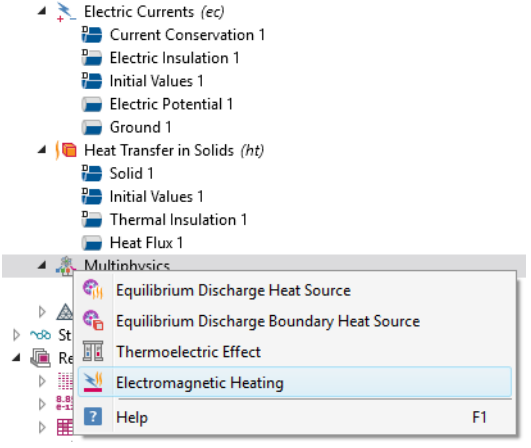


More advanced boundary conditions for electromagnetics analysis are available with the AC/DC Module, such as the capability to give the total current on a boundary.

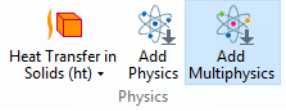
Combining Physics Interfaces Manually

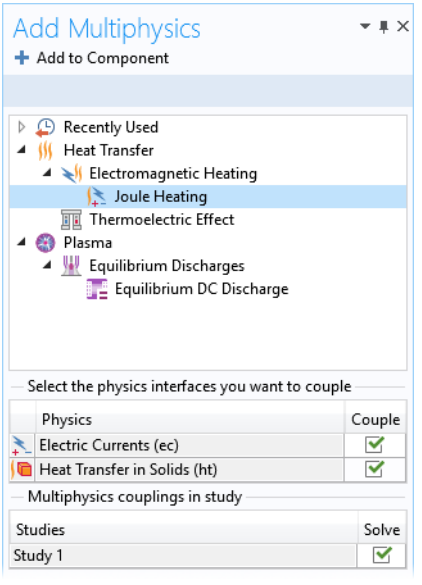
As an alternative to using the preconfigured **Joule Heating** multiphysics interface, you can manually combine the **Electric Currents** and **Heat Transfer in Solids** interfaces. For example, you can start by setting up and solving the model for

Electric Currents and then subsequently add **Heat Transfer in Solids**. In that case, you right-click the **Multiphysics** node to add the required multiphysics couplings.



As an alternative, you can click the **Add Multiphysics** button in the **Physics** tab in the ribbon and select a suggested multiphysics coupling in the **Add Multiphysics** window.





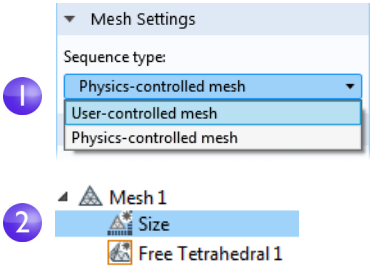
Mesh

The simplest way to mesh is to create an unstructured tetrahedral mesh, which is perfect for the busbar model. Alternatively, you can create several meshing sequences, as shown in “Adding Meshes” on page 116.

A physics-controlled mesh is created by default. In most cases, it is possible to skip to the **Study** node and just solve the model. For this exercise, the settings are investigated in order to parameterize the mesh settings.

1 In the **Model Builder**, click the **Mesh 1** node. In the **Settings** window for **Mesh**, select **User-controlled mesh** from the **Sequence type** list.

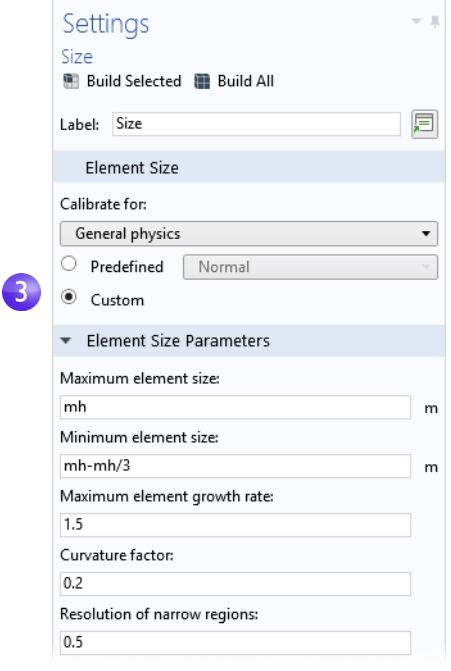
2 Under **Mesh 1**, click the **Size** node.



3 In the **Settings** window for **Size**, in the **Element Size** section, click the **Custom** button.

Under **Element Size Parameters**, enter:

- `mh` in the **Maximum element size** field. Note that `mh` is 3 mm — the value entered earlier as a global parameter. By using the parameter `mh`, element sizes are limited by this value.
- `mh-mh/3` in the **Minimum element size** field. The **Minimum element size** is slightly smaller than the maximum size.
- `0.2` in the **Curvature factor** field. The **Curvature factor** determines



the number of elements on curved boundaries; A lower value gives a finer mesh.

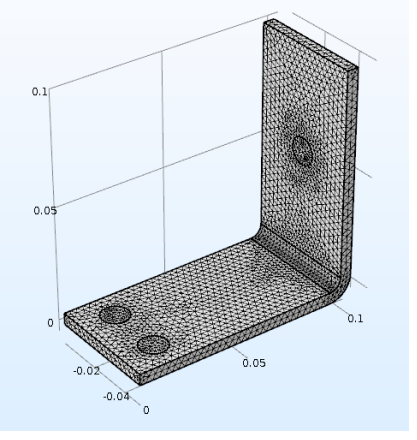
The other two parameters are left unchanged.

The Maximum element growth rate determines how fast the elements should grow from small to large over a domain. The larger this value is, the larger the growth rate. A value of 1 does not give any growth.

For Resolution of narrow regions, a higher value will generally result in a finer mesh.

The asterisk (*) that displays in the upper-right corner of the **Size** node indicates that the node is being edited.

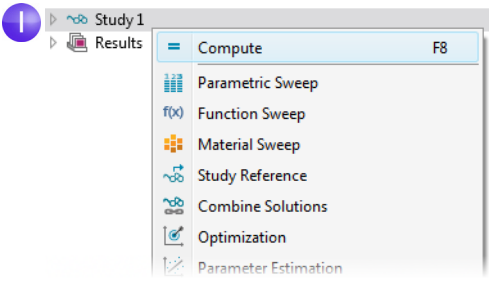
4 Click the **Build All** button in the **Settings** window for **Size** to create the mesh, as in this figure:



You can also click **Build Mesh** in the **Home** tab of the ribbon.

Study

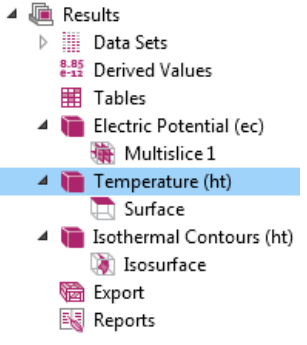
1. To run a simulation, in the **Model Builder**, right-click **Study 1** and choose **Compute** . You can also press F8 or click **Compute** in the ribbon **Home** tab.



The **Study** node automatically defines a solution sequence for the simulation based on the selected physics and study type. In this case, the simulation only takes a few seconds to solve. During the solution process, two **Convergence** plots are generated and are available from tabs next to the **Graphics** window. These plots show the convergence progress of the different solver algorithms engaged by the **Study**.

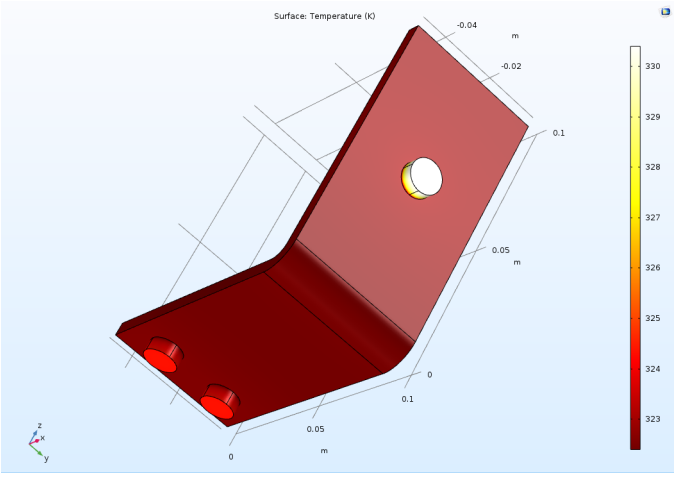
Results

By default, in the **Results** node, three plot groups are generated: a **Multislice** plot of the **Electric Potential**, a **Surface** plot of the **Temperature**, and a plot named **Isothermal Contours** containing an **Isosurface** plot of the temperature.



Click **Results > Temperature** to view the **Temperature** plot in the **Graphics** window. The temperature difference in the device is less than 10 K due to the high thermal conductivity of copper and titanium. The temperature variations are largest in the top bolt, which conducts double the amount of current compared to the two lower bolts. The temperature is substantially higher than the ambient temperature of 293 K.

1 Click and drag the image in the **Graphics** window to rotate and view the back of the busbar.



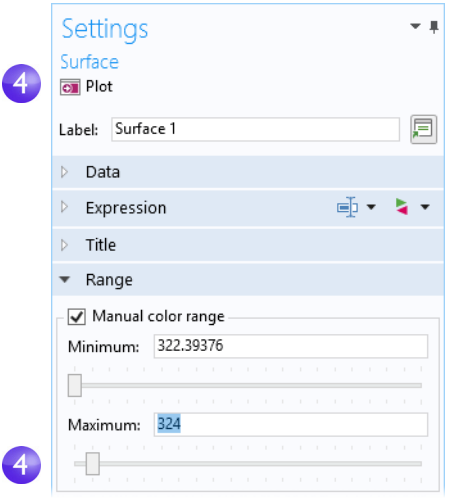
2 In the **Graphics** toolbar, click the **Go to Default 3D View** button .

You can now manually set the color table range to visualize the temperature difference in the copper part.

3 In the model tree, expand the **Results > Temperature** node and click the **Surface** node .

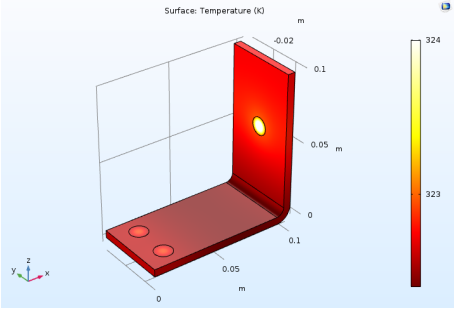
4 In the **Settings** window for **Surface**, click **Range** to expand the section. Select the **Manual color range** check box and enter `324` in the **Maximum** field (keep the

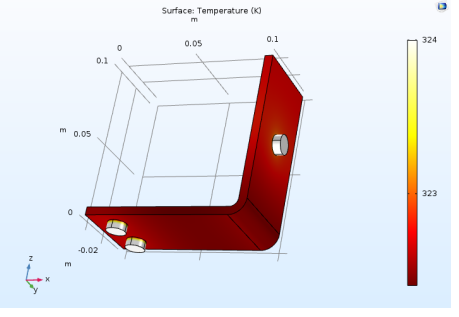
default for **Minimum** and replace the default for **Maximum**). Click the **Plot** button in the **Settings** window for **Surface**.



5 On the **Graphics** toolbar, click the **Zoom Extents** button to view the updated plot.

6 Click and drag in the **Graphics** window to rotate the busbar and view the back.



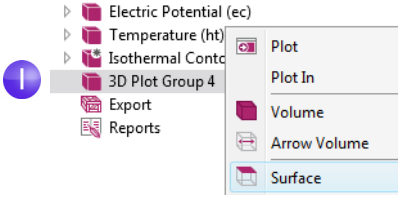


The temperature distribution is laterally symmetric with a vertical mirror plane running between the two lower titanium bolts and cutting through the center of the upper bolt. In this case, the model does not require much computing power and you can model the entire geometry. For more complex models, you can consider using symmetries to reduce the computational requirements.

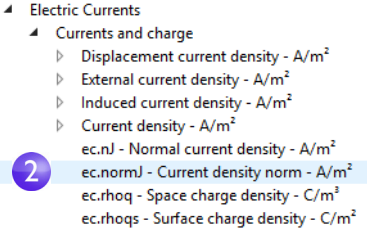
Select **Go to Default View** on the toolbar in the **Graphics** window.

Then, generate a **Surface** plot that shows the current density in the device.

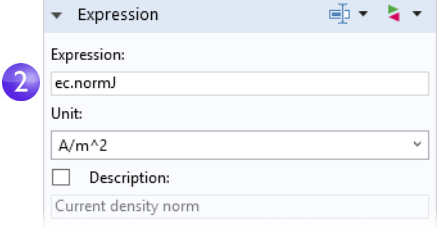
1 In the **Model Builder**, right-click **Results** and add a **3D Plot Group**. In the **Label** field enter `Current Density` to give the plot group a descriptive name. Right-click **3D Plot Group 4** and add a **Surface** node.



2 In the **Settings** window for **Surface** under **Expression**, click the **Replace Expression** button. Go to **Model > Component 1 > Electric Currents > Currents and charge > ec.normJ - Current density norm - A/m^2** and double-click or press Enter to select.



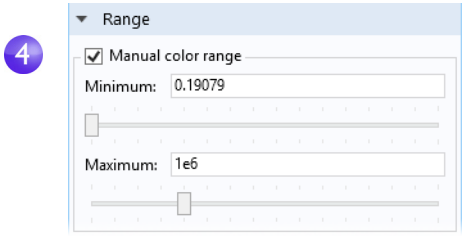
`ec.normJ` is the variable for the magnitude, or absolute value, of the current density vector. You can also type `ec.normJ` in the **Expression** field when you know the variable name or, alternatively, use the auto completion functionality by typing `norm` and then pressing Ctrl+Space to get a variable name suggestion.



3 Click the **Plot** button.

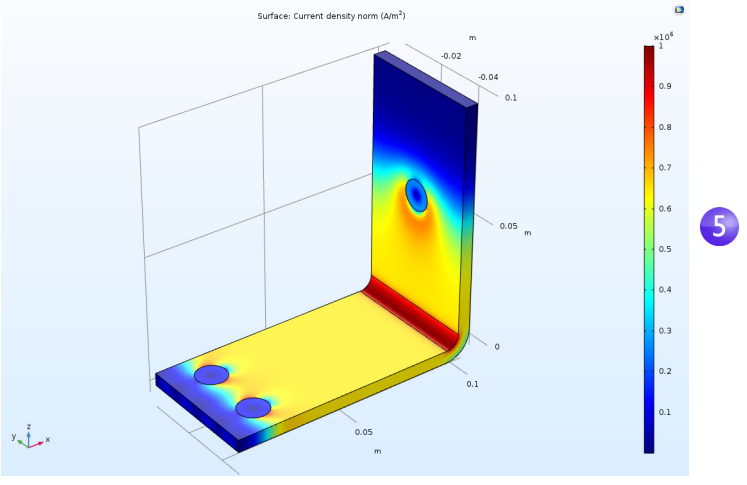
The plot that displays in the **Graphics** window is almost uniform in color due to the high current density at the contact edges with the bolts. The next step is to manually change the color table range to visualize the current density distribution.

4 In the Settings window for **Surface** under **Range**, select the **Manual color range** check box. Enter `1e6` in the **Maximum** field and replace the default.

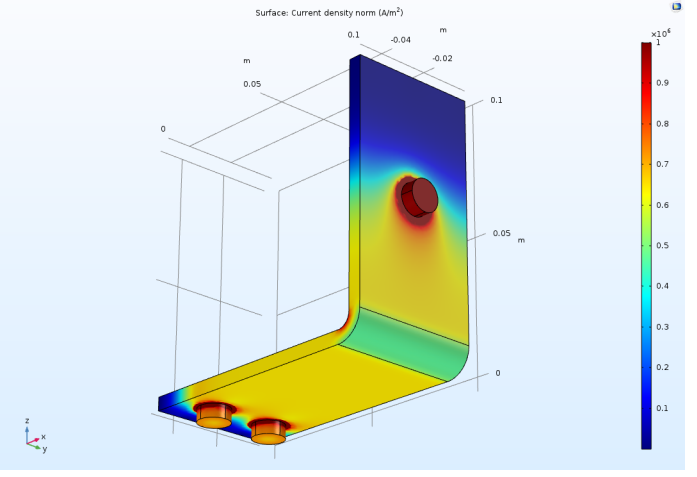


5 Click the **Plot** button .

The resulting plot shows that the current takes the shortest path in the 90-degree bend in the busbar. Notice that the edges of the busbar outside of the bolts are hardly carrying any current.



6 Click and drag the busbar in the **Graphics** window to view the back. Continue rotating the image to see the high current density around the contact surfaces of each of the bolts.

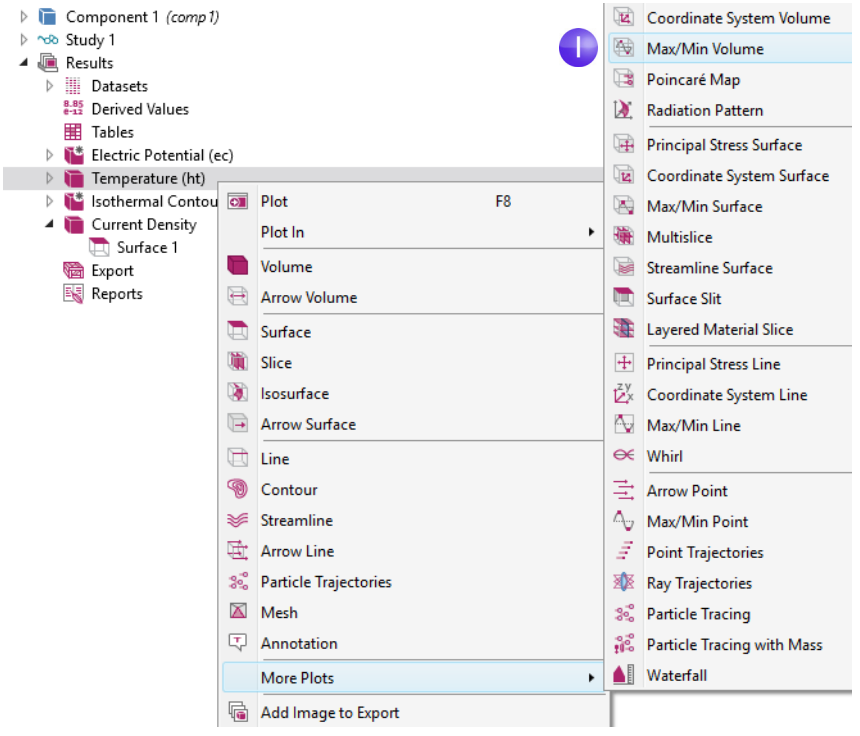


When you are done, click the **Go to Default View** button on the **Graphics** toolbar.

Getting the Maximum and Minimum Temperature

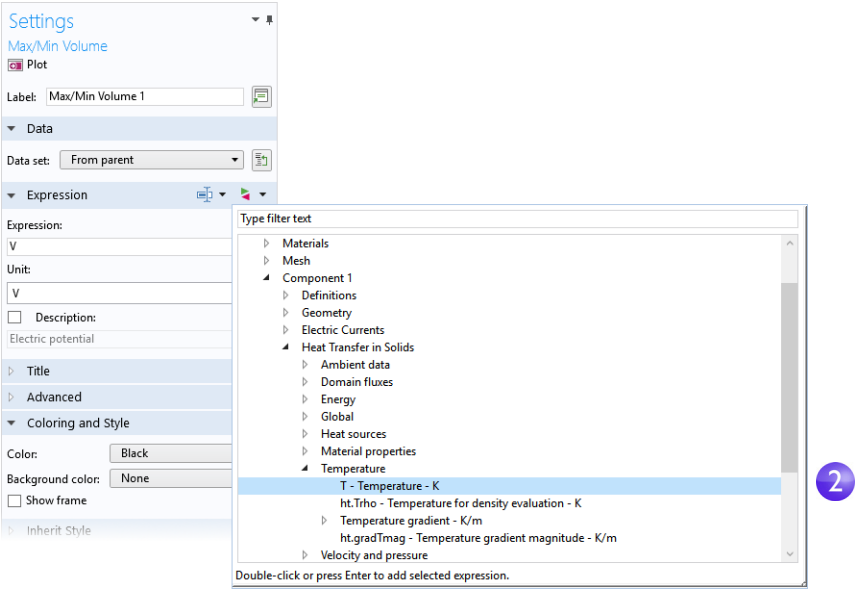
You can easily obtain the maximum and minimum temperature values in the busbar, including their locations.

1 Right-click **Results > Temperature** and select **More Plots > Max/Min Volume**.

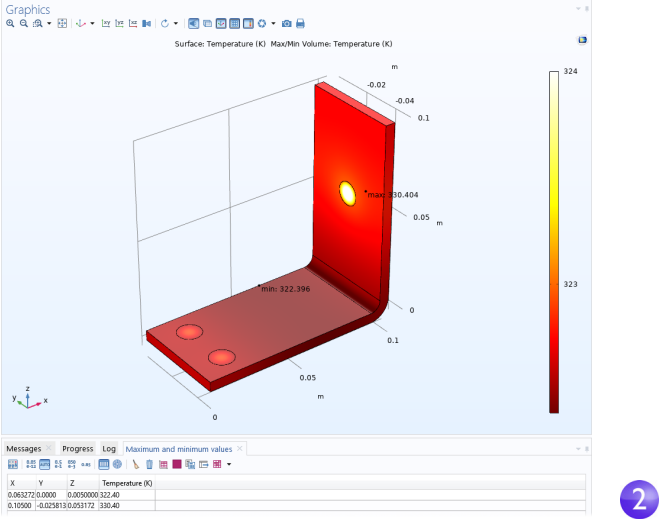


2 In the **Settings** window for **Max/Min Volume** under **Expression**, click the **Replace Expression** button . Go to **Model > Component 1 > Heat Transfer in Solids > Temperature > T - Temperature - K** and double-click or press Enter to select. You

can also search for `Temperature` in the search field above the tree of postprocessing variables or manually enter `T`. Click **Plot**.



The maximum and minimum values are plotted in the **Graphics** window and their locations and values are displayed in the **Maximum and minimum values** table. The values may vary a bit depending on the version you are running. For a finer mesh the results will be the same regardless of version.

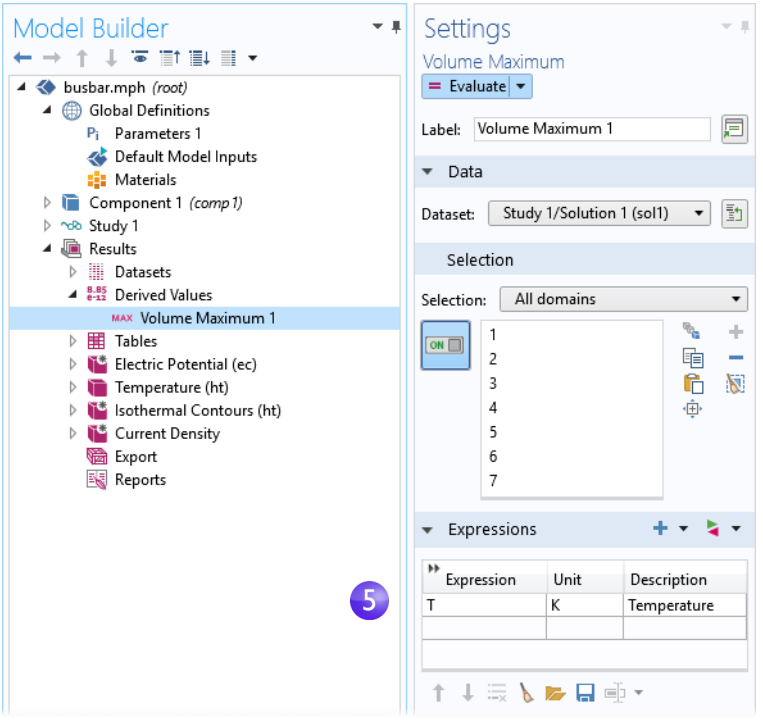


As an alternative to the method of plotting the maximum and minimum values described above, you can use the **Derived Values** node or an **Evaluation Group** for a pure numerical output. The following steps show how to use **Derived Values**.

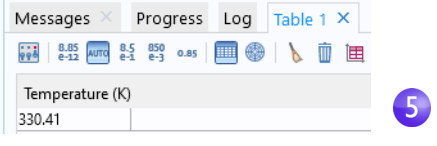
3 In the **Results** section of the model tree, right-click the **Derived Values** node and select **Maximum > Volume Maximum**.

4 In the **Settings** window for **Volume Maximum**, under **Selection**, choose **All domains**.

5 In the first row of the **Expression** table, enter `T` (unless it is automatically filled out) for the temperature and click **Evaluate** at the top of the **Settings** window.



The resulting maximum temperature is shown in **Table 1**.



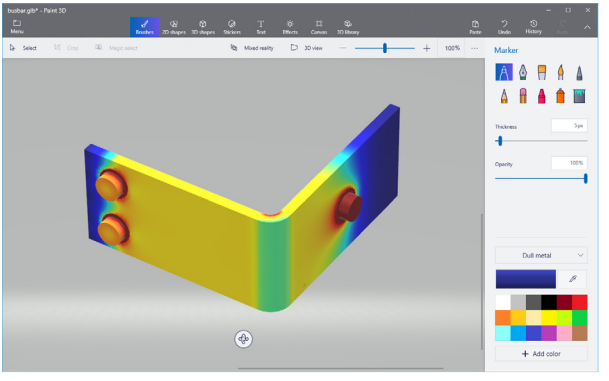
As a next step, you can create a model thumbnail image.

Creating Model Images from Plots

With any solution, you can create an image to display when browsing for model files. After generating a plot, click the corresponding plot node in the **Model Builder** under **Results**. Then, click the root node (the first node in the model tree). In the **Settings** window for the root node, under **Thumbnail**, click **Set from Graphics Window**.

There are two other ways to create images from a plot. One is to click the **Image Snapshot** button in the **Graphics** window toolbar to directly create an image. You can also add an **Image** node to the **Export** node to create an image file. Right-click the plot group of interest and then select **Add Image to Export**.

You can use the **Image Snapshot** button to export **Results** visualizations to glTF™ (file extension `.glb`). These files can then be opened in 3D visualization software such as Paint 3D from Microsoft®, as shown in the figure below. (If you are running Windows® 10, then this may be available in your system.) You can also find glTF™ viewers online that run in web browsers.



Make sure to save the model at this stage. This version of the model, `busbar.mph`, is reused and renamed during the next set of tutorials.

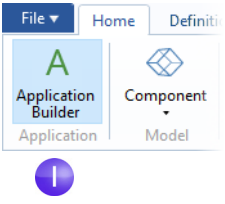
Building an Application with the Application Builder

This section gives a brief introduction to creating an application based on the busbar model by using the Application Builder. The application will have a specialized user interface dedicated to interact with the busbar model and include input fields for the length, width, and applied voltage. The outputs will be the maximum temperature and a temperature plot.

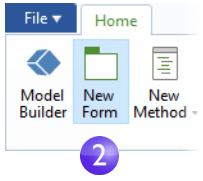
You will learn how to use the New Form wizard to quickly create an application. The New Form wizard is used to create a form with user interface components called form objects. The wizard gives quick access to a subset of all of the form objects available for creating applications with the Application Builder. In this example, you will learn how to add input fields, numerical output, graphics, and buttons. For detailed information on building applications, see the book *Introduction to Application Builder*.

Creating a New Form

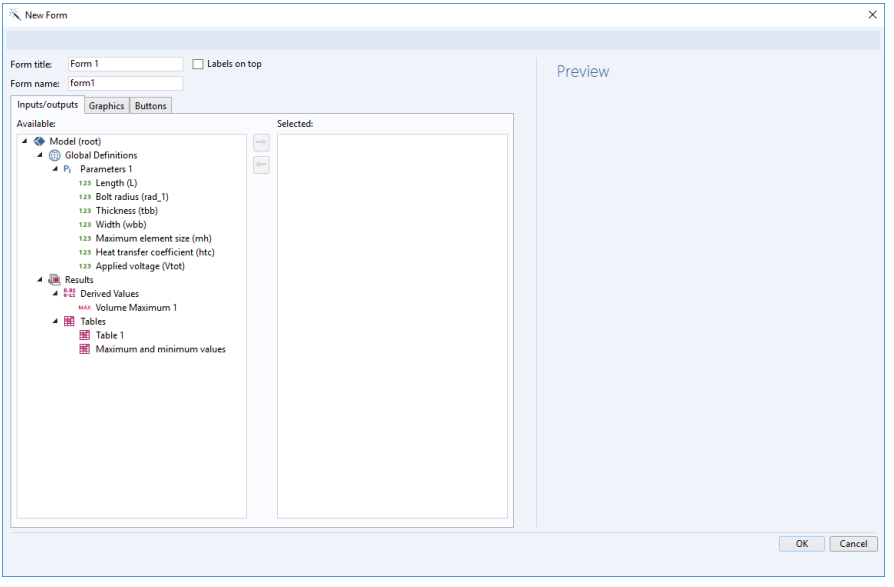
1 To switch from the Model Builder to the Application Builder, click the **Application Builder** button in the leftmost part of the **Home** tab in the ribbon.



