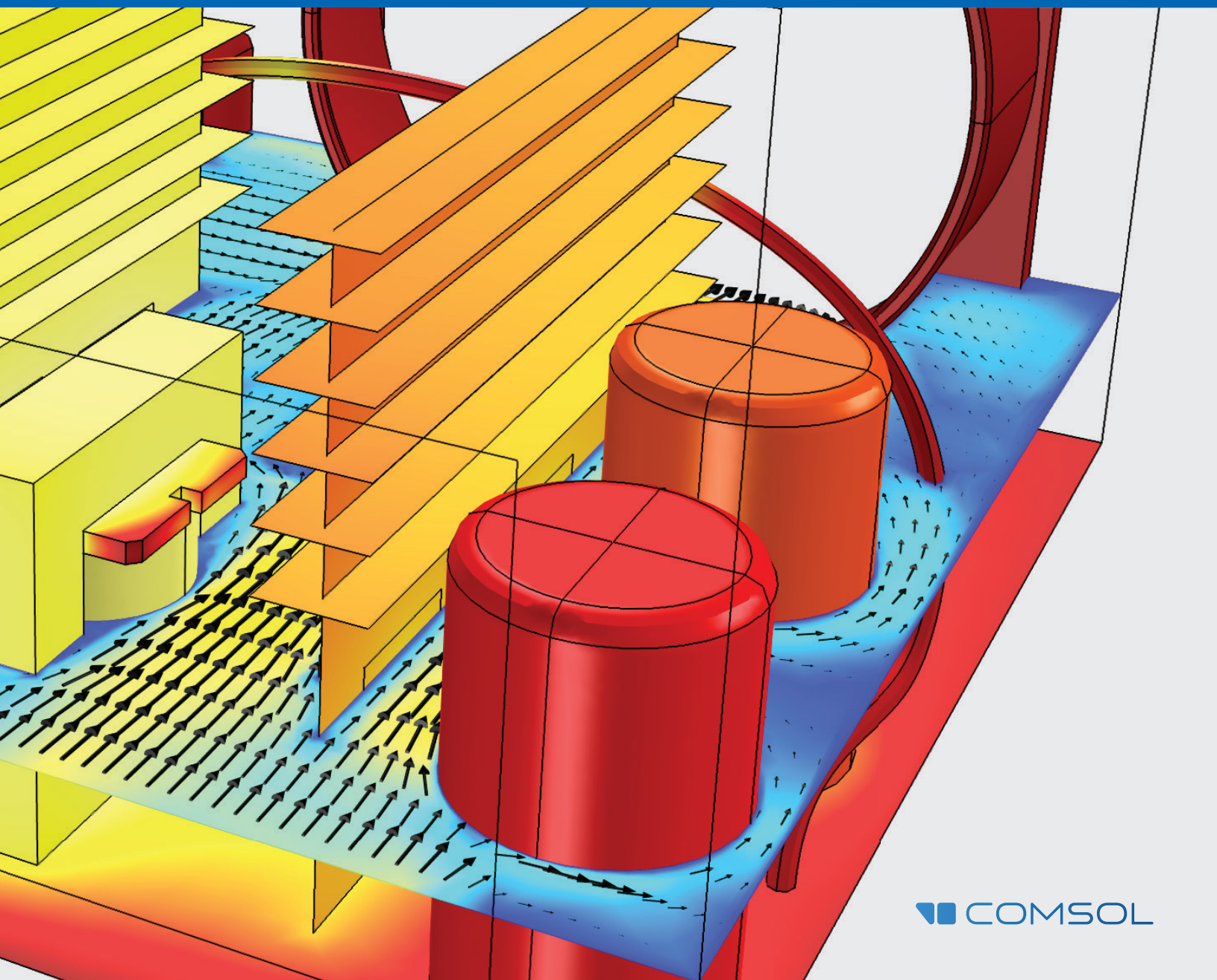
COMSOL HANDBOOK SERIES

Specialized Techniques for Postprocessing and Visualization in COMSOL Multiphysics®



COMSOL

Specialized Techniques for Postprocessing and Visualization in COMSOL Multiphysics

COMSOL, COMSOL Multiphysics, Capture the Concept, COMSOL Desktop, LiveLink, and Simulation for Everyone 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.
©2015-2016 COMSOL.

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.

Further Resources

Further writing and tutorials on postprocessing in COMSOL® software are available here:

PREVIOUS HANDBOOK: ESSENTIALS OF POSTPROCESSING AND VISUALIZATION IN COMSOL MULTIPHYSICS

http://www.comsol.com/offers/Postprocessing-and-Visualization-Handbook-Part-1

VIDEOS

http://www.comsol.com/videos?workflow=workflowstep-results

BLOG ARTICLES

http://www.comsol.com/blogs/category/all/postprocessing/

DISCUSSION FORUM

www.comsol.com/community/forums/results-and-visualization/

SUPPORT KNOWLEDGE BASE

www.comsol.com/support/knowledgebase/

TABLE OF CONTENTS

INTRODUCTION 1

PLOT EXPRESSIONS AND RESULTS INTERPRETATION

- Creating Your Own Expressions 2
- Using Derived Values to Verify Your Model 4
- Mesh Plots 5
- Deformations, Height Expressions, and Scaling 6
- Operations for Results Interpretation 7

VISUALIZATION TOOLS

- View and Camera Control 13
- Lighting and Scene 20

APPLICATION-SPECIFIC PLOT TYPES

- Far-Field Plots 23
- Particle Tracing 24

ADVANCED TIPS & TRICKS

- Ideas for Powerful Visualization 26
- Animations 30
- Rendering and Smoothing 31

LOWER-DIMENSION POSTPROCESSING

- Terminology Refresher 33
- Thin Layer Attributes 34
- Discontinuities 36
- The Up and Down Operators 36
- The Side Operator 37
- Comparing Domain and Boundary Plots 38
- Summary 40

CONCLUDING REMARKS 41

INTRODUCTION

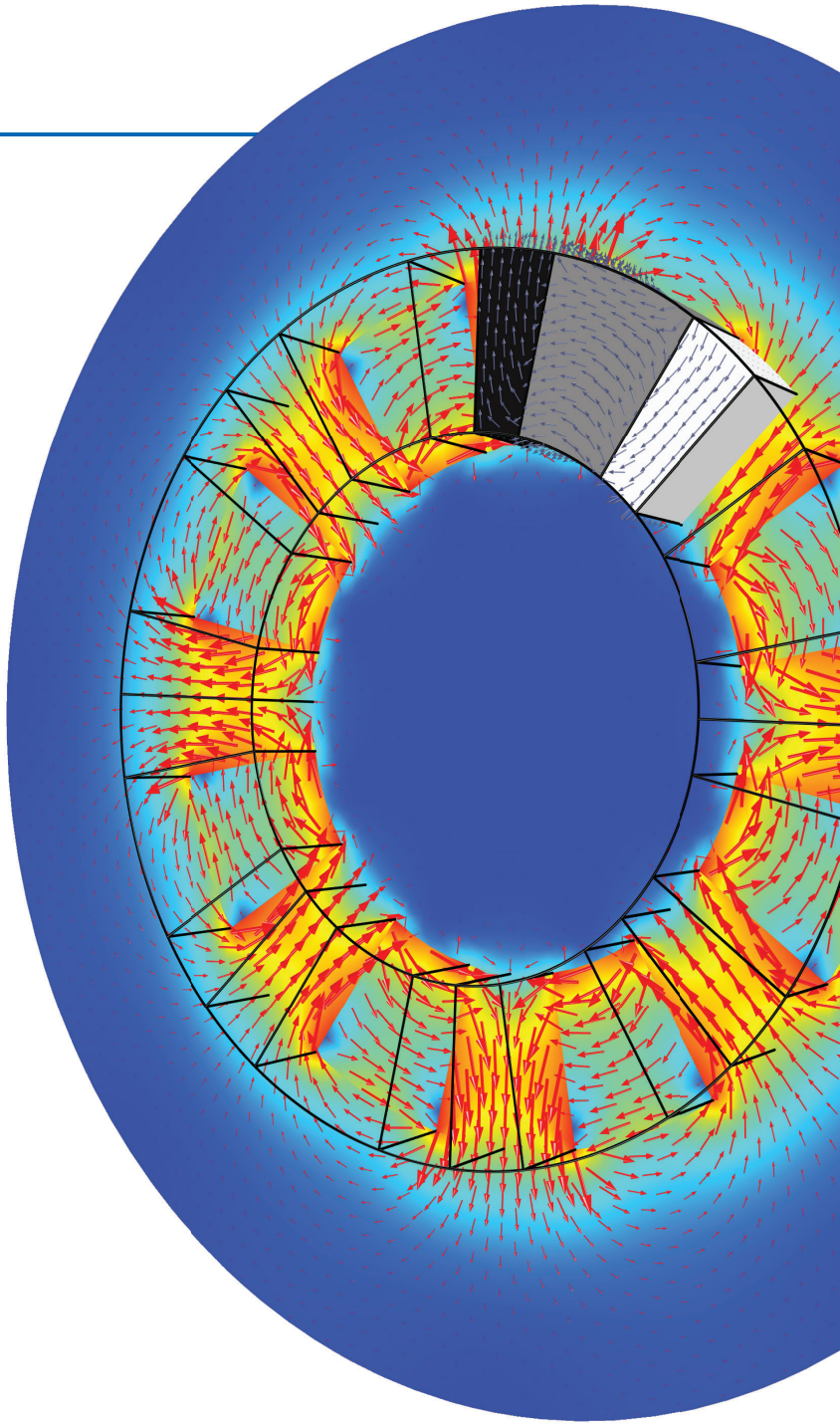
Welcome back! If you're reading this, hopefully that means you found the first part of this handbook set, Essentials of Postprocessing and Visualization in COMSOL Multiphysics®, helpful for understanding the postprocessing and visualization techniques used in the COMSOL Multiphysics® software.

In the first handbook, we explored ways to use data sets, derived values, and different plot groups. At the very end, we investigated some tips and tricks such as showing mesh on surfaces, interactive positioning tools, and shortcuts that can be used to manipulate the View node and entity selections.

In this second part, we will forge ahead to a more in-depth look at the flexible postprocessing capabilities of COMSOL® software that are relevant for a more experienced user. In particular, these techniques are meant to aid the verification, understanding, and validation of your model. These include performing operations on data sets, creating deformations that can be used to show realistic displacements, defining your own expressions, and interpreting results.

Lastly, we'll dive briefly into some application-specific questions, such as when it might (or might not) be appropriate to change the resolution and smoothing on a plot, demonstrate camera and lighting tools to augment visualization, showcase some advanced combinations used to create stunning visuals, and discuss techniques for postprocessing on entities with dimensions lower than that of the model geometry.

We hope that the material that follows will offer you new insight into your simulations and equip you to more easily analyze, visualize, and share your work.



STATIC FIELD MODELING
Simulation of a 4-pole pair Halbach rotor. Results show the magnetic flux density at a cross section and as a function of the rotor angle.

PLOT EXPRESSIONS AND RESULTS INTERPRETATION

CREATING YOUR OWN EXPRESSIONS

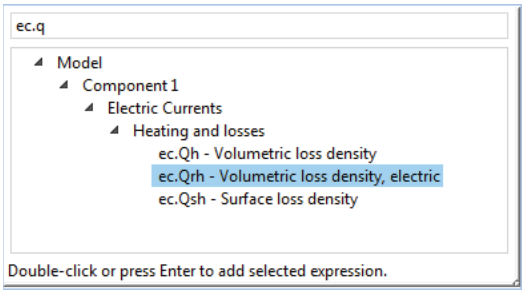
For every plot type used in the COMSOL software, an expression determines the quantity plotted. Before we demonstrate some tools for visualization, we'll look briefly behind the scenes of these plot expressions.

COMSOL offers a plethora of predefined plot quantities, but sometimes it is necessary to create your own expressions. As an example of a user-defined expression, let's re-create the formula for Joule heating using a model of a busbar experiencing 20 mV of electric potential.

Go to the application libraries (under *File* > *Application Libraries*) and open *COMSOL Multiphysics* > *Multiphysics* > *busbar*. This simulation shows the heating in the busbar caused by an applied electric current.

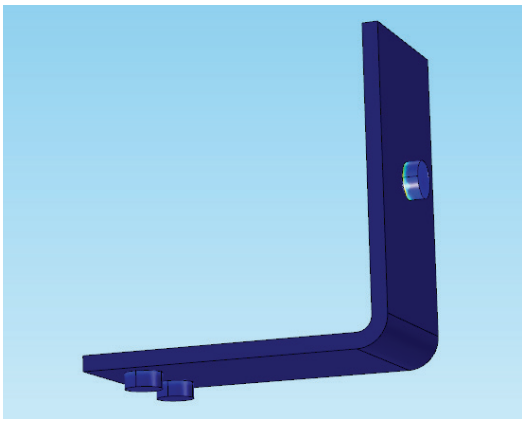
Mathematically, Joule heating is described by the power dissipated, $Q = \vec{J} \cdot \vec{E}$, where $\vec{J}$ represents current density and $\vec{E}$ the electric field.

This is equivalent to the power dissipation density, which is represented in COMSOL by the name *ec.Qrh* (the *ec* prefix refers to the Electric Currents interface in the model, and the quantity is called "Volumetric loss density, electric"). These quantities can be found by clicking Replace Expression in the plot settings, and then searching in the *Type filter text* field:

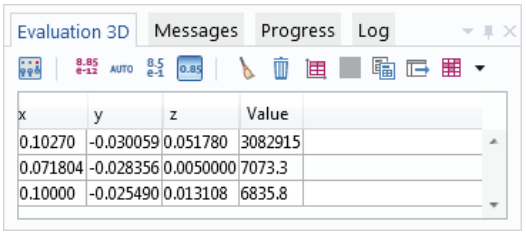


Add a 3D plot group to the Results node of the busbar model and add a surface plot to it. We will work within this node to examine several different quantities. First plot the expression *ec.Qrh*.

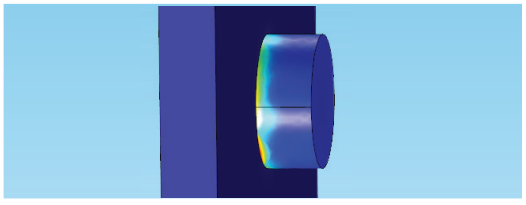
At first glance, there appears to be only a small amount of resistive losses produced in the busbar, localized in the single bolt at the back.



But click anywhere on the busbar, and you'll see an Evaluation 3D table appear below the Graphics window. This will show the coordinates clicked, as well as the value of the resistive losses at that point. These will range from very small—single digits—to thousands of watts per cubic meter (W/m^3); but since most of the busbar experiences far less heat than the bolt, the color gradient appears fairly uniform despite the wide range of values.



The expanded scalar product that represents Joule heating mathematically is $\vec{J} \cdot \vec{E} = J_x E_x + J_y E_y + J_z E_z$. The equivalent expression in COMSOL notation is *ec.Jx*ec.Ex + ec.Jy*ec.Ey + ec.Jz*ec.Ez*. If you plot this expression, it will look the same as the plot of *ec.Qrh*.



REFERENCE

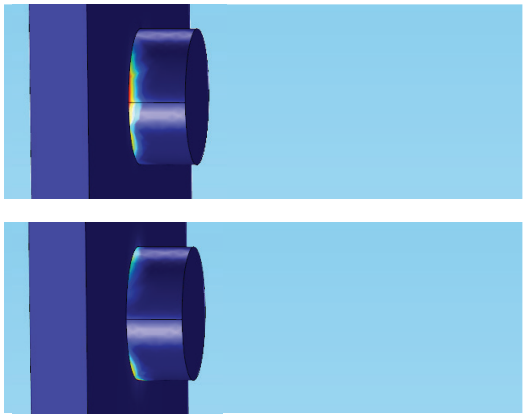
For an introduction to the different plot types available in COMSOL Multiphysics® software, head to the *Plot Types* chapter of Essentials of Postprocessing and Visualization in COMSOL Multiphysics®.

HINT

The keyboard shortcut *CTRL+spacebar* functions as an auto-complete command, opening the *Replace Expression* menu with a list of quantities whose names contain any terms already typed in the *Expression* field.

We can easily deconstruct this expression and check the contribution from each term. Change the expression to the product of the current and the electric field in the *x*-direction (*ec.Jx*ec.Ex*); it looks nearly the same as the previous image.

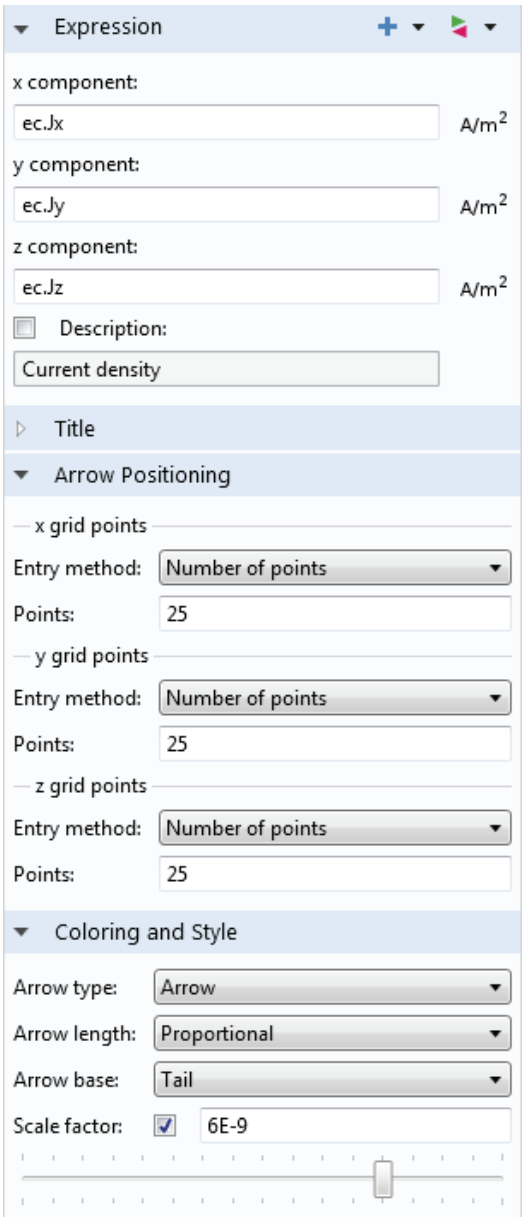
However, if we change it to *ec.Jy*ec.Ey*, we'll discover that there isn't as much power dissipation in the *y*-direction (shown below, top); it's still concentrated around that back bolt, but is lower than the power dissipation in the *x*-direction. It decreases even further in the *z*-direction (bottom, below):



Now plot the current density norm, *ec.normJ*. You'll see, once again, that the current is concentrated in the back bolt.

Disable the node for this surface plot and add a second surface to the plot group. Plot the electric field norm, *ec.normE*. In this case, the plot again shows that the electric field is concentrated in the bolts. Since Joule heating is described by the scalar product of these two quantities, it follows that this plot would also show the most heat in the bolt region.

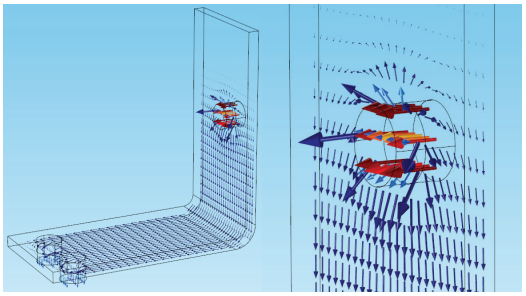
Now disable the electric field norm plot and add an arrow volume. The default plot expression is the current density. You can use the plot settings shown below or adjust them to your liking.



NOTES

Expressions follow the naming pattern of **interface.quantity**. For example, *ec.Jx*ec.Ex* is the product of the *x*-components of the current density and electric field vectors, as defined in the Electric Currents physics interface. Many other mathematical operators and functions are built in to the COMSOL® software, and a list is available in the *Global and Local Definitions* section of the COMSOL Multiphysics Reference Manual available with the COMSOL documentation. You can also create your own variables and operators and type them in manually, provided their definitions are accurate on the entity being plotted.

The resulting arrow plot will show the strength of the current at different locations in the busbar. Below, the arrow plot is shown with a color expression based on the strength of the electric field norm, *ec.normE*.



The plot confirms that the areas of highest current are in the bolts, and therefore that these areas experience the most heat. The arrows also show, as the Joule heating plots did, that the majority of the resistive losses happen in the *x*-direction.