2 To create a new form, click the **New Form** button in the ribbon.

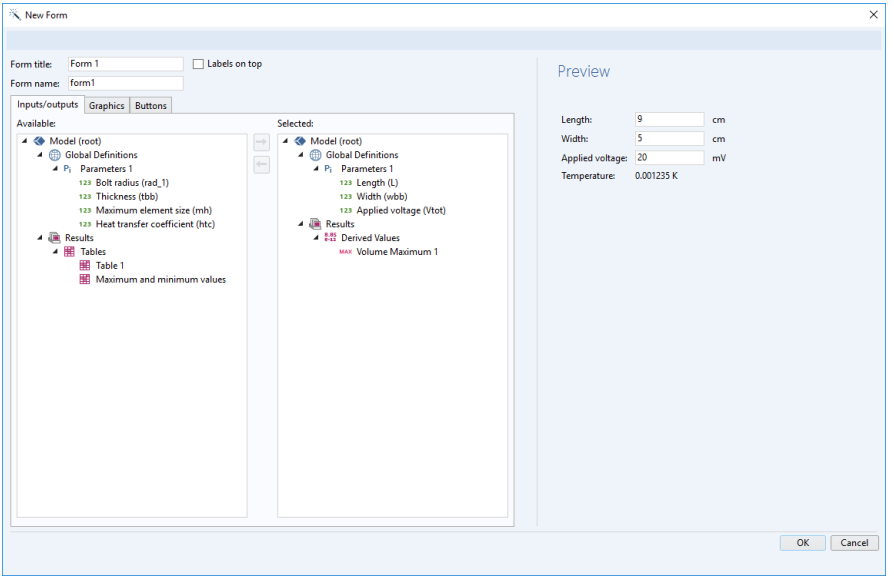


The wizard window appears.



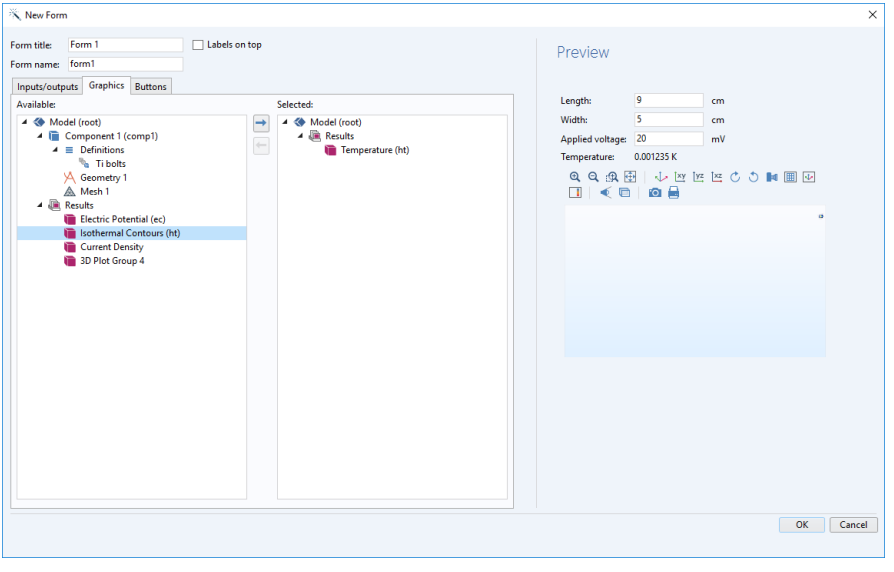
It consists of three tabs for **Inputs/outputs**, **Graphics**, and **Buttons**. In the **Inputs/outputs** tab, you can double-click the nodes in the tree to make these available in the user interface of the application. For this application, we will select a few of the parameters as input fields and the maximum temperature as output.

3 Double-click each of the parameters: **Length (L)**, **Width (wbb)**, and **Applied voltage (Vtot)**. Also, double-click the **Volume Maximum** under **Derived Values**.

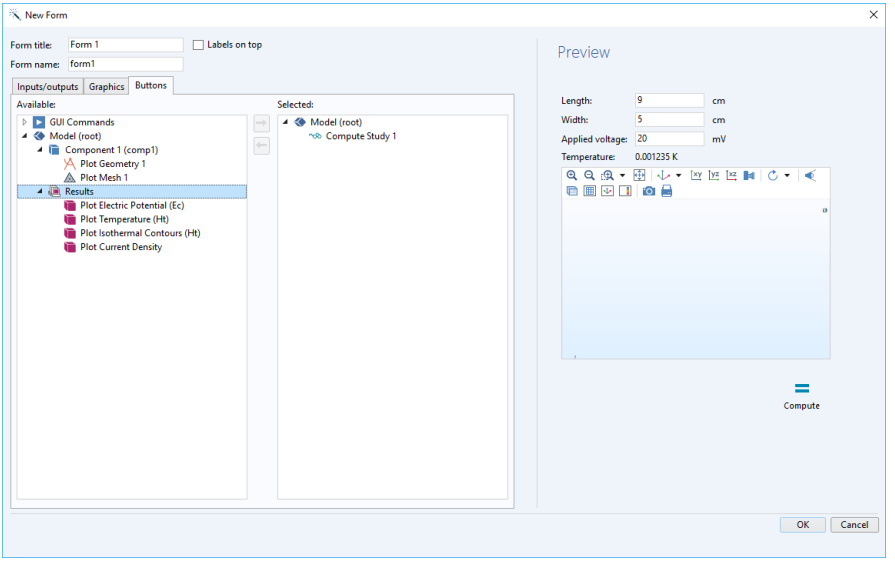


The **Preview** section to the right displays the user interface layout.

4 Click the **Graphics** tab in the wizard and double-click the **Temperature** plot. A graphics object placeholder appears in the **Preview** area.

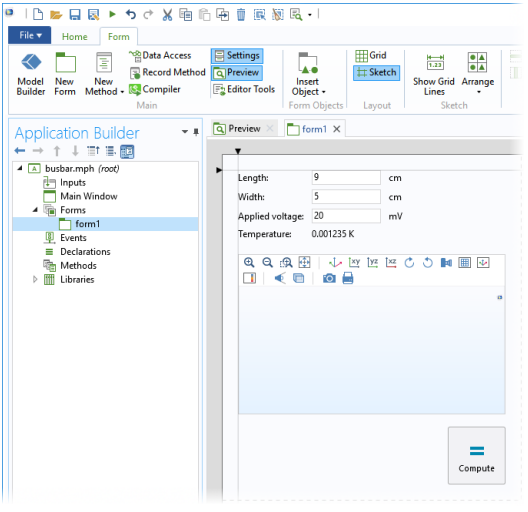


5 Click the **Buttons** tab in the wizard and double-click **Compute Study 1**. A button now appears in the **Preview** area beneath the graphics object.

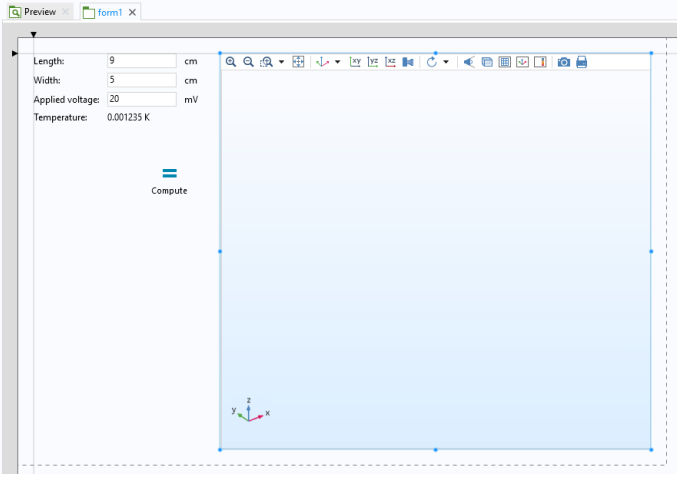


6 Finally, click **OK** to accept the settings and close the wizard.

After closing the New Form wizard, the Form editor is opened. The Form editor allows you to drag-and-drop form objects and add new ones.

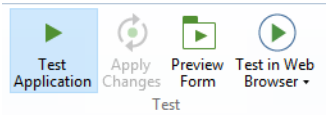


7 You can, for example, rearrange the user interface by dragging the graphics object to the right of the input fields and the button directly under the input fields, as shown in the figure below. Blue guidelines will aid you in aligning the objects.

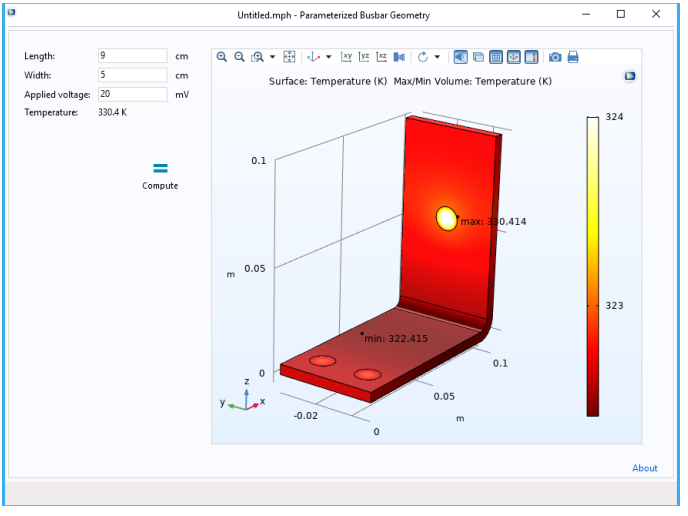


You can make the graphics object larger by dragging the blue handles.

8 You can test the application by clicking **Test Application** in the ribbon.



This creates a run-time copy of the application and lets you quickly try different designs during the process of developing an application.

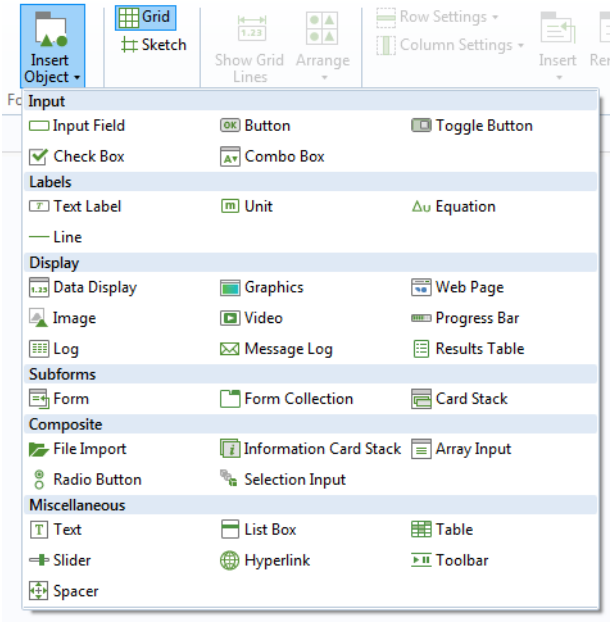


9 Now, try a different **Length** value and click **Compute**.

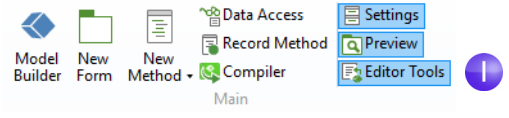
Adding a Button

You may wish to preview the geometry before starting the computation. This can easily be done by adding a button. There are two ways to add button. Either use

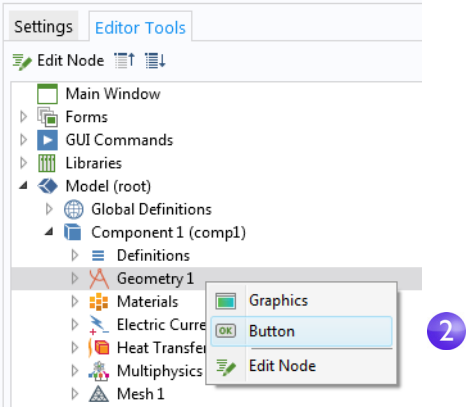
the **Insert Object** menu in the ribbon, as shown in the figure below, or use the **Editor Tools** window. In this example, you will use the latter.



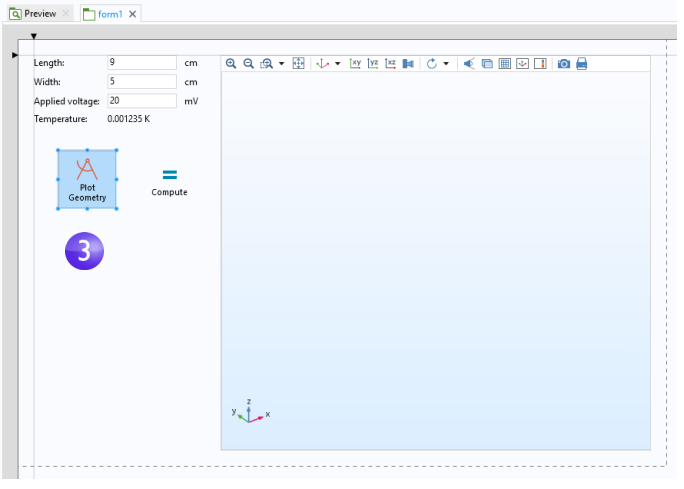
- First, make sure the **Editor Tools** window is visible by selecting the corresponding button in the ribbon.



2 In the **Editor Tools** tree, right-click the **Geometry** node and select **Button**.

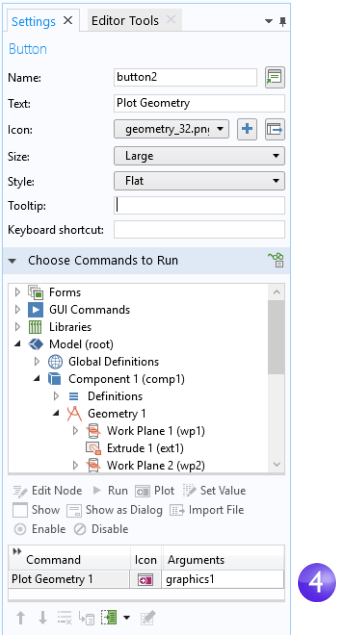


3 Drag the **Plot Geometry** button to the left of the **Compute** button.



Behind each button is a command sequence that controls the action taken when it is pressed. The command sequence is defined in the **Settings** window of the button and you can open the **Settings** window (if not already opened) by double-clicking the button.

4 Double-click the **Plot Geometry** button to see its **Settings** window. Command sequences appear below the tree in the section **Choose Commands to Run**.

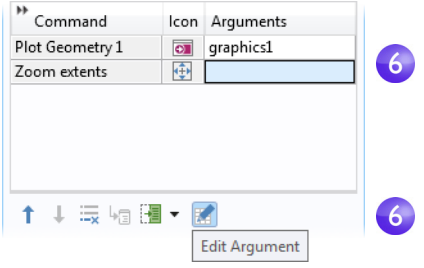


In this case, the command sequence for the **Plot Geometry** has one command for plotting the geometry. The input argument contains information on which graphics window to send the geometry plot, in this case, the graphics object `graphics1`.

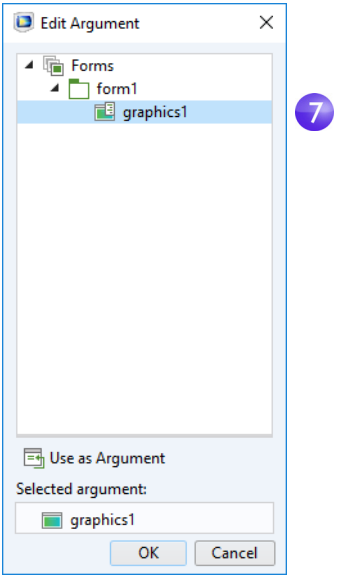
To automatically adjust the axis settings, you can add a **Zoom Extents** command.

5 In the **Choose Commands to Run** section, locate the **Zoom Extents** command under **GUI Commands > Graphics Commands**. Double click to add it to the command sequence.

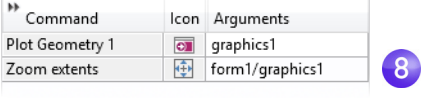
6 Now, create the input argument to the **Zoom Extents** command. Click the white field next to **Zoom extents** and click the **Edit Argument** button.



7 This opens a dialog box with possible input arguments. Select **graphics1** under **form1**.



8 The command sequence is now complete.

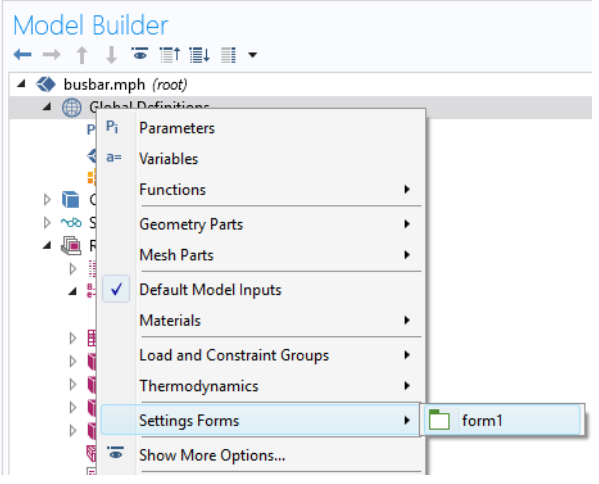


9 To make sure the new **Plot Geometry** button works, you can test the application again by clicking **Test Application** in the ribbon.

Using Forms in the Model Builder

Forms without graphics form objects can be used in the Model Builder. You can use this functionality to create customized **Settings** windows for common or repetitive tasks, for example. If you would like to reuse your customized **Settings** window between sessions or even with other users, you can create an add-in, as explained in the *Introduction to Application Builder*.

To use a form, right-click **Global Definitions** and select the form under **Settings Forms**.

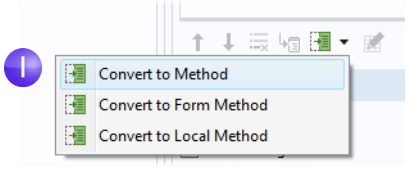


Methods

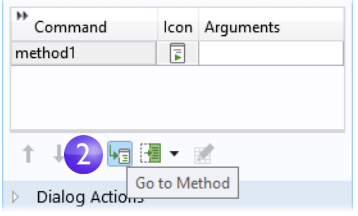
To perform a more advanced operation than what is possible with a command sequence, you can use the Method editor to write code that defines the actions to be taken when the user clicks a button, selects a menu item, etc. In addition to calling methods from applications or **Settings Forms**, you can call methods from the **Developer** tab in the ribbon or from **Method Calls** in the model tree. For more information, see *Introduction to Application Builder*.

An easy way to get started with the Method editor is to convert an existing command sequence into a method.

1 Locate or open the **Settings** window for the **Plot Geometry** button again. Click the **Convert to Method** button below the command sequence (the other two options are for creating two types of local methods).

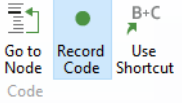


2 To open the method in the Method editor, click the **Go to Method** button below the command sequence.



The Method editor now opens with the code for **method1**, corresponding to the **Plot Geometry** and the **Zoom Extents** commands in the previous command sequence.

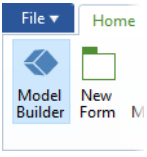
The Application Builder has several tools for automatically generating code. In addition to converting a command sequence to a method, you can, for example, click **Record Code** in the ribbon.



This will generate code for each operation performed in the COMSOL Desktop windows. Click **Stop Recording** in the ribbon to stop recording and go back to the Method editor.

To learn more about using the Form editor and the Method editor, see the book *Introduction to Application Builder*, which provides more information on writing methods. For more in-depth information on methods, see the *Application Programming Guide*.

This completes the busbar example. To switch back to the Model Builder, click the **Model Builder** button in the ribbon.



The next sections are designed to deepen your understanding of the steps implemented so far and to extend your simulation to include additional effects like thermal expansion and fluid flow. These additional topics begin on the following pages:

- "Parameters, Functions, Variables, and Couplings" on page 110
- "Material Properties and Material Libraries" on page 114
- "Adding Meshes" on page 116
- "Adding Physics" on page 119
- "Parametric Sweeps" on page 141
- "Parallel Computing" on page 152
- "COMSOL Multiphysics Client-Server" on page 156
- "Appendix A — Building a Geometry" on page 159

Parameters, Functions, Variables, and Couplings

This section explores working with parameters, functions, variables, and nonlocal couplings.

The **Global Definitions** and **Component > Definitions** nodes contain functionality that helps you to prepare model inputs and nonlocal couplings and organize simulations. You have already used the functionality for adding parameters to organize model inputs in "Global Definitions" on page 65.

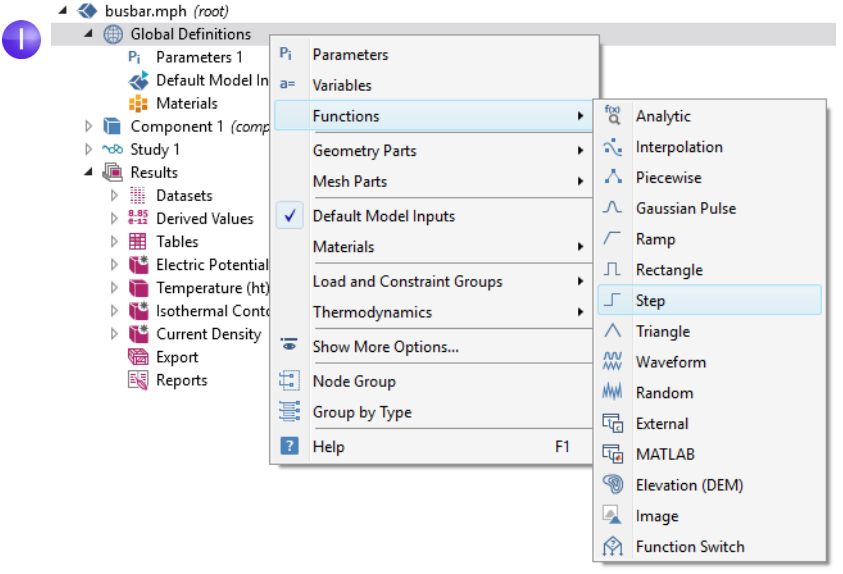
Functions, available as both **Global Definitions** and **Component > Definitions**, contain a set of predefined function templates that can be useful when setting up multiphysics simulations. For example, the **Step** function template can create a smooth step function for defining different types of spatial or temporal transitions.

To illustrate using functions, assume that you want to add a time-dependent study to the busbar model, where an electric potential is applied across the busbar that goes from 0 V to 20 mV in 0.5 seconds. For this purpose, you could use a step function to be multiplied with the parameter `Vtot`. In this section, you will add a step function to the model that goes smoothly from 0 to 1 in 0.5 seconds to find out how functions can be defined and verified.

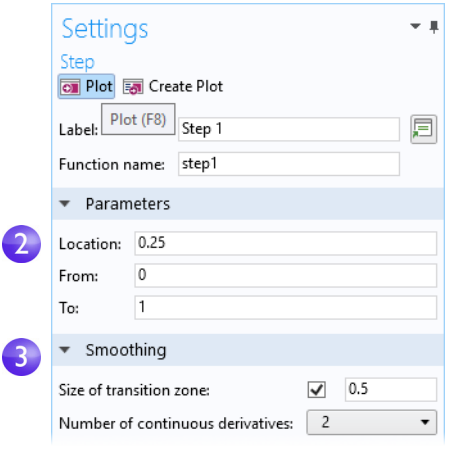
Defining Functions

For this section, you can continue working with the same model file created in the previous section. Locate and open the file `busbar.mph` if it is not already open on the desktop.

1 Right-click the **Global Definitions** node and select **Functions > Step** .



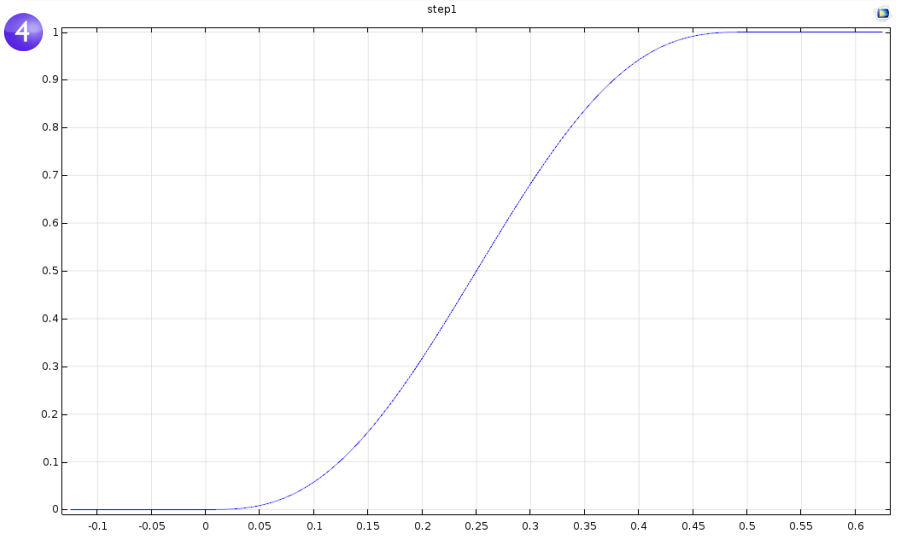
2 In the **Settings** window for **Step**, enter `0.25` in the **Location** field to set the location of the middle of the step, where it has the value of `0.5`.



3 Click **Smoothing** to expand the section and enter `0.5` in the **Size of transition zone** field to set the width of the smoothing interval. Keep the default **Number of continuous derivatives** at 2.

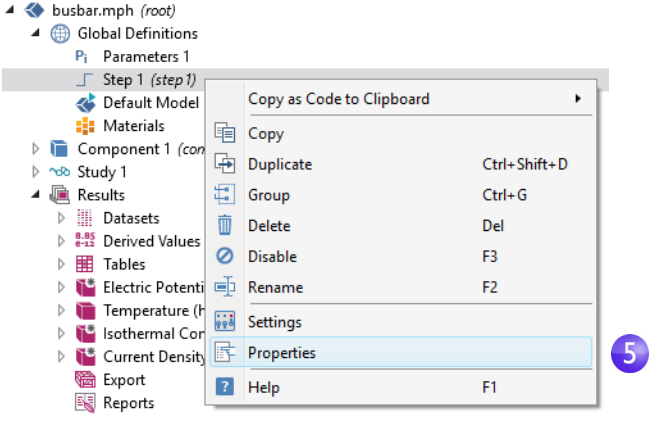
4 Click the **Plot** button in the **Settings** window for **Step**.

If your plot matches the one below, this confirms that you have defined the function correctly.

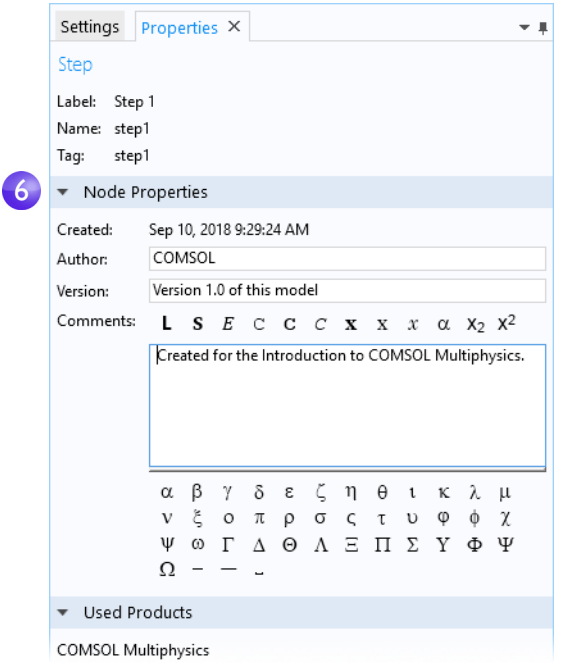


You can also add comments to the function to make it more descriptive.

5 Right-click the **Step 1** node in the **Model Builder** and select **Properties**.



6 In the **Properties** window, enter any information you want. Right-click the **Properties** tab and select **Close** when you are finished. Special characters and markup tags that you use here will be displayed in reports generated from the **Reports** node.

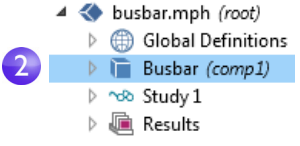


For the purpose of this exercise, assume that you want to introduce a second component to represent an electric device connected to the busbar through the titanium bolts.

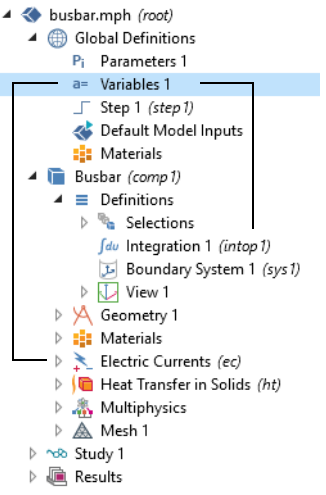
First, rename **Component 1** to specify that it represents the busbar.

1 Right-click the **Component 1** node and select **Rename** (or press F2).

2 In the **Rename Component** window, enter `Busbar`. Click **OK** and save the model.



Defining Nonlocal Couplings and Probes



Right-click the **Definitions** node ≡ under **Busbar (comp1)** to introduce a nonlocal coupling that computes the integral of any **Busbar (comp1)** variable at the bolt boundaries facing the electric device. You can use such a coupling, for example, to define a variable in a **Variables** node under the **Global Definitions** node that calculates the total current. This variable is then globally accessible and could, for example, form a boundary condition for the current that is fed to an electric device modeled as a second component.

The **Nonlocal Couplings** in **Definitions** have a wide range of uses. The **Average** AV, **Maximum** MAX, and **Minimum** MIN couplings have applications in generating results as well as in boundary conditions, sources, sinks, properties, or any other contribution to the model equations. The **Probes** are for monitoring the solution progress. For instance, you can follow the solution at a critical point during a time-dependent simulation or for each parameter value in a parametric study.