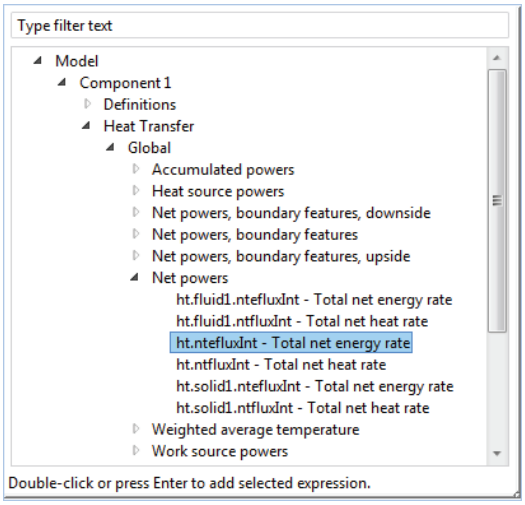
USING DERIVED VALUES TO VERIFY YOUR MODEL

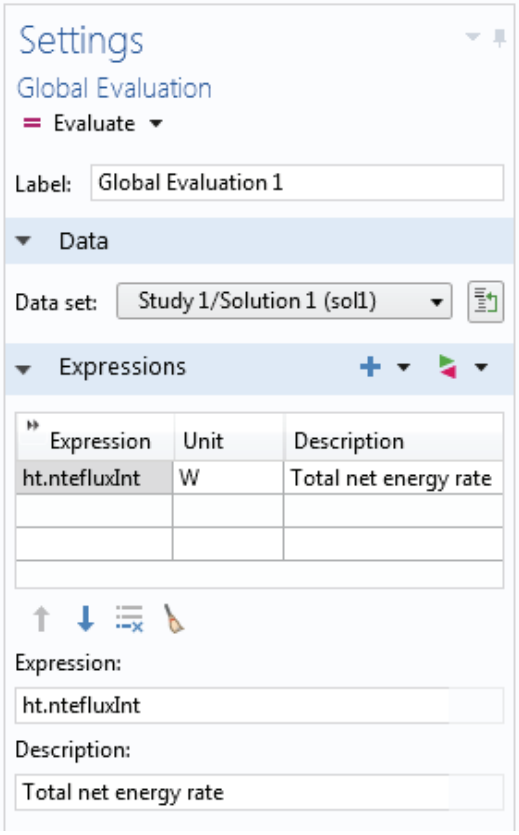
In Essentials of Postprocessing and Visualization in COMSOL Multiphysics, we briefly touched on derived values. In addition to, for example, investigating the figures of merit for your application, derived values can tell you a lot about the mesh you're working with. Verifying that a model is accurate confirms, among other things, that the mesh is fine enough to capture the behavior of the system. Global evaluations, one of several derived values available in COMSOL, are a good way to do this. We'll demonstrate a couple of these as examples.

Let's now open the heat sink model that was used in the first handbook. If you have the Heat Transfer Module or CFD Module installed, open the heat sink model from *File > Application Libraries > Heat Transfer Module > Tutorials, Forced and Natural Convection* or from *File > Application Libraries > CFD Module > Non-Isothermal Flow*.

If material properties and boundary conditions are properly set, verifying how well mass and energy are conserved is an important measure of the accuracy of the model. Under Results, expand the Derived Values node. After you've explored the existing subnodes, delete or disable them so that we can start from scratch to demonstrate the process. Then add a global evaluation using the right-click menu.

The first evaluation will be performed on the power flowing through the device. In the settings, navigate to *Replace Expression > Model > Component 1 > Heat Transfer > Global > Net powers* and choose *ht.ntefluxInt - Total net energy rate*.



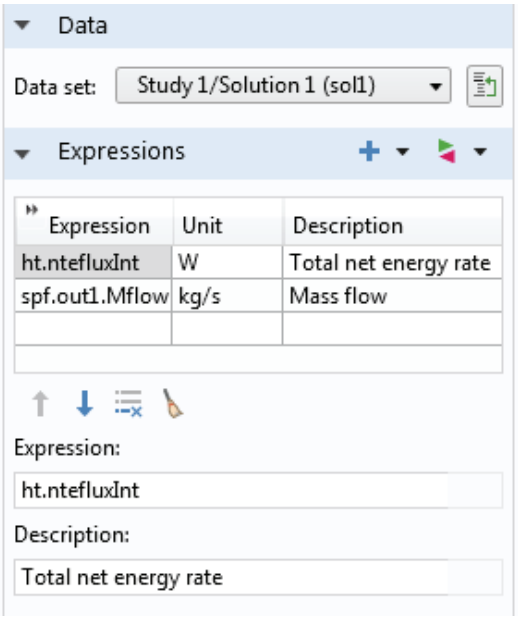


LIBRARY UPDATE

This handbook assumes that you have updated the COMSOL Application Libraries. This can be done from *File > Help > Update COMSOL Application Library*. Then click *Find Applications* and click *Uncheck all* on the next screen. Navigate to the model that you're looking for, select it, and click *Download*.

Click Evaluate. A table will open beneath the Graphics window showing the value of the total net energy power: 0.99997 W. Since the only heat flux input was set to 1W, we can see that energy is conserved with a reasonable level of accuracy.

The second global evaluation will be performed on the mass flow. Use the Add Expression button + to include a second expression in the Expressions table. In the menu, navigate to *Model 1 > Component 1 > Laminar flow > Auxiliary variables* and choose *spf.out1.Mflow - Mass flow*.



Click Evaluate. Both evaluations will now appear in the table together. The result for the mass flow evaluation is 2.709e-5 kg/s. If we multiply the inlet velocity (5 cm/s), fluid density (1.2754 kg/m3), and area of the inlet (3 cm by 1.5 cm), we get a value of 2.8e-5 kg/s and can see that mass is also well conserved.

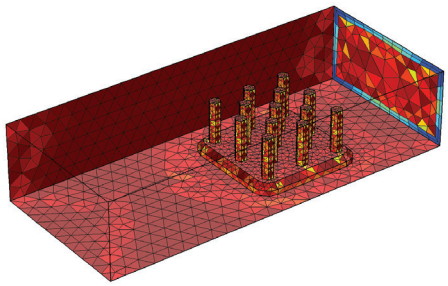
Since these evaluations matched the expected results, we know that the mesh is fine enough and the model is accurate given the operating conditions under investigation.

MESH PLOTS

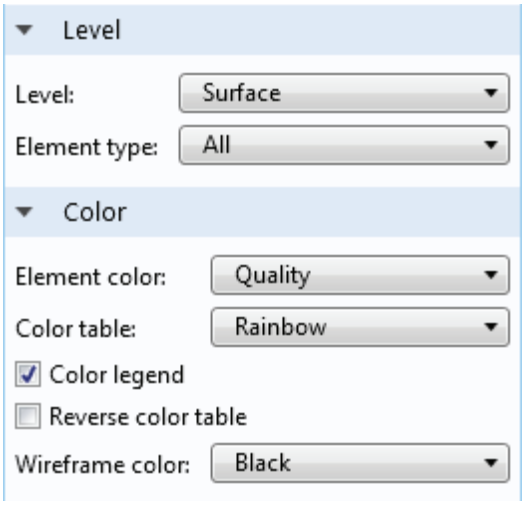
Postprocessing quality, as well as solution accuracy, depends greatly on the size and quality of the elements in a mesh. Too coarse a mesh will result in too few data points, and distorted elements will affect the geometric and numerical representation of your application. Both result in poor accuracy. We've previously discussed using derived values to check whether the mesh is fine enough. Now we'll focus on its quality.

One way to determine the quality of the mesh or visualize elements on the model geometry is to create a mesh plot. These don't even require solving the model in order to plot them.

In the heat sink model under the Component 1 node, right-click the Mesh node and choose Plot. You'll see something like this:



The accompanying color legend ranges from 0 to 1 and describes, by default, the quality of the elements. The color can also be based on element size, set to a uniform color, or disabled entirely.



The mesh quality is a measure of how isotropic a mesh cell is. An element with equal side lengths, for example a 2D equilateral triangular element, has the quality 1. A completely flattened triangle with no area has the quality 0. Similar geometric formulas are available for 3D tetrahedral elements. For other types of elements, like bricks and prisms, the quality does not have the same meaning.

REFERENCE

For an introduction to global evaluations, turn to the *Data Sets, Derived Values, and Tables* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics. For more information about derived values and interpreting results, turn to the section called *Operations for Results Interpretation* at the end of this chapter.

HINT

Sometimes the mesh needs to be refined in certain regions to improve accuracy, especially where there is a large gradient in the solution. This refinement can be done using a feature in COMSOL called *Adaptive Mesh Refinement*, which analyzes errors in the solution and refines the mesh accordingly. For details on how to show the mesh in a results plot, see the *Combining Plot Types* section in the *Advanced Tips and Tricks* chapter.

DEFORMATIONS, HEIGHT EXPRESSIONS, AND SCALING

There are several ways to create results plots that visually exaggerate changes such as displacement, bending, or shrinking. This helps to visualize deformations, especially in cases where they're not obvious when plotted on a 1:1 scale.

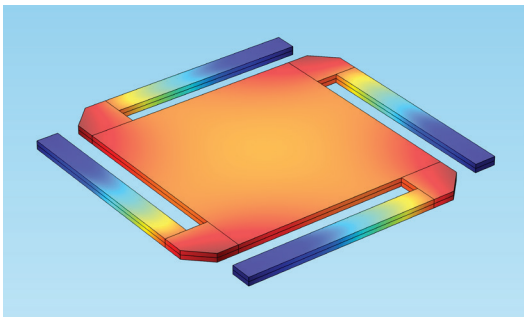
To demonstrate this, we'll use two models, a piezoacoustic transducer and a micromirror. Although you may not work with models of this type yourself, these techniques may be applied to any model dealing with many kinds of physics (for example, showing displacement on a structural mechanics plot is done in the same way for nonlinear and piezoelectric materials as it is for elastic materials).

DEFORMATIONS

If you have the MEMS Module installed, navigate to *File* > *Application Libraries* > *MEMS Module* > *Actuators* and open the micromirror model. This model simulates an electrostatically-controlled micromirror. These optical semiconductors often serve as reflection devices, such as the optical redirectors used in digital cinema production.

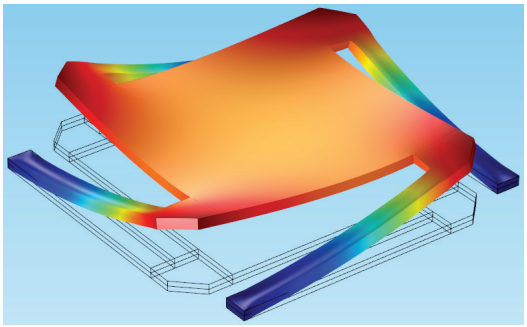
A 3D structural analysis determines the deformations and stresses necessary to lift the mirror by the correct amount. The analysis is performed for a device with steel plates and for one with aluminum plates, and the results will depict the mirror's response to different levels of prestress.

If you're following along in the model and would like to recreate the existing plot, add a 3D plot group with a surface plot to the Results node. The expression for displacement is automatically entered in the expression field. Use Study 1/ Solution 2 for the data set.



This plot describes the deformation shown on different regions of the micromirror. But we'll make it look a little more physically realistic to better depict the movement.

Add a deformation to the surface plot (right-click Surface 1 and choose *Deformation*). The software automatically enters *(u, v, w)* to plot the components of the displacement field. Change the scale factor to 1 and click Plot.



Now we can see the deformation, overlaid on the edges showing the original position of the micromirror. In the *Parameter value* entry field, test out the options to see how different parameter values result in different levels of deformation.

HEIGHT EXPRESSIONS

Next, let's take a look at a piezoacoustic transducer. This model is available under *File* > *Application Libraries* > *Acoustics Module* > *Piezoelectric Devices* if you have the Acoustics Module installed.

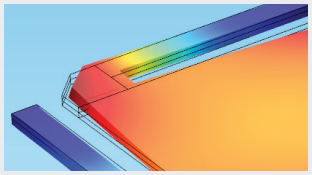
Piezoacoustic transducers are used to convert electric currents into acoustic pressure or vice versa. They're found in quite a range of devices and applications, such as ultrasound equipment, sonar, microphones, and bioimaging.

The transducer is represented by a rectangle made of a piezoelectric material and grounded on one side. The air domain is represented by one quadrant of a circle. We'll visualize the resulting acoustic pressure in the surrounding air when 100 V is applied to the transducer.

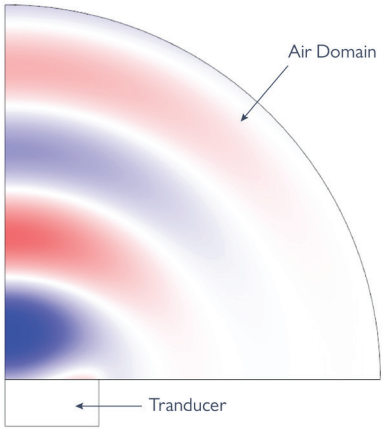
Create a new 2D plot group and add a surface plot to it. Navigate to *Replace Expression* > *Model* > *Component 1* > *Pressure Acoustics, Frequency Domain* > *Pressure and sound pressure level* and select *acpr.p_t - Total acoustic pressure field* for the plot expression.

HINT

You can also play around with the *x*-, *y*-, and *z*-components to see how the micromirror deforms in each direction. The image below shows the twisting deformation (in the *x*- and *y*-directions only) with the *z*-component set to zero.



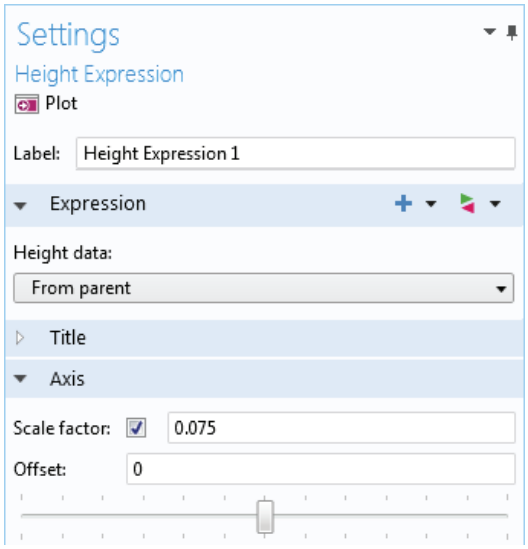
In the image below, the color settings are set to the WaveLight color table with a symmetrized color range:



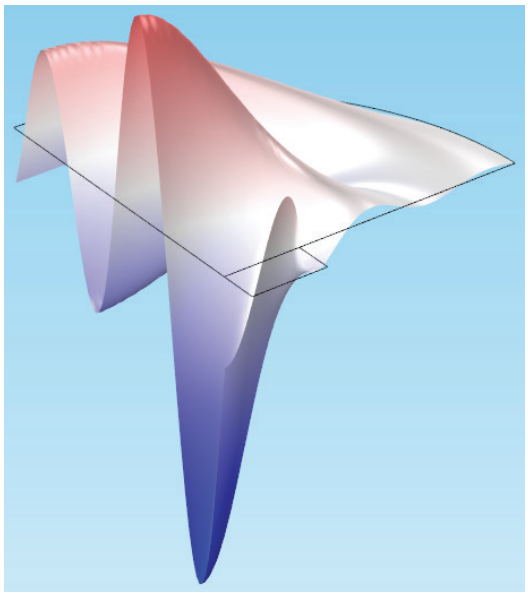
We can see the pressure field across the air domain, with the field growing weaker farther away from the transducer. But if we want to examine those waves a little more closely, we can add a type of deformation called a height expression, which will give us an extended 3D view of the pressure field.

Height expressions are particularly useful for cases where the results are wavy in nature or when a scalar or vector quantity varies mainly along one direction. Unlike other deformations, which require more setup, a height expression is an easy way to visualize peaks and troughs.

Right-click the Surface node and choose *Height Expression*. The data will automatically be taken from the parent node plotting the acoustic pressure.



Under the Scale tab, make sure to set the scale factor to 0.1 or less (you'll see why this is necessary if you set it to a larger quantity, like 1). Then click Plot.



OPERATIONS FOR RESULTS INTERPRETATION

In order to evaluate the accuracy of your model, understand the physics effects, and validate your design, it is often necessary to perform operations on a solution. These might take the form of computing averages, finding local maxima and minima for a relevant vector quantity, or checking your simulation results against experimental data.

Let's take a look at some of the available operations for data sets and derived values.

AVERAGES

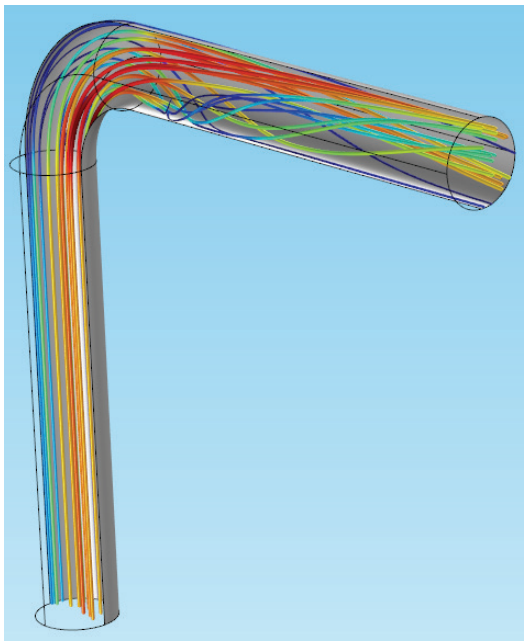
Averages can be calculated over a set of domains for a specific data set. We'll check out a model of turbulent flow in a pipe to demonstrate this (although once again, these techniques are applicable to many types of models). If you have the CFD Module installed, open the solved pipe elbow model from *File > Application Libraries > CFD Module > Single-Phase Benchmarks*.

This model simulates the flow in a 90° pipe elbow using a κ-ω turbulence model. It contains two different components; we will examine the second one.

REFERENCE

For an introduction to data sets and derived values, turn to the *Data Sets, Derived Values, and Tables* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics.

Take a look at the *Pressure (spf2)* plot group, which contains plots for the fluid velocity and the pressure inside the pipe. In the figure below, we can see streamlines showing the velocity of the water. This plot shows the separation zone after the bend and vortices caused by centripetal force.

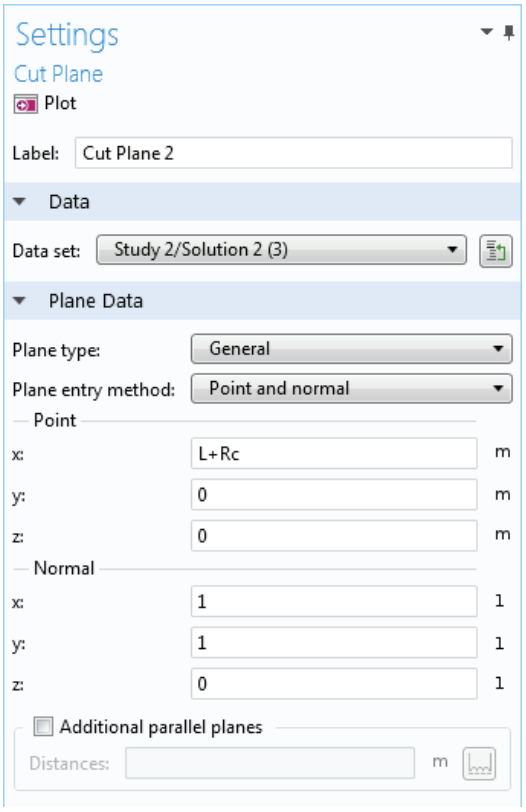


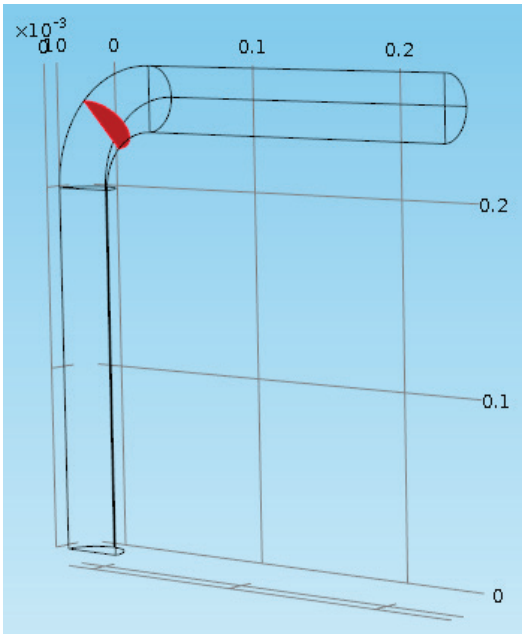
But suppose we want to determine the size of the pump necessary for a desired flow rate, or the contribution of the bend to the total pressure drop. Or suppose we need to measure the pressure on a specific plane in order to compare our results to an engineering correlation.

To this aim we can create a cut plane in the pipe bend to measure, for example, the pressure drop from the inlet to the bend and compute results using a surface average.

Although some of the data sets and evaluations already exist, we'll start from scratch to demonstrate the process.

Right-click the Data Sets node and choose *Cut Plane*. The plane's position should be defined by the length of the inlet pipe, *L*, plus the radius of the bend, R_c. In order to orient the plane perpendicular to the pipe walls and at the halfway point in the bend, set the normal vector to *(1,1,0)*.





Right-click the Derived Values node and choose *Average> Surface Average*. For the data set, choose the cut plane solution you just created. In the *Expression* field, type *p2* (the expression for pressure used for the second component of this model).

Click Evaluate. The table that appears will show the pressure, approximately 1970 Pa.