You can find an example of using the average operator in “Parametric Sweeps” on page 141. Also see “Functions” on page 187 for a list of available functions.

To learn more about working with definitions, in the **Model Builder,** click the **Definitions** ≡ or **Global Definitions** node and press F1 to open the **Help** window. This window displays helpful information about the selected item in the desktop and provides links to the documentation. It could take up to a minute for the window to load the first time it is activated, but will load quickly the next time it is opened.

Material Properties and Material Libraries

You have already used the functionality in the **Materials** node to access the properties of copper and titanium in the busbar model. In the **Materials** node, you are also able to define your own materials and save them in your material library. In addition, you can add material properties to existing materials. In cases where you define properties that are functions of other variables, typically temperature, the plot functionality helps you to verify the property functions in the range of

interest. You can also load Excel® spreadsheets and define interpolation functions for material properties using LiveLink™ *for* Excel®.

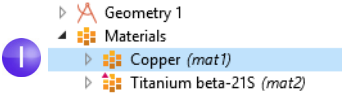
The Material Library add-on contains over 3800 materials with tens of thousands of temperature-dependent property functions. In addition, many of the add-on products include material libraries relevant to their application areas.

First, investigate how to add properties to an existing material. Assume that you want to add bulk modulus and shear modulus to the copper properties.

Customizing Materials

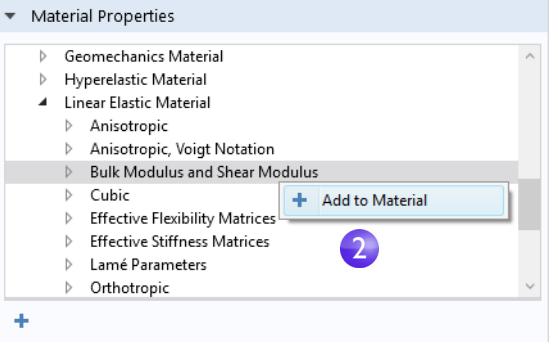
Let us keep working on the busbar.

1 In the **Model Builder**, under **Materials**, click **Copper** .



2 In the **Settings** window for **Material,** click to expand the **Material Properties** section, which contains a list of all of the definable properties.

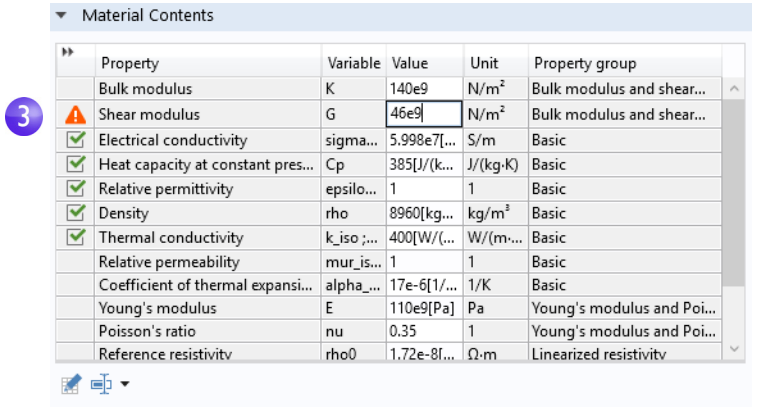
Expand the **Solid Mechanics > Linear Elastic Material** section. Right-click **Bulk Modulus and Shear Modulus** and select + **Add to Material**.



This lets you define the bulk modulus and shear modulus for the copper in your model.

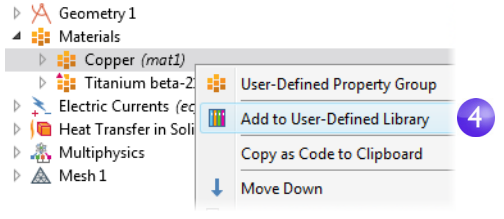
3 In the **Copper** node, locate the **Material Contents** section. **Bulk modulus** and **Shear modulus** rows are now available in the table. The warning sign ⚠ indicates that

the values are not yet defined. To define the values, click the **Value** column. In the **Bulk modulus** row, enter `140e9` and in the **Shear modulus** row, enter `46e9`.



By adding these material properties, you have changed the **Copper** material. You can save this information in your own material library, but not in the read-only Solid Mechanics materials library.

4 In the **Model Builder**, right-click **Copper** and select **Add to User-Defined Library**.



To see this library, click the **Browse Materials** in the **Materials** tab in the ribbon.

Adding Meshes

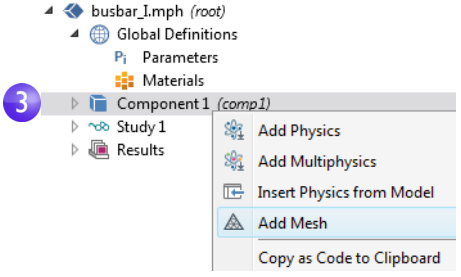
A model component can contain different mesh sequences for generating meshes with different settings. A mesh sequence is an ordered set of instructions for generating a mesh. Mesh sequences can be accessed by the study steps. In a study, you can select which mesh you would like to use in a particular simulation.

In the busbar model, a second mesh node is now added to create a mesh that is refined in the areas around the bolts and the bend.

Adding a Mesh

1 Open the model `busbar.mph` that was created earlier.

2 In order to keep this model in a separate file for later use, select **File > Save As** and rename the model `busbar_I.mph`.

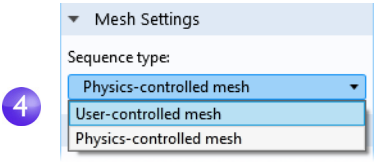


3 To add a second mesh node, right-click the **Component 1** node and select **Add Mesh**. (If you followed the instructions in "Parameters, Functions, Variables, and Couplings" on page 110, the name of **Component 1** is **Busbar** instead.)

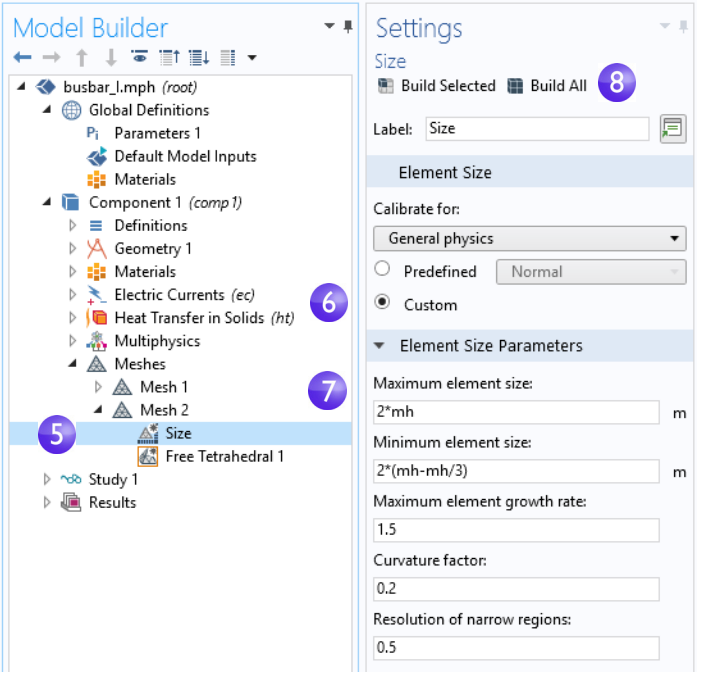
By adding another **Mesh** node, a **Meshes** parent node is created that contains both **Mesh 1** and **Mesh 2**.

4 Click the **Mesh 2** node. In the Settings window for **Mesh** under **Mesh Settings**, select **User-controlled mesh** as the **Sequence type**.

A **Size** and **Free Tetrahedral** node are now available under **Mesh 2**.



5 In the **Model Builder**, under **Mesh 2**, click **Size** .



The asterisk in the upper-right corner of a node icon indicates that the node is being edited.

6 In the **Settings** window for **Size**, under **Element Size**, click the **Custom** button.

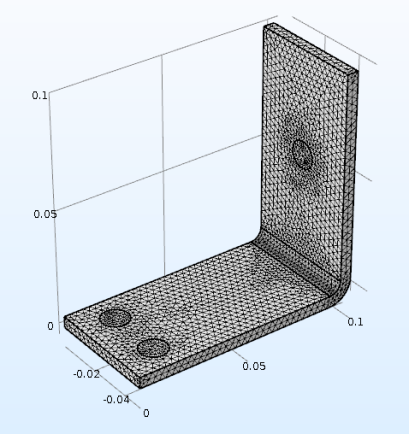
7 Under **Element Size Parameters**, enter:

- `2*mh` in the **Maximum element size** field, where `mh` is 3 mm — the mesh control parameter defined previously.
- `2*(mh-mh/3)` in the **Minimum element size** field.
- `0.2` in the **Curvature factor** field.

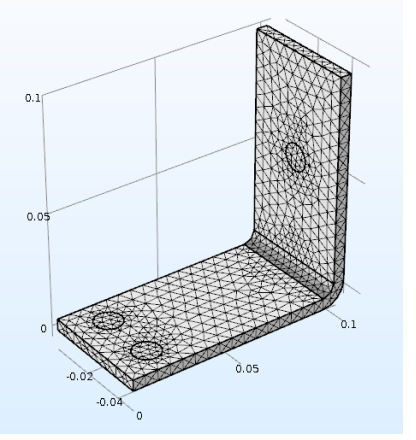
8 Click **Build All** . Save the file as `busbar_I.mph`.

Compare **Mesh 1** and **Mesh 2** by clicking the **Mesh** nodes. The mesh is updated in the **Graphics** window. An alternative to creating multiple meshes in this way is to

run a parametric sweep of the parameter for the maximum mesh size, `mh`, that was defined in the section “Global Definitions” on page 65.



Mesh 1



Mesh 2

Adding Physics

The distinguishing characteristics of COMSOL Multiphysics, particularly adaptability and compatibility, are prominently displayed when you add physics to an existing model.

In this section, you will experience the ease with which this seemingly difficult task is performed. By following these directions, you can add structural mechanics and fluid flow to the busbar model.

Structural Mechanics

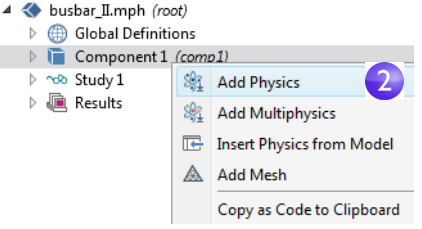
After completing the busbar Joule heating simulation, we know that there is a temperature rise in the busbar. What kind of mechanical stress is induced by thermal expansion? To answer this question, let us expand the model to include the physics associated with structural mechanics.

To complete these steps, either the Structural Mechanics Module or the MEMS Module (which enhances the core Solid Mechanics interface) is required.

If you want to add cooling by fluid flow, or do not have the Structural Mechanics Module or MEMS Module, read this section and then go to “Cooling by Adding Fluid Flow” on page 128.

1 Open the model `busbar.mph` that was created earlier. From the main menu, select **File > Save As** and rename the model `busbar_II.mph`. Alternatively, you could load the `busbar.mph` model from the Application Libraries as shown earlier.

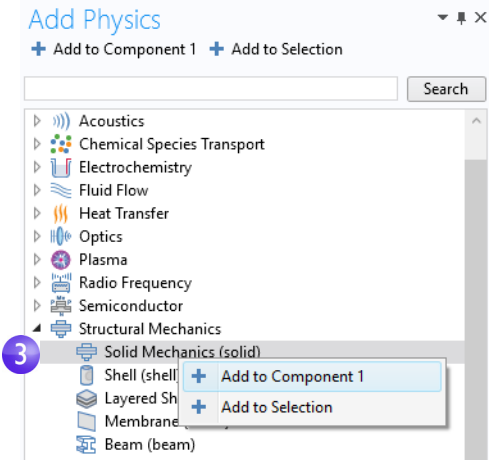
2 In the **Model Builder**, right-click the **Component 1** node and select **Add Physics**.



3 In the **Add Physics window,** under **Structural Mechanics**, select **Solid Mechanics**.

To add this interface, you can right-click and select **Add to Component 1** or click the **Add to Component 1** button at the top of the window.

4 Close the **Add Physics** window and save the file.



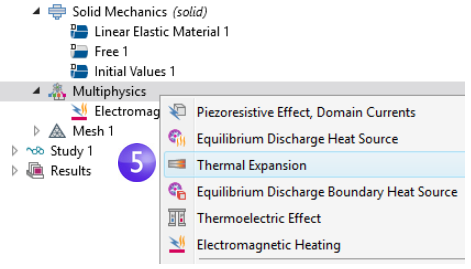
When adding additional physics, you need to make sure that materials included in the **Materials** node have all of the required properties for the selected physics. In this example, all properties are already available for copper and titanium.

You can start by adding the effect of thermal expansion to the structural analysis.

5 In the **Model Builder**, right-click the **Multiphysics** node and select **Thermal Expansion** .

A **Thermal Expansion** node is added to the model tree.

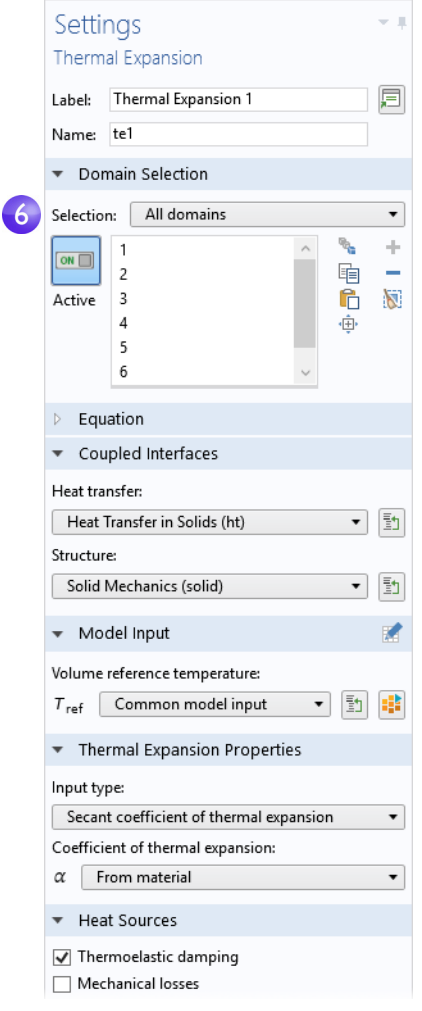
You can also use the ribbon and select, from the **Physics** tab, **Multiphysics Couplings> Thermal Expansion**.



6 In the **Settings** window for **Thermal Expansion**, from the **Selection** list, select **All domains**. This will enable thermal expansion in the copper as well as in the titanium bolts.

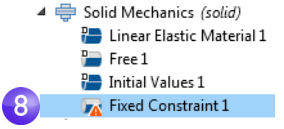
The **Thermal Expansion Properties** section of this window shows information about the **Coefficient of thermal expansion** and the **Strain reference temperature** (there are also some more advanced settings.) The **Coefficient of thermal expansion** takes its value from the **Materials** node. In the **Model Input** section, the **Volume reference temperature** (strain reference temperature) has a default value of 293.15 K (room temperature) taken from the **Default model inputs** node, available in **Global Definitions,** and defines the temperature for which there is no thermal expansion. The **Coupled Interfaces** section shows which two physics interfaces define the physics for heat transfer and solid mechanics. This is useful in the case where there is more than one physics interface for heat transfer or solid mechanics in the model component. Keep all default settings in this window.

Next, constrain the busbar at the position of the titanium bolts.

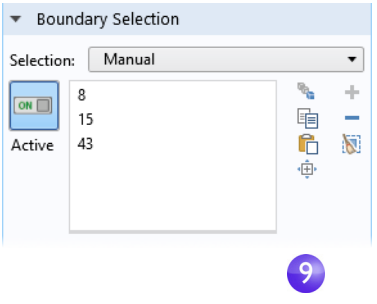


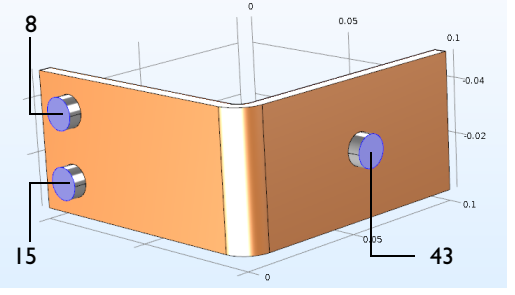
7 In the model tree, right-click **Solid Mechanics** and from the boundary level, select **Fixed Constraint**. A node with the same name is added to the tree.

8 Click the **Fixed Constraint** node. In the **Graphics** window, rotate the busbar to view the back. Click the circular surface of one of the bolts to add it to the **Selection** list.



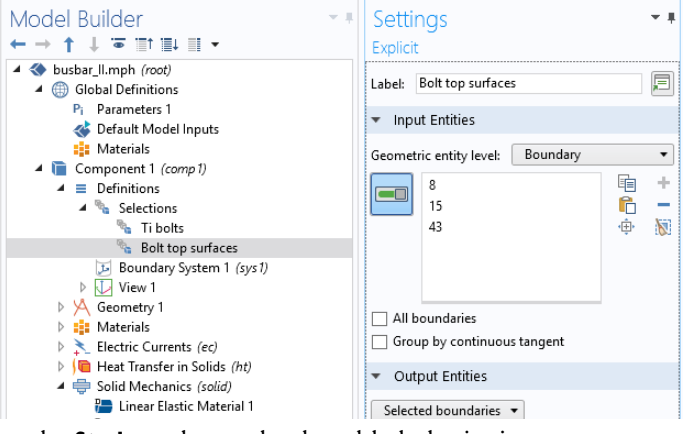
9 Repeat this procedure for the remaining bolts to add boundaries 8, 15, and 43.





For easier management of multiple boundary selections, you can group boundaries into a user-defined selection. Once you have selected boundaries 8, 15, and 43, click the **Create Selection** button and give it a name such as `Bolt top surfaces`.

This selection is then added as a node under **Component 1 > Definitions** and is accessible from the **Selection** list in the **Settings** window for all types of boundary conditions. You can use this technique to group domains, boundaries, edges, and points.



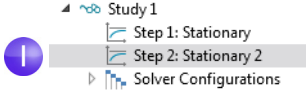
Next, we update the **Study** node to take the added physics into account.

Solving for Joule Heating and Thermal Expansion

The Joule heating effect is independent of the stresses and strains in the busbar, assuming temperature invariant materials, small deformations, and ignoring the effects of electric contact pressure. This means that you can run the simulation using the temperature as merely an input to the structural analysis. In other words, the extended multiphysics problem is weakly coupled. As such, you can solve it in two separate study steps to save computation time — one for Joule heating and a second one for structural analysis. In a more advanced analysis, you can include all of the aforementioned effects.

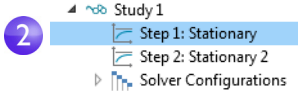
For this example, you can skip these steps and just click **Compute**. However, the following technique can be useful for saving computational time and memory resources, especially for larger simulations.

1 In the **Model Builder**, right-click **Study 1** and select **Study Steps > Stationary > Stationary** to add a second stationary study step.



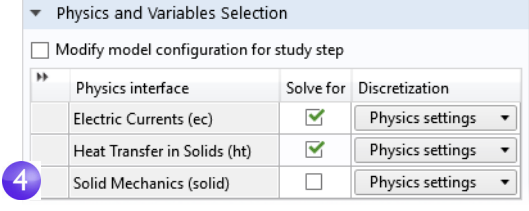
When adding study steps, you need to manually connect the respective physics with the correct study step. We start by disabling the structural analysis from the first step.

2 Under **Study 1**, click the **Step 1: Stationary** node.



3 In the **Settings** window for **Stationary,** locate the **Physics and Variables Selection**.

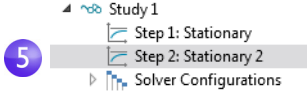
4 In the **Solid Mechanics (solid)** row under **Solve for**, click to change the check mark to an to remove **Solid Mechanics** from **Study Step** 1.



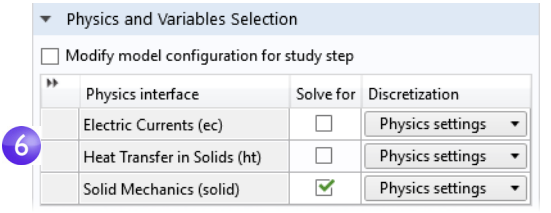
In the **Multiphysics couplings** section, keep all of the default settings.

Now, repeat these steps to remove **Electric Currents (ec)** and **Heat Transfer in Solids (ht)** from the second study step.

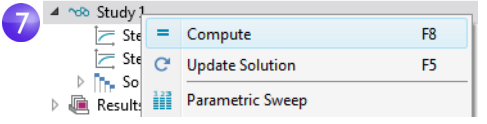
5 Under **Study 1**, click **Step 2: Stationary 2**.



6 Under **Physics and Variables Selection**, in the **Electric Currents (ec)** and **Heat Transfer in Solids (ht)** rows, click to change the check mark ☑ to an ☐ to remove Joule heating from **Step 2: Stationary 2**.



7 Right-click the **Study 1** node and select **Compute** (or press F8 or click **Compute** in the ribbon) to solve.

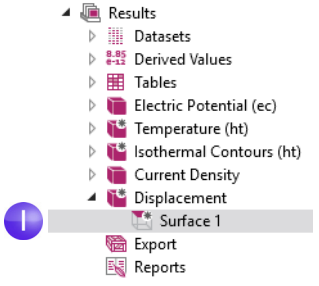


Save the file as `busbar_II.mph`, which now includes the **Solid Mechanics** interface and the additional study step.

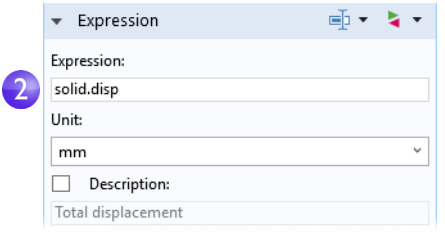
Resulting Deformation

Now that a physics interface has been added, additional plots for **Solid Mechanics** need to be added manually. First, a displacement plot.

1 In the **Model Builder**, right-click **Results** and add a **3D Plot Group**. In the **Label** field enter `Displacement` to give the plot group a descriptive name. Right-click **Displacement** and add a **Surface 1** node. Click the **Surface 1** node.



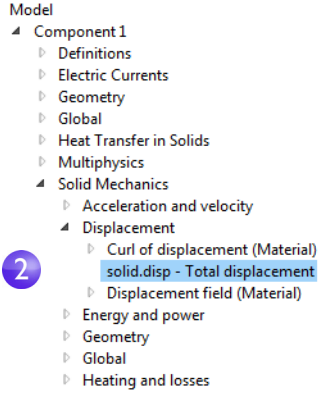
2 In the **Settings** window for **Surface** in the **Expression** section, click the **Replace Expression** button .



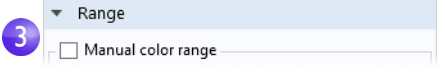
From the context menu, select **Model > Component 1 > Solid Mechanics > Displacement > solid.disp - Total displacement - m**.

You can also enter `solid.disp` in the **Expression** field.

Now, change to a more suitable unit. In the **Settings** window under **Expression**, from the **Unit** list, select `mm` (or type `mm` in the field).



3 Click **Range** to expand that section. Click to clear the **Manual color range** check box.

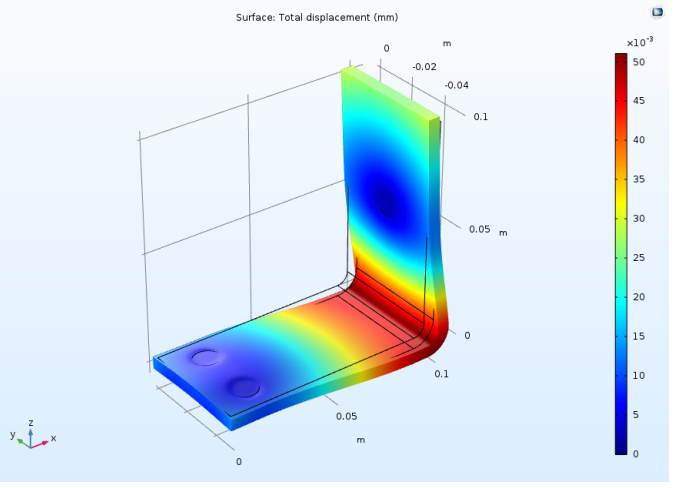


The local displacement due to thermal expansion is displayed as a surface plot. Next, we will add information about the busbar deformation.

4 In the **Model Builder**, under **Results > Displacement**, right-click the **Surface 1** node and add a **Deformation** node . The plot automatically updates in the **Graphics** window. Click the **Go to Default View** button so your view appears similar to the plot shown below.

The deformations shown in the figure are highly amplified to make the very small distortions that actually take place more visible.

4



5 Save the `busbar_II.mph` file, which now includes a **Surface** plot with a **Deformation**.

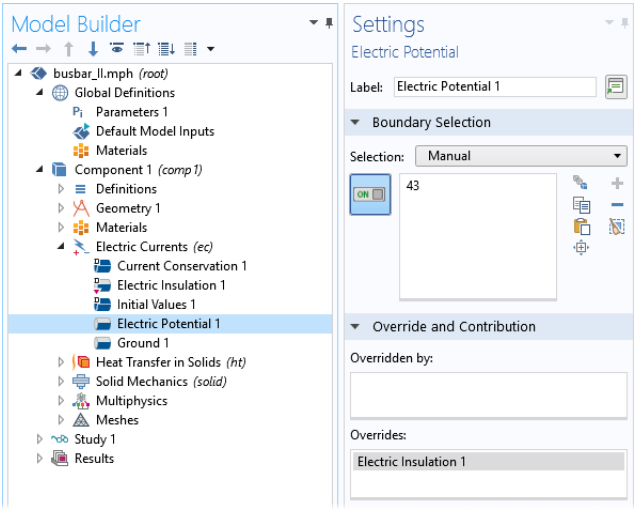
You can also plot the von Mises and principal stresses to assess the structural integrity of the busbar and the bolts.

Override and Contribution: Exclusive and Contributing Nodes

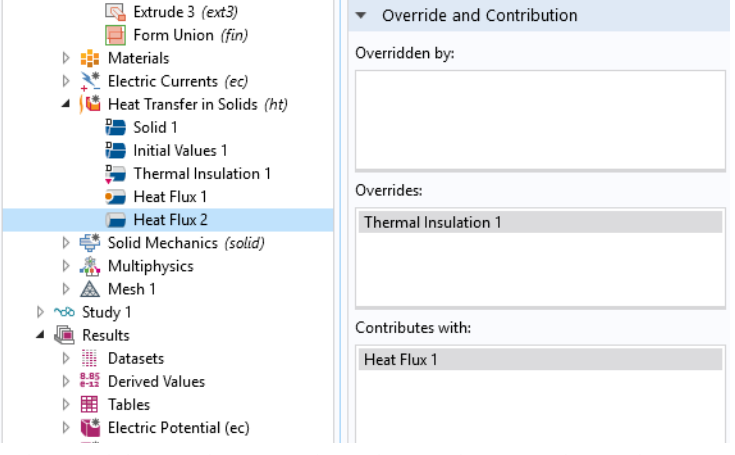
The model tree nodes for a physics interface are traversed in a downward sequence starting from the top. Depending on the selection, a node can totally or partially override, or shadow, a node that comes earlier in the sequence. There are two types of physics interface nodes: exclusive and contributing. How the nodes are treated depends on the type.

A constraint boundary condition is a typical example of an exclusive node. Examples of constraint boundary conditions include **Electric Potential** or **Fixed Constraint**. In the case of **Electric Currents**, for example, if there are two or more **Electric Potential** nodes for the same boundary, then the last **Electric Potential** node in the sequence will override the settings of the other nodes. The **Ground** node is also an exclusive node and will similarly override the boundary conditions specified by any previous node.

The figure below shows the **Override and Contribution** section in the **Settings** window for **Electric Potential**. The **Overrides** list indicates that this boundary condition takes precedence over the default **Electric Insulation** boundary condition.



A flux boundary condition, such as one specifying a heat flux, is a typical example of a contributing node. You can stack several heat flux boundary conditions and they will all contribute to the total heat flux. The figure below shows an example where a **Heat Flux 2** boundary condition overrides the default **Thermal Insulation** boundary condition and contributes to the heat flux together with **Heat Flux 1**.



In the model tree, the override and contribution relationships are indicated by a red arrow and an orange circle, respectively.

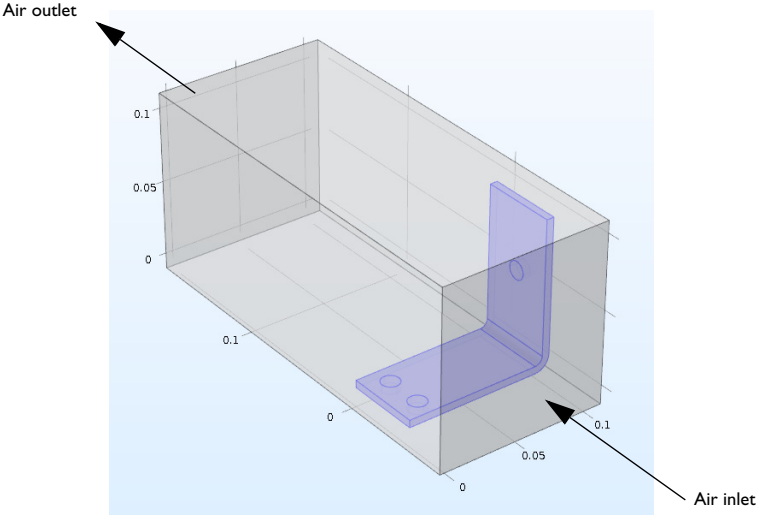
Cooling by Adding Fluid Flow

After analyzing the heat generated in the busbar and possibly the induced thermal expansion, you might want to investigate ways of cooling the busbar by letting air flow over its surfaces. These steps do not require any additional modules, as the model only includes Joule heating without the previous thermal expansion analysis.

If you have the CFD Module, the **Non-Isothermal Flow** multiphysics interface is available. If you have the Heat Transfer Module, the **Conjugate Heat Transfer** multiphysics interface is available. Either one of these two interfaces automatically defines coupled heat transfer in solids and fluids including laminar or turbulent flow, whereas in this example, this is done manually and with limited functionality.

Adding fluid flow to the Joule heating model creates a new multiphysics coupling. To simulate the flow domain, you need to create an air box around the busbar for the external flow. You can do this manually by altering the geometry from your first model or by opening an Application Library file. To save time, open a file with the box already created.

After loading the geometry, you will learn how to simulate air flow according to this figure:



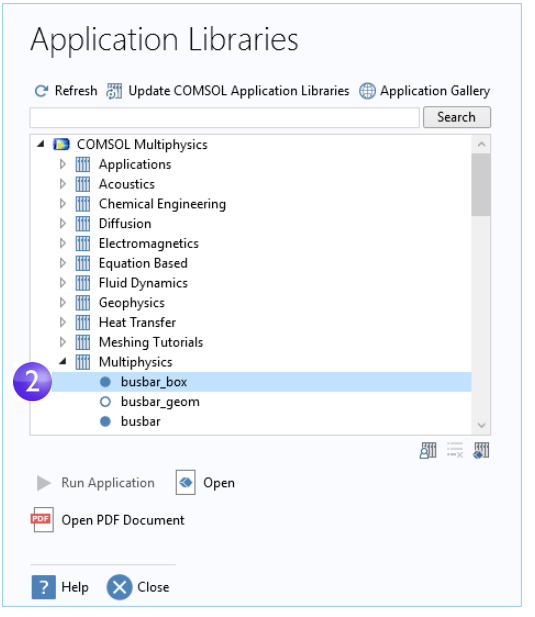
Defining Inlet Velocity

Start by loading the geometry and adding a new parameter for the inlet flow velocity.

1 If you have just reopened the software, close the **New** window that opens automatically by clicking the **Cancel** button.

2 Click the **Home** tab, and from the **Windows** menu select **Application Libraries**. Navigate to **COMSOL Multiphysics > Multiphysics > busbar box**. (You can also open the Application Libraries directly from the **File** menu.)

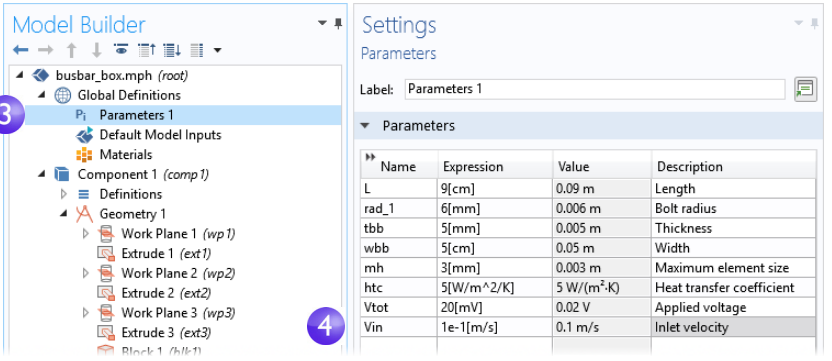
Double-click to open the file, which contains the geometry in addition to the physics modeling steps completed through the end of the section "Customizing Materials" on page 115.



3 Under **Global Definitions,** click the **Parameters** node.

4 In the **Settings** window for **Parameters**, click the empty row just below the `Vtot` row. In the **Name** column, enter `Vin`. Enter `0.1[m/s]` in the **Expression** column

and a description of your choice in the **Description** column, such as `Inlet velocity`.



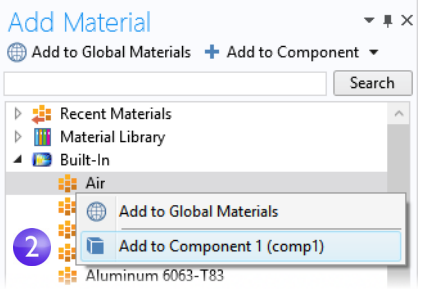
5 Select **File > Save As** and save the model with a new name, `busbar_box_I.mph`.

Adding Air

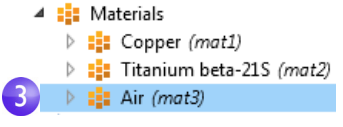
The next step is to add the material properties of air.

1 From the **Home** tab, select **Add Material** (or right-click the **Materials** node and select **Add Material from Library**.)

2 In the **Add Material window**, expand the **Built-In** node. Right-click **Air** and select **Add to Component 1**. Close the **Add Material** window.

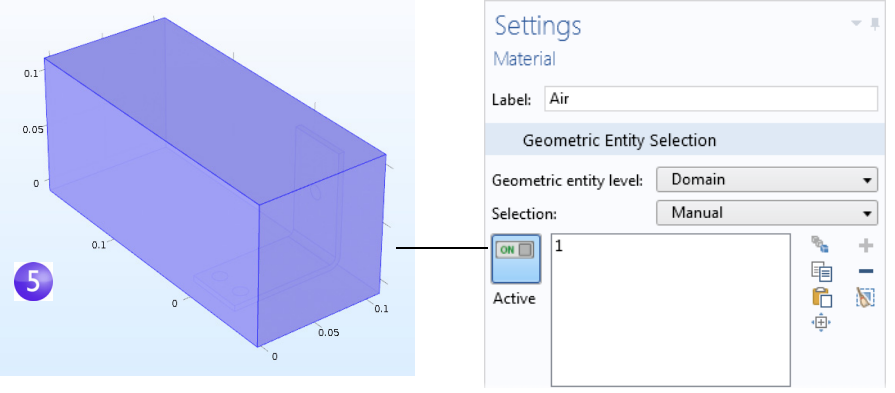


3 In the **Model Builder** under **Materials**, click the **Air** node .



4 On the **Graphics** window toolbar, click **Zoom Extents**.

5 In the **Graphics** window, click the air box (domain 1) to add it to the **Selection** list, which changes the color to blue. This step assigns the air material properties to the air box.

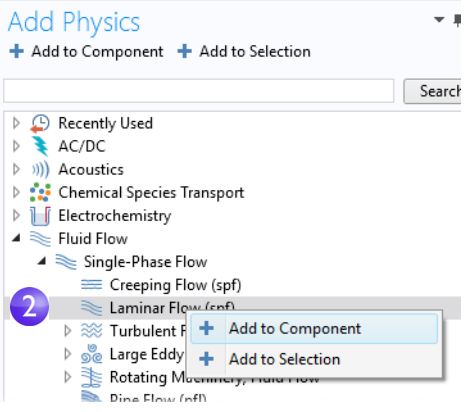


Adding Fluid Flow

Now, add the physics of fluid flow.

1 In the model tree, right-click **Component 1** and select **Add Physics**.

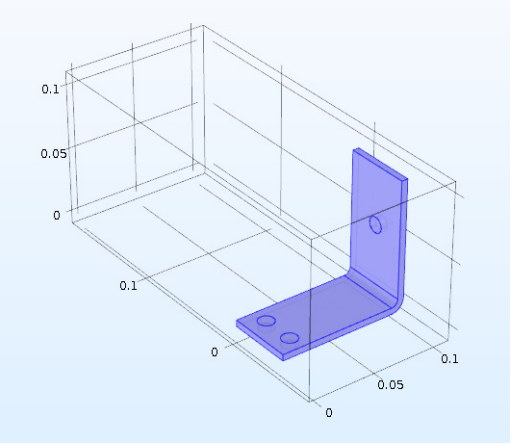
2 In the **Add Physics** window under **Fluid Flow > Single-Phase Flow,** right-click **Laminar Flow** and select **Add to Component**. **Laminar Flow** will appear under **Component 1** in the model tree. Close the **Add Physics** window.



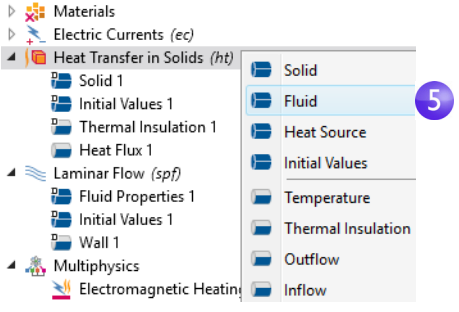
3 In the **Graphics** toolbar, click the **Transparency** button. In addition, click the **Wireframe Rendering** button. These two settings make it easier to see inside the box. Toggle these on and off as needed during the modeling process to control the type of rendering used.

Now that you have added fluid flow to the model, you need to remove the air domain (domain 1) from the **Electric Currents (ec)** interface (assuming that the air has no electric conductivity and hence no currents) and then couple the heat transfer part of the **Joule Heating** multiphysics interface to the fluid flow.

4 In the model tree, select the **Electric Currents (ec)** node . In the **Graphics** window, move the mouse pointer over the air domain and click to remove it from the selection list. At this point, only the busbar should be selected and highlighted in blue.

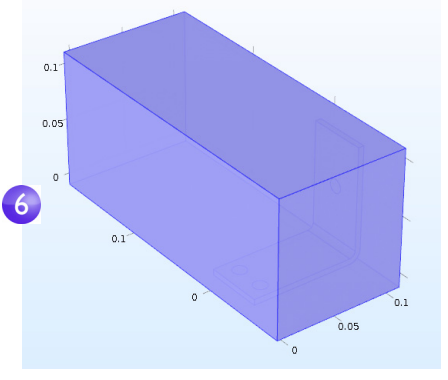


5 In the **Model Builder**, right-click **Heat Transfer in Solids** . In the first section of the context menu, the domain level , select **Fluid**.

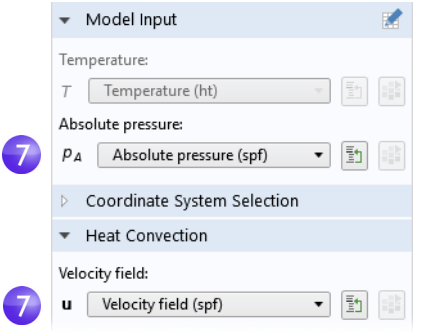


6 In the **Graphics** window, click the air domain (domain 1) to add it to the **Selection** list.

Now, couple the fluid flow and heat transfer phenomena.



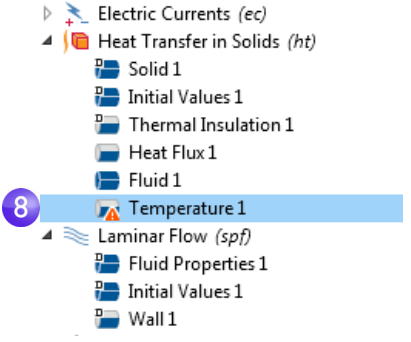
7 In the **Settings** window for **Fluid**, under **Heat Transfer in Solids**, in the section **Model Inputs**, select **Absolute pressure (spf)** from the **Absolute pressure** list. In the section **Heat Convection**, select **Velocity field (spf)** from the **Velocity field** list.



This identifies the pressure and flow field from the **Laminar Flow** interface and couples it to heat transfer.

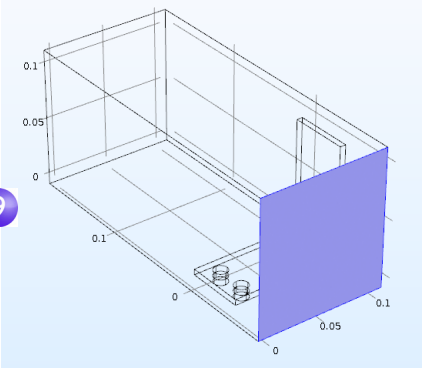
Now, define the boundary conditions by specifying the inlet and outlet for the heat transfer in the fluid domain.

8 In the **Model Builder**, right-click **Heat Transfer in Solids**. In the second section of the context menu, the boundary section, select **Temperature**.



A **Temperature** node is added to the model tree.

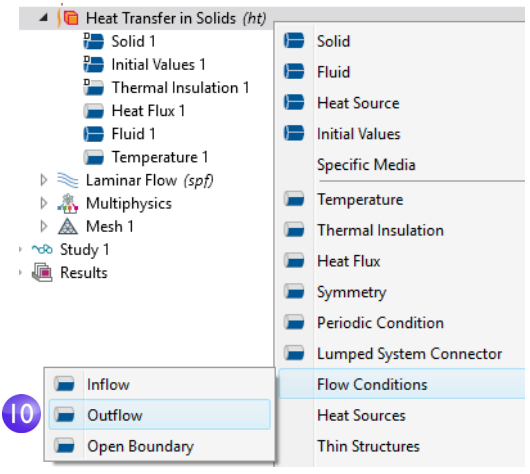
9 In the **Graphics** window, click the inlet boundary, boundary number 2, to add it to the **Selection** list.



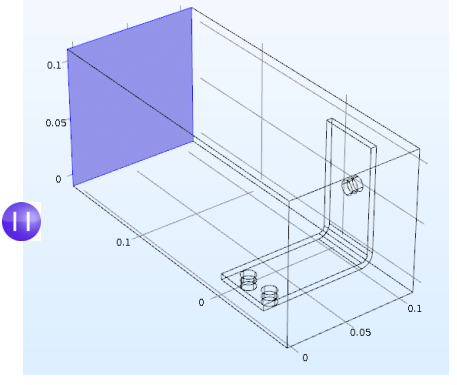
This sets the inlet temperature to 293.15 K, the default setting.

The graphics should look like the image to the right. (It may look slightly different depending on whether you have **Transparency** and **Wireframe Rendering** on or off.) Continue by defining the outlet.

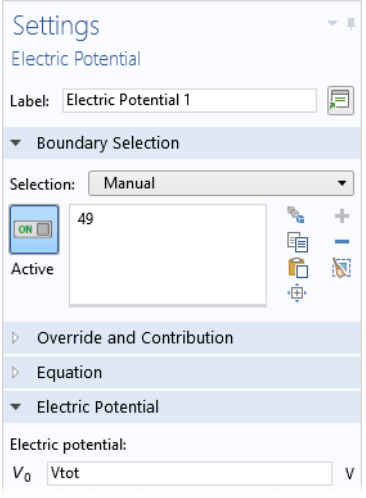
10 In the **Model Builder**, right-click **Heat Transfer in Solids** . At the boundary level, select **Flow Conditions>Outflow**. An **Outflow** node is added to the model tree.

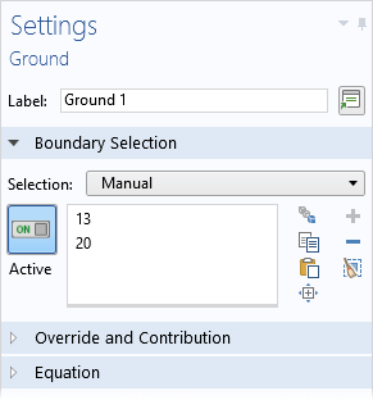


11 In the **Graphics** window, click the outlet boundary, boundary number 5, to add it to the **Selection** list. Use the mouse scroll wheel to scroll in and highlight the boundary before selecting it, or use the up and down arrow keyboard buttons.



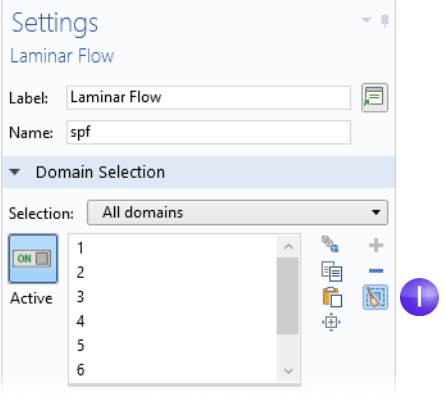
The settings for the busbar, the bolts, and the **Electric Potential 1** and **Ground 1** boundaries have retained the correct selection, even though you added the box geometry for the air domain. To confirm this, in the model tree under **Electric Currents**, click the **Electric Potential 1** and the **Ground 1** nodes to verify that they have the correct boundary selection.



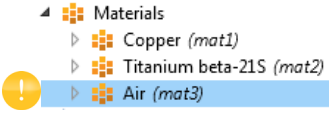


Next, continue with the flow settings. You need to indicate that fluid flow only takes place in the fluid domain and then set the inlet, outlet, and symmetry conditions. Do this by first removing all domains from the selection and then adding the air domain.

1. In the model tree, click the **Laminar Flow** node. In the **Settings** window for **Laminar Flow**, click the **Clear Selection** button.



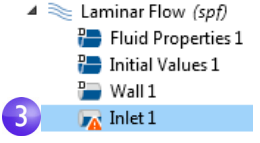
2 In the **Graphics** window, click the air box (domain 1) to add it to the **Selection**.



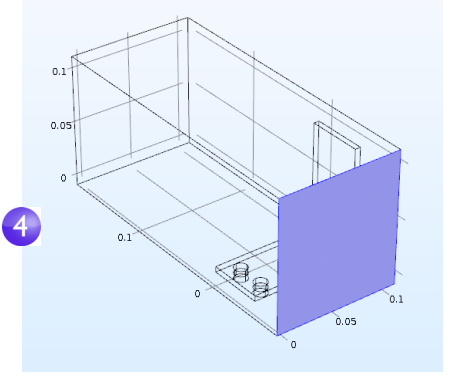
It is good practice to verify that the **Air** material under the **Materials** node has all of the properties that this multiphysics combination requires. In the model tree under **Materials**, click **Air**. In the Settings window for **Material** under **Material Contents**, verify that there are no missing properties, which are marked with a warning sign. The section "Materials" on page 73 has more information.

Now, we continue with the boundaries.

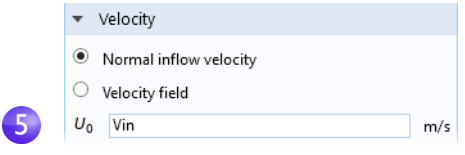
3 In the **Model Builder**, right-click **Laminar Flow** and at the boundary level, select **Inlet**. An **Inlet** node is added to the model tree.



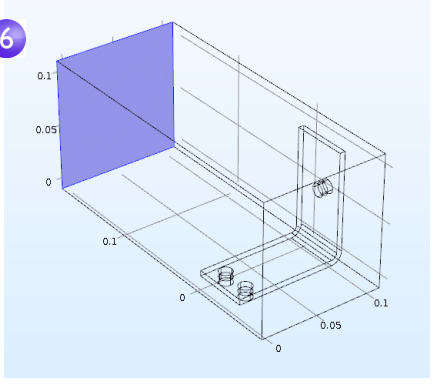
4 In the **Graphics** window, click the inlet (boundary 2) to add it to the **Selection** list.



5 In the **Settings** window for **Inlet** under **Velocity** in the U_0 field, enter `Vin` to set the **Normal inflow velocity**.



6 Right-click **Laminar Flow** and at the boundary level, select **Outlet**. In the **Graphics** window, click the outlet (boundary 5) to add it to the **Selection** list. Use the mouse scroll wheel or keyboard arrows to scroll in and highlight the boundary before selecting it.

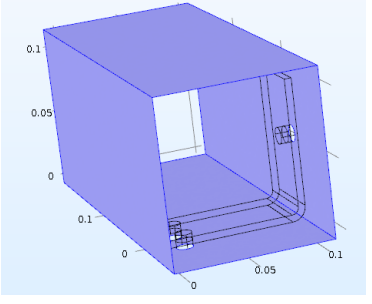


The last step is to add symmetry boundaries. For simplicity, assume that the flow just outside of the faces of the channel is similar to the flow just inside these faces. This assumption can be correctly expressed by the symmetry condition.

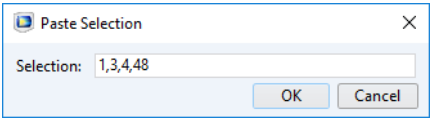
7 Right-click **Laminar Flow** and select **Symmetry**. A **Symmetry** node is added to the sequence.

8 In the **Graphics** window, click each of the blue faces in the figure below (boundaries `1`, `3`, `4`, and `48`) to add all of them to the **Selection** list. You may need to use the mouse scroll wheel or rotate the geometry to select all of them.

Save the `busbar_box_I.mph` file, which now includes the **Air** material and **Laminar Flow** interface settings.



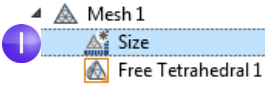
When you know the boundary numbers, you can click the **Paste Selection** button and enter the information. In this example, enter `1,3,4,48` in the **Paste Selection** window. Click **OK** and the boundaries are automatically added to the **Selection** list.



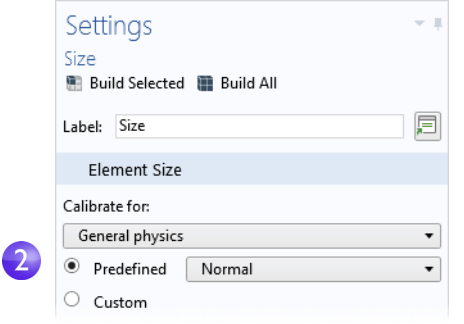
Coarsening the Mesh

To get a quick solution, we will change the mesh slightly and make it coarser. The current mesh settings would take a relatively long time to solve, and you can always refine it later.

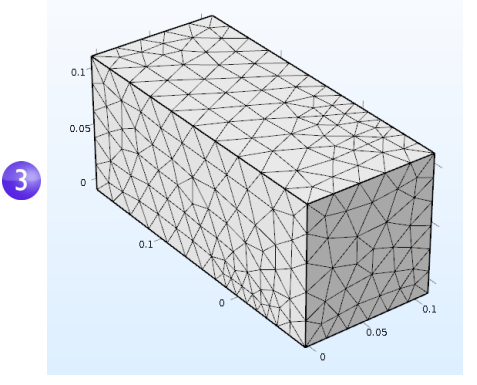
1 In the **Model Builder**, expand the **Mesh 1** node and click the **Size** node .



2 In the **Settings** window for **Size** under **Element Size**, click the **Predefined** button and ensure that **Normal** is selected.



3 Click the **Build All** button . The geometry displays with a mesh in the **Graphics** window (you may need to turn **Transparency** off to see the picture below.)



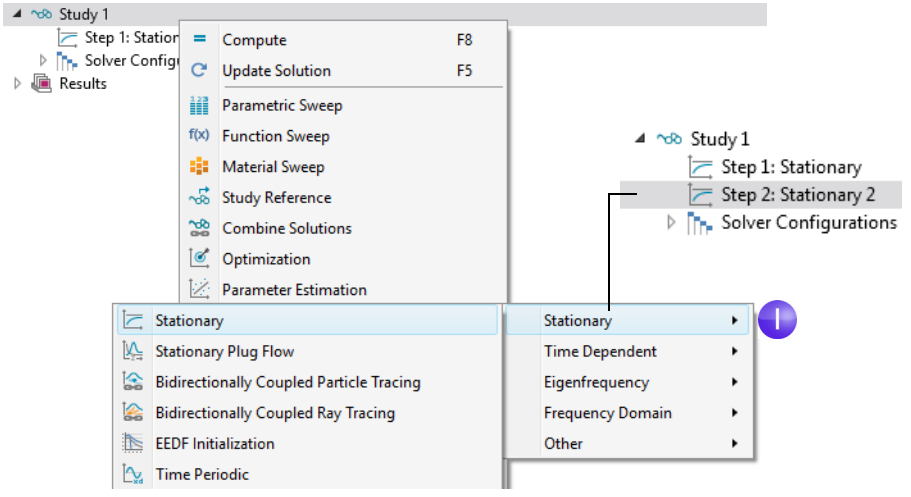
You can assume that the flow velocity is large enough to neglect the influence of the temperature increase in the flow field.

It follows that you can solve for the flow field first and then solve for the temperature using the results from the flow field as input. This is implemented with a study sequence.

Solving for Fluid Flow and Joule Heating

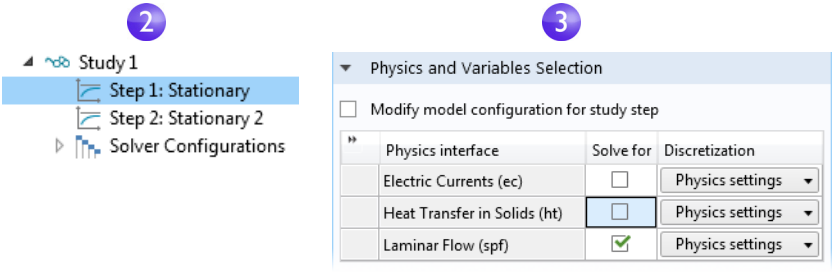
When the flow field is solved before the temperature field, it yields a weakly coupled multiphysics problem. The study sequence described in this section automatically solves such a weak, or one-directional, coupling.

1 In the model tree, right-click **Study 1** and select **Study Steps > Stationary > Stationary** to add a second stationary study step to the **Model Builder**.



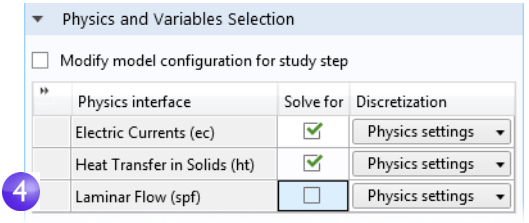
Next, the correct physics needs to be connected with the correct study step. Start by disabling the **Electric Currents (ec)** and **Heat Transfer in Solids (ht)** interfaces associated with Joule heating from the first step.

2 Under **Study 1**, click **Step 1: Stationary**.

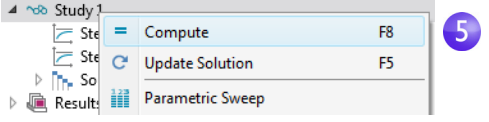


3 In the **Settings** window for **Stationary**, locate the **Physics and Variables Selection** section. In both the **Electric Currents (ec)** and the **Heat Transfer in Solids (ht)** rows, click to change the check mark ☑ to an ☐ in the **Solve for** column, removing the Joule heating effect from **Step 1**.

4 Repeat the step. Under **Study 1,** click **Step 2: Stationary 2** . Under **Physics and Variables Selection**, in the **Laminar Flow (spf)** row, click in the **Solve for** column to change the check mark ☑ to an ☐.

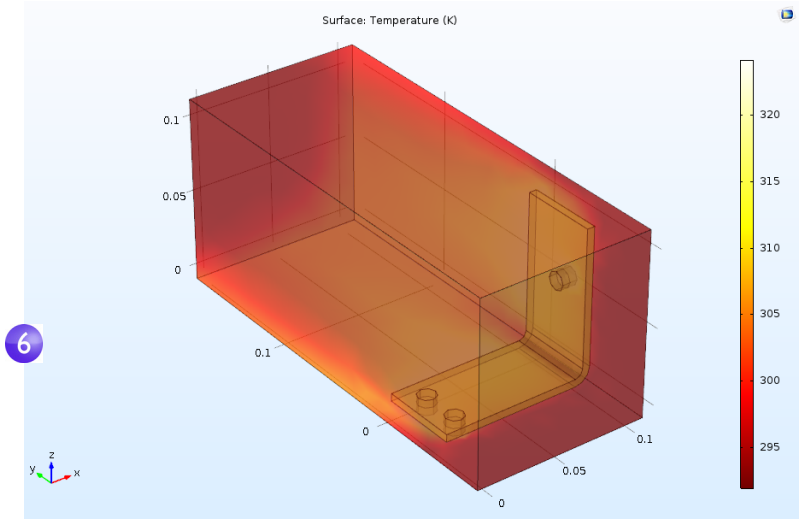


5 Right-click the **Study 1** node and select **Compute** (or press F8 or click **Compute** in the ribbon). This will now automatically create a new solver sequence that solves for laminar flow and then Joule heating.



6 After the solution is complete, select the **Temperature (ht)** plot under the **Results** node in the **Model Builder**. If you do not already have transparency on, click the **Transparency** button on the **Graphics** toolbar to visualize the temperature field

inside the box. To zoom, middle-click and hold down the mouse button (or scroll wheel) while dragging the mouse.



The **Temperature Surface** plot that displays in the **Graphics** window shows the temperature in the busbar and in the surrounding box. You can also see that the temperature field is not smooth due to the relatively coarse mesh. A good strategy to get a smoother solution would be to refine the mesh to estimate the accuracy.

7 At this point, save the `busbar_box_I.mph` file so you can return to it at any time. The next steps use the original `busbar.mph` file.

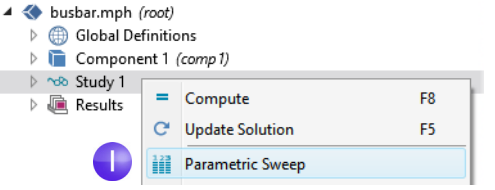
Parametric Sweeps

Sweeping a Geometric Parameter

It is often useful to generate multiple instances of a design with the objective of meeting specific constraints. In the previous busbar example, a design goal might be to lower the operating temperature or decrease the current density. We will demonstrate the former. Since the current density depends on the geometry of the busbar, varying the width, `wbb`, should change the current density and, in turn, have some impact on the operating temperature. Let us run a parametric sweep on `wbb` to study this change.