Table 1

visc_fact	Pressure (Pa)
1	1970.0236095933185

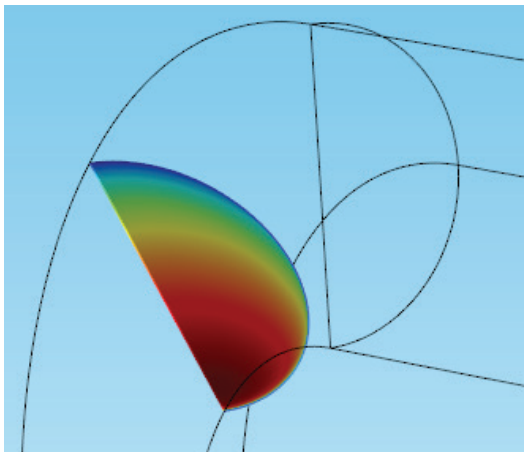
We'll find the pressure at the inlet using another surface average. Add the second surface average and use Study 2/Solution 2 (3) for the data set, which contains the solution from Component 2.

Then select the inlet boundary (boundary 1) and evaluate it using the expression *p2* again.

Table 2

visc_fact	Pressure (Pa)
1	3444.152484074552

Between these two evaluations, we've shown that the pressure drop from the inlet to the middle of the bend is quite high—almost 1500 Pa. If we plot the velocity magnitude on the cut plane, we'll also see that the velocity changes rapidly from the outer radius of the pipe bend to the inner radius. This explains the pressure drop.

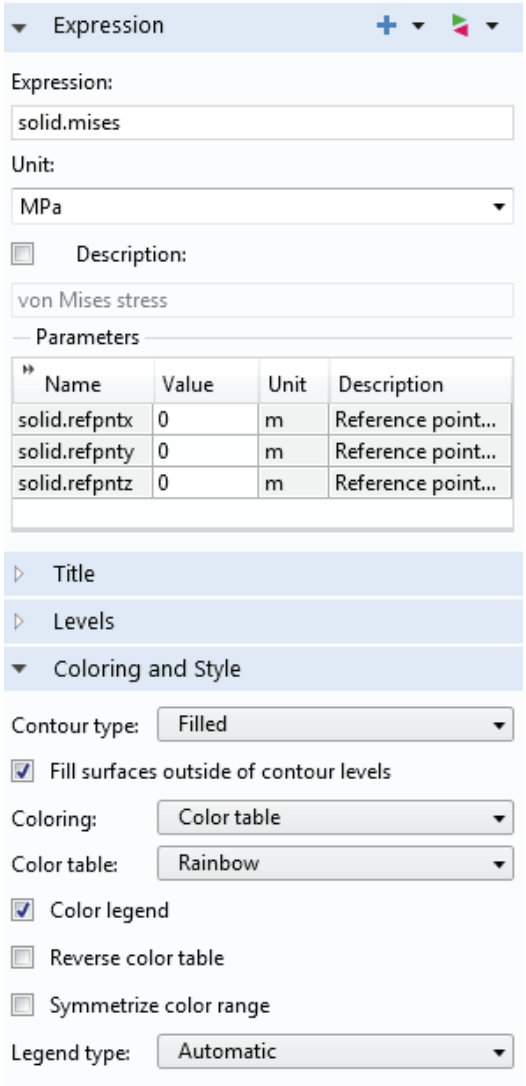


CONTOUR VALUES

Contour plots are one of the quickest and most useful ways to tell if a device will hold up to the intended operating conditions—for instance, whether an area of it will experience a temperature above the limit or stress greater than a material's yield strength.

We'll test this on the pulley model we used in the Essentials of Postprocessing and Visualization in COMSOL Multiphysics handbook, where we introduced readers to solutions, cut points, and tables. Open the model from *File > Application Libraries > COMSOL Multiphysics > Structural Mechanics > stresses in pulley*.

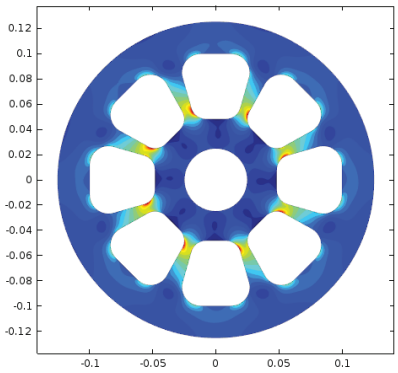
Under the Results node is a list of 2D and 1D plots. Add another 2D plot group and add a contour plot to it. Use the Study 1/Parametric Solutions 1 data set as the solution to draw from. Change the expression to *solid.mises* in order to see the stress instead. Under the Coloring and Style tab, set the contour type to *Filled*. By default, the expression will describe the displacement experienced by the pulley at 9000 rpms.



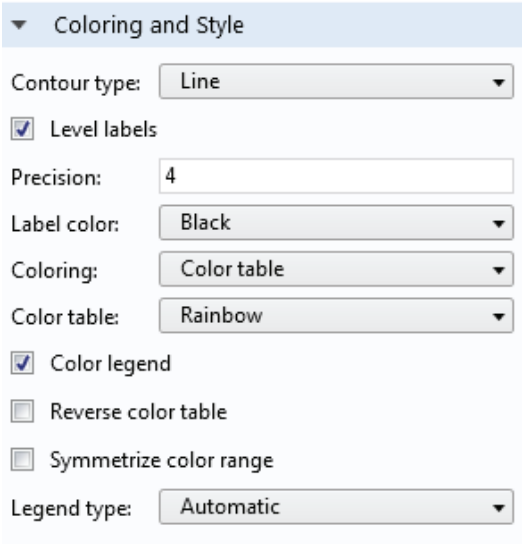
VON MISES STRESS

The plot expression *solid.mises* for von Mises stress is found under *Replace Expression > Model 1 > Component 1 > Solid Mechanics > Stress*.

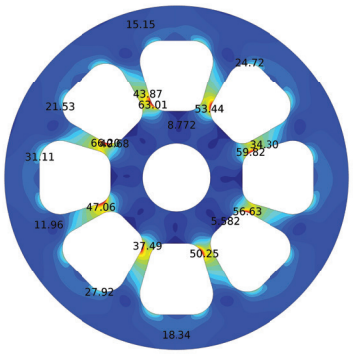
The contour colors show clear distinction between the areas of high and low stress; the contour of highest stress can be used to determine whether the pulley will fail.



To see the exact value more clearly at each contour boundary, duplicate this plot. In the new plot, set the contour type to *Lines* and check the *Level labels* box.



Then click Plot.



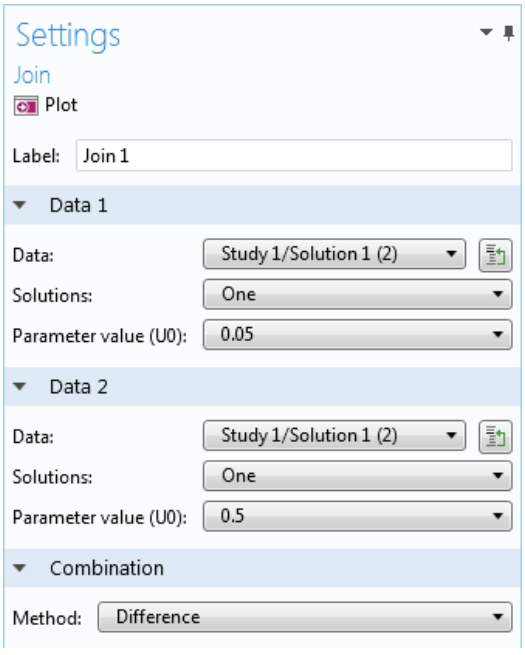
The new plot shows the value of each contour, with the highest stress appearing at the thinnest area of the joint (around 63 MPa). These labels are very helpful for an at-a-glance look at the data; if the stress is too high in one area, you'll know immediately.

JOINING

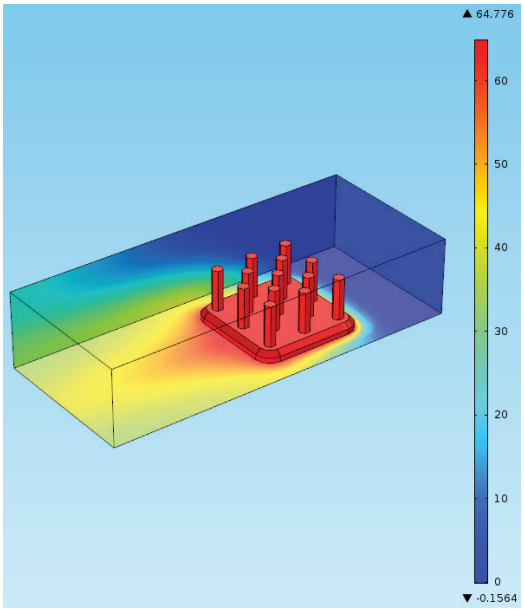
The Join function combines solutions from two data sets, allowing you to compare them. This is helpful for visualizing the difference between two solutions, creating sums that contain multiple solutions, or seeing how two different meshes affect the results.

We'll use the heat sink model again. In the model shown below, we've added a parametric sweep that evaluates the study over the mean inlet velocities ranging from 0.05 m/s to 0.5 m/s in steps of 0.05 m. (These values are purely for demonstration purposes and not necessarily realistic for most standard cooling fans.)

Join solutions can be added under the Data Sets node. We'll use the new parametric study and compare data from two different velocities. The first will be data for *U0 = 0.05* (where *U0* is the mean inlet velocity of the air blowing across the heat sink). For the second, we'll use the same data set, but with data from *U0 = 0.5*. The combination method should be set to *Difference*.

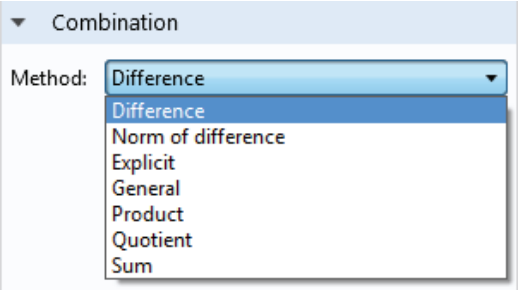


If we create a new 3D plot group using a surface based on Join 1, we'll find that the slowest air flow (*U0 = 0.05 m/s*) results in temperatures around 64 K hotter than the fastest air flow.



This means that, as expected, the fan speed has a pretty large effect on the cooling, and simulation can help in accurately quantifying such an effect. For a device like this, these results can be used to determine the optimal air flow velocity from the fan by studying the temperature results for different flow speeds.

A final note: although there are many combination methods for the Join operator, two call for a deliberate mention, *General* and *Explicit*. Most combination methods are a variation on the General combination. Like mixing ingredients, the Join solution will not recognize the individual parts once joined; these cases limit the user to the same variable for each solution. To draw a parallel to simple functions, we could perform *f(v)-g(v)* or *f(u)-g(u)*, but not *f(u)-g(v)*.



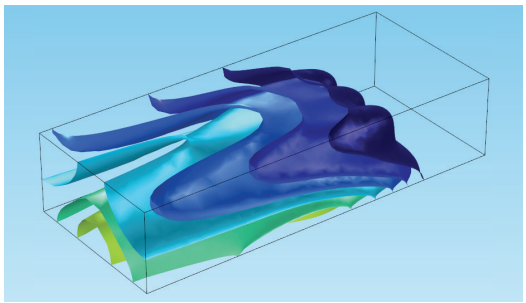
The Explicit combination method allows you to implement operations while still preserving individual data sets within the join function—they remain usable, for example, in the expression of a plot that uses a Join solution. COMSOL creates two new operators that access the quantities from the data sets; therefore a function *f(u)-g(v)* could be implemented using the Explicit method, as long as *u* is a valid variable available in *f* and *v* is available in *g*.

FILTERS

Sometimes, an entire plot area will actually block your view of the inside of a model. But COMSOL makes it easy to "peel" the plot so that you can see better. This function is called a filter and can be added to 2D and 3D plots (except the streamline type).

Simpler and less computationally heavy than adding multiple data sets with selections, filtering allows you to limit the plot expression by specifying a range. For the purposes of interpreting your results, it can be a very effective way of seeing physics effects up close and without obstruction.

In the heat sink model, the 3D plot group *Isothermal Contours (ht)* contains an isosurface plot to show the temperature. In the isosurface plot settings under the Level tab, the number of levels is set to 10.



This plot shows temperature isosurfaces. But we can't see the heat sink itself. In the image above, the color table has been modified from the default.

HINT

As demonstrated here, joining doesn't actually require two solvers—you can also use a Join data set to examine one solution evaluated at two different parameter values.

REFERENCE

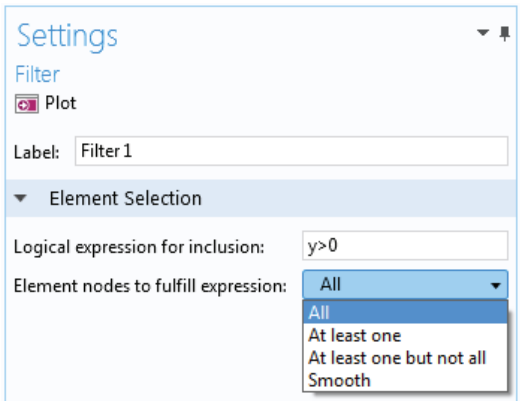
For a brief introduction to isosurfaces, see the *Plot Types* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics.

In this case, adding a solution that contains only half the geometry would require creating a cut plane and a lot of careful selecting. We'll use a filter instead.

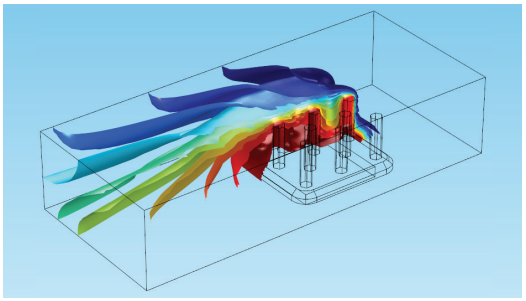
Right-click the isosurface plot node and choose *Filter*. The filter settings allow you to choose a logical expression to designate an area of the plot surface to include. For instance, setting the expression to *y>0* will exclude areas of the isosurfaces that lie in the *-y* quadrants of the coordinate system.

The *Element nodes to fulfill expression* dropdown list offers choices for which individual elements of the mesh to include. The options are:

- *Smooth,* which is the default setting and will compute the points within elements where the logical expression changes from true to false, creating a smooth transition for continuous functions
- *All,* which will include elements for which the logical expression applies to every node
- *At least one*, which will include elements for which the logical expression applies to at least one node, and may apply to every node
- *At least one but not all*, which will include elements for which the logical expression applies to at least one node but does not apply to every node

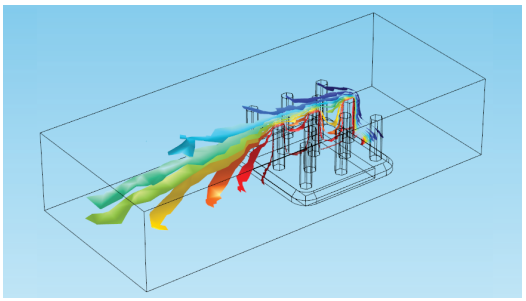


Change the expression to *All* for now, and click Plot.



We can now see the temperature surfaces everywhere that *y>0*.

Change the *Element nodes to fulfill expression* field to *At least one but not all*.



This option will plot a zone around the logical expression—in this case, plotting elements that overlap with the line *y=0* so that some nodes do not satisfy the *y>0* condition. This can be especially useful for plotting around lines or boundaries where there is a sudden change in values.

This concludes the demonstration of some different ways postprocessing can help you interpret results by isolating or selecting specific portions of a solution, evaluating variables at specific points, and joining data sets together. Read on for an introduction to using the camera and viewing tools in the COMSOL environment.

VISUALIZATION TOOLS

VIEW AND CAMERA CONTROL

The View node is one of the most powerful tools available for orienting your object in the Graphics window. Found under the Definitions node, views can help you set up the exact perspective and scene style that you want. This is where lights are added, angled, and dimmed, where color and position are arranged, and where regions of a model can be hidden or revealed. To draw a parallel to the world of theater, the View node lets you set the stage before the curtain rises and the show begins.

To reach the View node that applies to a particular plot group, click the plot group node (such as *3D Plot Group 1*) and check the Plot Settings tab. In the *View* field, you can change which view applies to the plot group, or navigate directly to its view by clicking the *Go to Source* button next to the drop-down list .

THE CAMERA AND ITS SHORTCUTS

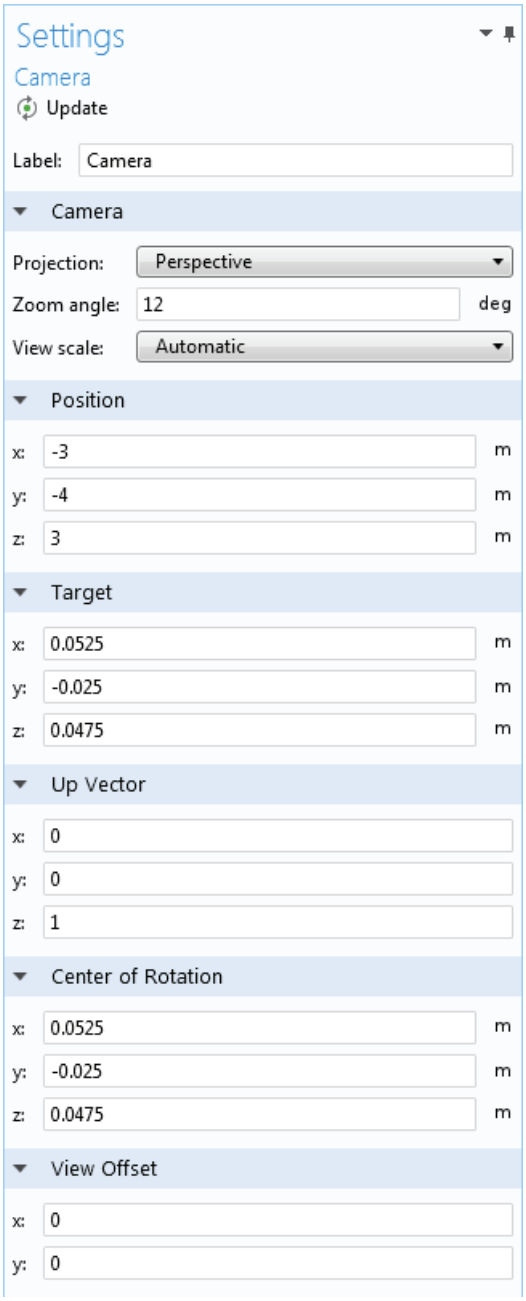
Camera controls are located under the View node. Just like in photography or film, positioning the camera correctly can help you highlight certain features to get the look you want. COMSOL provides mouse and keyboard shortcuts that control the camera, and these are some of the most efficient ways to set up a good shot.

Open up any model and click on the Camera node under View 1. Camera settings control the angle, position, zooming, and perspective of the camera. One way to imagine the camera is to picture a physical camera sitting in front of the Graphics window; when you look at the object, you're looking straight through the camera's viewfinder at the scene.

This means that when you move something in the Graphics window, you're actually moving the *camera*, not the geometry itself, unless you are performing a geometry operation like a rigid transformation.

The pages that follow show some common shortcuts used for orienting the camera.

The cards in this chapter will demonstrate shortcuts using the busbar model (*File* > *Application Libraries* > *COMSOL Multiphysics* > *Multiphysics* > *busbar*), but we encourage you to try these out on one of your own simulations. Here's what the Camera settings look like:



INFORMATION

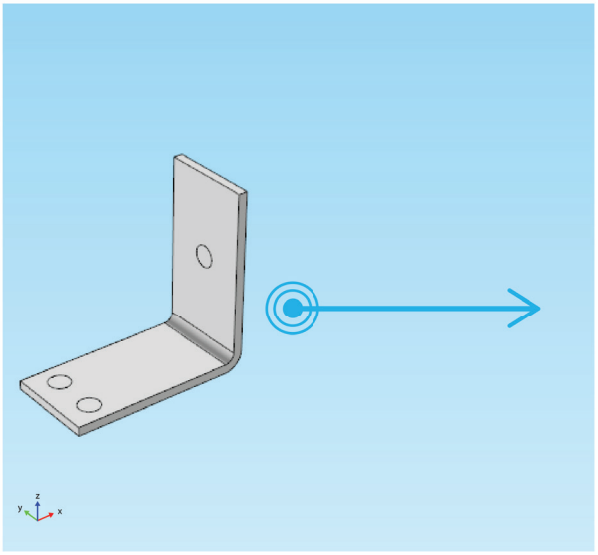
For Mac users: the third and sixth shortcuts will only work on a two-button mouse or a mouse with a center scroll wheel.

Right-click and drag. This first shortcut moves the visible frame left, right, up, or down on the camera's viewfinder plane (the plane of the Graphics window). This changes the View Offset.