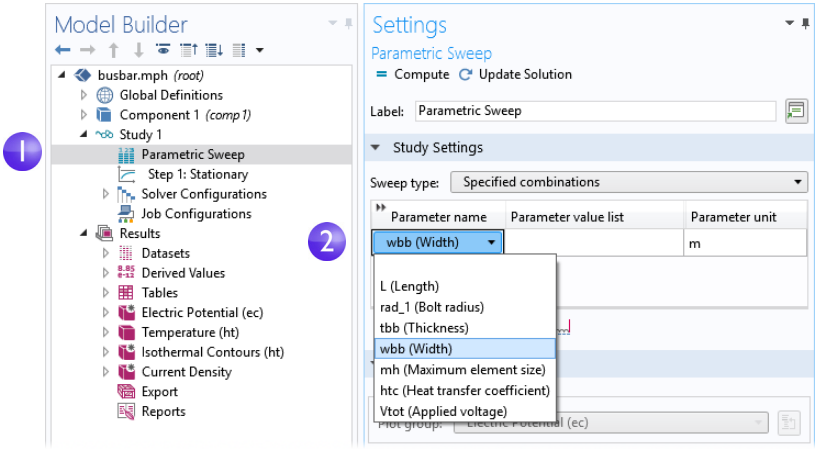
Adding a Parametric Sweep

1 From the **File** menu, open the model file `busbar.mph`. If you did not save the model, you can also open it from the Application Libraries: **File > Application Libraries > COMSOL Multiphysics > Multiphysics > busbar**.

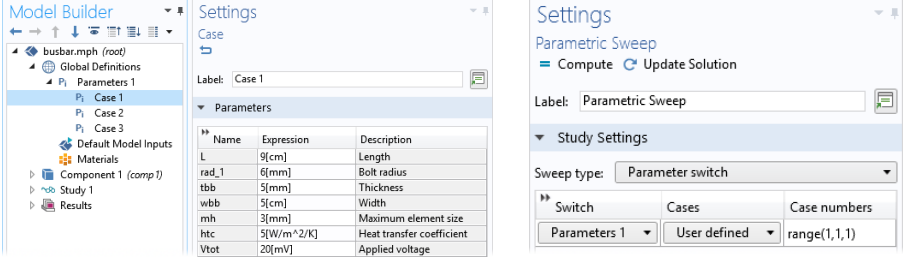


In the **Model Builder**, right-click **Study 1** and select **Parametric Sweep**. A **Parametric Sweep** node is added to the **Model Builder** sequence.

2 In the **Settings** window for **Parametric Sweep**, under the empty parameter table, click the **Add** button +. From the **Parameter names** list in the table, select `wbb`.

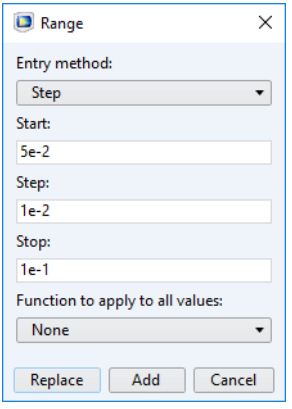


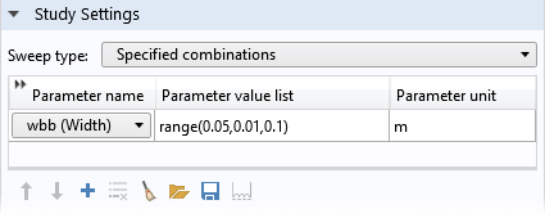
The **Sweep type** is used to control parametric sweeps with multiple parameters. You select between sweeping for **All combinations** of the given parameters, a subset of **Specified combinations**, or **Parameter switch**. The option **Parameter switch** makes it possible to sweep over **Cases** for **Parameters** as defined in **Global Definitions**, as shown below.



3 Enter a range of parameter values to sweep the width of the busbar from 5 cm to 10 cm in 1 cm increments. There are different ways to enter this information:

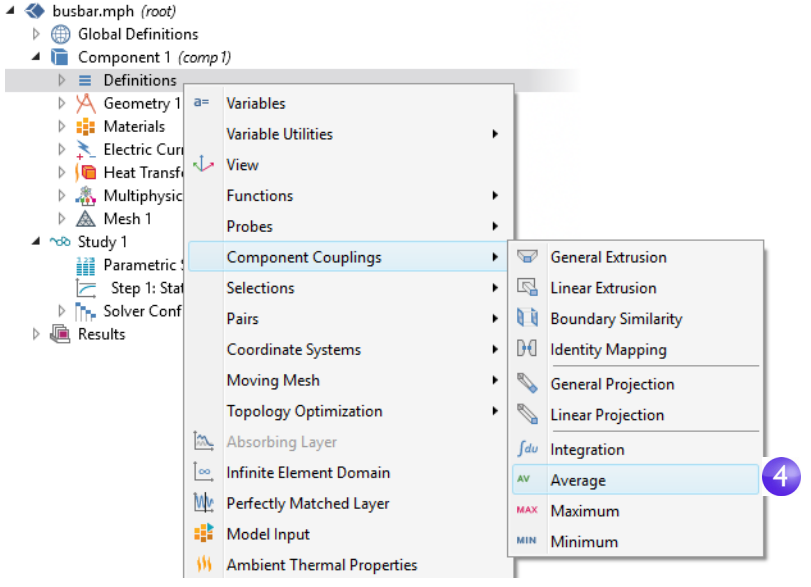
- Copy and paste or enter `range(0.05,0.01,0.1)` into the **Parameter value list** field.
- Click the **Parameter value list** field and then click the **Range** button and enter the values in the **Range** dialog box. In the **Start** field, enter `5e-2`. In the **Step** field, enter `1e-2`, and in the **Stop** field, enter `1e-1`. Click **Replace**.
- In any of the methods, you can also use length units to override the default SI unit system using meters. Instead of `5e-2`, you can enter `5[cm]`; similarly, `1[cm]` instead of `1e-2` and `10[cm]` instead of `1e-1`. You can also change the default unit system from the **Settings** window of the root node in the model tree.



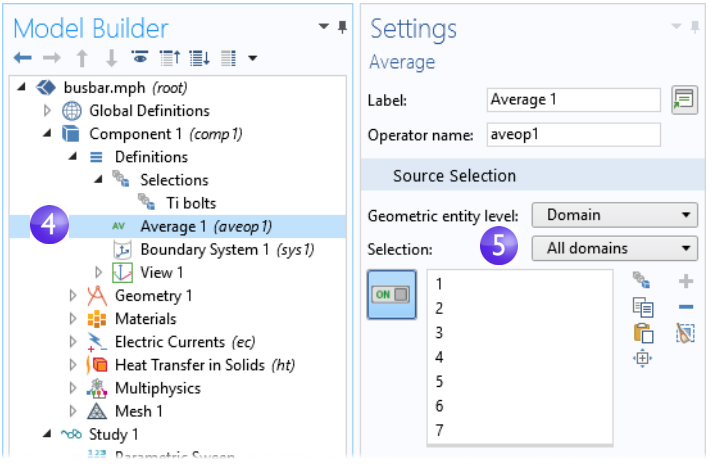


Next, define an **Average** nonlocal coupling that can be used later to calculate the average temperature in the busbar.

4 Under **Component 1**, right-click **Definitions** ≡ and select **Nonlocal Couplings > Average** AV.



5 In the **Settings** window for **Average**, select **All domains** from the **Selection** list.



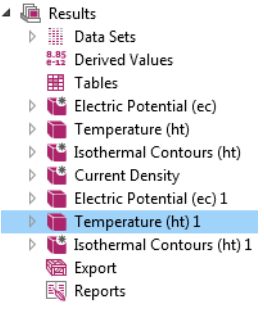
This creates an operator called `aveop1`. This operator is now available to calculate the average of any quantity defined on the selected domains. Later, we use this tool to calculate the average temperature, but it can also be used to calculate average electric potential, current density, and so forth.

6 Select **File > Save As** to save the model with the new name, `busbar_III.mph`.

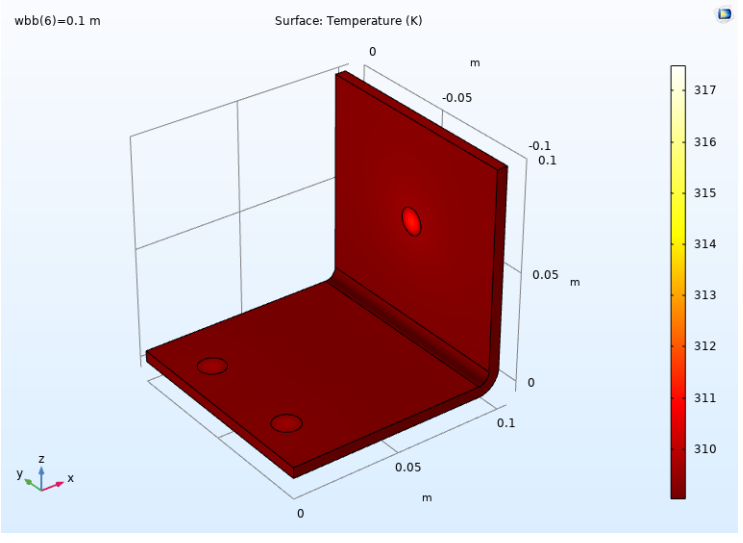
7 Right-click **Study 1** and select **Compute** to run the sweep or click **Compute** in the **Home** tab.

Parametric Sweep Results

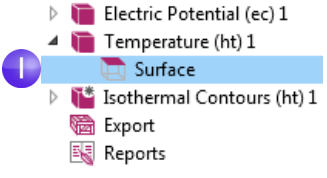
Click the **Temperature (ht) 1** node (the second temperature plot node) located under **Results** in the model tree.



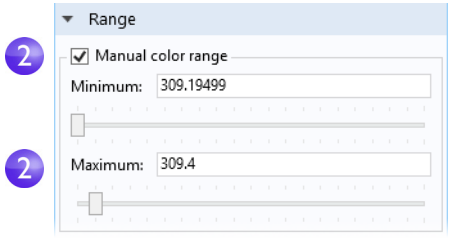
The plot that displays in the **Graphics** window shows the temperature in the wider busbar using the last parameter value, `wbb=0.1[m] (10[cm])`. Select **Zoom Extents** from the **Graphics** window toolbar so that you can see the entire plot. The resulting plot is rather uniform in color, so we need to change the maximum color range.



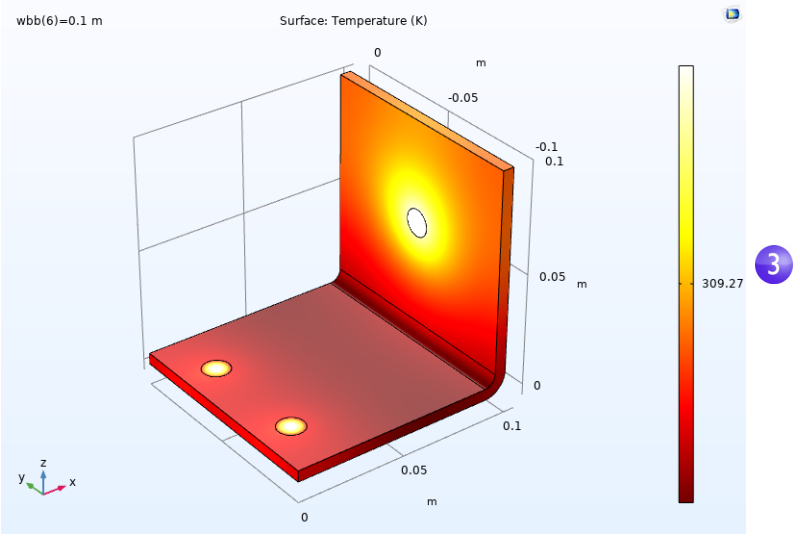
1 Under the **Temperature (ht) 1** node, click the **Surface** node.



2 In the **Settings** window for **Surface**, click **Range** to expand the section. Select the **Manual color range** check box. Enter `309.5` in the **Maximum** field (replace the default).

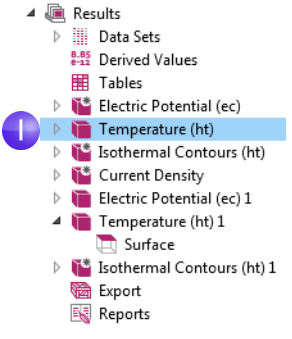


3 The **Temperature (ht) 1** plot is updated in the **Graphics** window for `wbb=0.1[m] (10[cm])`.



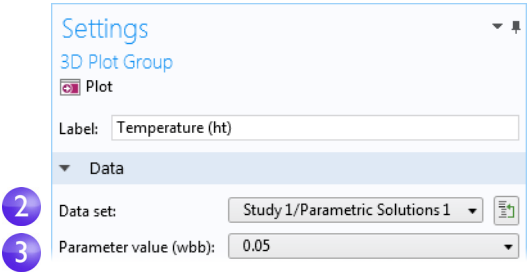
The plot may look a bit different depending on which version you are running. Compare the wider busbar plot to the temperature for `wbb=0.05[m] (5[cm])`. For this purpose, you can reuse one of the previously defined plot groups.

1 In the **Model Builder**, click the first **Temperature (ht)** node .

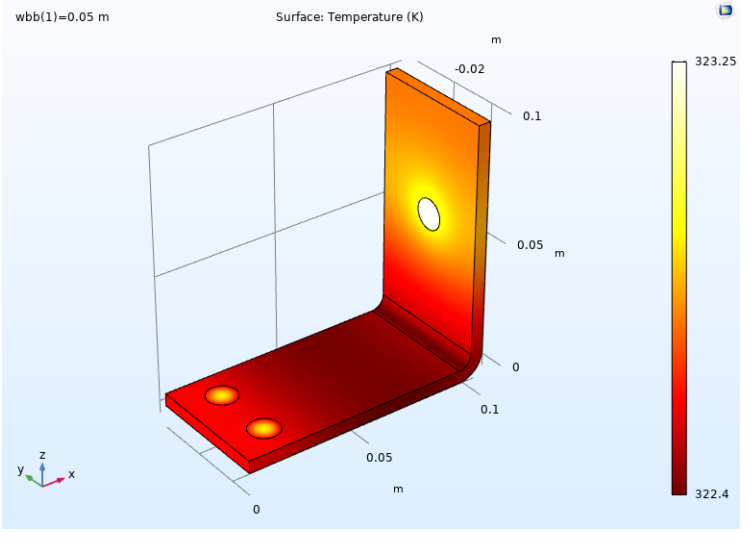


2 In the **Settings** window for **3D Plot Group**, select **Study 1/Parametric Solutions 1** from the **Dataset** list. This dataset contains the results from the parametric sweep.

3 In the **Parameter value** list, select **0.05** (which represents `wbb=5 cm`). Click the **Plot** button . Click the **Zoom Extents** button on the **Graphics window** toolbar.



The **Temperature (ht)** plot is updated for `wbb=0.05[m] (5[cm])`. Note that if you have updated the color range for this plot already, your plot should look similar to the one below. If not, follow the subsequent steps.



(The plot may include the max and min points, depending on which version of the model you started from.) Like the wider busbar, the plot may be quite uniform in color, so change the maximum color range.

1 Under the first **Temperature (ht)** node, click the **Surface** node .

2 In the Settings window for **Surface**, click **Range** to expand the section (if it is not already expanded). Select the **Manual color range** check box.

3 Enter `323.25` in the **Maximum** field (replace the default).

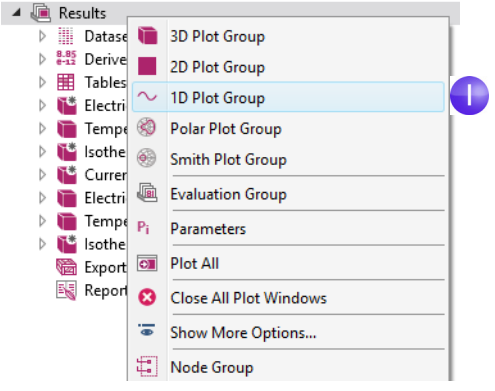
The **Temperature (ht)** plot is updated in the **Graphics** window for `wbb=0.05[m] (5[cm])`.

Click the first and second **Temperature** plot nodes to compare the plots in the **Graphics** window. The maximum temperature decreases as the width of the busbar increases from 5 cm to 10 cm.

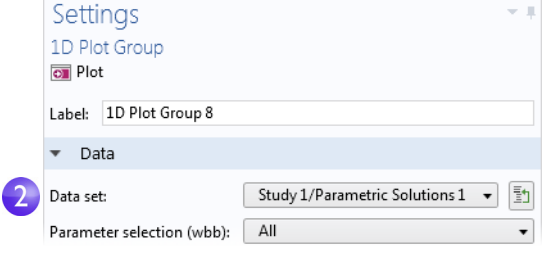
Adding More Plots

To further analyze these results, you can plot the average temperature for each width.

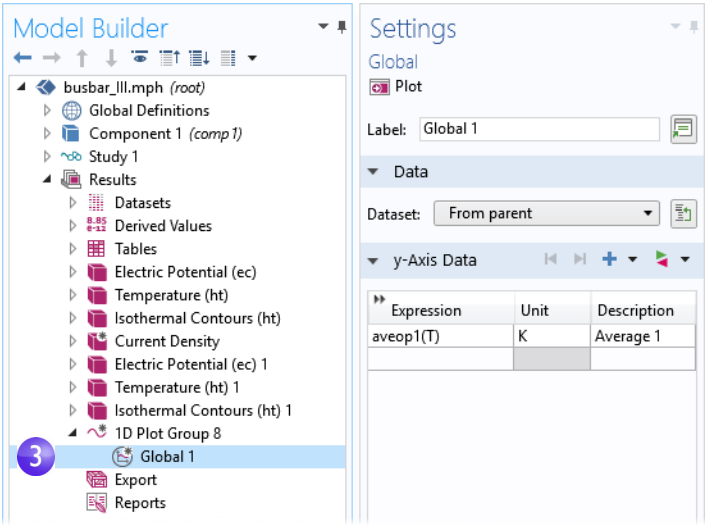
1 Right-click **Results** and add a **1D Plot Group**.



2 In the **Settings** window for **1D Plot Group 8**, select **Study 1/Parametric Solutions 1** from the **Dataset** list.

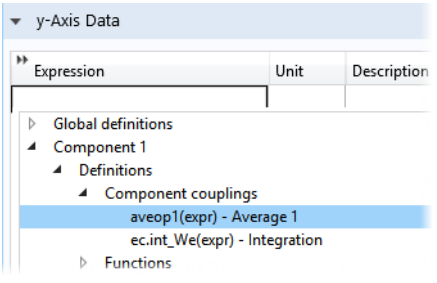


3 In the **Model Builder**, right-click **1D Plot Group 8** and add a **Global** node.

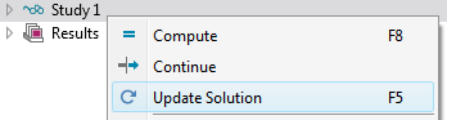


4 In the **Settings** window for **Global** under **y-Axis Data**, click the first row in the **Expressions** column and enter `aveop1(T)`. This is the operator that we defined on page 144 for later use. You use a similar syntax to calculate the average of other quantities.

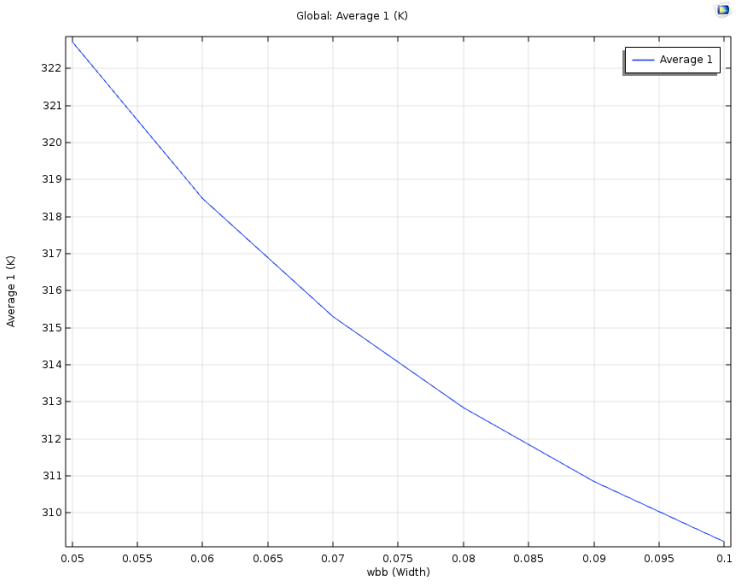
You can also use auto-completion by using Ctrl+Space after clicking the first row or after typing a few characters.



To use a nonlocal coupling operator without solving, you can select the option **Update Solution** that is available by right-clicking a **Study** node.



5 Click the **Plot** button and save the `busbar_III.mph` model with these additional plots that use the parametric sweep results.

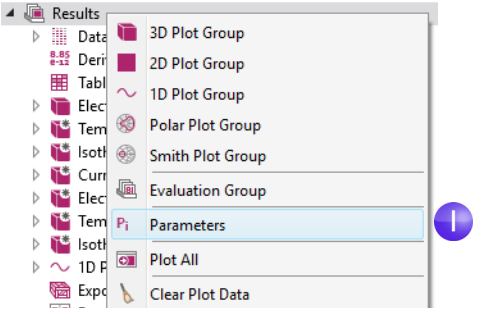


In the plot, the average temperature decreases as the width increases.

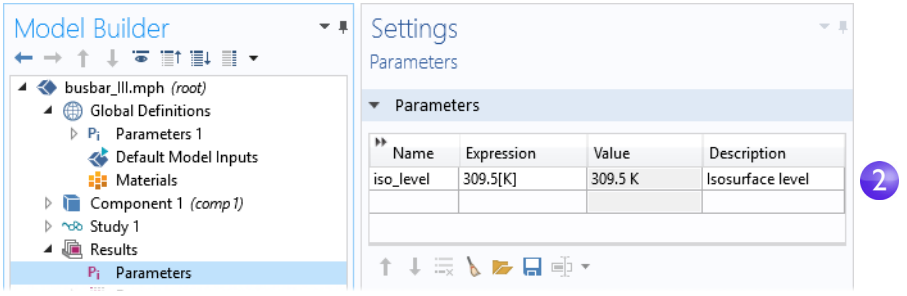
Parameters in Results

For greater flexibility, it is possible to define parameters that are only used in the **Results** node. Using these parameters does not require resolving the model. The following example shows how you can animate over a **Parameter** defined under **Results**.

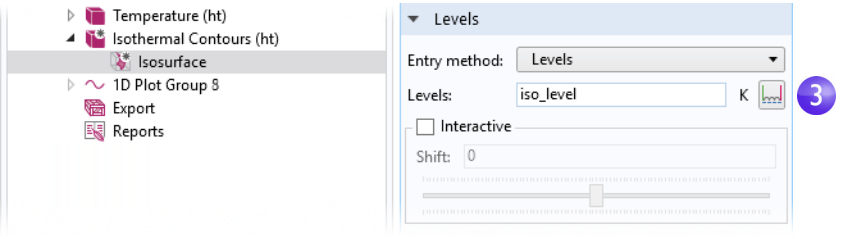
1 Right-click the **Results** node and select **Parameters**.



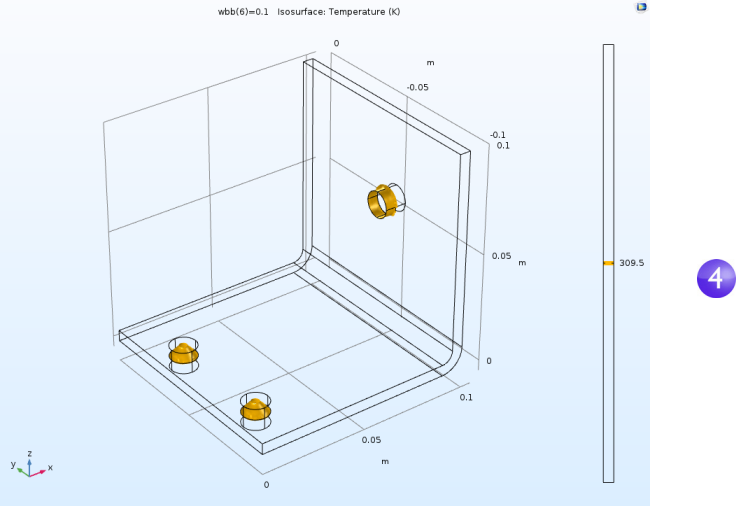
2 Define a parameter, `iso_level`, with **Expression** `309.5[K]`.



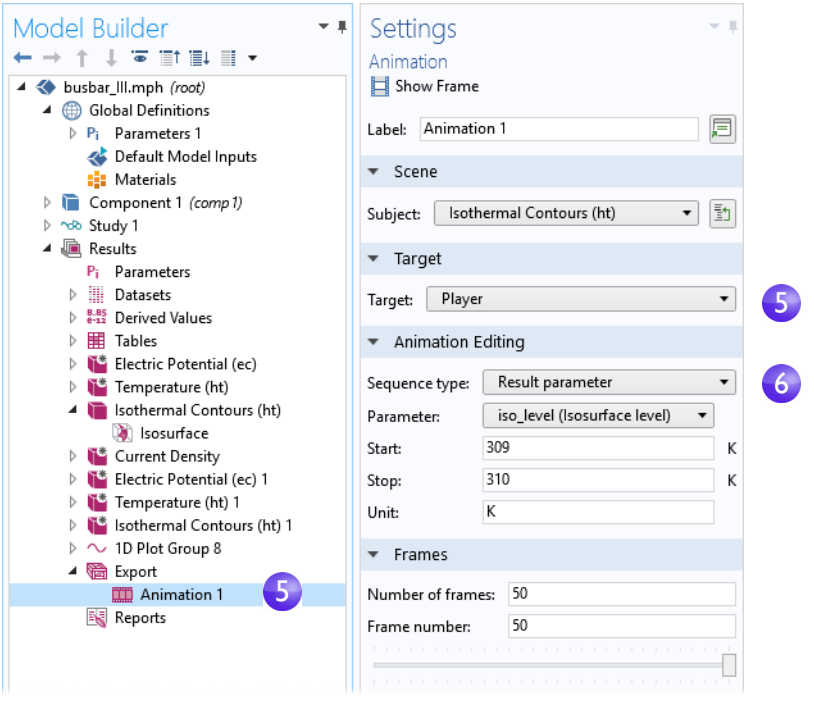
3 For the **Isothermal Contours (ht) 1** plot group, in the **Settings** window for the **Isosurface** plot, change the **Entry** method to **Levels**. For the **Levels** expression field, type `iso_level`.



4 In the same **Settings** window, click **Plot**.



5 Generate an **Animation** by either selecting the **Player** option in the **Animation** menu of the **Isothermal Contours (ht) I** tab in the ribbon, or right-click the Export node and select **Animation > Player**.



6 In the **Settings** window for **Animation**, change the **Sequence type** to **Result parameter** and select the `iso_level` parameter. Type `309` for **Start** and `310` for **Stop**. Click the **Play** button in the **Graphics** toolbar or the **Show Frame** button in the **Settings** window for the **Animation**. To get a smoother animation, change the **Number of frames** to a higher number; for example, `50`.

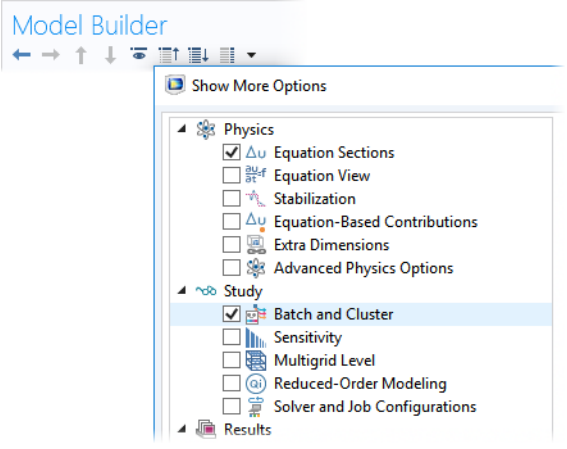
The subject of parametric sweeps raises the question of parallel computing; it would be efficient if all parameters were solved simultaneously.

Parallel Computing

COMSOL Multiphysics and COMSOL Server support most forms of parallel computing, including shared memory parallelism for multicore processors and high-performance computing (HPC) for clusters and clouds. All COMSOL

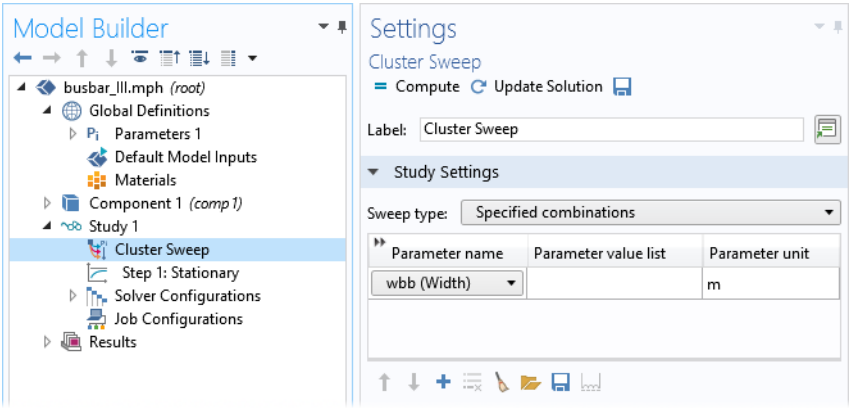
licenses are multicore enabled. For cluster or cloud computing, including parallelized sweeps, a Floating Network License (FNL) is needed.

You can use clusters or clouds with one of two options: either for **Cluster Sweep** or for **Cluster Computing**. If you have a Floating Network License, these two options are available by right-clicking the **Study** node. However, you first need to enable the **Batch and Cluster** study options by clicking the **Show More Options** button on the **Model Builder** toolbar and selecting **Batch and Cluster** in the **Show More Options** dialog box.



Cluster Sweep

The **Cluster Sweep** option is used for solving several models in parallel where each model has a different set of parameters. This can be seen as a generalization of **Parametric Sweep**. Right-click the **Study** node to add a **Cluster Sweep** node.



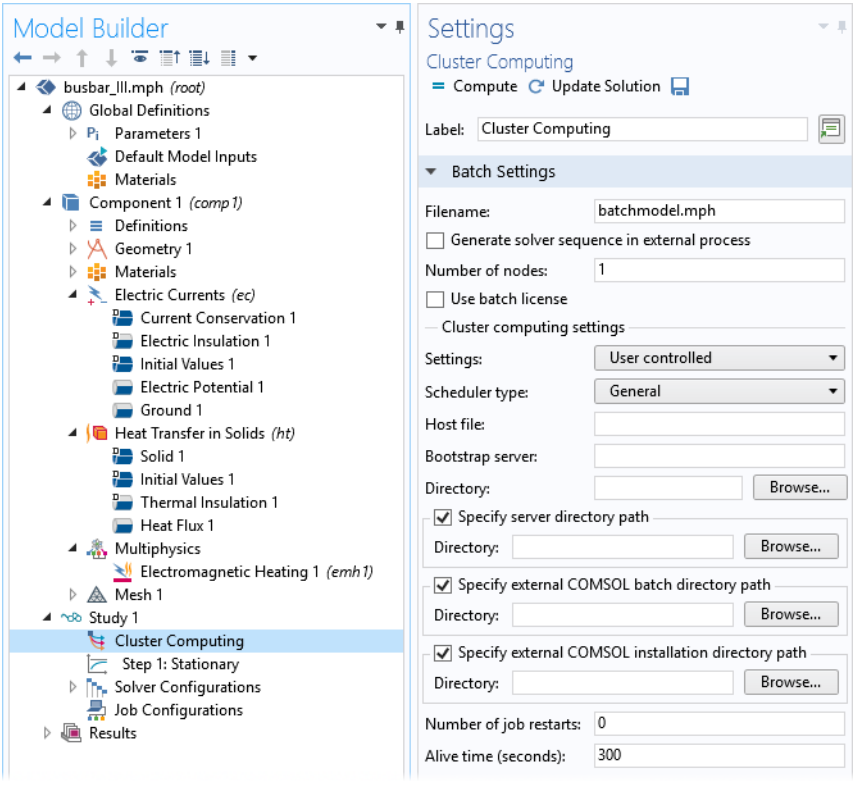
The **Study Settings** for **Cluster Sweep** are similar to that of **Parametric Sweep**, but additional settings are required for the cluster or cloud being used. The picture above shows how the top of the **Settings** window for **Cluster Sweep** would look for the same sweep as defined in "Parametric Sweeps" on page 141. To run a cluster sweep in this case, you would first remove the **Parametric Sweep** node.

Cluster Computing

You can also utilize a cluster or cloud to solve a single large model using distributed memory. For maximum performance, the COMSOL software cluster implementation can utilize shared-memory multicore processing on each node in combination with the message passing interface (MPI) based distributed memory model. This is also known as hybrid parallelism and brings a major performance boost by making the most out of the computational power available.

Right-click the **Study** node to add a **Cluster Computing** node. A **Cluster Computing** node cannot be used in combination with a **Cluster Sweep**. You will be asked if you want to remove the **Cluster Sweep** before proceeding. Select **Yes**.

The **Settings** window for **Cluster Computing**, shown below, helps to manage the simulation with settings for the cluster or cloud.



The **Settings** menu under **Cluster computing settings** has two options: **Preferences** or **User controlled**. The figure above shows the **User controlled** option. For the **Preferences** option the settings are defined in the **Preferences** dialog box in the **File** menu.

You choose the type of cluster job you want to perform from the **Scheduler type** list. COMSOL Multiphysics and COMSOL Server support General, Windows® HPC Server (HPCS) 2008/2012/2016, Open Grid Scheduler/ Grid Engine (OGS/GE), SLURM, PBS, or Not distributed. The **General** option is a multipurpose option for Linux® clusters. Additional cluster and scheduler settings, such as **Prepend command** and **Postpend command**, are available in the **Preferences** window available from the **File** menu.

To learn more about running in parallel, see the *Reference Manual*.

The **Cluster Sweep** and **Cluster Computing** options are only visible if you have a floating network license and have enabled the **Batch and Cluster** study options.

COMSOL Multiphysics Client-Server

When using COMSOL Multiphysics with a Floating Network License (FNL), it is possible to use a client-server mode of operation to access remote computing resources for solving large models, while still using the graphics card on a local machine to display graphics. (This mode of operation should not be confused with the COMSOL Server product for use with COMSOL applications.)

Using the COMSOL Multiphysics client-server mode can have some significant advantages, since engineering and scientific computing problems involve working with large amounts of data — anywhere from megabytes to gigabytes of data are commonly generated during a simulation. To generate and store this data, you may want to use computers with fast processors, large amounts of Random Access Memory (RAM), and a large hard drive. To visualize large datasets, it is important to have a high-end graphics card on your local computer.

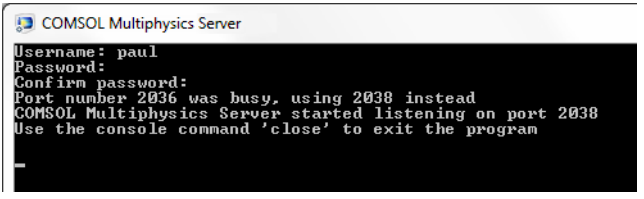
Under ideal conditions, you would always be working on a high-end computer with more than enough memory and processing power for all of the simulation steps. But if this is not the case and you need to solve larger models, you may instead access a shared computing resource across a network.

At any time while using COMSOL Multiphysics, it is possible to connect to a remote computing resource via the client-server mode of operation. This is a two-step process. First, log onto the remote system and invoke the COMSOL Multiphysics server, which will start the COMSOL Multiphysics server process and open a network connection. Second, on the local machine, simply enter the network connection information into an open session of COMSOL Multiphysics. The software then transparently streams the model data and results back and forth over the network and uses the remote computing resource for all computations.

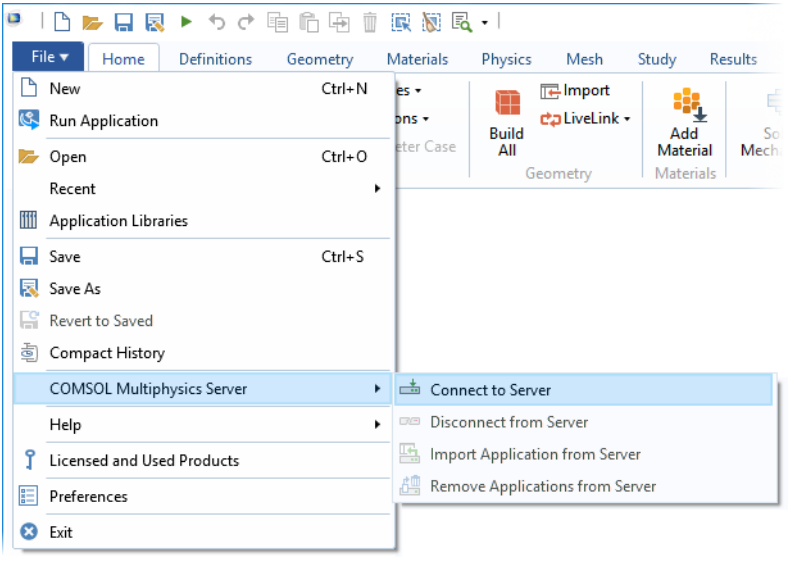
There are several ways to start a COMSOL Multiphysics server session. In Windows® 7, you can, for example, start a server session from the start menu under **COMSOL Multiphysics 5.5 > Client-Server**. In Windows® 10, you can start a server session from the **COMSOL Launchers** option from the Windows start menu. In Linux®, you can use the `comsol mphserver` command. The server and client sessions can be run on different operating systems. For example, the server can run in Linux® and the client session in Windows®.

The first time you start a COMSOL Multiphysics server on a computer, you are asked for a username and password that are associated with the client-server mode

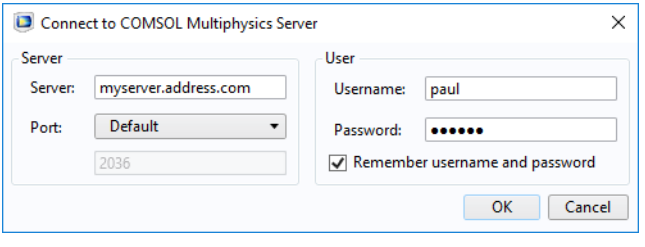
of operation and stored for future connections. The figure below shows the command window for starting a server session in Windows.



The figure below shows how to connect to a server session from the COMSOL Desktop user interface. To connect, from the **File** menu, select **COMSOL Multiphysics Server > Connect to Server**.



You are now prompted for the username and password that you gave the first time you started a server session.



For more information on running the COMSOL Multiphysics client-server, see the *Reference Manual*.

Appendix A — Building a Geometry

This section details how to create the busbar geometry using the built-in geometry tools in COMSOL Multiphysics. The step-by-step instructions take you through the construction of the geometry using the parameters set up in the **Global Definitions** section.

All geometry operations under the geometry node in the model tree are organized in a parametric sequence of operations called a geometry sequence. Using parameterized dimensions helps to produce *what-if* analyses and geometric parametric sweeps.

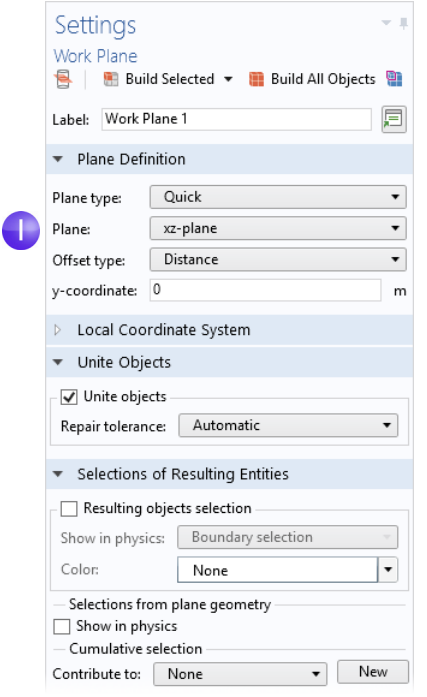
As an alternative to building the geometry using the tools available in COMSOL Multiphysics, you can import a geometry created with a CAD package. The optional CAD Import Module and Design Module support many CAD file formats. Moreover, several add-on products are available that provide bidirectional interfaces to common CAD software packages. See "Appendix E — Connecting with LiveLink™ Add-Ons" on page 202 for a list. Note that the Design Module, in addition to the features available in the CAD Import Module, adds 3D functionality for dimensions, constraints, loft, fillet, chamfer, midsurface, and thicken.

If you have not done so already, start with "Example 2: The Busbar — A Multiphysics Model" on page 59. Follow the steps in the section "Model Wizard" on page 34 to add the physics and study and then follow the steps under **Global Definitions** to add the parameters. Return to this section to learn about geometry modeling. The first step in the geometry sequence is to draw the profile of the busbar.

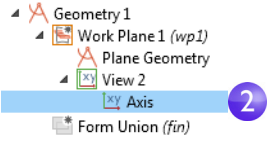
1 Under **Component 1**, right-click **Geometry 1** and select **Work Plane**. In the **Settings** window for **Work Plane**:

- Select **xz-plane** from the **Plane** list (the last option in the list).
- Click the **Show Work Plane** button on the **Settings** window for **Work Plane** toolbar.

Continue by editing the axis and grid settings in **Work Plane 1**.



2 In the **Model Builder**, expand the **View 2** node and click **Axis**.



3 In the **Settings** window for **Axis**:

Under **Axis**:

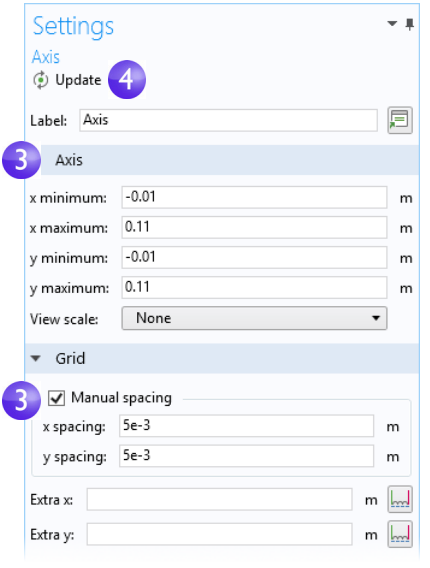
- In the **x minimum** and **y minimum** fields, enter `-0.01`.
- In the **x maximum** and **y maximum** fields, enter `0.11`.

Under **Grid**:

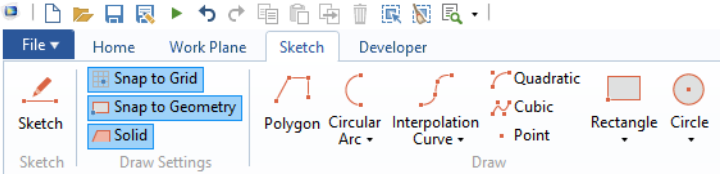
- Select the **Manual Spacing** check box.
- In the **x spacing** and **y spacing** fields, enter `5e-3`.

4 Click the **Update** button on the toolbar.

Note that the values you type will automatically adjust slightly after you click **Update** to adapt to the screen aspect ratio.



You can use interactive drawing to create a geometry using the drawing tools from the **Sketch** tab in the ribbon while pointing and clicking in the **Graphics** window.



You can also right-click the **Plane Geometry** node under **Work Plane 1** to add geometry objects to the geometry sequence. For clarity, this example will demonstrate the non-interactive way of creating the geometry (not using the **Sketch** tab).

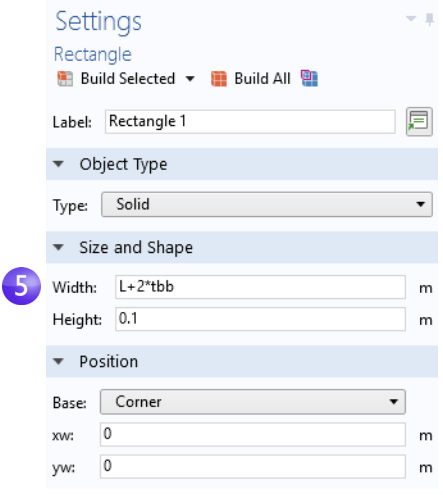
In the next few steps, we create a profile of the busbar.

5 In the **Model Builder** under **Work Plane 1**, right-click **Plane Geometry** and select **Rectangle**.

In the **Settings** window for **Rectangle** under **Size**, enter:

- `L+2*tbb` in the **Width** field.
- `0.1` in the **Height** field.

Click the **Build Selected** button.



6 Create a second rectangle. Under **Work Plane 1**, right-click **Plane Geometry** and select **Rectangle**.

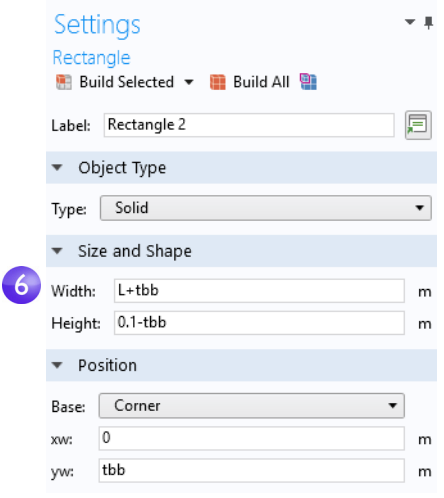
Under **Size**, enter:

- `L+tbb` in the **Width** field
- `0.1-tbb` in the **Height** field.

Under **Position**, enter:

- `tbb` in the **yw** field.

Click the **Build Selected** button.



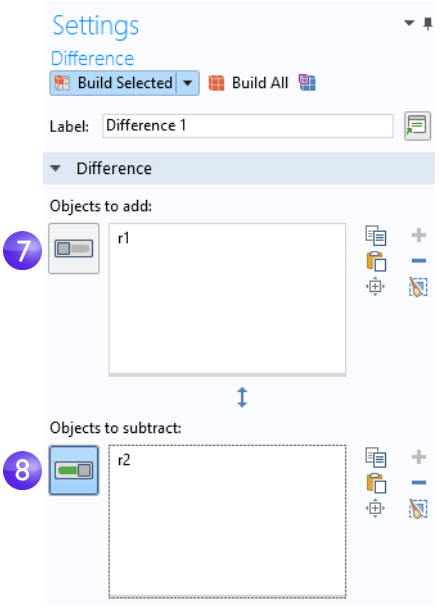
Use the Boolean **Difference** operation to subtract the second rectangle from the first one.

7 Under **Work Plane 1**, right-click **Plane Geometry** and select **Booleans and Partitions > Difference**. In the **Graphics** window, click `r1` (the larger of the two rectangles) to add it to the **Objects to add** list in the **Settings** window for **Difference**.

To help select the geometry, you can display geometry labels in the **Graphics** window. In the **Model Builder** under **Geometry 1 > WorkPlane 1**, click the **View 2** node. Go to the **Settings** window for **View** and select the **Show geometry labels** check box.

8 Click the **Difference** node. In the **Settings** window for **Difference**, click the **Active selection** button to the left of the **Objects to subtract** list. Select the smaller rectangle, `r2`, by using the mouse scroll wheel or arrow keys to cycle through the overlapping rectangles to first highlight it and then click on it to select it.

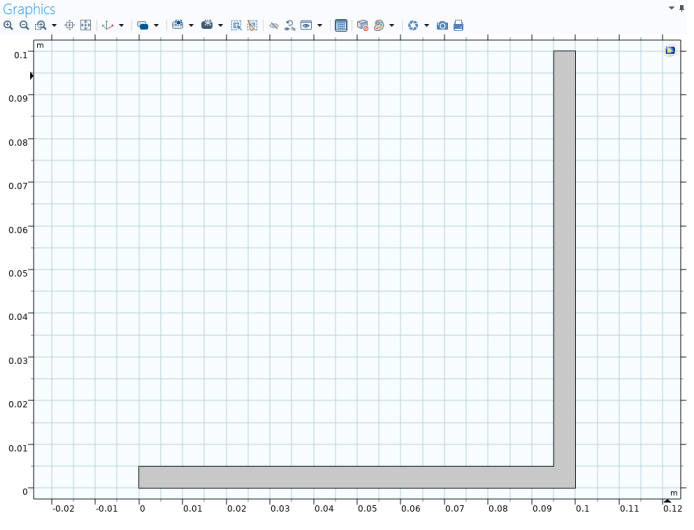
Click **Build Selected** .



Another way to select `r2` in the **Graphics** window is to use the **Selection List** feature. Go to the **Home** tab in the ribbon and select **Windows > Selection List**. In the **Selection List**, click to highlight `r2` (solid). Then, right-click `r2` (solid) in the list and select **Add to Selection** to add it to the **Objects to subtract** list. Right-click on the **Selection List** window title and select **Close**.

After building the selected geometry, you should have a backward-facing, L-shaped profile. Continue by rounding the corners of the L-shaped profile.

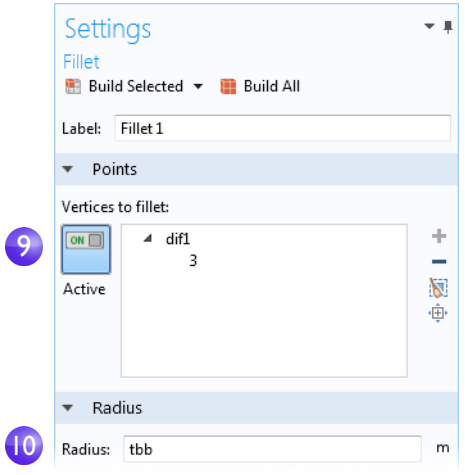
8



9 Under **Work Plane 1**, right-click **Plane Geometry** and select **Fillet**.

Select point 3 (in the inner-right corner) to add it to the **Vertices to fillet** list. There are different ways to add points:

- In the **Graphics** window, click point 3 to add it to the **Vertices to fillet** list.
- From the **Home** tab, select **Windows > Selection List**. In the **Selection List** window, click 3. The corresponding point is highlighted in the **Graphics** window. Click the **Add to Selection** button + in the **Settings** window for **Fillet** or right-click in the **Selection List**.



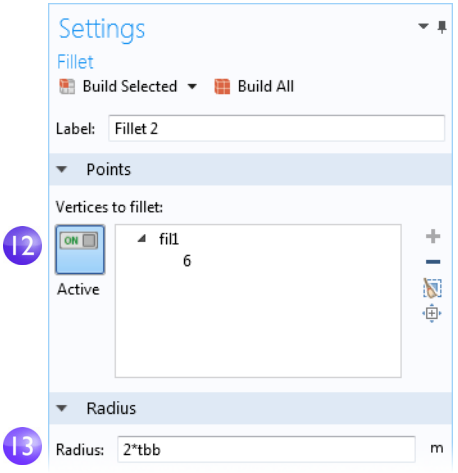
10 Enter `tbb` in the **Radius** field. Click **Build Selected**.

This creates a fillet at the inner corner.

11 For the outer corner, right-click **Plane Geometry** and select **Fillet**.

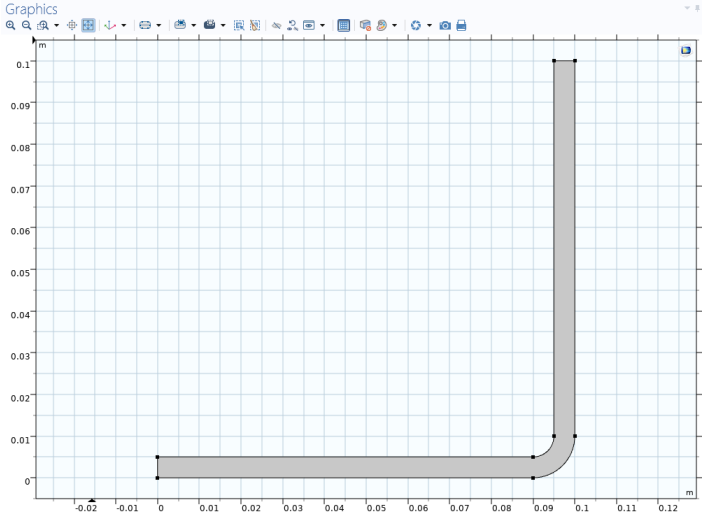
12 In the **Graphics** window, click point 6, the outer corner, to add it to the **Vertices to fillet** list.

13 Enter `2*tbb` in the **Radius** field. Click **Build Selected**.



The result should match this figure:

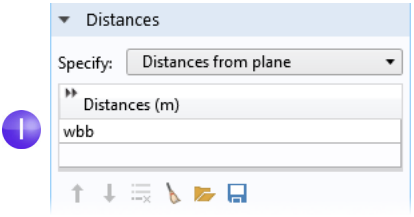
13



In the Design Module you will find tools for interactive drawing and dimensioning of fillets and general 2D sketches.

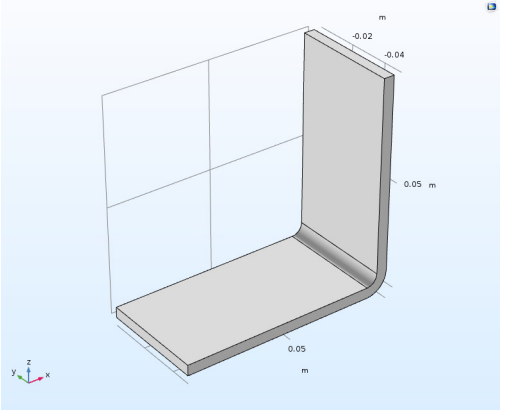
Next, extrude the work plane to create the 3D busbar geometry.

1 In the **Model Builder**, right-click **Work Plane 1** and select **Extrude**. In the Settings window for **Extrude**, enter `wbb` in the **Distances from Plane** table (replace the default) to extrude to the width of the profile.



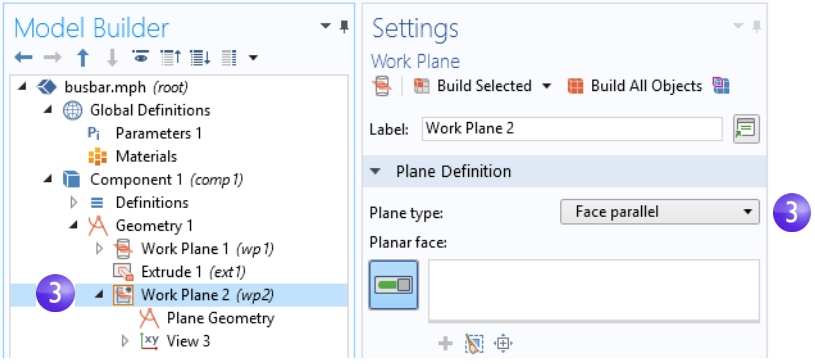
The table allows you to enter several values in order to create sandwich structures with different materials. In this case, only one extruded layer is needed.

2 Click **Build Selected** and then click the **Zoom Extents** button on the **Graphics** toolbar. Click the **Save** button and name the model `busbar.mph` (if you have not already done so).



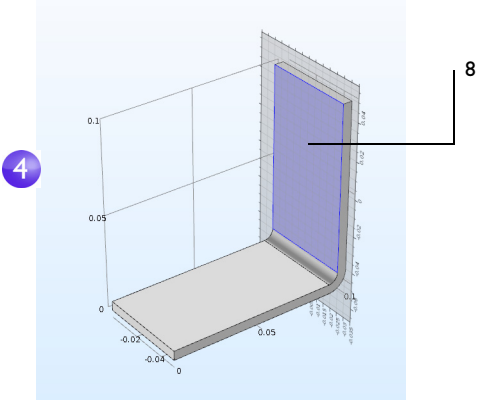
Next, create the titanium bolts by extruding two circles drawn in two work planes.

3 In the **Model Builder**, right-click **Geometry 1** and add a **Work Plane**. A **Work Plane 2** node is added. In the **Settings** window for **Work Plane**, under **Plane Definition**, select **Face parallel** as the **Plane type**.



4 In the **Graphics** window, click face 8, as shown in the figure below, to add it to the **Planar face** list in the Settings window for **Work Plane**.

Face 8 is now highlighted in blue and the work plane is positioned on top of the face.



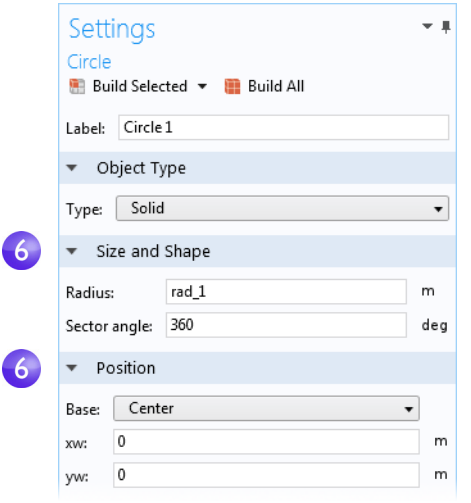
5 Click the **Show Work Plane** button to star drawing the first circle representing the position of the first bolt. Click the **Zoom Extents** button on the **Graphics** toolbar.

6 Under **Work Plane 2**, right-click **Plane Geometry** and select **Circle** .

In the Settings window for **Circle**:

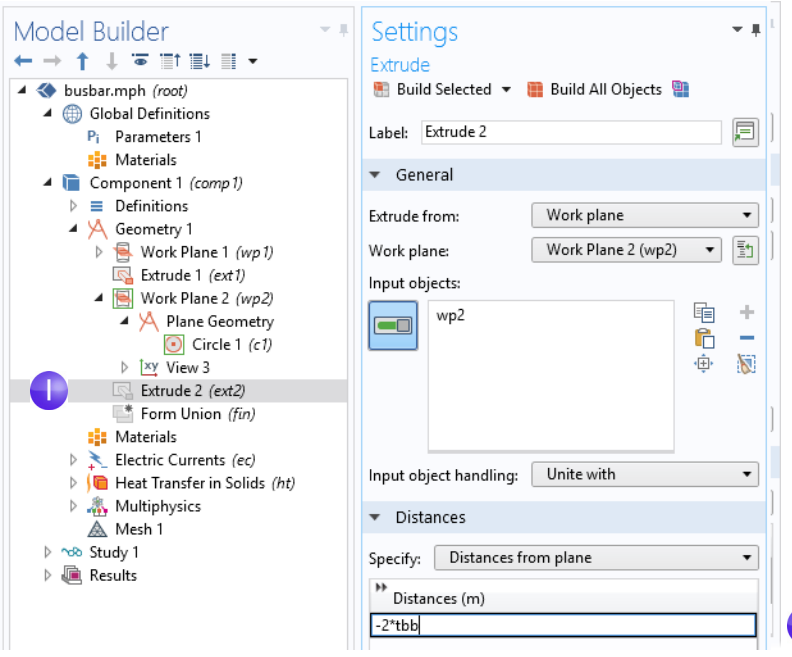
- Under **Size and Shape,** in the **Radius** field, enter `rad_1`.
- Under **Position**, leave the default **xw** and **yw** coordinates (0, 0).