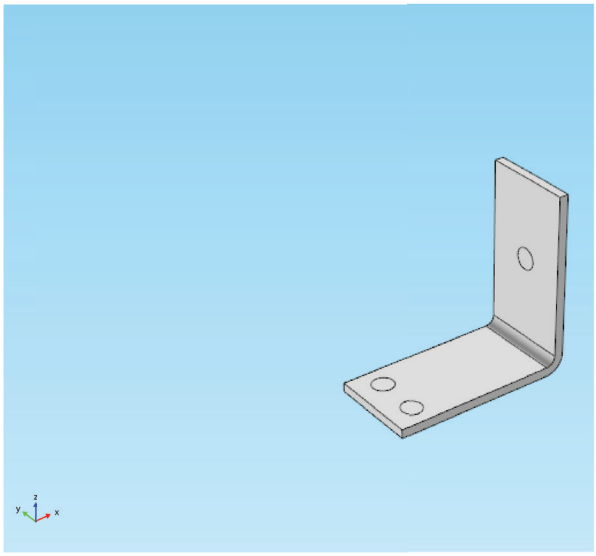
ACTION

Move the visible frame around on the plane of the Graphics window.

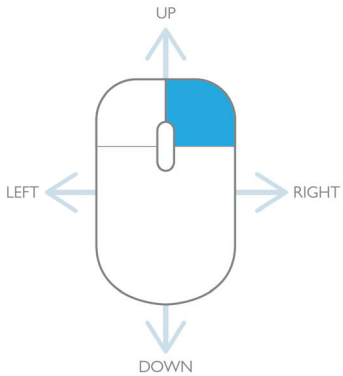
Action



Final Effect



SHORTCUT



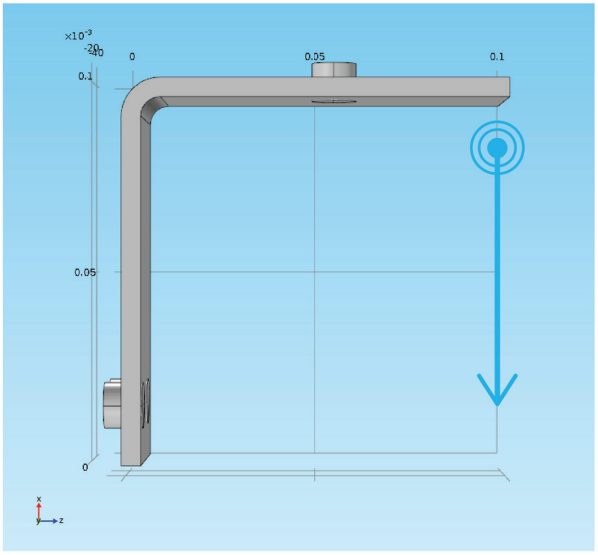
CAMERA

	Initial	Final
VIEW OFFSET		
x	-0.3	-0.1
y	-0.2	-0.1

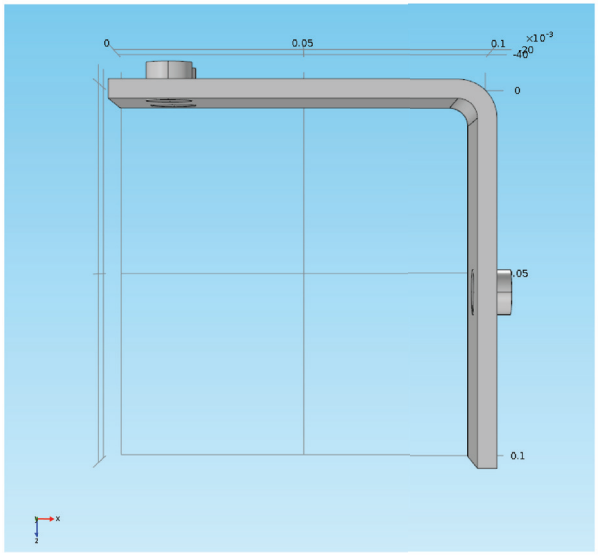
Left-click and drag while holding the Alt key. This shortcut rotates the camera around an axis perpendicular to the plane of the Graphics window:

ACTION Roll the camera.

Action



Final Effect

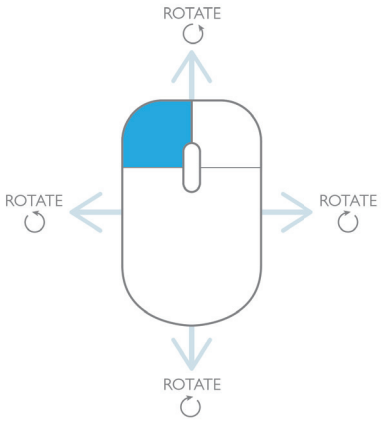


SHORTCUT

WINDOWS/LINUX

MAC

Use the Alt key



CAMERA

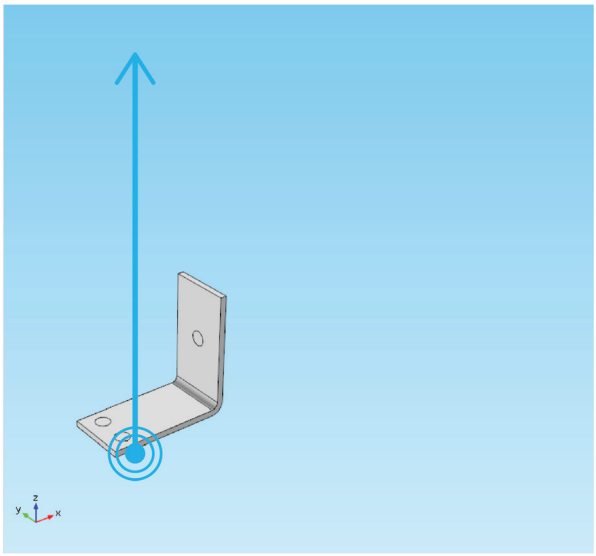
	Initial	Final
UP VECTOR		
x	1	0
y	0	0
z	0	-1

Center-click and drag. Another very simple action is zooming in on the area where the mouse is located. While the zoom buttons in the Graphics tools are quick and easy, they zoom in on the center of the visible window by specific increments and are not affected by the location of the mouse. This action gives you more precise control over the zoom angle:

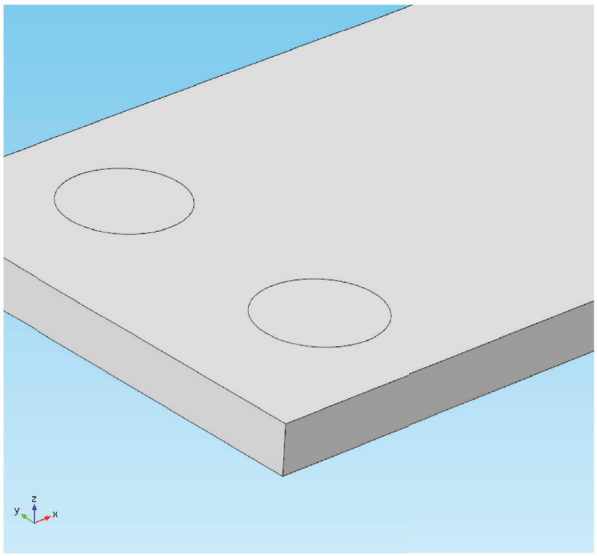
ACTION

Zoom in and out at the place where the mouse was clicked.

Action

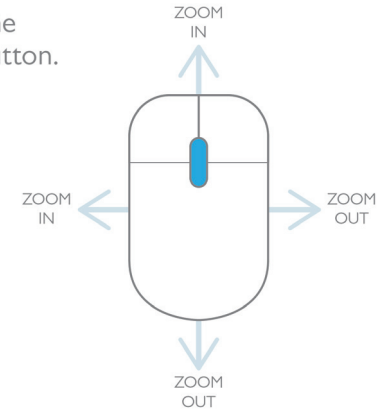


Final Effect



SHORTCUT

Click and hold the center mouse button.



CAMERA

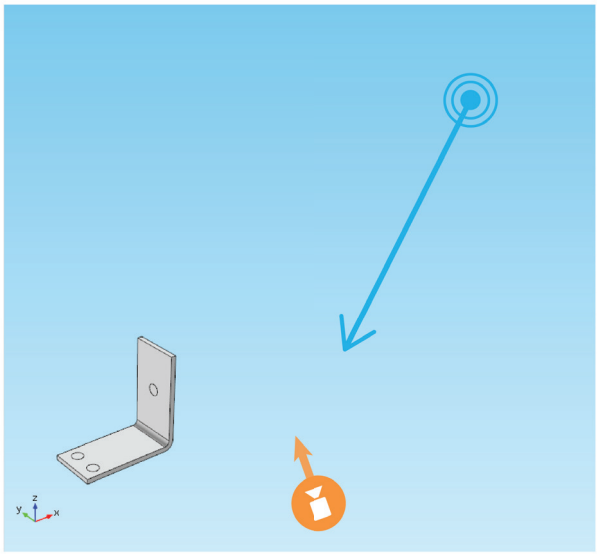
	Initial	Final
ZOOM ANGLE	42°	8.75°

Left-click and drag while holding the CTRL key. The following shortcut changes the pan and tilt of the camera. This effect changes the position and angle from which the camera sees the component geometry (in contrast to the first shortcut, where the visible frame moves within the camera's viewfinder plane). The orange icon indicates the direction of the camera. Since this shortcut moves and rotates the camera with respect to the component geometry, the model will appear to move in the opposite direction of the mouse.

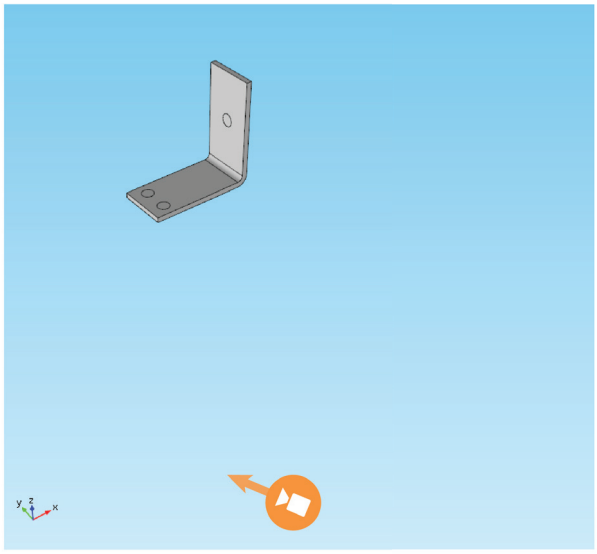
ACTION

Pan and tilt the camera without changing its location.

Action

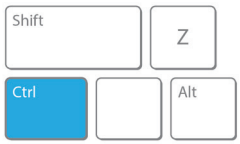


Final Effect



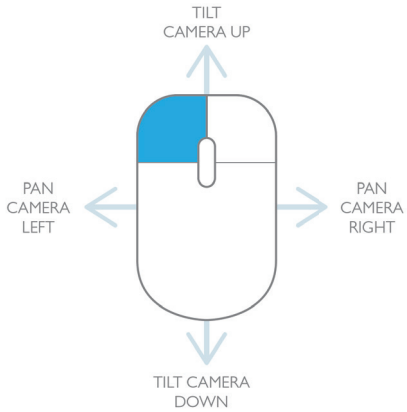
SHORTCUT

WINDOWS/LINUX



MAC

Use the Control key



CAMERA

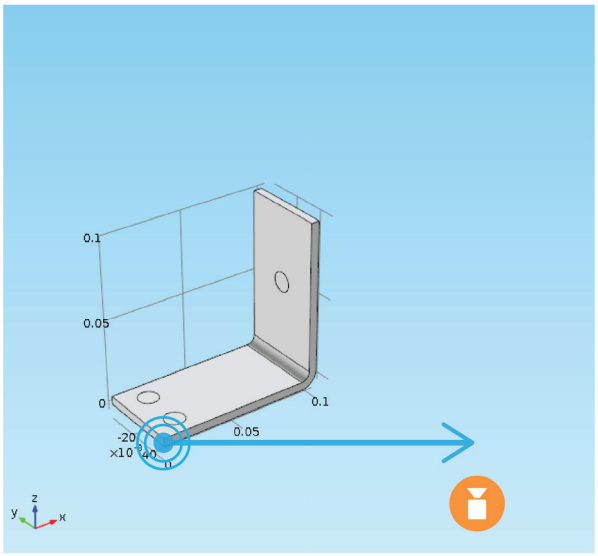
	Initial	Final
TARGET		
x	0.0525 m	0.075 m
y	-0.025 m	-0.075 m
z	0.0475 m	-0.175 m

Right-click and drag while holding the CTRL key. This shortcut moves the camera up, down, left, and right, but keeps it facing the same direction. The camera moves with respect to the component's geometry, so the geometry will appear to move in the opposite direction as the camera (but the same direction as the mouse).

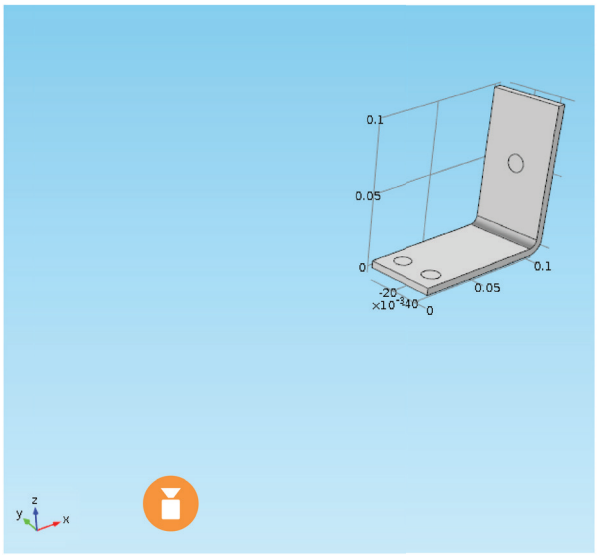
ACTION

Move the camera up/down/left/right.

Action

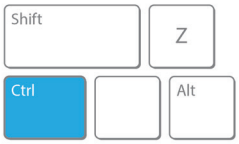


Final Effect



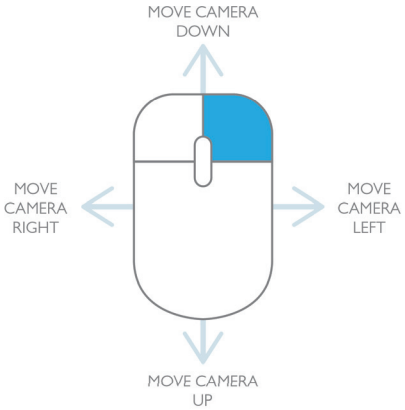
SHORTCUT

WINDOWS/LINUX



MAC

Use the Control key



CAMERA

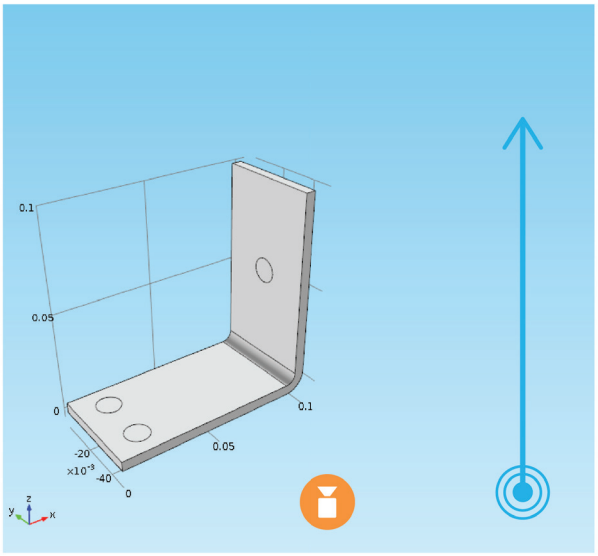
	Initial	Final
POSITION		
x	-0.35 m	-0.5 m
y	-0.6 m	-0.5 m
z	0.45 m	0.35 m
TARGET		
x	0.06 m	-0.2 m
y	-0.04 m	0.07 m
z	0.04 m	-0.06 m

Center-click and drag while holding the Alt key. This last shortcut keeps the component geometry in the center of the camera view and dollies the camera in and out (forward and back). Unlike zooming, it is not affected by the location of the mouse, nor does this action change the zoom angle of the lens.

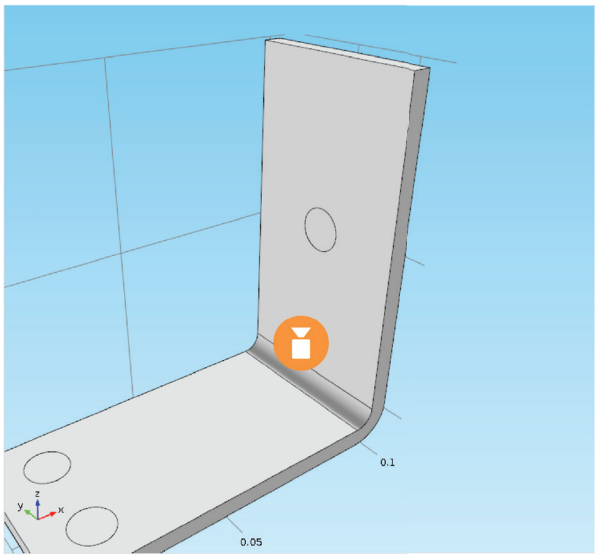
ACTION

Dolly the camera in and out around the center of the model.

Action



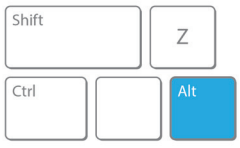
Final Effect



SHORTCUT

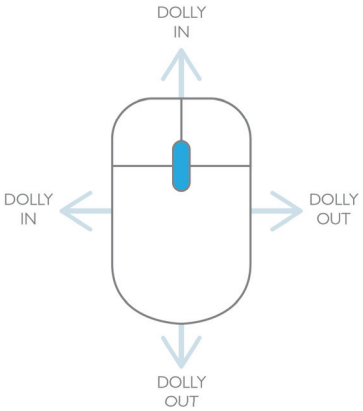
Click and hold the center mouse button.

WINDOWS/LINUX



MAC

Use the Alt key

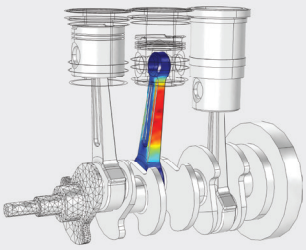


CAMERA

	Initial	Final
POSITION		
x	-0.1 m	-0.03 m
y	-0.25 m	-0.15 m
z	0.2 m	0.13 m
TARGET		
x	0.3 m	0.37 m
y	0.3 m	0.4 m
z	-0.2 m	-0.27 m

HINT

The reciprocating engine has a great geometry for experimenting with multiple surfaces, deformations, and different lighting approaches. For hints on creating the image below, turn to page 29 in the *Advanced Tips and Tricks* chapter.



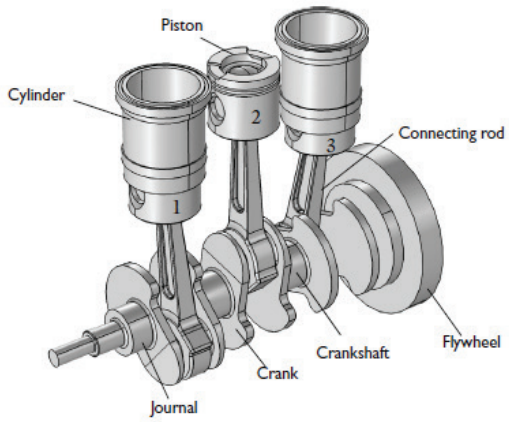
LIGHTING AND SCENE

Lighting can help to illuminate regions of a model where it's hard to tell what's going on, putting new perspective on a results plot. These finishing touches can cast a device in a whole new light.

To experiment with the lighting and scene tools in COMSOL, we'll use the Reciprocating Engine model. If you have the Multibody Dynamics Module installed, this model is found under *File > Application Libraries > Multibody Dynamics Module > Automotive and Aerospace*. (If you don't have the already-solved model in your libraries, download it from *File > Help > Update COMSOL Application Library*.) The techniques shown here are applicable to any geometry, but we'll use the engine because it has curves, nooks, and crannies that make the effects easy to see.

This example analyzes the multibody dynamics of a three-cylinder engine and the thermodynamics of the air-fuel mixture in the cylinders. The simulation results are used to determine the pressure variation in the combustion chamber, rotational speed (rpm), the motion of different components of the assembly, and the power output from the cylinders. The geometry contains a flywheel, crankshaft, journal, and crank and piston for each of the three cylinders.

There are two components in this model; we will work within the second component, called Multibody Analysis, which analyzes the multibody dynamics.

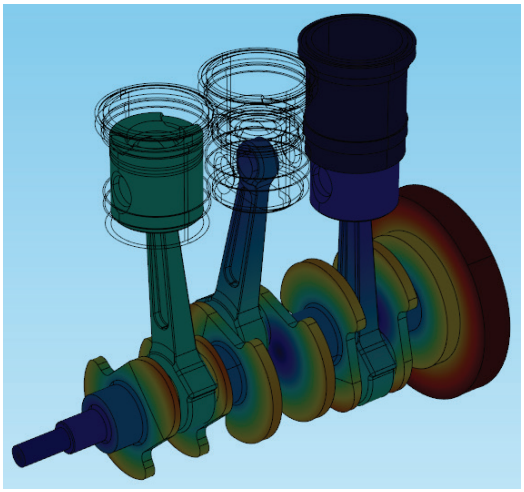


To create good lighting for this engine, we'll need to use several lights aimed in different directions.

Expand the Displacement plot group under Results. This plot shows the displacement of the different components after the engine has begun running. If you expand the Surface 1 subnode, you'll see a deformation. This deformation moves the visible geometry to the engine's position at 0.16 seconds, but keeps the outline of the original position.

Navigate to the View node for the Displacement plot group. Once you've explored the existing lights and the Hide Geometric Entities 1 node (which hides the middle engine piston), delete Directional Lights 1, 2, and 3.

If we go back to the displacement plot we'll see the dark engine with all the lights off, as shown below. In the View node, the geometry should appear in plain gray.



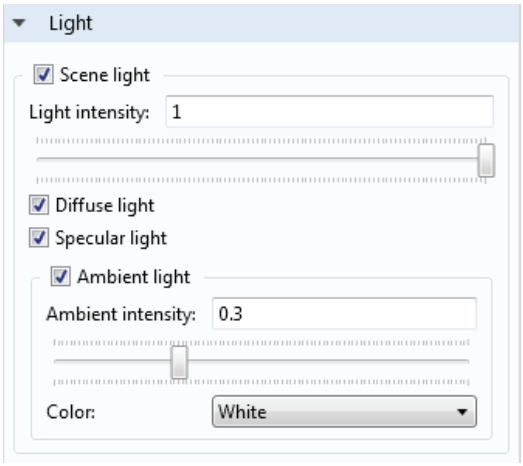
Right-click View 2 and choose *Directional Light*. The new light that appears in the Graphics window, indicated by an arrow inside of a tetrahedral box, is automatically aimed close to *(0,0,0)*.

The position of **directional lights** is defined by their direction; they behave like rays of sunlight, falling on every part of the geometry in their path, and can be blocked by walls and protrusions. Unlike spotlights (which will be

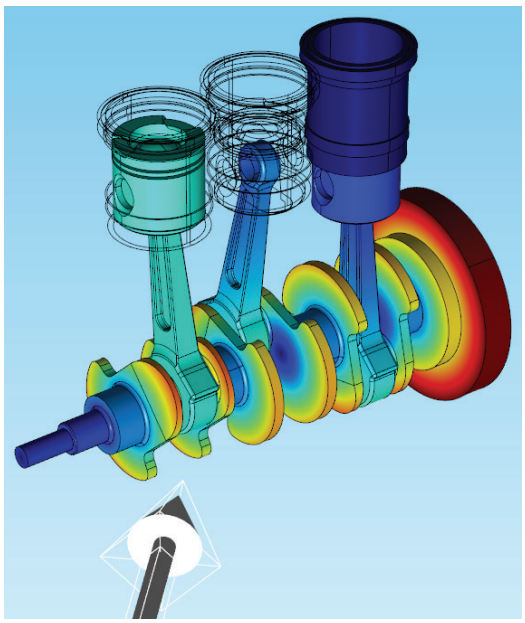
mentioned later), the position of a directional light cannot be changed.

Try playing around with some different coordinates by changing the entries under the Direction tab in the light's settings. Once you've got a feel for how the lighting changes with the coordinates, create another directional light. We'll illuminate some different areas of the engine.

We'll need one light to make a nice splash that brightens the overall scene. Later, we'll add a third light for the top and sides, and one to create emphasis on the curved surfaces.



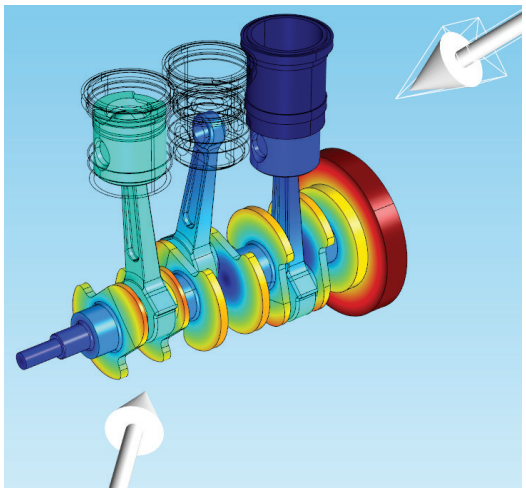
For the first light, try setting the direction coordinates to *(3,3,0)* and the light intensity to 0.75. This will illuminate most of the geometry.



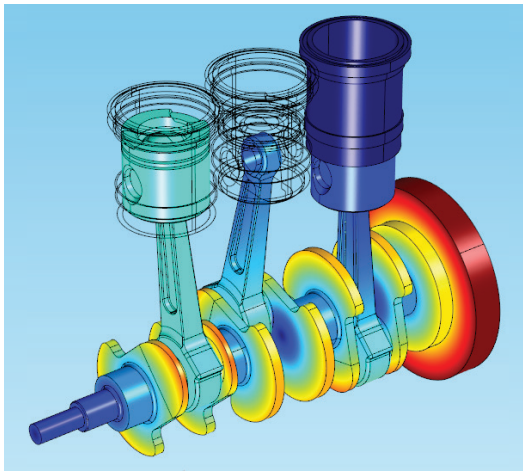
HINT

When editing lighting after setting up a model, it can be especially helpful to keep only one light enabled at a time. This lets you see exactly which effect is caused by which light as you add each one into the scene. You can also expand the Light tab in the main View node settings to change the ambient lighting and the overall scene light intensity. When you're working in the View node, arrows are displayed for each light, but only the light being edited is surrounded by a box.

For the second directional light, test what happens when you shift the coordinates—try highlighting the top, the bottom, or the side of the engine. The coordinates used to create the scene below are *(-0.5,1,-1)*.

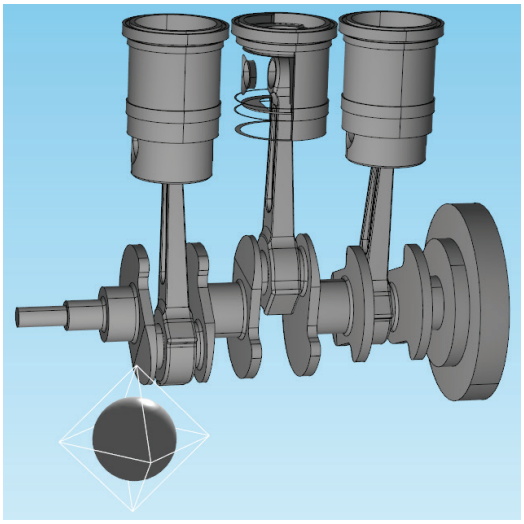


When you've found the exact position that you're happy with, create the third directional light. This is the light we'll use for the curved surfaces. The coordinates of the third light in the figure below are *(1,1,1)*, but see what you like best:

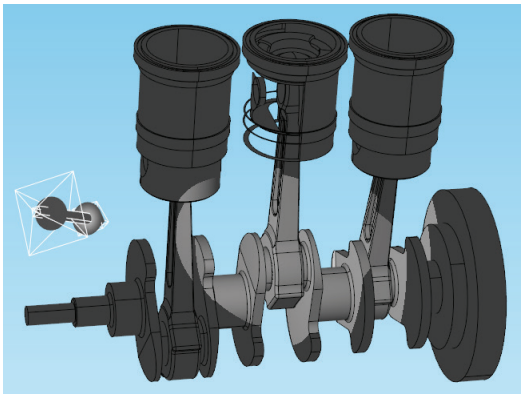


Directional lights like the ones we've added here are the most commonly used lights, but there are several other types of lights available in COMSOL. If you right-click View 2 again, you can add a point light, a spotlight, or a headlight.

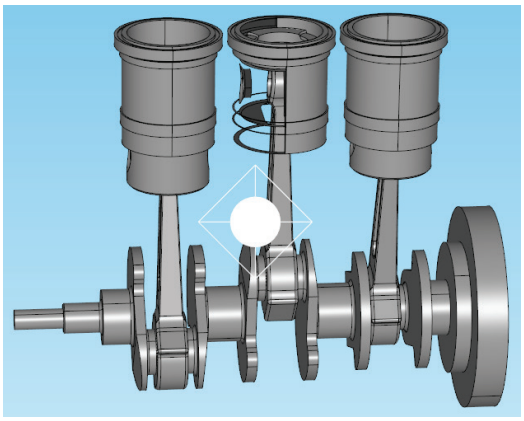
A **point light** is like a lightbulb; it emits light in all directions, and its settings control its position. This point light is positioned at *(-0.25, -0.5, -0.25)*:



A **spotlight** is like a flashlight, with both direction and position specified by the user. This spotlight is positioned at *(-0.28, -0.33, 0)* and aimed at *(1,1,-0.75)*:



A **headlight** points from the camera position; it is locked to the camera's coordinate system, and its position and direction cannot be changed.



This concludes our explanation of the camera and lighting tools available in COMSOL. With these in your hands, you'll be able to emphasize the most important region of your results plots, clearly see what's happening in your model, and produce powerful visual reports.

APPLICATION-SPECIFIC PLOT TYPES

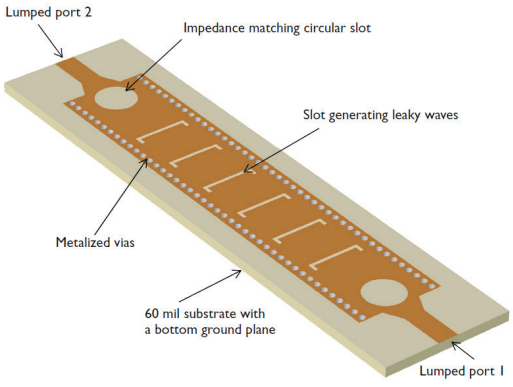
There are certain plots that make most sense to use with certain physics and for this reason are most frequently used with add-on products to the COMSOL software that are specific to the application area. Let's detour for a moment to cover a few of these application-specific plot types.

FAR-FIELD PLOTS

Radio-frequency (RF) applications deal with electromagnetic waves being radiated and absorbed at radio frequencies. Simulation can be used to design a multitude of RF devices, and one of the most common figures of merit considered by engineers is the radiation pattern. RF and microwave applications include devices such as antennas.

We'll now take a look at an example of an RF device where you might want to know its radiation pattern.

The model shown below is a substrate-integrated waveguide (SIW) used in antenna applications. Leaky waves escape through slots on the top surface to generate a beam that can be steered by changing the operating frequency.



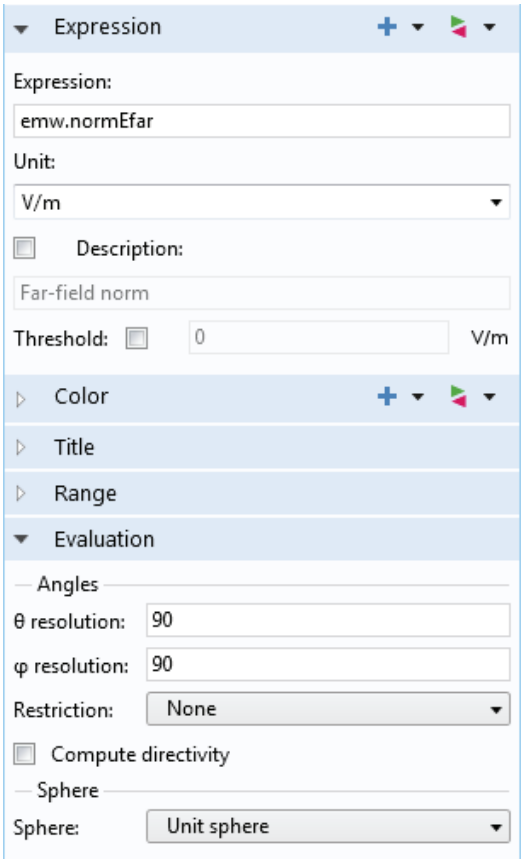
This model, available from *File* > *Application Libraries* > *RF Module* > *Antennas* if you have the RF Module installed, is one where a far-field calculation is used to determine the electromagnetic field distribution at distances far from the SIW.

Far-field plots are used to plot global variables for the far-field of electromagnetic or acoustic pressure fields, so the plot tools we'll use for the SIW are applicable to many RF and acoustics applications.

In a far-field plot, the variable is plotted for a specified range of angles (Θ and ϕ in the settings below), and the surface is created by radially deforming a specified sphere from the location of evaluation points lying on it.

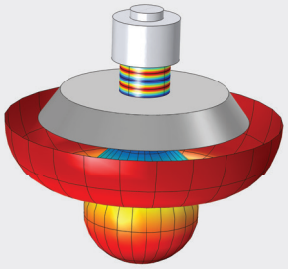
One benefit to using far-field plots is that the deformed sphere is not part of the model geometry, and so the number of directions plotted is not affected by the discretization of the solution domain. The shape of the deformed sphere depends on the specified expression for the evaluation points.

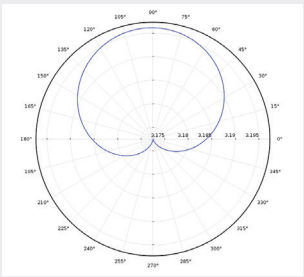
The settings for a far-field plot in the SIW model are shown below.



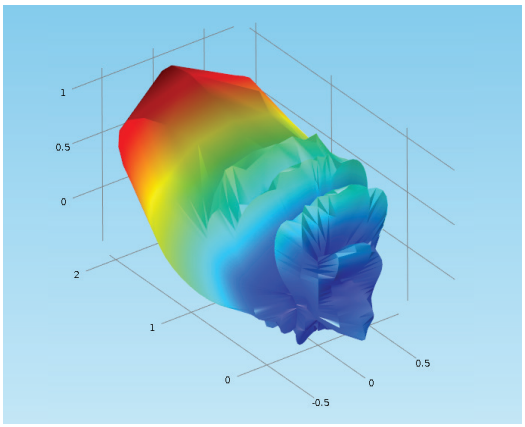
ACOUSTICS MODELING

You might use a similar process in an acoustic simulation, to visualize the acoustic pressure field at distances far away from the object you're modeling. The example below shows the far-field sound pressure levels around a tonpilz piezo transducer. Such devices are used, for example, in sonar systems where high-power, low-frequency sound generation is needed.

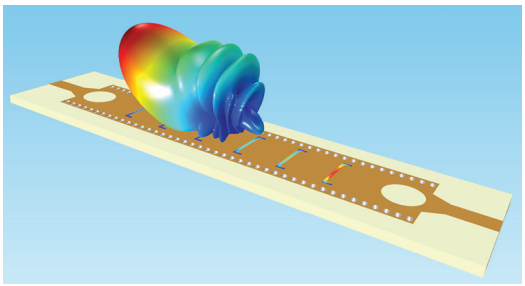




For this SIW, we can plot the far-field norm variable, *emw.normEfar*. The plot below is scaled up to make details more visible and depicts the far-field norm of the electric field for an operating frequency of 12.25 GHz.



Increasing the angle resolutions will give a smoother and clearer depiction of the far-field. In the image below, a rigid translation is used to move the waveguide down so that the entire far-field radiation pattern is visible.



The maximum resolution for Θ and ϕ is 180 (shown above). As might be expected, the higher the resolution, the more computationally heavy the postprocessing operations.

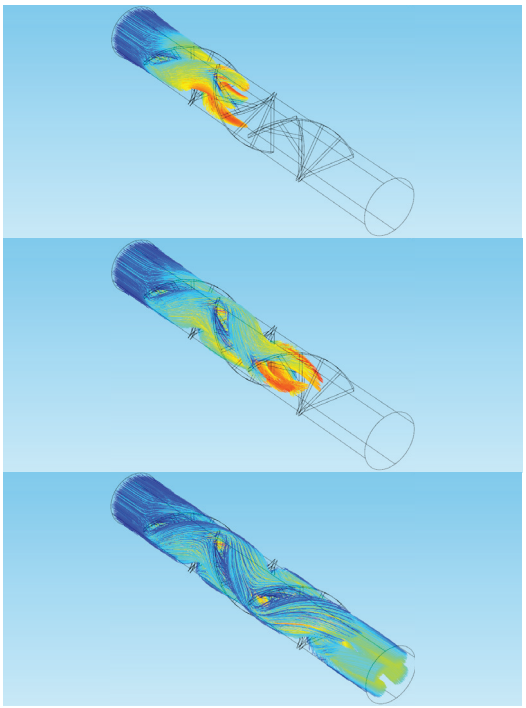
PARTICLE TRACING

The Particle Tracing Module is used to solve a set of ordinary differential equations for the motion of distinct particles. As the particles move, they can be affected by external fields through a wide variety of forces, such as electric, magnetic, drag, and dielectrophoretic forces. There are many options for including particle-boundary, particle-particle, and particle-field interactions. The physics interfaces are usable for a range of simulation topics, such as ion transport in electromagnetic fields, large-scale planetary movement, and separation of biological cells in small microfluidic systems. We'll now demonstrate some options for postprocessing using the Particle Tracing Module in a simulation of a static mixer.

If you have the Particle Tracing Module installed, open the tutorial from Application Libraries > Particle Tracing Module > Fluid Flow > laminar mixer particle. The geometry is a flow channel containing a series of static blades that are twisted in alternating directions. In this example, the mixing performance is evaluated by computing the trajectories of solid particles carried by water through the mixing blades.

We will start from some of the plots that are included in this tutorial. Click on the 3D plot group named Particle Trajectories (fpt). This plot displays the particle trajectories up to a selected time, with a color expression indicating the shear rate at the particles' current positions.

This plot can be adjusted easily to visualize the particles' progression at different times. The sequence below shows trajectories for three different times, t=1 s (left), t=1.6 s (center), and t=3 s (right), with the color expression changed to spf.U to indicate the magnitude of the particle velocities. To control the color range manually, expand the Range tab in the settings for the Color Expression 1 subnode.

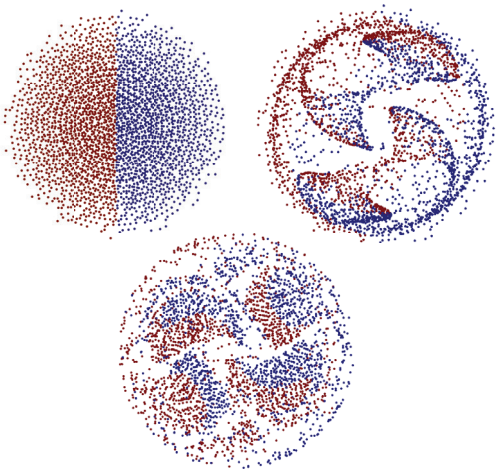


The Particle Trajectories plot provides a more realistic depiction of the mixing performance than a streamline plot would. Streamline plots are used to visualize vector quantities using curves tangent to an instantaneous vector field. (See the Plot Types chapter of the Essentials of Postprocessing and Visualization in COMSOL Multiphysics handbook for an introduction to streamline plots.) The Particle Tracing for Fluid Flow physics interface, on the other hand, treats the particles as distinct entities instead of a continuous field, and solves a distinct set of ordinary differential equations for each particle so that it correctly accounts for the inertia of the particles.

Another useful way to visualize how particles are mixed is to use a Phase Portrait plot. In a Phase Portrait, each particle is depicted as a colored dot at some location in a user-defined phase space. For example, the two coordinate axes can be two different components of the particle position; alternatively you can plot particle position along one axis and particle speed along the other.

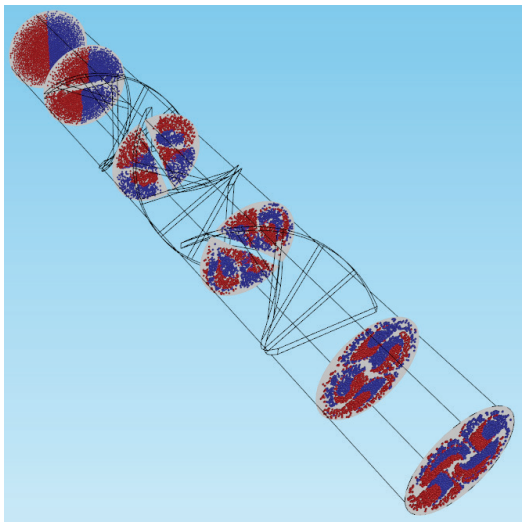
Click on 2D Plot Group 5, which includes a plot called Phase Portrait 1. In this example, the two coordinate axes of the phase portrait are the two transverse components of the particle position.

The time value can be adjusted in the Data tab of the Phase Portrait settings window. Shown below are phase portraits at t=0 s (left), t=1 s (center), and t=4.2 s (right). A Color Expression indicates the initial position of each particle. At the final time, some pockets consisting entirely of red or blue particles can still be seen, indicating that the mixing performance could still be improved.



The main drawback of the phase portrait is that it plots all particle positions in phase space at the same time, not at the same position along the length of the mixer. A better way to evaluate the mixing performance at specific locations in the mixer geometry is to use a Poincaré Map. This type of plot shows the locations where particles intersect with one or more surfaces defined using a Cut Plane data set.