Click **Build Selected** .

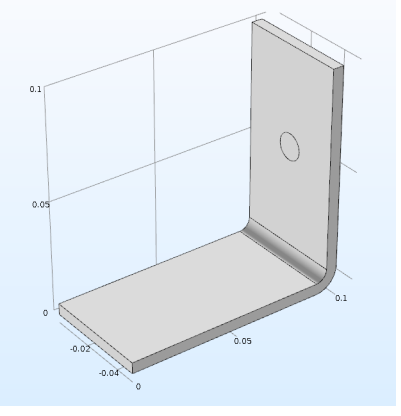


Continue creating the bolt by adding an extrude operation.

1 In the **Model Builder**, right-click **Work Plane 2** and select **Extrude**. In the Settings window for **Extrude**, in the first row of the **Distances from Plane** table, enter `-2*tbb` to extrude the circle.



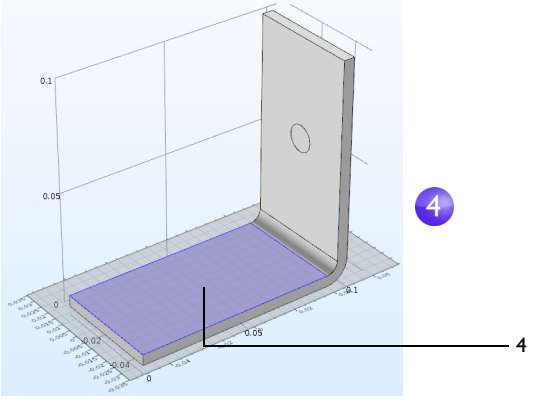
2 Click the **Build Selected** button to create the cylindrical part of the titanium bolt that runs through the busbar.



Draw the two remaining bolts.

3 Right-click **Geometry 1** and select **Work Plane**. A **Work Plane 3** node is added. In the **Settings** window for **Work Plane**, for **Work Plane 3**, select **Face parallel** as the **Plane type**.

4 In the **Graphics** window, click face 4, as shown in the figure, to add it to the **Planar face** list in the **Settings** window for **Work Plane**.



5 Click the **Show Work Plane** button on the **Settings** window for **Work Plane** and the **Zoom Extents** button on the **Graphics** toolbar to get a better view of the geometry.

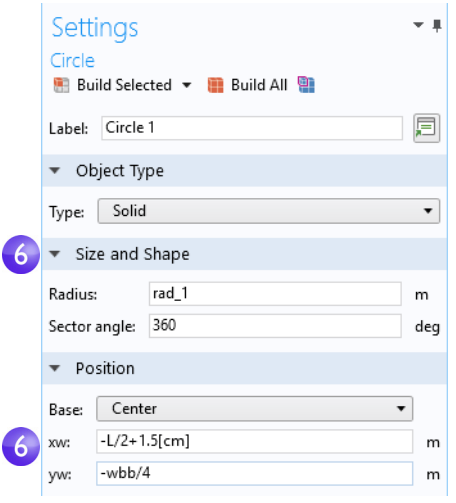
To parameterize the position of the two remaining bolts, add the circles that form the cross sections of the bolts.

6 Under **Work Plane 3**, right-click **Plane Geometry** and select **Circle**.

In the **Settings** window for **Circle**:

- Under **Size and Shape**, enter `rad_1` in the **Radius** field.
- Under **Position**, enter `-L/2+1.5[cm]` in the **xw** field and `-wbb/4` in the **yw** field.

Click **Build Selected**.

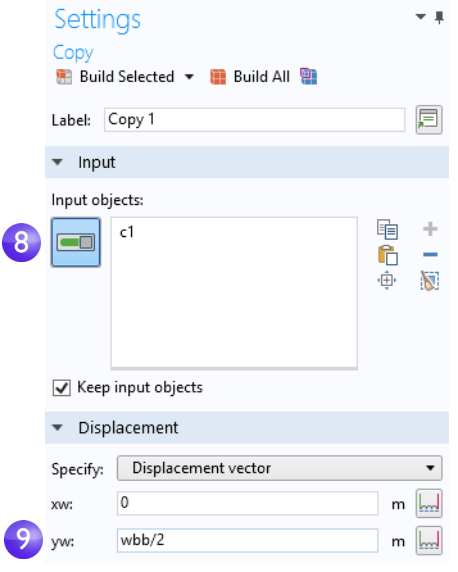


Copy the circle that you just created to generate the third bolt in the busbar.

7 Under **Work Plane 3**, right-click **Plane Geometry** and select **Transforms > Copy**.

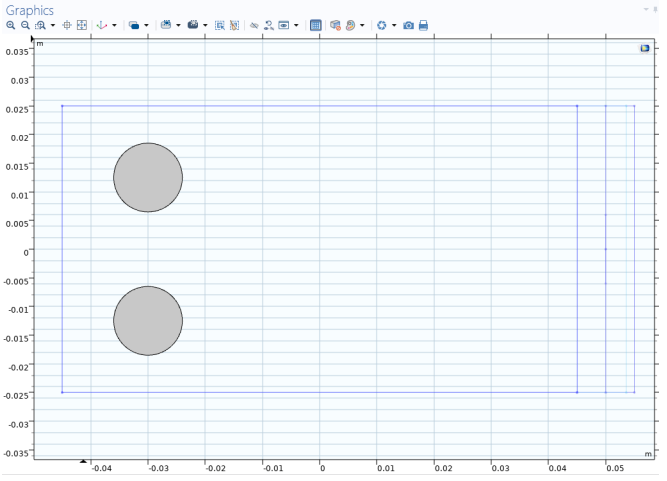
8 In the **Graphics** window, click the circle `c1` to select and add the circle to the **Input objects** list in the **Settings** window for **Copy**.

9 In the **Settings** window for **Copy** under **Displacement**, enter `wbb/2` in the **yw** field.



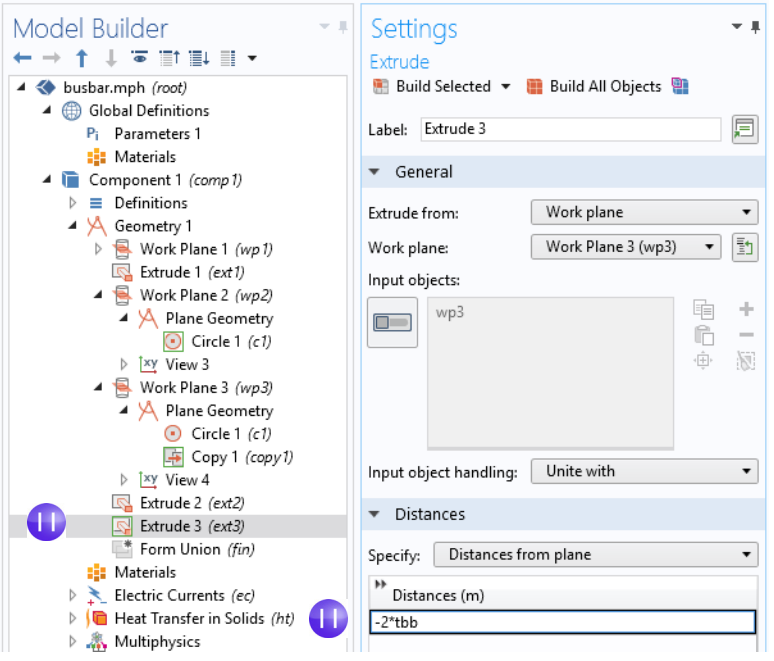
10 Click **Build Selected** and click the **Zoom Extents** button on the **Graphics** toolbar.

Your geometry, as shown in the work plane, should match this figure so far.

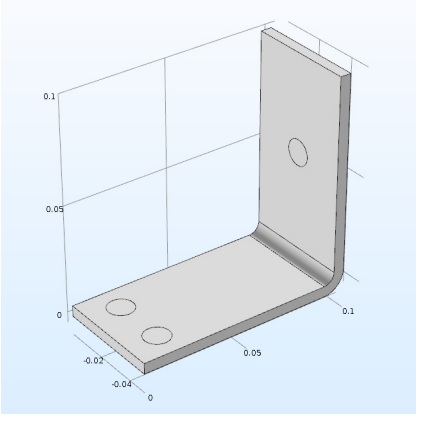


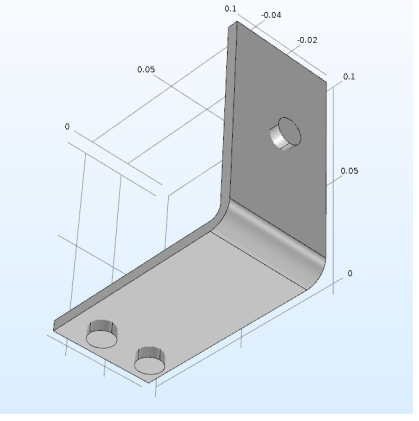
Continue by extruding the circles.

II In the **Model Builder**, right-click **Work Plane 3** and select **Extrude**. In the **Settings** window for **Extrude**, in the first row of the **Distances from Plane** table, enter `-2*tbb` (replace the default). Click **Build All Objects** .



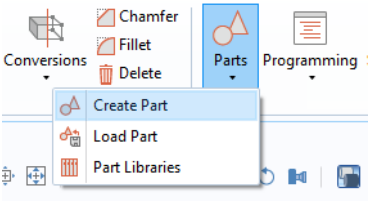
The geometry and geometry sequence should match the figures below. Click the **Save** button and name the model `busbar.mph`.





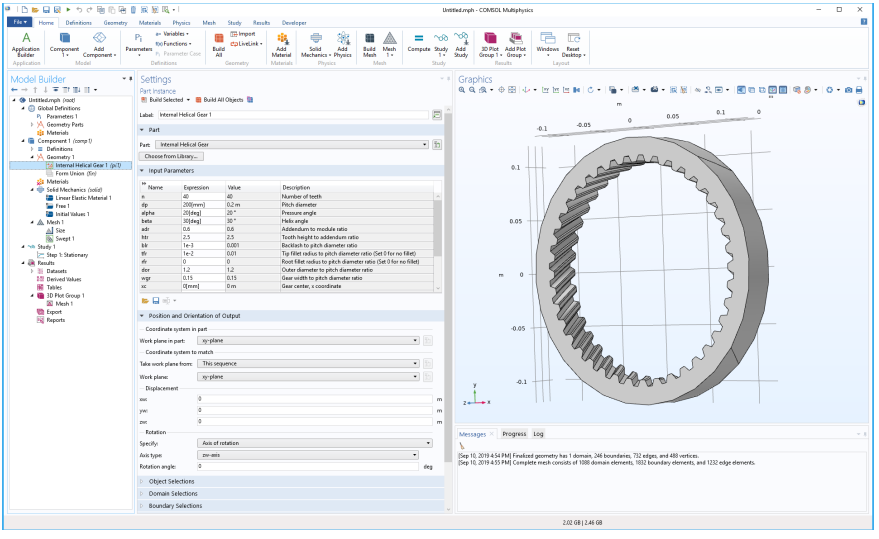
After setting up the busbar or other geometry model, it is convenient to be able to save it for future use, avoiding unnecessary work. In the steps you have just completed, the geometry was stored directly in the COMSOL model file that will also be used to set up the complete busbar model. Instead, you can also create a reusable and parameterized part that can serve as a building block of a more complex COMSOL model geometry, which is stored in a separate file accessible through the Part Libraries.

While building the busbar geometry, you were using features from the **Geometry** and **Workplane** tabs. A **Parts** menu is located in the **Other** group in these tabs.



Through the **Parts** menu, you can choose to create or load a part, or add a part to the model geometry by selecting one from the Part Libraries. Several Part Libraries are already provided with the software by default. When new parts are created, they are added under a part's parent node in the **Global Definitions** section of the model tree.

The figure below shows an **Internal Helical Gear** part from the part library of the Multibody Dynamics Module. The part comes with 18 **Input Parameters** that control the dimensions.



For additional information on working with parts and the Part Libraries, see the *Reference Manual*.

To continue with the busbar tutorial, return to the section “Materials” on page 73.

Appendix B — Keyboard and Mouse Shortcuts

These shortcuts are primarily applicable to the Model Builder and the Graphics window. For information on shortcuts in the Application Builder, see the *Introduction to Application Builder*.

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACOS)	ACTION
F1 (Fn+F1 on certain tablets)	F1	Display help for the selected node or window
Ctrl+F1	Command+F1	Open the COMSOL Documentation front page in an external window
F2	F2	Rename the selected node, file, or folder
F3	F3	Disable selected nodes
F4	F4	Enable selected nodes
F5	F5	Update the Datasets Solutions with respect to any new Global Definitions and Definitions without re-solving the model
F6	F6	Build the preceding node in the geometry branch or plot the previous plot for a time-dependent, eigenfrequency, or eigenvalue solution.
F7	F7	Build the selected node in the geometry and mesh branches, compute the selected study step, compute the selected node in the solver sequence, or **plot the next plot for a time-dependent, eigenfrequency, or eigenvalue solution.**
F8	F8	Build the geometry, build the mesh, compute the entire solver sequence, update results data, update the plot, or run a method call.
Del	Del	Delete selected nodes or objects
Left arrow (Windows); Shift + Left arrow (Linux)	Left arrow	Collapse a branch in the model tree

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACOS)	ACTION
Right arrow (Windows); Shift + Right arrow (Linux)	Right arrow	Expand a branch in the model tree
Up arrow	Up arrow	Move to the node above in the model tree
Down arrow	Down arrow	Move to the node below in the model tree
Alt+left arrow	Ctrl+left arrow	Move to the previously selected node in the model tree
Alt+Right arrow	Ctrl+Right arrow	Move to the next selected node in the model tree
Ctrl+A	Command+A	Select all domains, boundaries, edges, or points; Select all cells in a table
Ctrl+C	Command+C	Copy text or objects
Ctrl+D	Command+D	Clear the selection of domains, boundaries, edges, or points
Ctrl+Shift+D	Command+Shift+D	Duplicate the selected node in the model tree
Ctrl+F	Command+F	Find a search string
Ctrl+G	Command+G	Group nodes.
Ctrl+Shift+G	Command+Shift+G	Ungroup nodes.
Ctrl+K		Create shortcut for use in the Application Builder
Ctrl+L	Command+L	Take a quick image snapshot.
Ctrl+N	Command+N	New model
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 or object
Ctrl+X	Command+X	Cut copied text or object
Ctrl+Y	Ctrl+Shift+Z	Redo the last undone operation
Ctrl+Z	Command+Z	Undo the last operation

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACOS)	ACTION
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+Shift+A	Not available	Go to Application Builder
Ctrl+Shift+M	Not available	Go to Model Builder
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+Dow n arrow	Switch focus to the next section in the Settings window.
Ctrl+F4	Command+W	Close the active window in COMSOL Desktop, if it is closable.
Shift+F10 or (Windows only) Menu key	Ctrl+F10	Open the context menu.
Ctrl+Space or Ctrl+/	Ctrl+Space	Open list of predefined quantities for insertion in Expression fields.
Left-click and hold down the mouse button while dragging the mouse.	Same as for Windows; only available for two-button mouse.	Rotate the scene around the axes parallel to the screen's x- and y-axes with the origin at the scene rotation point.
Right-click and hold down the mouse button while dragging the mouse.	Same as for Windows; only available for two-button mouse.	Move the visible frame on the image plane in any direction (pan).

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACOS)	ACTION
Middle-click outside model and hold down the mouse button while dragging the mouse.	Same as for Windows; only available for two-button mouse.	The scene is zoomed in/out around the mouse position where the action started.
Press Ctrl and left-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Tilt and pan the camera by rotating about the x- and y-axes in the image plane.
Press Ctrl and right-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Move the camera in the plane parallel to the image plane.
Press Ctrl and middle-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Move the camera into and away from the object (dolly in/out).
Press Ctrl and middle-click.	Same as for Windows; only available for two-button mouse.	Zoom to the area around the point on the model determined by the mouse position.
Press Ctrl+Alt and left-click. While holding down the keys and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Rotate the objects in the scene about the camera look-at direction
Press Alt and left-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Rotate the objects in the scene about the axis that intersects the camera origin and the scene's rotation center (roll direction)
Press Alt and right-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Move the scene in the plane orthogonal to the axis between the camera and the scene rotation point.
Press Alt and middle-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Move the camera along the axis between the camera and the scene rotation point.
Middle-click on model.	Same as for Windows; only available for two-button mouse.	Set the center of rotation.
Press X, Y, or Z and left-click. While holding down the key and button, drag the mouse.	Same as for Windows; only available for two-button mouse.	Rotate around the X, Y, or Z-axis, respectively.

SHORTCUT (WINDOWS, LINUX)	SHORTCUT (MACOS)	ACTION
R	R	Toggle between manual and automatic rotation center.
+ or - keyboard button alt. Up arrow or Down arrow button	+ or - keyboard button alt. Up arrow or Down arrow button	Cycle between geometric entities (domains, boundaries, edges, points) for selection. When focus is in the graphics window.

Appendix C — Language Elements and Reserved Names

Building a model tree in COMSOL Multiphysics is equivalent to graphically programming a sequence of operations. Using the Record Code functionality in the Application Builder or saving as a model file for MATLAB® or Java® outputs the sequence of operations as a list of traditional programming statements. In this section, we will give an overview of the following element categories as defined by the underlying COMSOL language:

- Constants
- Variables
- Functions
- Operators
- Expressions

These language elements are built-in or user-defined. Operators cannot be user-defined. Expressions 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 will turn orange and you will 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. In the following pages, we list the most commonly used built-in elements and hence those reserved names. A more exhaustive list of built-in elements can be found in the *Programming Reference Manual* and the *Application Programming Guide*.

Variables used in Applications

Model parameters and variables can be used in applications. For example, you can let the user of an application change the value of a parameter. In addition, variables to be used in applications can be defined in the Application Builder under the **Declarations** node. Such variables are available globally in form objects and methods but cannot be used in the Model Builder.

Constants and Parameters

There are three different types of constants accessible from the Model Builder: built-in mathematical and numerical constants, built-in physical constants, and parameters. Parameters are user-defined constants, which 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 Mathematical and Numerical Constants

DESCRIPTION	NAME	VALUE
Floating point relative accuracy for double floating point numbers, also known as machine epsilon	`eps`	2^{-52} ($\sim 2.2204 \cdot 10^{-16}$)
The imaginary unit	`i`, `j`	`i, sqrt(-1)`
Infinity, ∞	`inf`, `Inf`	A value larger than what can be handled with floating point representation
Not-a-number	`NaN`, `nan`	An undefined or unrepresentable value, such as the result of `0/0` or `inf/inf`
π	`pi`	`3.141592653589793`

Built-in Physical Constants

DESCRIPTION	NAME	VALUE
Acceleration of gravity	`g_const`	`9.80665[m/s^2]`
Avogadro constant	`N_A_const`	`6.02214076e23[1/mol]`
Boltzmann constant	`k_B_const`	`1.380649e-23[J/K]`
Characteristic impedance of vacuum (impedance of free space)	`Z0_const`	`376.73031346177066[ohm]`
Electron mass	`me_const`	`9.10938291e-31[kg]`
Elementary charge	`e_const`	`1.602176634e-19[C]`
Faraday constant	`F_const`	`96485.3365[C/mol]`
Fine-structure constant	`alpha_const`	`7.2973525698e-3`
Gravitational constant	`G_const`	`6.67384e-11[m^3/(kg*s^2)]`
Molar volume of ideal gas (at 273.15 K and 1 atm)	`V_m_const`	`2.2413968e-2[m^3/mol]`
Neutron mass	`mn_const`	`1.674927351e-27[kg]`
Permeability of vacuum (magnetic constant)	`mu0_const`	`2*alpha_const*h_const/c_const/e_const/e_const`
Permittivity of vacuum (electric constant)	`epsilon0_const`	`1/mu0_const/c_const/c_const`
Planck's constant	`h_const`	`6.62607015e-34[J*s]`
Planck's constant over 2 pi	`hbar_const`	`1.05457172533629e-34[J*s]`
Proton mass	`mp_const`	`1.672621777e-27[kg]`
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]`

Parameters

Parameters are user-defined constant scalars in the **Global Definitions** branch in the model tree. Example uses are:

- Parameterizing geometric dimensions
- Parameterizing mesh element sizes

- Defining parameters to be used in parametric sweeps

A parameter can be defined as an expression in terms of numbers, parameters, built-in constants, and built-in functions of parameters and built-in constants. Parameters should be assigned a unit, using `[]`, unless they are dimensionless.

Variables

There are two types of variables used in the Model Builder: built-in and user-defined. Variables can be scalars or fields. Variables can have units.

Note: Spatial coordinate variables and dependent variables are user-defined variables of special interest. These variables have default names based on the space dimension of the geometry and the physics interface, respectively. As a result of the names chosen for these variables, a list of built-in variables will be created by COMSOL Multiphysics: the first- and second-order derivatives with respect to space and time.

BUILT-IN VARIABLES

NAME	DESCRIPTION	TYPE
`curv`	The curvature of a boundary in 2D	Field
`curv1,curv2`	The principal curvature components of a boundary in 3D	Field
`dom`	The domain number for a domain, boundary, edge, or point	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
`freq`	Frequency	Global variable
`geomapproxdist`	For each element, this variable indicates how far a node point in each element was moved from the geometry.	Field
`h`	Mesh element size (length of the longest edge of the element)	Field
`lambda`	Eigenvalue	Global variable
`linearizedelem`	Elements forced to be linear	Field
`numberofdofs`	Number of degrees of freedom	Global variable
`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) element	Field
`nx,ny,nz`	Normal component for boundaries	Field
`phase`	Phase angle	Global variable
`qual`	Mesh quality	Field
`realdetjac`	The determinant of the Jacobian matrix for the mapping from a 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.	Field

realdetjacmin	A scalar field variable for each element defined as the minimum value of the reldetjac variable for the corresponding element. A reldetjacmin value less than zero for an element means that the element is wrapped inside-out; that is, the element is an inverted mesh element.	Field
s,s1,s2	Curve and surface parameterization variables	Field
t	Time	Global variable
tcurvx,tcurvy,tcurv1x, curv1y,tcurv1z,tcurv2x, tcurv2y,tcurv2z	Tangential directions for the corresponding curvature variables	Field
tx,ty,t1x,t1y,t1z, t2x,t2y,t2z	Tangent vector components corresponding to the curve and surface parameterizations	Field
qual	A mesh quality measure between 0 (poor quality) and 1 (perfect quality)	Field

User-defined Variables That Generate Built-in Variables

DEFAULT NAME	DESCRIPTION	TYPE
x, y, z	Spatial coordinates (Cartesian)	Field
r, phi, z	Spatial coordinates (Cylindrical)	Field
u, T, etc.	Dependent variables (Solution)	Field

Example: T is the name for the temperature in a 2D, time-dependent heat transfer model; x and y are the spatial coordinate names. In this case, the following built-in variables will be generated: T, Tx, Ty, Txx, Txy, Tyx, Tyy, Tt, Txt, Tyt, Txxt, Txyt, Tyxt, Tyyt, Ttt, Txtt, Tytt, Txxtt, Txytt, Tyxtt, and Tyytt. Here, Tx corresponds to the partial derivative of the temperature T with respect to x and Ttt corresponds to the second-order time derivative of T, and so on. If the spatial coordinate variables have other names — for example, psi and chi — then Txy would be Tpsichi, and Txt would be Tpsit. (The time variable t is built-in; the user cannot change its name.)

Functions

In the Model Builder, there are two types of functions: built-in and user-defined. Functions can be scalar valued or field valued depending on the input argument(s). Some functions can have units for both input and output arguments.

Built-in Mathematical Functions

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

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)`
`binomial`	Binomial coefficients	`binomial(n,k)`
`ceil`	Nearest following integer	`ceil(x)`
`conj`	Complex conjugate	`conj(x)`
`cos`	Cosine	`cos(x)`
`cosh`	Hyperbolic cosine	`cosh(x)`
`cot`	Cotangent	`cot(x)`

NAME	DESCRIPTION	SYNTAX EXAMPLE
`coth`	Hyperbolic cotangent	`coth(x)`
`csc`	Cosecant	`csc(x)`
`csch`	Hyperbolic cosecant	`csch(x)`
`erf`	Error function	`erf(x)`
`exp`	Exponential	`exp(x)`
`factorial`	Factorial of nonnegative integer	`factorial(n)`
`floor`	Nearest previous integer	`floor(x)`
`gamma`	Gamma function	`gamma(x)`
`imag`	Imaginary part	`imag(u)`
`legendre`	Legendre polynomials	`legendre(l,x),`
`legendre`	Associated Legendre functions	`legendre(l,m,x)`
`log`	Natural logarithm	`log(x)`
`log10`	Base-10 logarithm	`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)`
`sphericaly`	Spherical harmonic function	`sphericaly(l,m,theta,phi)`
`sphericalyr`	Real spherical harmonic function	`sphericalyr(l,m,theta,phi)`
`tan`	Tangent	`tan(x)`
`tanh`	Hyperbolic tangent	`tanh(x)`
`zernike`	Zernike polynomials	`zernike(n,m,r,phi)`

Built-in Operator Functions

These built-in functions behave differently than the built-in mathematical functions. They may not belong in an introductory text but are listed to complete the list of reserved names. For more information, see the *Reference Manual*.

NAME	NAME	NAME	NAME
adj	dtang	mean	timeavg
at	emetric	noenv	timeint
atlocal	env	noxd	timemax
attimemax	error	pd	timemin
attimemin	fsens	ppr	treatasconst
atxd	fsensimag	pprint	try_catch
atonly	if	prev	uflux
ballavg	integrate	reacf	up
ballint	isdefined	realdot	var
bdf	isinf	residual	with
bndenv	islinear	*scope*.atent*i*	withsol
centroid	isnan	*scope*.at*i*	
circavg	jacdepends	sens	
circint	lindev	setconst	
circumcenter	linper	setind	
d	linpoint	setval	
depends	linsol	shapeorder	
dest	lintotal	side	
dflux	lintotalavg	sphavg	
diskavg	lintotalpeak	sphint	
diskint	lintotalrms	subst	
down	linzero	sum	
dtang	nojac	test	

User-Defined Functions

A user-defined function can be defined in the model tree under the **Global Definitions** node or, for each **Component**, under the **Definitions** node. Select a template from the **Functions** menu and enter settings to define the name and detailed shape of the function.

TEMPLATE NAME	ARGUMENTS AND DEFINITION	SYNTAX EXAMPLE
Analytic	The function name is its identifier; for example, `an1`. The function is a mathematical expression of its arguments. Example: Given the arguments `x` and `y`, its definition is `sin(x)*cos(y)`. The function has an arbitrary number of arguments.	The name of the function with comma-separated arguments within parentheses. For example: `an1(x,y)`
Elevation	The function name is its identifier; for example, `elev1`. Used to import geospatial elevation data from digital elevation models 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.	The name of the function with comma-separated arguments within parentheses. For example: `elev1(x,y)`
Gaussian Pulse	The function name is its identifier; for example, `gp1`. The Gaussian pulse function defines a bell-shaped curve according to the expression $y(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-x_0)^2}{2\sigma^2}}$ It is defined by the mean parameter, x_0, and the standard deviation, σ. The function has one argument.	The name of the function with a single argument within parentheses. For example: `gp1(x)`

TEMPLATE NAME	ARGUMENTS AND DEFINITION	SYNTAX EXAMPLE
Image	The function name is its identifier; for example, `im1`. Used to import an image (in BMP, JPEG, PNG, or GIF format) and map the image's RGB data to a scalar (single channel) function output value. By default the function's output uses the mapping (R+G+B)/3.	The name of the function with comma-separated arguments within parentheses. For example: `im1(x,y)`
Interpolation	The function name is its identifier; for example, `int1`. An interpolation function is defined by a table or file containing the values of the function in discrete points. The file formats are the following: spreadsheet, grid, or sectionwise. The function has one to three arguments.	The name of the function with comma-separated arguments within parentheses. For example: `int1(x,y,z)`
Piecewise	The function name is its identifier; for example, `pw1`. 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. This function has one argument with different definitions on different intervals, which must not overlap or have any holes between them.	The name of the function with a single argument within parentheses. For example: `pw1(x)`
Ramp	The function name is its identifier; for example, `rm1`. A ramp function is a linear increase with a user-defined slope that begins at some specified time. The function has one argument. It can also be smoothed.	The name of the function with a single argument within parentheses. For example: `rm1(x)`

TEMPLATE NAME	ARGUMENTS AND DEFINITION	SYNTAX EXAMPLE
Random	The function name is its identifier; for example, `rn1`. A random function generates white noise with uniform or normal distribution and has one or more arguments to simulate white noise. The function has arbitrary number of arguments.	The name of the function with comma-separated arguments within parentheses. For example: `rn1(x,y)` The arguments x and y are used as random seeds for the random function.
Rectangle	The function name is its identifier; for example, `rect1`. A rectangle function is `1` in an interval and `0` everywhere else. The function has one argument.	The name of the function with a single argument within parentheses. For example: `rect1(x)`
Step	The function name is its identifier; for example, `step1`. A step function is a sharp transition from `0` to some other value (amplitude) at some location. The function has one argument. It can also be smoothed.	The name of the function with a single argument within parentheses. For example: `step1(x)`
Switch	The function name is its identifier; for example, `sw1`. A switch is used to switch between global functions during a solver sweep. You add the functions as subnodes under the Switch node. The switch for functions acts essentially as a switch statement in a programming language; that is, it dynamically chooses one of its underlying branches depending on a parameter that can be controlled from the solvers using a Function Sweep study. You can also add a Switch node under Materials and then use a Material Sweep to sweep over a discrete set of materials.	Switch is a container node for functions.