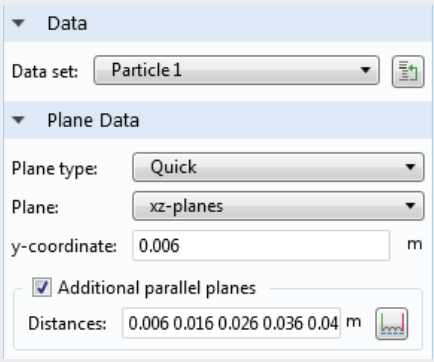
Click on 3D Plot Group 4. In this plot we can see the particle groups on different planes in space after mixing has begun, and can tell that they are not fully mixed by the time they reach the outlet. This Poincaré Map uses planes located at 0.006, 0.016, 0.026, 0.036, and 0.042 meters along the y-axis.



This concludes the application-specific plot section of this handbook.

CUT PLANES

For plots on cut planes, we use Cut Plane data sets, which are created on 3D data sets and can be visualized in either 2D or 3D plot groups. A cut plane can be used to create many parallel planes. Use the *Additional parallel planes* checkbox and choose an appropriate distance or range to specify the offsets of extra planes, as shown in the image to the right.



ADVANCED TIPS AND TRICKS

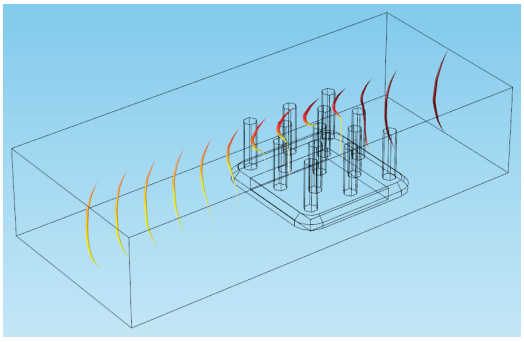
IDEAS FOR POWERFUL VISUALIZATION

This section is meant to help you brainstorm by showcasing different ways to create some unique graphics. We will show examples of a few techniques that can help you create new perspectives on your results, such as deformations, cut planes, different plot combinations, and advanced camera control.

CUT PLANES AND CUT LINES

In Essentials of Postprocessing and Visualization in COMSOL Multiphysics, we briefly touched on using cut planes (and analogously, cut lines) to visualize results on planes (or lines) that cut through a model geometry. COMSOL offers the ability to combine these, which can considerably improve and simplify a results plot.

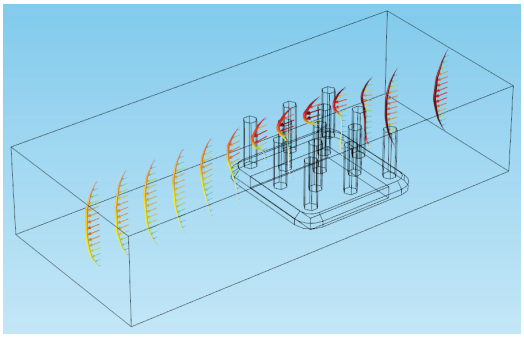
Last time, we saw a line plot that ran through the heat sink model along its *x*-axis, showing the air temperature and flow velocity in the channel. The line plot shown below is deformed according to the velocity field *(u,v,w)*, whose magnitude also controls the radius expression for the tubes.



REFERENCE

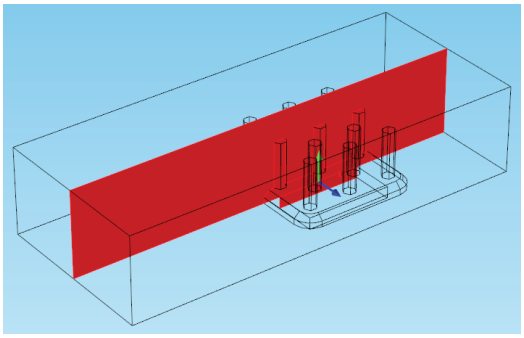
See the *Data Sets, Derived Values, and Tables* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics for an introduction to cut points and, by extension, cut lines and cut planes.

We can also add an arrow plot alongside the deformed lines, to show the velocity field. The arrow color in the following plot is based on the velocity magnitude, *spf.U*.



But it may be hard to see what's going on right around the heat sink pillars because there are so many lines in the way. Let's create some new data sets to show a sliced view—a cross-section of the heat sink and air domain so that the region nearest the heat sink is easier to see.

Right-click the Data Sets node and add a cut plane. Under the Plane Data tab, set the *Plane* field to *xz*-planes. Click Plot. Based on the default coordinates, this will create a plane running down the center of the air domain and the heat sink.

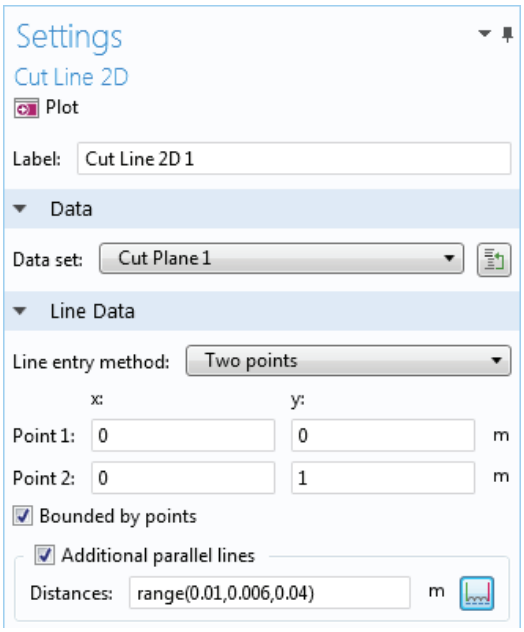


Now right-click the Data Sets node and choose *Cut Line 2D*. For the data set, choose the cut plane you just created (Cut Plane 1). Set the location of Point 1 to *(0,0)* and the location of Point 2 to *(0,1)*. Then check the *Additional parallel lines* box.

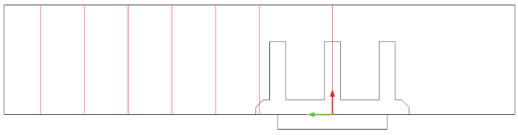
We'll add a range to this, so that we create a series of vertical cut lines in the new plane. Click the Range button next to the *Distances* field to open a dialogue box for the range. Enter the following settings:

- Start: 0.01
- Step: 0.006 (this will create a line every 6 millimeters from the start point)
- Stop: 0.04

Click Add, and the range will be added to the *Distances* field.



Click Plot to show the cut lines on the slice created by the cut plane.



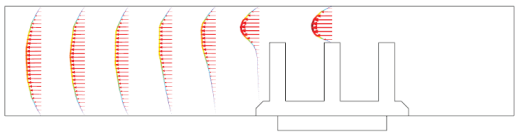
Now add a new 2D plot group to the Results node using Cut Plane 1 as the data set. Add both a line plot and an arrow line plot.

For the **line plot** use the Cut Line 2D 1 data set (the cut line solution you just created) and the plot expression *spf.U* (the velocity magnitude). Set the line type to *Tube* and set the tube radius expression to *spf.U* as well.

Add a deformation to this line plot, with the *x*-component set to *u* and the *y*-component set to *v* (the components of the velocity field).

For the **arrow line plot**, set the data set to Cut Line 2D 1. Enter the expression for the velocity field and set the scale factor to 0.025.

Click Plot. Now we have a 2D cross-sectional view equivalent to the 3D plot we saw earlier.



Although the 3D plot is more useful for visualizing certain aspects of the solution, this plot makes it much easier to see how the air flow develops on a specified plane as the air moves past the heat sink.

USING DEFORMATIONS TO STACK YOUR RESULTS

One way to use deformations that we haven't touched on yet is to compare results within the same plot. This is a great way to see different quantities at the same time. This involves using several different plots showing multiple quantities.

We'll demonstrate using a 3D example. If you have the Heat Transfer Module installed, open the heating circuit model under *File > Application Libraries > Heat Transfer Module > Power Electronics and Electronic Cooling*.

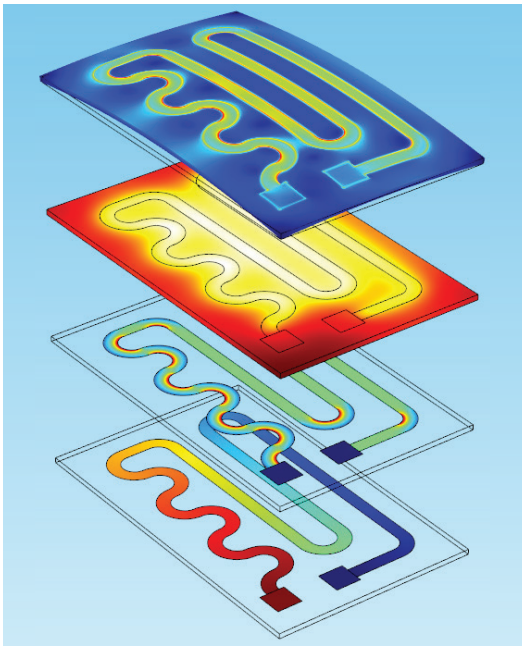
Heating circuits find their way into many applications, especially in manufacturing. This particular device includes an electrically resistive layer on a glass plate, which experiences Joule heating when a voltage is applied. The solved model contains results plots showing stress, isothermal contours, temperature, electric potential, and surface losses in the circuit.

Suppose we want to visualize many of these at once, to take the bulk of our results and array them all together. We can use the existing plot groups to create our stack.

ARROW PLOTS

In the same way that an arrow surface plot uses arrows on a planar surface to depict a vector quantity, an arrow line plot depicts a quantity using arrows on lines.

The new plot group will contain surfaces from the Stress (solid), Temperature (ht), Electric Potential (ecs), and Surface Losses plot groups, spaced 50 mm apart along the *z*-axis. In the image below, the Surface Losses plot group has been modified to include only the heating element.



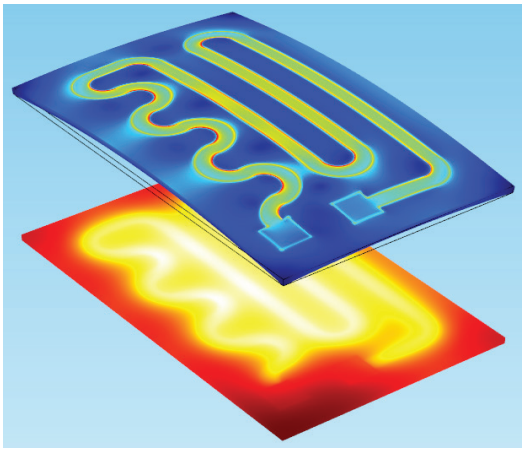
To start, expand the Stress (solid) node and take a look at the existing deformation—it bends the geometry according to the displacement field.

Create a new 3D plot group. Right-click the surface plot in Stress (solid) and choose *Copy*. (The deformation will also be copied, since it is a subnode of the surface.) Right-click your new plot group and choose *Paste Surface*.

Also copy the surface under the Temperature (ht) node. After you've pasted it into your plot group, do a few things before adding a new deformation: under the Title tab set the title type to *None*, and under the Coloring and Style tab uncheck the *Color Legend* box. This will ensure that when you duplicate this surface, you won't end up with multiple titles and color legends, which can really crowd up the Graphics window.

Now add a deformation to the temperature plot. In the image shown below it is shifted down by -50 mm in the *z*-direction. Set the scale factor to 1 and the *x*- and *y*-components to zero.

The array will initially look blurry:



We'll add a line plot to the temperature surface to distinguish the different areas of the device. Go to the Data Sets node and duplicate Study 1/ Solution 1 (1). This will create a data set called Study 1/Solution 1 (3). Add a selection that includes all of the plot edges (choose *Edge* for the geometric entity level, then select *All edges* in the selection field).

Now add a line plot to your 3D plot group and choose Study 1/Solution 1 (3) as the data set. Set the expression to 1 and the coloring to uniform black.

You won't see anything change. This is because your line plot doesn't contain the deformation that you added to the temperature plot, so it's located right on top of the lines for the stress surface.

Copy the deformation node from your temperature surface and paste it into the line plot to shift the plot edges. Click Plot.

There are several ways to finish the array, but it is often easier to duplicate the surfaces that already exist in a plot group than copy and paste everything from the other plot groups. Create two more surface plots: one showing the electric potential and another showing the surface losses. The following table includes the expressions and deformations we've used for each, but adjust the deformation settings to your liking:

Plot	Expression	Deformation
Surface Losses	ecs.Qsh	-100
Electric Potential	V	-150

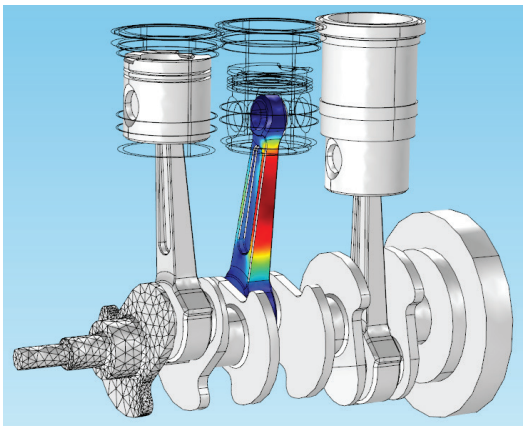
TRANSLATIONS

You can only add a single deformation to each surface plot. Therefore, the plot containing the deformation that shows displacement in the circuit cannot also be translated in the *z*-direction. Each of the other surfaces uses a deformation to translate them in the *z*-direction.

COMBINING PLOT TYPES

In Essentials of Postprocessing and Visualization in COMSOL Multiphysics, we explored examples of plotting several surfaces on a model at the same time. For powerful visuals, sometimes you might want three, four, five surfaces on your geometry. How do you do this without overcrowding the look and feel of the graphic?

The image of the reciprocating engine that we saw in the *Visualization Tools* chapter is a combination of surface plots, line plots, deformations, wireframe rendering, and different color styles.



The colored surface in this image is limited to just one of the engine's connecting rods. Most remaining regions of the engine are simple gray, with one area meshed.

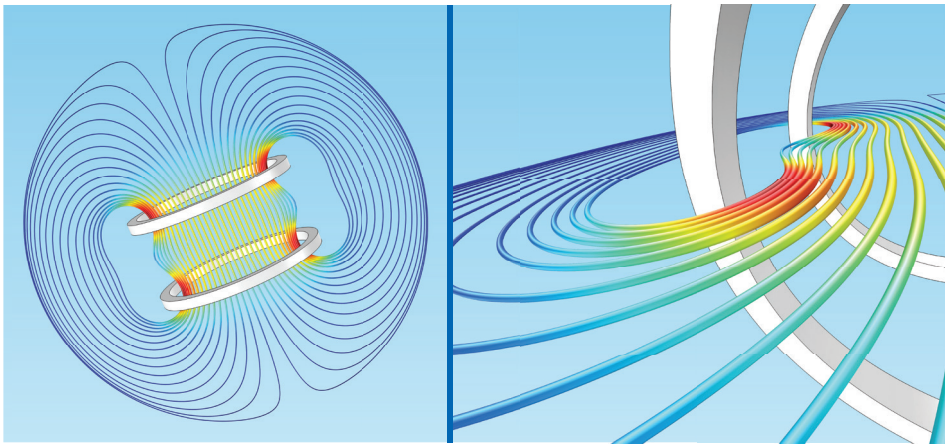
This results plot contains:

- A surface plot with a geometric selection that excludes the middle piston and cylinder and the left cylinder. The coloring is a uniform light metallic gray. This surface has a deformation of *(u, v, w)*, which is repeated for most of the following plots to show the position at *t=0.16* seconds.
- A surface plot based on a data set containing only the center connecting rod. This surface plots the von Mises stress.
- A line plot showing all of the data set edges. This plot does not contain the deformation, so it shows the outline of the engine's starting position.
- A surface plot based on a data set containing only the leftmost crank, journal, and end of the connecting rod, with wireframe rendering turned on and the color set to black to show the edges of the mesh elements.

EXAMPLES OF ADVANCED CAMERA CONTROL

One trick that the camera can help you with is showing interesting perspectives on a model. Whether you're looking for a close-up of the mesh, to see how different surfaces intersect, or to examine a boundary in detail, these shots can give a new perspective to your visualization and let you look at the model from the inside of the geometry.

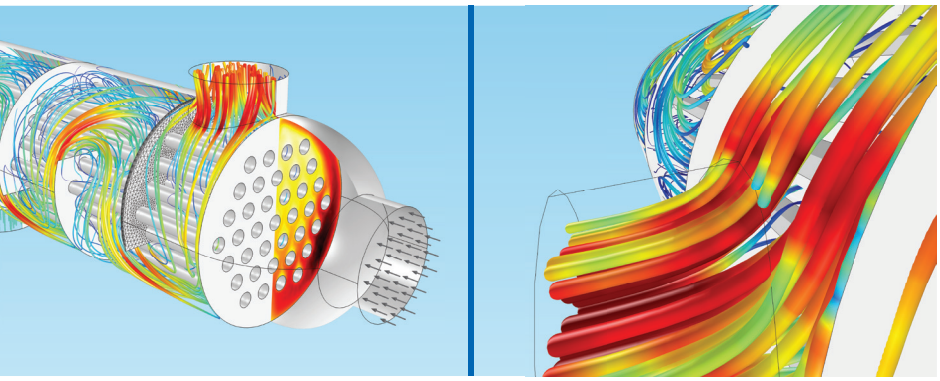
Created through various combinations of rotations and zooming, the following examples show some different views.



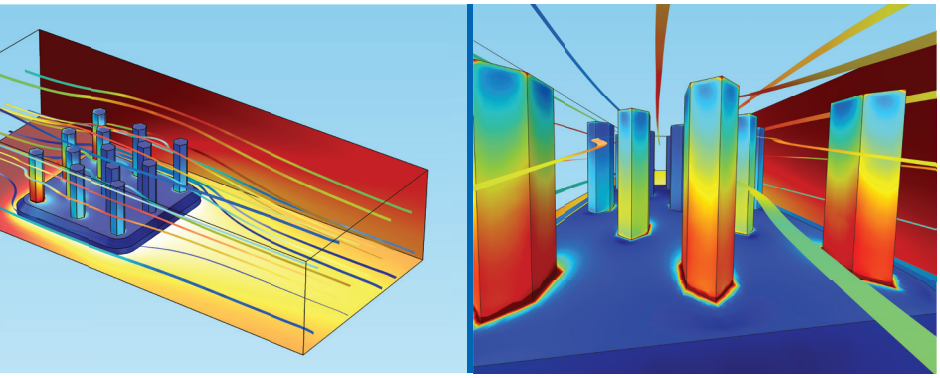
The images above depict a simulation of a Helmholtz coil with streamlines showing the magnetic field. These effects were created using a combination of panning and tilting, dollying, zooming, and moving the camera up, down, left, and right (camera shortcuts 3-6 in the *Visualization Tools* chapter).

WIREFRAME RENDERING

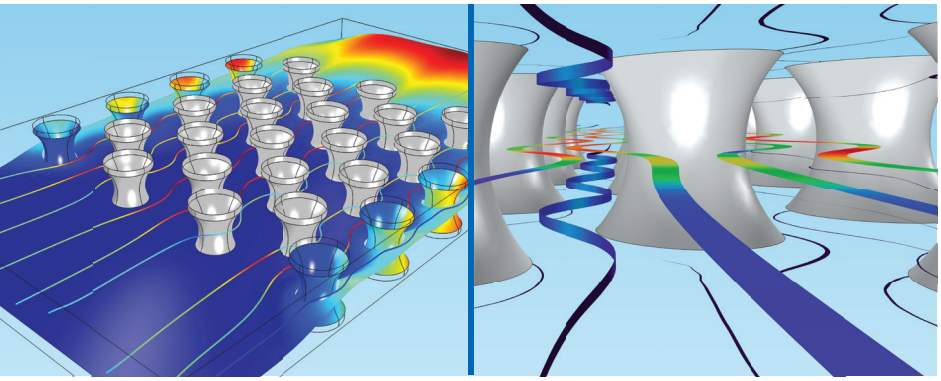
To show mesh elements in a plot, add a new surface to a plot group and set the color to uniform black. In the surface settings, check the *Wireframe* box under the Coloring and Style tab. This is usually used in combination with a surface of another color, so that the mesh is overlaid on top of the first surface.



Above is a shell-and-tube heat exchanger with fluid flow streamlines. This plot includes streamline, surface, arrow, and wireframe mesh plots. The perspective on the right was created using combinations of rotating and zooming movements.



Above is a view looking through the columns of the heat sink plot containing surfaces, and streamlines. The perspective required moving the camera down to the height of the base of the pillars, dollying, and using center-click zooming.



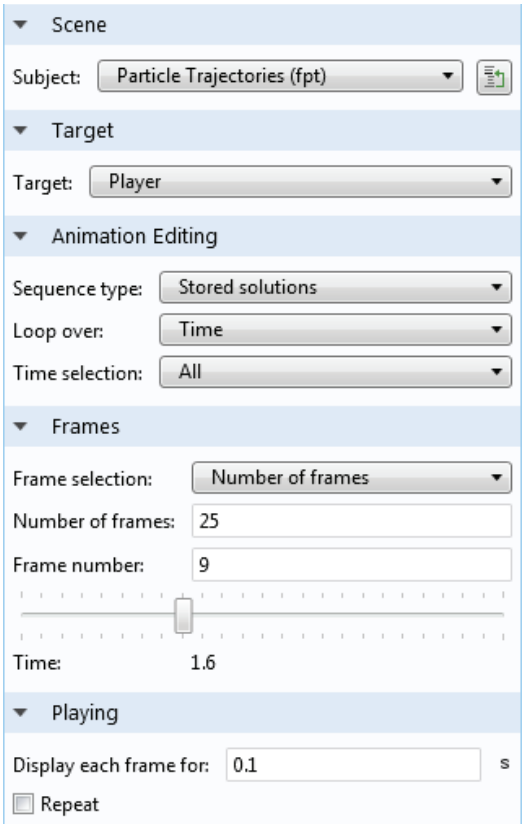
Shown above are reacting pillars in a biosensor where fluid velocity is depicted. This plot required shifting the camera angle, zooming, and dollying.

ANIMATIONS

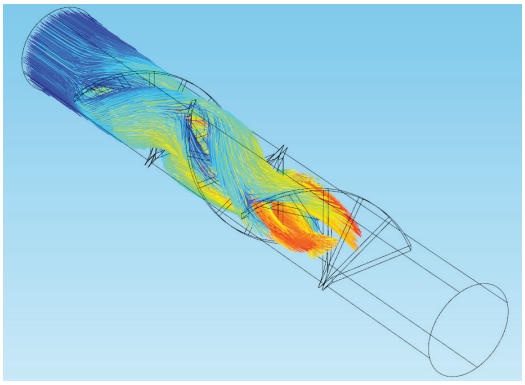
One of the final capabilities of COMSOL that we'll touch on here is exporting results. This can take the form of plots, tables, animations, and reports. Some techniques for exporting data and generating results are discussed in the *Exporting Results* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics. In this section, we will focus on animations.

We'll use the laminar mixer model, but you can do this with many different applications featuring a time-dependent study. Right-click the Export node under the Results node, below all of your plot groups, and choose *Animation > Player*. The Player creates interactive animations directly in the COMSOL Desktop® environment. Under the Scene tab, set the subject to *Particle Trajectories (fpt)*.

When you click the Play button that appears in the Graphics window toolbar, you should see the particle trajectory lines appear at the inlet and trace across the mixer until they reach the outlet. In the Player settings, you can also control the number of frames and the speed of the animation.



Here's an in-progress shot:

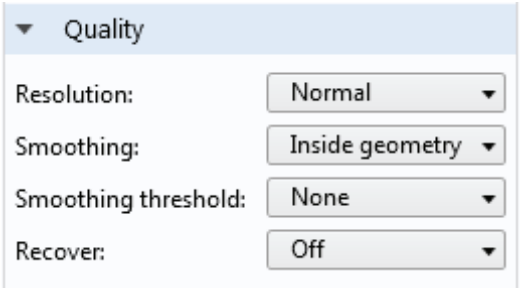


Right-click the Export node again and choose *Animation* > *File*. This is where you can export a video or series of images to save as a file—great for presentations or websites where you want to show your results in just a few seconds.

Set the subject to *Particle Trajectories (fpt)* again, choose a name in the *Filename* field under the Output tab, and change the frame and layout settings to your liking. Then click Export at the top of the settings window to make a video. If you open the new file in a browser or player that supports GIFs, you'll be able to watch it.

RENDERING AND SMOOTHING

If you click a node for almost any of the individual plot types we've used in this guide, you'll see a Quality section toward the bottom of the Settings window, underneath the Coloring and Style tab. The Quality tab lets you select a plot resolution, enforce continuity with smoothing operations, and specify the use of accurate derivative recovery.

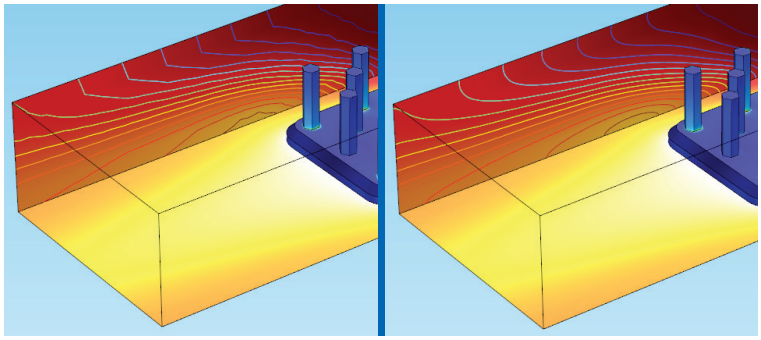


REFINEMENT

When we explored the heat sink model in the Essentials of Postprocessing and Visualization in COMSOL Multiphysics handbook, we added a contour plot showing the temperature change on the back wall of the air domain. That demonstration used a fine mesh to achieve that level of smoothness, but also included a certain amount of quality refinement.

By default, the resolution settings in a plot are set to *Normal*. A resolution higher than *No refinement*, which is an option in the *Resolution* drop-down list, means that elements can be split into smaller patches during rendering.

Take a look at the difference between the contours with no refinement (left) and high refinement (right), below:



Increasing the resolution works well when a finer mesh is not needed to improve the accuracy of a solution that is expected to be continuous. With higher refinement, the field being plotted is rendered sampling a larger number of points within each element, which improves visualization for smooth fields.

For cases such as this conjugate heat transfer example, high refinement is a good choice; refining the visualization of the continuous temperature distribution doesn't affect the correct rendering of the results.

If, however, the quantity plotted has a strong variation or even a discontinuity within an element, the refinement process is likely to produce overshoots or oscillations. Hence, it is important to examine the rendering of your solution as you increase the refinement, and to find a setting that enables better visualization without altering the behavior of the quantity you're investigating.

For example, many structural mechanics analyses require examining stress levels with high accuracy; in such cases, refinement could

RECOVERY

Recovery is a polynomial-preserving process that recovers fields with derivatives (such as fluxes or stresses) of known quantities, and has a higher theoretical convergence than smoothing. For more information about the Recovery settings, see the *Results and Analysis* chapter of the COMSOL Multiphysics Reference Manual available with your COMSOL documentation.

actually render the data incorrectly and might disguise important information, such as an area where stress or strain in a mechanical component passes the yield limit or changes abruptly.

Refinement may also be unsuitable when a plotted expression contains a combination of different fields. The fields may have different sources and different representations, and a high-order sampling may cause artificial oscillations. If you see continuities where you would expect discontinuities that you want to study—such as when rendering a phase transition occurring in a material—this means that the visualization is hiding important information. In such cases, set the refinement to *None*.

SMOOTHING

The options in the Smoothing dropdown list enforce continuity on discontinuous data—like refinement, this is helpful when you know that your solution is smooth but due to a chosen discretization, mesh settings, or modeling approach, simulation results appear discontinuous.

The default option smooths the quantity everywhere except across borders between domains, where there is often a sharp transition from one material to another, or between different types of physics.

There are five choices in the Smoothing dropdown list:

- *None*: plots the results within each individual element
- *Inside geometry domains*: the default option, this smooths within each domain in the geometry, but not across boundaries between domains with different settings
- *Inside material domains*: smooths within domains that share the same material, but not between different materials
- *Everywhere*: smooths over the entire geometry
- *Expression*: smooths in regions where a specified expression is continuous; this is particularly helpful when the expression is defining some material property

There are also options for a Smoothing threshold where you may specify *None* (the default), or *Manual*. The *Manual* option allows you to enter a number between zero and 1 to specify the maximum "jump" that can be smoothed over. In places where a parameter has a relative growth rate between elements that is higher than the given threshold, the software will not smooth between those elements in order to preserve the values or discontinuities that may be important. A higher number for the threshold will results in smoothing in more places.

LOWER-DIMENSION POSTPROCESSING

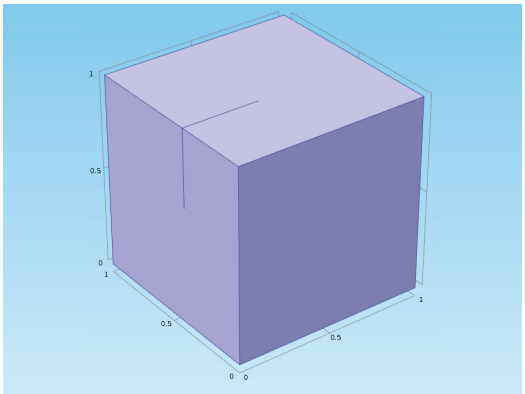
When postprocessing a simulation that contains thin walls modeled as surfaces or layers with an assigned mathematical thickness (rather than a geometric thickness), you will occasionally see different results when plotting on different types of entities. This means it's important to be careful in cases that we refer to as *lower-dimension postprocessing*.

TERMINOLOGY REFRESHER

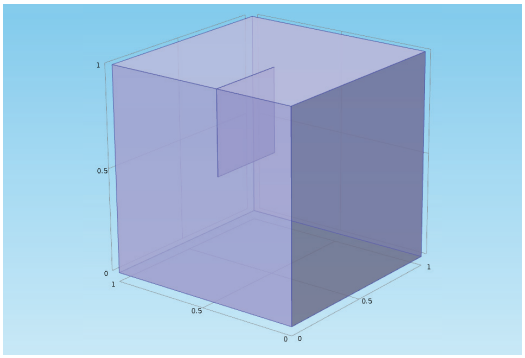
First let's do a quick refresher of some terminology. The COMSOL software uses the following conventions when referring to geometric entities:

- *Edges* refer to curves (including straight lines) in the geometry.
- *Points* refer to the intersections of two or more edges, with one point created for every intersection. *Cut points*, i.e. points not originally present in the geometry, can also be created for postprocessing purposes.
- *Boundaries* refer to surfaces that are completely enclosed by edges. While these surfaces often enclose entire domains, internal surfaces are also referred to as boundaries.
- *Domains* refer to volumes enclosed by boundaries.

For instance, in the cube geometry below, all of the external walls are boundaries:



However, the internal surface jutting out from two of the walls is also considered a boundary even though it does not act as a divider between domains.



When performing postprocessing operations under the Results node, the following facts apply:

- *Volume plots* will plot results on domains
- *Surface plots* will plot results on boundaries
- *Line plots* will plot results on edges

With certain plot types it may be wise to take note of which type of entity they plot on. For instance, arrow plots come in three styles—line, surface, and volume—with each corresponding to the appropriate geometric entity. Streamline plots may apply to both 3D and 2D plot groups, but they plot results on domains in 3D and on boundaries in 2D. Meanwhile, mesh plots are very general (applicable to any plot group except polar plots), while 1D plots contain options for plot types that are not applicable to other dimensions.

This terminology will be particularly relevant to this final section of the handbook.

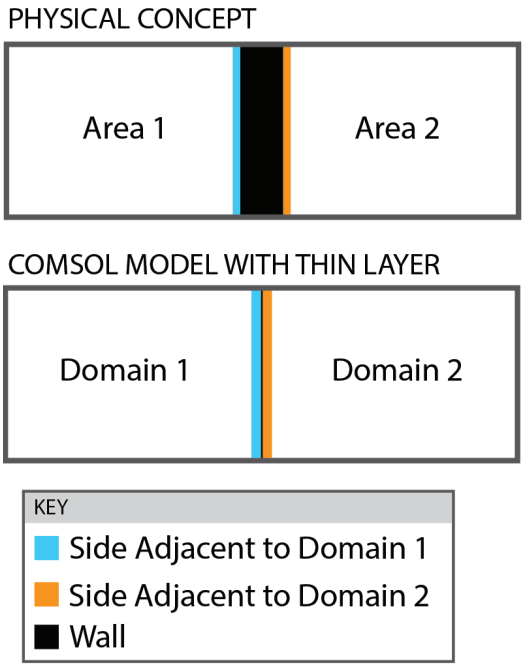
THIN LAYER ATTRIBUTES

In the COMSOL software, a "thin layer" refers to a surface where physics will be solved by mathematically accounting for the real thickness of a wall instead of meshing it. These layers are treated as boundaries, while the surrounding area or areas are treated as domains.

The benefit of using thin layer modeling—especially in multiphysics scenarios—is that it reduces computational time by circumventing the need to mesh a thin 3D layer.

Lower-dimension postprocessing is a powerful tool that enables you to analyze results on one or both sides of a thin layer and is used for modeling electrical, mechanical, fluid, and chemical applications. In this section, we'll discuss different operations that can be used in COMSOL to correctly postprocess such scenarios and avoid falling into traps.

In modeling a three-dimensional wall as a two-dimensional layer, COMSOL treats the thin layer as a boundary that is adjacent to as many domains as touch the wall.



Such a boundary touched by two domains can have results plotted for either domain, or a function of results from both.

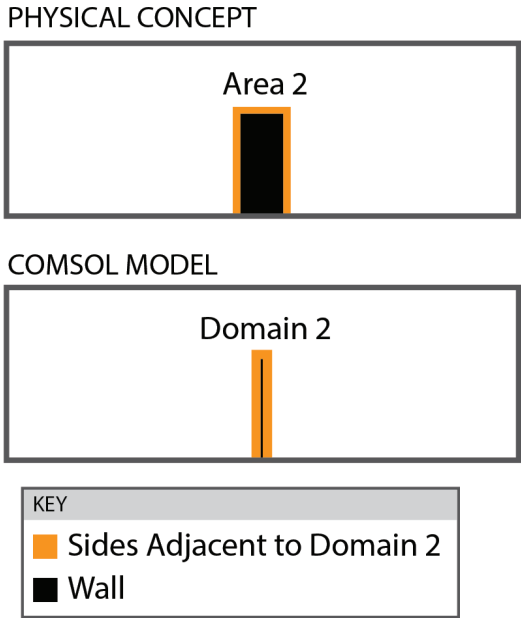
When plotting domain quantities for a thin layer, COMSOL treats the results slightly differently depending on the type of plot being used:

- When using a surface plot to represent results on a boundary, the software will usually display the average of the results from both domains using information from the solver about the boundary layer close to the wall, if any.
- When using a domain plot that falls on the location of the boundary, the software chooses results from one domain to display based on the location of the plot, i.e., which domain it falls inside of.

The latter are treated differently because the domain plot—for example, a 3D slice plot—will by definition fall on either side of the thin layer's geometry. Even plots located "exactly" on the interface between two domains will fall on one side, since plotting on the exact intersection is not numerically possible. This means that domain plots will show values from one domain or the other, but not a combination.

The implications of this functionality are that results may look slightly different depending on the plot type used.

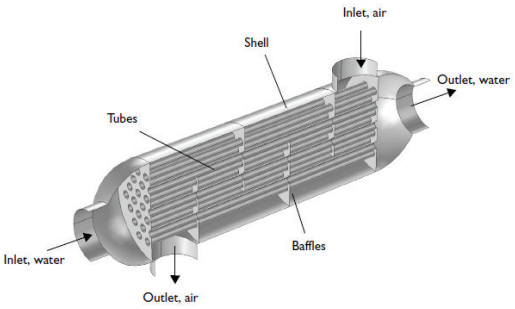
In the case of boundaries that do not divide a domain, a boundary can be adjacent to the same domain on both sides:



EXAMPLE: HEAT EXCHANGER

Let's look at a model of a heat exchanger that contains walls defined as thin layers and interior walls for thermal and fluid problems. The shell and tube heat exchanger can be found under *File > Application Libraries > Heat Transfer Module > Heat Exchangers* if you have the Heat Transfer Module installed.

The geometry consists of a series of tubes through which water flows, and a shell section where air circulates. They have different inlet temperatures, but as they flow past each other the two fluids exchange heat and reach an equilibrium temperature. This type of heat exchanger is often used in chemical processes and other industrial applications.

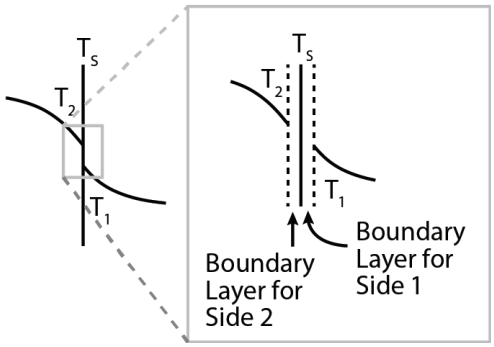


The physics setup in the Model Builder contains physics interfaces for heat transfer and turbulent flow, and a multiphysics coupling for non-isothermal flow.

In the heat transfer problem, a thin layer condition is used to define the thermal and geometric properties of every wall, where a thickness of 5 mm is assigned mathematically and the walls are considered thermally conductive.

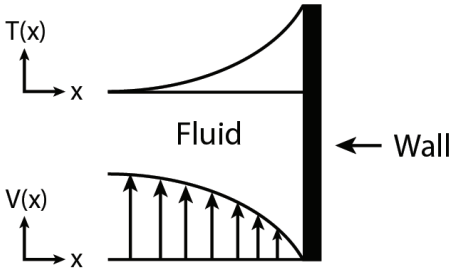
For walls adjacent to two domains, the software will solve within each domain; within the boundary layers close to the wall; and at the interface between the domains and the boundary layers.

The temperatures in each domain are represented by T1 and T2 in the image below.



Boundary layers, where there is a high temperature gradient, are accounted for mathematically using thermal wall functions.

BOUNDARY LAYER



The expression for the wall's surface temperature is *ht.Twall*. Temperature distribution from one side of the wall to the other is considered constant since in this application the wall is considered conductive and so thin that the gradient is assumed to be negligible compared to the gradient in the surrounding domain.

A thin layer boundary condition called an "interior wall" is used when fluids are involved. This makes it possible to model a thin region with large gradients in the results near a solid wall, across which the flow properties can change dramatically.

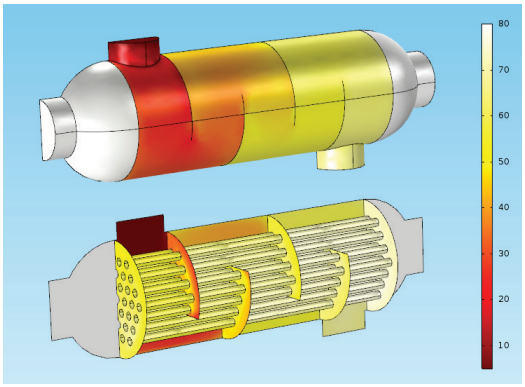
In this model, an interior wall is applied to the turbulent flow settings in order to allow discontinuities in velocity, pressure, and turbulence variables across the thin layer. The software will solve for the fluid velocity within each domain and use a wall function to account for the boundary layer forming close to the interior wall, as it did in the thermal problem.

REFERENCE

For further information about the temperature variables and boundary calculations, see the Boundary Wall Temperature section in the Heat Transfer Modeling chapter of the Heat Transfer Module User's Guide available with your COMSOL documentation.

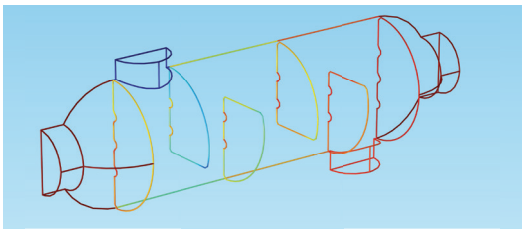
DISCONTINUITIES

We can easily see apparent discontinuities if we plot the temperature on all the boundaries of the heat exchanger's geometry, which is a default plot in the solved model. The temperature appears to change abruptly at the interfaces between certain walls, and the inlet and outlet areas where the hot water enters appear to have noticeably higher temperatures than the adjacent boundaries near the airflow.

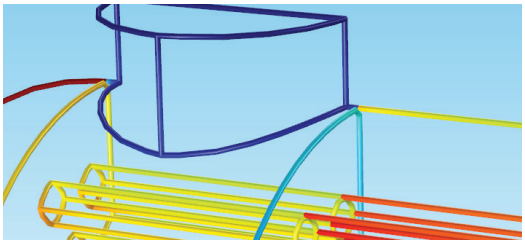


Such discontinuities are expected because there is no physical thickness in the geometry of the conductive wall bridging the temperature difference from one side to the other, and therefore no visible gradient when plotting on the boundaries.

To see similar discontinuities occurring on adjacent domains, we can create a 3D line plot showing the temperature along the edges of the geometry:



At the intersection of certain edges, it is clear that different temperatures are being plotted. If we zoom in at certain locations, it is easy to see the distinct change from edge to edge:



THE UP AND DOWN OPERATORS

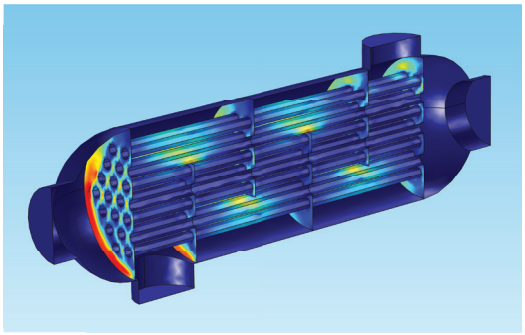
As mentioned previously COMSOL can evaluate quantities on one side of a geometric boundary. For thin layers, you may sometimes want to explicitly plot results from one domain or the other depending on which side you want to render.