TEMPLATE NAME	ARGUMENTS AND DEFINITION	SYNTAX EXAMPLE
Triangle	The function name is its identifier; for example, `tri1`. A triangle function is a linear increase and linear decline within an interval and `0` everywhere else. The function has one argument. It can also be smoothed.	The name of the function with a single argument within parentheses. For example: `tri1(x)`
Waveform	The function name is its identifier; for example, `wv1`. A waveform function is a periodic function with one of several characteristic shapes: sawtooth, sine, square, or triangle. The function has one argument. It can also be smoothed.	The name of the function with a single argument within parentheses. For example: `wv1(x)`
External (Global Definitions only)	An external function defines an interface to one or more functions written in the C language (which can be a wrapper function interfacing source code written in, for example, Fortran). Such an external function can be used, for example, to interface a user-created shared library. Note that the extension of a shared library file depends on the platform: .dll (Windows), .so (Linux), or .dylib (macOS).	The name of the function and the appropriate number of arguments within parentheses. For example: `myextfunc(a,b)`
MATLAB® (Global Definitions only)	A MATLAB® function interfaces one or more functions written in the MATLAB® language. Such functions can be used as any other function defined in COMSOL Multiphysics provided LiveLink™ for MATLAB® and MATLAB® are installed. (MATLAB® functions are evaluated by MATLAB® at runtime.)	The name of the function and the appropriate number of arguments within parentheses. For example: `mymatlabfunc(a,b)`

Unary and Binary Operators

In the Model Builder, the following table contains the unary and binary operators that can be used in expressions.

PRECEDENCE LEVEL	SYMBOL	DESCRIPTION
1	`() {} .`	Grouping, Lists, Scope
2	`^`	Power
3	`! - +`	Unary: Logical NOT, Minus, Plus
4	`[]`	Unit
5	`* /`	Multiplication, Division
6	`+ -`	Binary: Addition, Subtraction
7	`< <= > >=`	Comparisons: Less-Than, Less-Than or Equal, More-Than, More-Than or Equal
8	`== !=`	Comparisons: Equal, Not Equal
9	`&&`	Logical AND
10	`||`	Logical OR
11	`,`	Element Separator in Lists

Expressions

Parameters

A parameter expression can contain: numbers, parameters, built-in constants, built-in functions of parameter expressions, and unary and binary operators. Parameters can have units.

Variables

A variable expression can contain: numbers, parameters, constants, variables, functions of variable expressions, and unary and binary operators. Variables can have units.

Functions

A function definition can contain: input arguments, numbers, parameters, constants, functions of parameter expressions, including input arguments, unary and binary operators.

COMSOL File Formats

The COMSOL Model MPH file type, with the extension `.mph`, is the default file type containing the entire model tree and application tree from the Model Builder and Application Builder, respectively. The file contains both binary and text data. The mesh and solution data is stored as binary data, while all other information is stored as plain text.

The COMSOL binary and text file types, with the extension `.mphbin` and `.mphtxt`, respectively, contain either geometry objects or mesh objects that can be imported directly to the **Geometry** or **Mesh** nodes in the model tree. Note that if one of the products CAD Import Module, Design Module, or one of the LiveLink™ add-on products for CAD was used to create a geometry model then the same product may be required to open the corresponding file. The reason is that the geometry model may require features that are only available in one of the add-on products.

The Physics Builder file type, with the extension `.mphphb`, contains one or more user-defined physics interfaces that you can access from the Model Builder. See the *Physics Builder Manual* for more information.

See Supported External File Formats for more information about all the other formats supported by COMSOL Multiphysics.

FILE TYPE	EXTENSION	READ	WRITE
COMSOL Model and Application	`.mph`	Yes	Yes
Binary Data	`.mphbin`	Yes	Yes
Physics Builder	`.mphphb`	Yes	Yes
Text Data	`.mphtxt`	Yes	Yes

Supported External File Formats

CAD

The CAD Import Module and Design Module allow for import of a range of industry-standard CAD file types. Additional file types are available through the

bidirectional functionality of the LiveLink products for CAD as well as with the File Import for CATIA® V5 add-on.

The DXF (2D), VRML (3D), STL (3D), PLY (3D), and 3MF (3D) file types are available for import with COMSOL Multiphysics and do not require any add-on products. Unless noted otherwise in the table below, import of the listed file types is available on all supported versions of Linux®, macOS, and Windows® operating systems.

FILE TYPE	EXTENSION	READ	WRITE
AutoCAD®[1,2,3]	.dwg	Yes	Yes[9]
Inventor®[1,2,3]	.ipt,.iam	Yes	Yes[9]
NX®[1,4]	.prt	Yes	No
PTC® Creo® Parametric™[1,3]	.prt,.asm	Yes	Yes[9]
PTC® Pro/ENGINEER®[1,3]	.prt,.asm	Yes	Yes[9]
Revit® Architecture[3,5]	.rvt	Yes[9]	Yes[9]
Solid Edge®[3,6]	.par,.asm	Yes[9]	Yes[9]
SOLIDWORKS®[1,2,3]	.sldprt,.sldasm	Yes	Yes[9]
DXF (3D[1,2] and 2D)	.dxf	Yes	Yes[10]
Parasolid®[1]	.x_t,.xmt_txt,.x_b,.xmt_bin	Yes	Yes
ACIS®[1]	.sat,.sab,.asat,.asab	Yes	Yes
STEP[1]	.step,.stp	Yes	Yes
IGES[1]	.iges,.igs	Yes	Yes
CATIA® V5[2,7]	.CATPart,.CATProduct	Yes	No
VRML, v1[8]	.vrml,.wrl	Yes	No
STL	.stl	Yes	Yes
3MF	.3mf	Yes	No

FILE TYPE	EXTENSION	READ	WRITE
PLY	`.ply`	Yes	No

[1]Requires one of the LiveLink™ products for AutoCAD®, Revit®, PTC® Creo® Parametric™, Inventor®, PTC® Pro/ENGINEER®, Solid Edge®, SOLIDWORKS®, CAD Import Module, or Design Module

[2]Import supported on Windows® operating systems only

[3]Synchronization of files between COMSOL Multiphysics and a linked CAD program supported on Windows® 7, 8, 8.1 and 10 operating systems only

[4]Import supported on Windows® and Linux operating systems only

[5]Requires LiveLink™ *for* Revit®

[6]Requires LiveLink™ *for* Solid Edge®

[7]Requires the CAD Import Module (or the Design Module or one of the LiveLink™ products for AutoCAD®, PTC® Creo® Parametric™, Inventor®, PTC Pro/ENGINEER®, Solid Edge®, or SOLIDWORKS®) and the File Import for CATIA® V5

[8]Limited to a single geometric domain

[9]From/To file using linked CAD package, provided the original CAD geometry is created in the CAD package

[10]Writing to file is supported for 2D geometry only

ECAD

The ECAD Import Module allows for import of 2D layout files with automatic conversion to 3D CAD models. The Touchstone file type is used for exporting S-parameters, impedance, and admittance values from simultaneous port and frequency sweeps. The SPICE Circuit Netlist file type is converted at import to a series of lumped circuit element nodes under an **Electrical Circuit** node.

FILE TYPE	EXTENSION	READ	WRITE
IPC-2581[1]	`.cvg,.xml`	Yes	No
ODB++[1]	`.zip,.tar,.tgz,.tar.gz`	Yes	No
GDS[1]	`.gds`	Yes	No
Touchstone[2]	`.s2p,.s3p,.s4p,...`	No	Yes
SPICE Circuit Netlist[3]	`.cir`	Yes	Yes

[1]Requires the ECAD Import Module

[2]Requires one of the AC/DC Module, RF Module, MEMS Module, or Wave Optics Module

[3]Requires one of the AC/DC Module, RF Module, MEMS Module, Plasma Module, or Semiconductor Module

Material Databases

The Chemical Reaction Engineering Module and the Plasma Module can read CHEMKIN® files to simulate complex chemical reactions in the gas phase. The Plasma Module can read LXCAT files for sets of electron impact collision cross sections.

FILE TYPE	EXTENSION	READ	WRITE
CHEMKIN®[1]	`.dat,.txt,.inp`[3]	Yes	No
CAPE-OPEN[1] (direct connection)	`n/a`	n/a	n/a
LXCAT file[2]	`.lxcat,.txt`	Yes	No

[1]Requires the Chemical Reaction Engineering Module or the Plasma Module. The Plasma Module only supports thermodynamic and transport data.

[2]Requires the Plasma Module

[3]Any extension is allowed; These are the most common extensions

Mesh

The NASTRAN® Bulk Data file types are used to import a volumetric mesh. The VRML, STL, PLY, and 3MF file types are used to import a triangular surface mesh that can serve as a starting point for creating a volumetric mesh.

FILE TYPE	EXTENSION	READ	WRITE
3MF	`.3mf`	Yes	Yes
NASTRAN® Bulk Data	`.nas,.bdf,.nastran,.dat`	Yes	Yes
PLY	`.ply`	Yes	Yes
Sectionwise (COMSOL)	`.txt,.csv,.dat`	Yes	Yes
STL	`.stl`	Yes	Yes
VRML, v1	`.vrml,.wrl`	Yes	No
VTK	`.vtu`	No	Yes

Images and Movies

Results visualization can be exported to a number of common image file types; See the table below. Images can also be read and used as interpolation functions for

physics modeling. Animations can be exported to one of the Animated GIF, Adobe® Flash®, and AVI file types.

FILE TYPE	EXTENSION	READ	WRITE
Adobe® Flash®	`.swf`	No	Yes
Animated GIF	`.gif`	No	Yes
AVI[1]	`.avi`	No	Yes
BMP	`.bmp`	Yes	Yes
EPS (1D graphs only)	`.eps`	No	Yes
GIF	`.gif`	Yes	Yes
glTF™	`.glb`	No	Yes
JPEG	`.jpg,.jpeg`	Yes	Yes
MP4[2]	`.mp4`	Yes	No
OGV[2]	`.ogv`	Yes	No
PNG	`.png`	Yes	Yes
TIFF	`.tif,.tiff`	No	Yes
WEBM	`.webm`	Yes	No

[1] Available for Windows® only

[2] Available in the Application Builder, for Windows® only, in the video form object

Programming Languages and Spreadsheet

Model files for Java® are editable files with the extension `.java` that contain sequences of COMSOL commands as Java® code. Edit the files in a text editor to add additional commands. You can compile these Java® files into Java® Class files with the extension `.class` and run them as separate applications. Program code used in model files for Java® is essentially identical to that of the Method editor in the Application Builder. However, the Method editor comes with an integrated Java® compiler.

Model files for MATLAB® are editable script files (M files), similar to the model files for Java®, for use with MATLAB®. These model files, which have the extension `.m`, contain a sequence of COMSOL commands as a MATLAB® M file. You can run the model files in MATLAB® like any other M file scripts. It is also possible to edit the files in a text editor to include additional COMSOL commands or general MATLAB® commands. Running model files in the M file format requires LiveLink™ *for* MATLAB®.

With a license for LiveLink™ *for* Excel®, you can save models on the Visual Basic for Applications (VBA) format for use with Excel®.

Compiled C-code can be linked with a model or application in several ways including the External Material and External Function interfaces in the Model Builder and as the External Library interface in the Application Builder.

FILE TYPE	EXTENSION	READ	WRITE
MATLAB®: Model file for MATLAB®	`.m`	No	Yes
MATLAB®: Function[1]	`.m`	Yes	No
Java®: Archive file	`.jar`[6]	Yes	No
Java®: Compiled model file for Java®	`.class`	Yes	No
Java®: Model file for Java®	`.java`	No	Yes
C: Function	`.dll`[3],`.so`[4],`.dylib`[5]	Yes	No
Fortran[7]: Function	`.dll`[3],`.so`[4]	Yes	No
Excel®[2]	`.xlsx`,`.vba`	Yes	Yes

[1]Requires LiveLink™ *for* MATLAB®

[2]Requires LiveLink™ *for* Excel®, available for Windows® only

[3]Available for Windows® only

[4]Available for Linux® only

[5]Available for macOS only

[6]Can be linked to a method in an application using the Application Builder

[7]Requires a C-wrapper to the Fortran code and to include the C-code DLL as an External C Library in the Application Builder. Alternatively, you can include a Fortran DLL directly as an External C Library in the Application Builder. Note: You need to consider the naming convention for exported Fortran function names. For example, in Windows®, it is typically required that Fortran function names are in uppercase.

Numerical and Interpolation Data Formats

The grid, sectionwise, and spreadsheet file types can be read for defining interpolation functions. The sectionwise and spreadsheet file types can furthermore be read and used for defining interpolation curves and written for exporting results. In addition, tables can be copy-pasted on spreadsheet format.

Parameters and variables can be imported and exported to the plain text, comma-separated values, or data file types.

The continuous and discrete color table text file types are used for user-defined color tables for results visualization.

Digital elevation model (DEM) files can be read and used as a parametric surface for defining a geometry.

FILE TYPE	EXTENSION	READ	WRITE
Copy and paste spreadsheet format	`n/a`	Yes	Yes
Excel® spreadsheet[1]	`.xlsx`	Yes	Yes
Table	`.txt,.csv,.dat`	Yes	Yes
Grid	`.txt`	Yes	Yes
Sectionwise	`.txt,.csv,.dat`	Yes	Yes
Spreadsheet	`.txt,.csv,.dat`	Yes	Yes
Parameters	`.txt,.csv,.dat`	Yes	Yes
Variables	`.txt,.csv,.dat`	Yes	Yes
Continuous and Discrete color table	`.txt`	Yes	No
Point data for Cut Point datasets	`.txt`	Yes	No
DEM	`.dem`	Yes	No
VTK	`.vtu`	No	Yes

[1] Requires LiveLink™ *for* Excel®, available for Windows® only

Appendix E — Connecting with LiveLink™ Add-Ons

The following table shows the options to start COMSOL and the different linked partner software using the LiveLink™ add-on products.

COMSOL® Software	Can Start COMSOL from Partner Software	Can Start Partner Software from COMSOL	Can Connect Running Sessions
LiveLink™ *for* Excel®	Yes[1]	Yes[2]	No
LiveLink™ *for* MATLAB®	Yes[3]	Yes[4]	Yes[5]
LiveLink™ *for* AutoCad®	No	No	Yes
LiveLink™ *for* Revit®	No	No	Yes
LiveLink™ *for* PTC®Creo® Parametric™	No	No	Yes
LiveLink™ *for* Inventor®			
- Bidirectional Mode	No	No	Yes
- One Window Mode	Yes	No	No
LiveLink™ *for* PTC® Pro/ENGINEER®	No	No	Yes
LiveLink™ *for* Solid Edge®	No	No	Yes
LiveLink™ *for* SOLIDWORKS®			
- Bidirectional Mode	No	No	Yes
- One Window Mode	Yes	No	No

[1]When you open a model from Excel®, a model window starts and a link is established automatically. The model window is an output window that displays geometry, mesh, and results.
[2]A model that includes a table reference to an Excel® spreadsheet automatically starts an Excel® process in the background when the model is run in the COMSOL Desktop environment.

[3]You can start a COMSOL Multiphysics server from a MATLAB® session using the `system` command and then connect to it using `mphstart` in the MATLAB® command prompt.

[4]The "COMSOL with MATLAB®" desktop shortcut starts a COMSOL Multiphysics server and MATLAB®, then connects them automatically. When you run a COMSOL Multiphysics model in the COMSOL Desktop interface that includes a MATLAB® function (Global Definitions > Functions), a MATLAB® engine and connection is started automatically.

[5]You can connect a MATLAB® session to a running COMSOL Multiphysics server using the command `mphstart` in the MATLAB® command prompt.

Index

A
AC/DC Module *60, 83*
accuracy
convergence analysis *52*
Add Material window
busbar model *73*
opening *39, 73*
Add Multiphysics
button *84*
window *84*
add-in *107*
add-on modules
AC/DC Module *60, 81, 83*
Application Libraries *28*
CAD Import Module *159, 195*
CFD Module *128*
Chemical Reaction Engineering Module *198*
Design Module *159, 165, 195*
ECAD Import Module *197*
MEMS Module *119*
physics list *35, 62*
physics list, and *62*
Plasma Module *198*
Structural Mechanics Module *33, 44*
study types *63*
advanced topics *110*
all combinations
parametric sweep *142*
analysis
convergence *51*
example, parametric sweep *55*
animation *19, 152*
application *12, 96*
Application Builder *6, 8, 11, 12, 23, 97, 181*
Application Gallery *30*
Application Libraries
examples *28*
file types *29*
MPH files *10, 28*
window *30*
Application Library update *30*
autocompletion
parameter and variable names *149*
parameters and variables *66*
average temperature *144*

B
blank model, creating *14, 16*
boundaries *80*
adding to selection *82*
variables scope *24*
boundary condition *77*
automatically defined *44*
boundary load *43*
busbar, setting *77*
electric current *77, 81*
fixed constraint *42*
free *41*
ground, electrical *83*
heat transfer *77*
insulating *80*
material interface *44*
settings window *8*
boundary load *43*
boundary section
context menu *79*
boundary selection, busbar model *80*
Build All button
geometry *70*
meshes *46, 86*
built-in
constants, functions, and variables *28*

materials *39, 73*
variables *90*
button object *100*

C CAD Import Module *159, 195*
Cancel button *10, 27*
case
parameter *22, 142*
CFD Module *128*
chamfer *159*
Chemical Reaction Engineering Module *198*
cloud computing *152*
Cluster Computing node *153, 156*
cluster job *155*
Cluster Sweep node *153, 156*
coloring
materials *76*
selections *76*
command sequence *104*
Compact MPH files *29*
compiler *12*
component coupling *144*
Component node, adding materials *73*
computing studies *55*
COMSOL Client *11*
COMSOL Compiler *12*
COMSOL Desktop environment
customizing *25*
on Linux *18*
on macOS *18*
overview *6*
COMSOL Multiphysics *11*
native CAD format *37*
opening *34*
COMSOL Server *11*
constants
mathematical and physical types *28*
constraints *159*
context menu
domain and boundary sections *79*
contextual tab *17*
contributing node *127*
convergence analysis *51, 57*
Convergence plot *9, 87*
cooling
air stream *59*
natural convection *65*
coupling operator *144*
Create Selection *25*
Create Selection button *122*
custom Settings window *107*
custom studies *63*
customized desktop *25*

D datasets, defined *19*
default feature *78*
degrees of freedom *52, 57*
derivatives *28*
Derived Values *57, 95*
defined *19*
Global Evaluation *56*
Volume Maximum *52, 56, 95*
Design Module *159, 165, 195*
dimensions *159*
DirectX *13*
discretize *46*
documentation, models *10, 28*
domain level *79*
domain section
context menu *79*
domains
materials *75*
remove from selection *75*
variables scope *24*
dynamic help *10, 27*

E ECAD Import Module *197*
edges
variables scope *24*

Editor Templates window *103*
eigenfrequency analysis *20*
Electric Currents interface
defining boundary conditions *79*
multiphysics *63, 83*
overview *78*
electric potential *60*
physics node *81*
voltage drop, parameter *65*
electrical heating *59*
empty
selection *43*
equation
built-in *78*
user-defined *49*
evaluating
volume maximum *52, 93, 95*
von Mises stress *53*
Evaluation Group *57, 95*
automatic re-evaluation *57*
example
advanced, electrical heating *59*
basic, structural mechanics *33*
Excel® *199, 202*
exclusive node *126*
executable *12*
Export node *19*
exporting images *96*
expressions
Boolean *50*
manual entry *53, 65*
replacing *50, 90, 93*
units, specifying *52*
user-defined *49*
External Process window *9*

F
fillet *159*
fillets *165*
find
parameters and variables *67*
finite element
mesh *46*
preconditioning *55*
tetrahedra *46*
Fixed Constraint node *42*
Floating Network License *153*
Form editor *11, 100*
form objects *97*
form union, geometry *70*
frequency response *20*
frequency-domain study *20*
functions
advanced topics *110*
built-in *28*
mathematical *28*
scope *65*
Functions node *65*

G
geometric dimensions
parameters, and *10, 21, 174*
parametric sweep *65*
geometry
building *65*
CAD format *36*
importing *36*
in Application Libraries *68*
loading from file *65, 68*
operations *159*
parameterized *10, 21, 65, 69, 174*
sequence *159*
settings window *8*
Geometry node *31*
Global Definitions node *19, 22*
functions *65*
parameters *65*
scope *24, 65*
variables *23*
global parameters *22, 40, 54, 65*
expression *21*
glTF-file format *96*

graphics
rendering and hardware *13*
graphics object *99, 101*
Graphics toolbar
default view button *42, 49, 83, 92*
image snapshot button *96*
zoom extents button *71, 89*
Graphics window *7, 9, 27*
geometry, and *37*
plot *47*
rotate geometry *42, 88, 89*
selecting boundary *42, 43, 83*
toolbar *7*
using *38, 71*
zoom box *43*
Ground, boundary condition *82*
grouping
nodes *23*

H Heat Transfer in Solids interface
defining boundary conditions *79*
multiphysics *63, 83*
overview *78*
Heat Transfer Module *128*
Help window *10*
opening *18*
high performance computing *152*
Home tab *17*
HPC *152*
hybrid parallelism *154*

I Image Snapshot button *96*
images, creating thumbnails *96*
importing geometry *36, 37*
information windows *7, 9, 27*
initial conditions *8*
Initial Values node
Electric Currents interface *79*
Heat Transfer in Solids interface *79*
Solid Mechanics interface *41*
input fields *98*
Input Parameters
for part *175*
Insert Object menu *103*
Isosurface plot *87, 151*
iterative solver
preconditioning *55*

J Java file *32, 199*
Joule heating
equations *78*
multiphysics coupling *83*
multiphysics interface *62*

L Laminar Flow interface *131*
Linux
clusters *155*
version *18*
LiveLink™ add-ons *202*
loft *159*
Log window *9, 47*

M M file *32, 199*
macOS version *18*
Material Browser
defined *10*
material contents section *39, 74*
material color and texture *76*
material interface
mechanical contact *44*
materials
copper *59*
domains, assigning *75*
model tree *31*
settings window *8*
steel *39, 49*
titanium alloy *59*
Materials node *39, 73*
mathematical constants and functions *28*
MATLAB® *199*

Max/Min Volume plot *53*
Maximum and minimum values table *94*
MEMS Module
  thermal expansion, and *119*
mesh
  convergence analysis, and *52*
  density *54*
  finite element *46*
  model tree, and *31*
  parameterizing *54, 85*
  physics-controlled, default *85*
  refining *54*
  settings *46*
  unstructured tetrahedral *85*
  user-controlled *54, 85*
mesh element size
  parameters, defining *21, 85*
  settings *46*
message passing interface (MPI) *154*
Messages window *9, 47*
method *107*
  model *23*
Method Call *107*
Method editor *11, 107*
midsurface *159*
modal tab *17*
Model Builder *6, 97*
  defined *10*
  example *31*
  expanding sections *78*
  geometry import *37*
  node sequence example *41*
  the ribbon, and *17*
  toolbar *6*
  window *26*
model history *32*
model method *23*
model tree *6, 26*
  building *18*
  defined *65*
  example *31*
  geometry *31*
  Global Definitions node *19, 40, 65*
  Materials node *31, 39*
  Model Builder *11*
  nodes *18*
  Results node *19*
  ribbon *17*
  root node *19*
  Study node *20, 31*
Model Wizard
  adding physics *20, 35, 62*
  Component node, and *20*
  creating a new model *14*
  custom studies *36, 63*
  opening *34, 61*
  preset studies *36, 63*
  space dimension *34, 61*
  study *20, 63*
models
  defining *10*
  discretization *46*
  documentation *10, 28*
  saving *72*
  structural mechanics *33*
  symmetry *89*
  workflow *31*
MPH file *195*
  full and compact *29*
  saving *72, 96*
MPHBIN file *195*
MPHPHB file *195*
MPHTXT file *195*
MPI *154*
multicore processors *152*
multiphysics interface *63*
multiphysics models *59*
Multiphysics node *78, 79, 84, 121*

multiphysics phenomena *59*
Multislice plot *87*

N New Form wizard *97*
no solutions stored, message *30*
node groups *23*
nodes *18*
  default feature *41*
  reordering *32*

O OpenGL *13*
Override and Contribution *41, 127*

P parallel computing *152*
parameter
  case *22, 142*
parameter switch
  parametric sweep *142*
parameters *21, 99*
  defining *54, 65*
  editing *70*
  expression *21, 54*
  global *22*
  meshes, and *54*
  name *54*
  range of values *55*
  results *22, 150*
  scope *65*
  using, referencing *81*
Parameters node *65, 110*
parametric geometry model *159*
parametric study *65*
parametric sweep *21*
  all combinations *142*
  example *54*
  meshes, and *54*
  parameter switch *142*
  range, defining *55*
  specified combinations *142*
part *10, 174*
  Input Parameters *175*
Part Libraries *174*
physics
  adding *62*
  boundary conditions *77*
  electromagnetic heating *62*
  heat transfer *62*
  Joule heating *59, 78*
  laminar flow *131*
  model tree *31*
physics interface *14, 63*
  Electric Currents *63*
  Heat Transfer in Solids *63*
  Joule Heating *62*
  Laminar Flow *131*
  Solid Mechanics *35*
Plasma Module *198*
player *152*
plot
  regenerate after solving *58*
plot group *19*
  3D, adding *50*
Plot window *26*
plots
  expression, user-defined *49*
  max/min volume *53*
  model tree *31*
  regenerating *53*
  surface *48, 50*
  windows *9*
points, variables scope *24*
Poisson's ratio *75*
polynomial functions *46*
preconditioning *55*
preferences *13*
preset studies *63*
Preview MPH files *30*
principal stress *126*
program code
  model file for Java *32*

model M-file *32*
progress bar *10, 27*
Progress window *9, 47*

Q Quick Access Toolbar *6, 8, 18, 26*

R Record Code *108*
renaming plot groups *51*
rendering options *13*
reports, defined *19*
reserved name *28*
resolution of curvature, mesh *85*
results
color table range *88, 90*
derived values *52, 56, 95*
displaying *48*
parameters *22, 150*
surface settings, modify *88*
units, changing *48*
Results node
3D plot group *53, 90, 124*
busbar defaults *87*
subnodes *19*
surface node, adding *90, 124*
ribbon *6, 26*
Add Material *39, 77, 130*
Build All *70*
Build Mesh *86*
Compute *55, 124, 140*
defined *8, 17*
example *6*
importing geometry *37*
Model Builder *17*
Physics tab *42, 121*
Windows *10, 48, 163*
Work Plane *161*
root node
default units *143*
defined *19*
model thumbnails *96*
Model Wizard *34*
running simulations *87*

S saving files *72, 96*
scheduler
Grid Engine *155*
Open Grid Scheduler *155*
PBS *155*
SLURM *155*
Windows HPC Server *155*
scope
global definitions *65*
parameters and variables *24*
parameters and variables, defined *21*
variable name *25*
search
parameters and variables *67*
Select Study window *16*
selecting
boundaries *44*
multiple boundaries *122*
using the graphics window context menus *44*
selecting internal surfaces *44*
Selection *25, 44, 76*
empty *43*
selection *122*
Selection List window *10, 163*
selection of geometry objects, domains, boundaries, edges, and points *44*
Settings Form *107*
Settings Forms *107*
Settings window *6, 8, 18, 26*
customized *107*
shared memory parallelism *152*
Show Material Color and Texture *76*
Show Selection Colors *76*
simulation, running *87*
software rendering *13*
Solid Mechanics interface *35*

add-on modules *119*
Solved MPH files *29*
solvers
configurations *55*
iterative *54, 55*
stationary *47, 55*
using *47*
space dimension *14*
specified combinations
parametric sweep *142*
Stationary study *36, 47*
steady-state study *20*
structural analysis *33*
structural displacement field *46*
structural mechanics
design *51*
plastic deformation *49*
stresses and strains *59, 75*
von Mises stress *48*
Structural Mechanics Module
application library *33*
mechanical contact *44*
thermal expansion *119*
studies
computing solutions *55, 87*
defining *47*
example, multigrid iterative *55*
example, stationary *47*
model tree *31*
preset *63*
types *20*
Study node *20*
solution sequence *87*
Suppress Selection Highlight *76*
Surface plot
busbar results *87*
color table range *91*
deformation *48*
electrical current density *89*
replace expression *90*
settings *50*
updating *89, 90*

T table
Maximum and minimum *94*
Table window *9*
graph plot *56*
wrench results *53*
Tables node *19*
tables, evaluating *56*
temperature *87, 140*
temperature, average *144*
Test Application *102*
tetrahedron, polynomial functions *46*
thermal expansion *59, 75*
thicken *159*
thumbnail image *96*
time-dependent study *20*
torque, applied *33*

U unit, changing *48, 125*
update COMSOL Application Libraries *30*
Update Solution *149*
user interface
COMSOL Desktop *6*
overview *6*
user-controlled mesh *54*

V variables
advanced topics *110*
built-in *28*
example, built-in *56*
expression *23*
scope *65*
scope, limit *24*
Variables node *23, 65*
visualization *9*
color table *88, 90*
Volume Maximum, evaluating *52, 93, 95*

von Mises stress *48, 126*

W Windows list *48, 163*
wizard
New Form *97*
workflow *28*

Y yield stress *33, 49*
Young's modulus *75*

Z zoom *141*
Zoom Extents command *105*