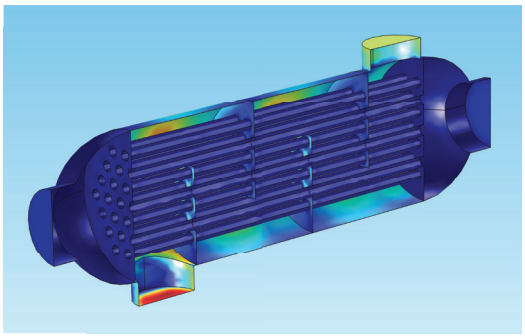
When using a boundary plot (such as a surface plot) rather than a domain plot, the up and down operators are helpful here.

These expressions take the form of *up(expression)* and *down(expression)*: *up(T)*, for example, will plot the temperature from the boundary layer on the "upside" of the wall.

As long as a boundary has an upside, *up(expression)* can be evaluated. If there is no upside—for instance, as in a case where the wall touches only one domain—the operator returns a value of zero.

Suppose we want to plot the fluid flow velocity, which will be different on each side of a wall. The expressions *up(spf.U)* and *down(spf.U)* will plot on the upside and the downside of a wall, respectively. Plots of *up(spf.U)* (top) and *down(spf.U)* (bottom) are shown below:





The up and down operators are available on geometric entities of dimension one lower than the model itself. For instance, in a three-dimensional model, *up(expression)* and *down(expression)* can only be plotted on two-dimensional entities—boundaries.

This means, for instance, that COMSOL will plot *up(spf.U)* on a surface plot, but will return an error if the expression is attempted on a domain plot such as a 3D slice plot, which has the same dimension as the model.

THE SIDE OPERATOR

Another expression that may be used in a similar fashion to the up and down evaluations is the side operator. The side operator is more flexible and precise, and can be applied to entities many dimensions lower than the model. For example, you can extract values from two adjacent domains for a point that lies at the intersection of four domains. This is helpful, for instance, if you want to integrate surface currents on a boundary in 3D.

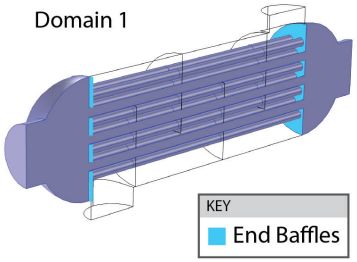
Variations on *side(m, expression)* will render results from an entity of dimension *n* adjacent to domain *m*, plotting on an entity of dimension *n-1*. Hence, at times using the side operator effectively may require knowing the entity labels of the domains. For instance, in the case where a boundary is adjacent to domain 2 but not domain 1, the quantity *side(2,expression)* will plot results in domain 2 near the selected boundary, while *side(1,expression)* will return an error.

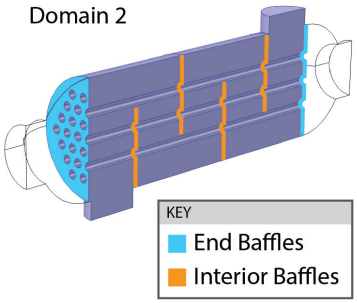
For example, creating a surface plot to render *side(1,T)* on a boundary will plot the temperature from the side of the boundary adjacent to domain 1. If the boundary is not adjacent to domain 1, *side(1,T)* will return an error.

For a surface adjacent to the domain on both sides, plots operate the same way; however, the operation will then compute the average results from the two physical sides—the upside and downside.

We can demonstrate this using plots of the end baffles in the shell and tube heat exchanger. The end baffles touch both domain 1 and domain 2.

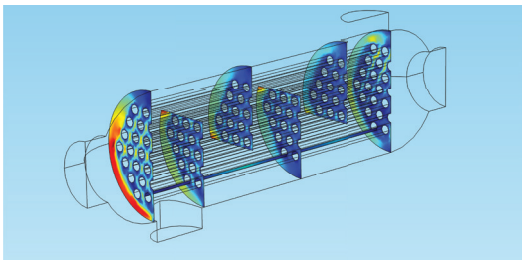
The interior baffles, on the other hand, face domain 2 on both sides:



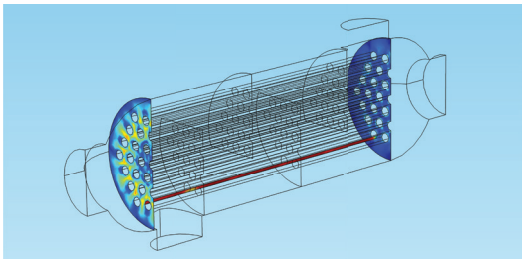


Something interesting happens if the selection in the data set includes a combination of boundaries that are adjacent to different domains. If we try to plot the fluid velocity on every baffle using *side(2, spf.U)* on a selection that contains all six baffles, we'll see all of the baffles as expected.

Below are the results for the velocity on the air momentum boundary layer:

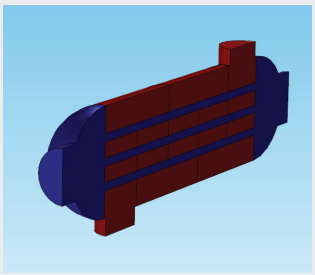


However, if we try to plot *side(1, spf.U)*, we will see only the baffles that are actually adjacent to domain 1. Below is a plot showing velocity for the water momentum boundary layer using *side(1, spf.U)*:



ENTITY INDICES

COMSOL can also plot entity indices such as domains. In the figure below, the expression *dom* has been plotted on a 3D volume, showing the two different domains in red and blue.

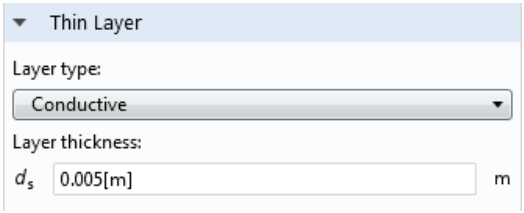


Since the interior baffles are adjacent only to domain 2, they disappear in this plot. The outer baffles are the only ones adjacent to both domain 1 and domain 2, and so they are the only ones with data available for domain 1.

You may notice a similar effect if you plot variations on *side(m,expression)* on 3D line plots containing only certain selections of the geometry's edges—depending on which results you plot on which edges, certain lines may disappear, or you may see an error message when you try to plot on edges that do not have data for that domain.

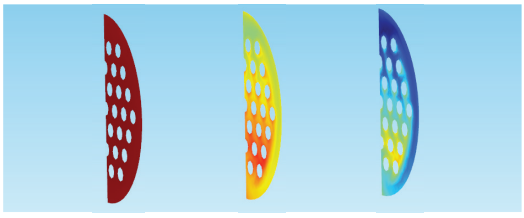
We might also expect to see differences between the temperature results on two sides of a boundary, depending on what's happening on each side. For a very conductive layer—as in the case of these walls—the gradient would be almost nonexistent since the layer is thin and the heat can diffuse quickly. For a layer made of a resistive material, however, there will be a noticeable difference in the temperatures from one side to the other.

These boundary conditions are chosen in the Settings window for the Thin Layer node:

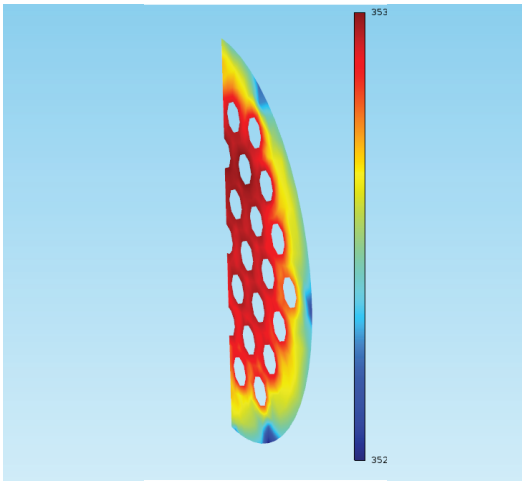


EXAMPLE: BAFFLE PLOTS

The image below depicts the temperature on one of the end baffles—boundary 334, if you're following along in the model—using surface plots showing the temperature of the side adjacent to domain 1 (left); the average, plotted using the expression *T* (center); and the temperature of the side adjacent to domain 2 (right). Style inheritance ensures that continuity in the color range is forced across all of the plots. (Note that the wall temperature, *ht.Twall* is not plotted here.)



But we have to be careful here. The plot showing the temperature from domain 1 appears to be monochromatic—this is not an error, but occurs because the color is saturated with values very close together, while the other plots contain values that are much farther apart. If we disable the other plots to view the left plot alone (showing results from domain 1), we can see clearly that there is a gradient:

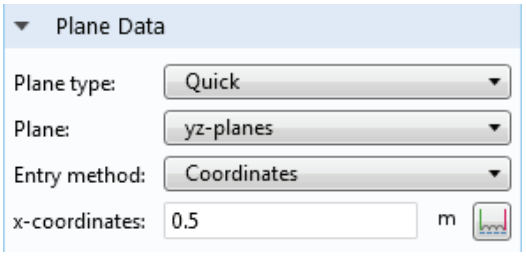


COMPARING DOMAIN AND BOUNDARY PLOTS

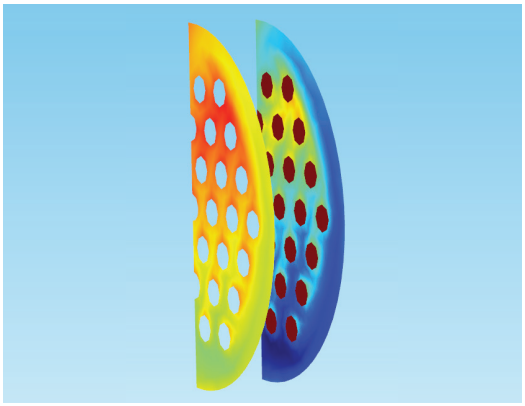
The *side*, *up*, and *down* operators are all powerful expressions that, when used correctly, can reveal important physical behavior in a system. We've discussed the differences between these two methods; now we'll briefly compare them side by side.

Let's return to the surface plot of a single baffle where *T* is plotted to show the temperature average of both sides.

Suppose we add a *yz*-slice plot—note that a slice plot is a 3D domain plot—located at the *x*-coordinate of the baffle, *x=0.5*.

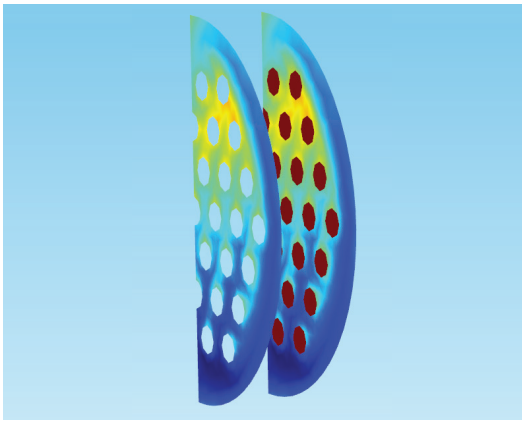


As we discussed before, the software still has to choose one domain to plot in. Using style inheritance will once again force a common color range between the plots, so that we can see if the slice plot looks different than the surface plot:



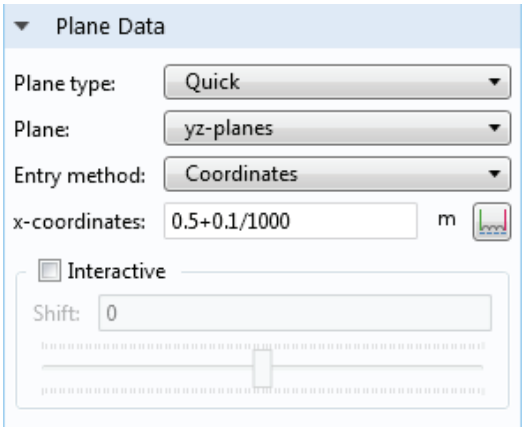
It does; the domain plot is displaying results from one side, while the surface plot is showing an average.

We cannot use *up* and *down* on the slice plot since it's a domain plot; however, if we change the expression in the surface plot from *T* to *side(2,T)*, we'll see results very similar to those shown in the slice plot, but without the tube cross-sections:

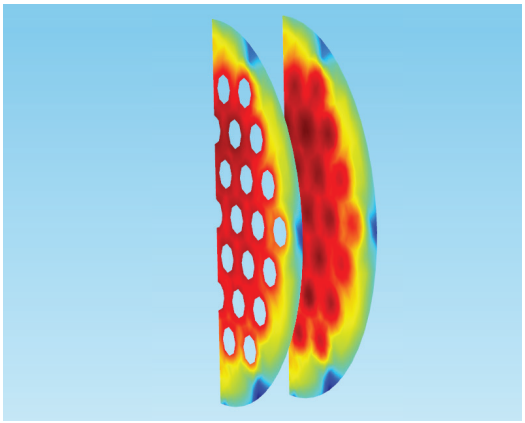


This tells us that the slice plot is showing results from the side of the baffle touching domain 2. Since the domain includes not only the baffle but also the tubes, we can see the temperature in the tube cross-sections where they intersect the baffle (shown as the saturated red areas that are much hotter than the baffle itself).

What if we now shift the slice plot just slightly so that it plots the other side of the domain? If you're following along, change the *x*-coordinates of its plane to be slightly farther than *x=0.5*.



Then change the surface plot to show results evaluated for *side(1,T)* on the boundary instead, and you'll see two new similar plots:



The slice plot once again matches the data from the surface plot, but we're now examining the other side of the wall.

REFERENCE

The slice plot (right) has been translated using a rigid deformation so that we can see both plots at once. For information on using deformations, turn to the *Deformations, Height Expressions, and Scaling* section. For information on creating a slice plot, see the *Plot Types* chapter in Essentials of Postprocessing and Visualization in COMSOL Multiphysics.

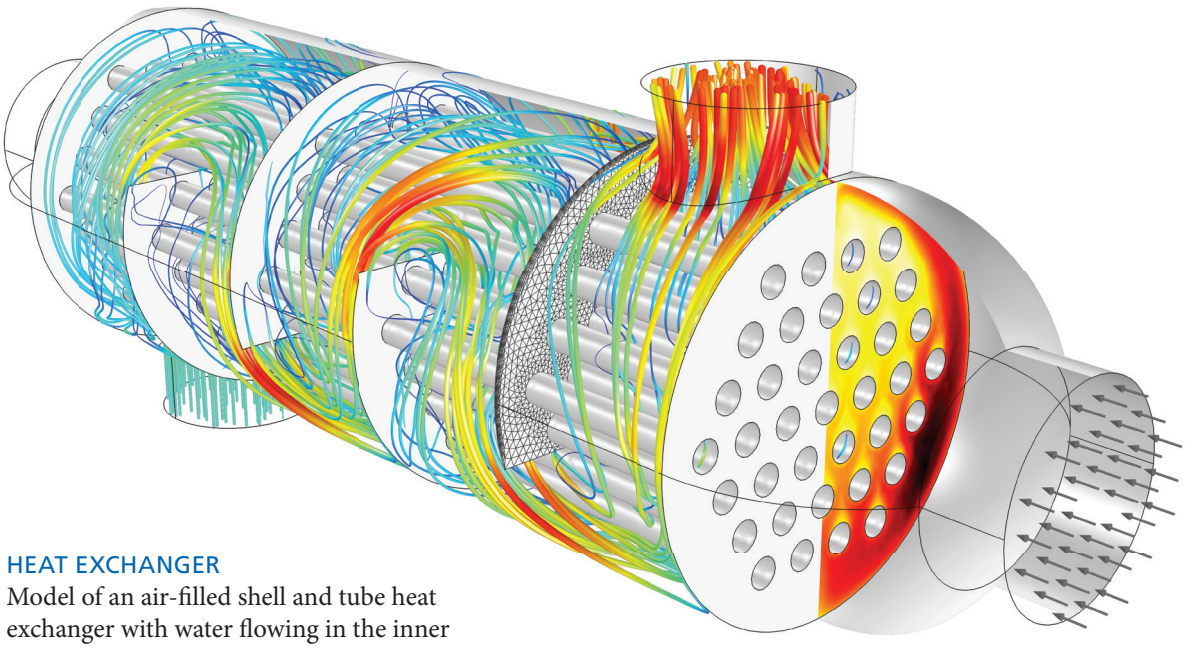
SUMMARY

We have now discussed several different uses of thin-layer modeling and distinguished between operators that can be used when postprocessing results on a single side. To recap:

- The *up* and *down* operators can be applied to show results on one side of a geometric entity, provided that the entity is of dimension one lower than the model itself and has an upside/downside.
- The *side* operator can be applied to entities of many dimensions lower than that of the model. This expression plots on entities of dimension *n* using results from the specified adjacent entity of dimension *n+1*.

Also recall that unexpected discontinuities may appear in a plot for several reasons when using lower-dimension postprocessing:

- The average of the temperatures from each side of a boundary may not smoothly transition from one boundary to the next; therefore adjacent boundaries may appear to have discontinuities between them.
- Since domain plots "choose" results from one side or the other to display, adjacent domains may also appear to have discontinuities when transitioning from one to the next if they are not showing results from the same side.
- When viewing results with a combination of domain plots and surface plots, there may be apparent differences between them—since the default surface plot shows an average, it is unlikely to have the same values as the results shown on the domain plot.

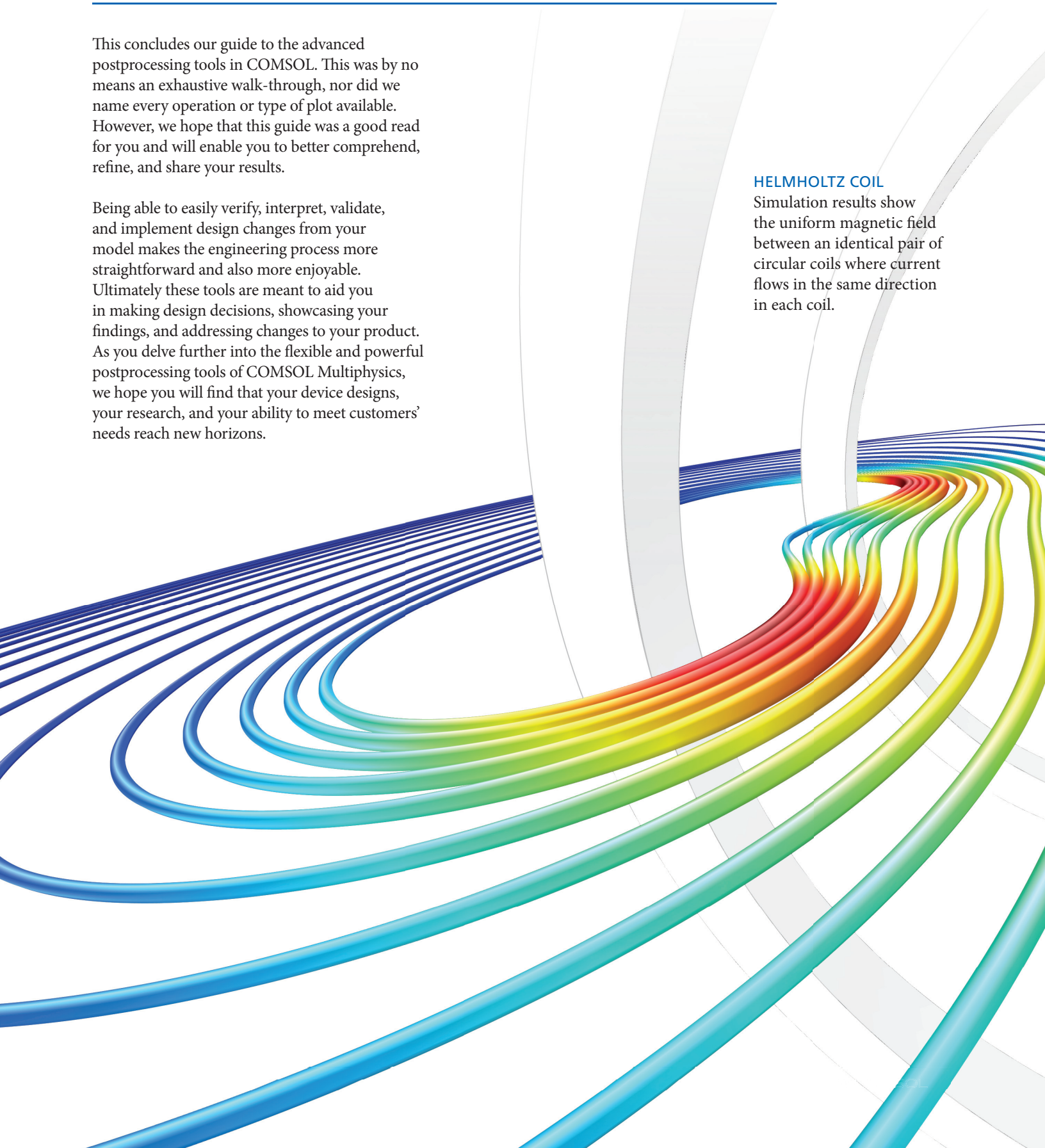


HEAT EXCHANGER
Model of an air-filled shell and tube heat exchanger with water flowing in the inner tubes. Simulation results reveal flow velocity, temperature distribution, and pressure within the vessel.

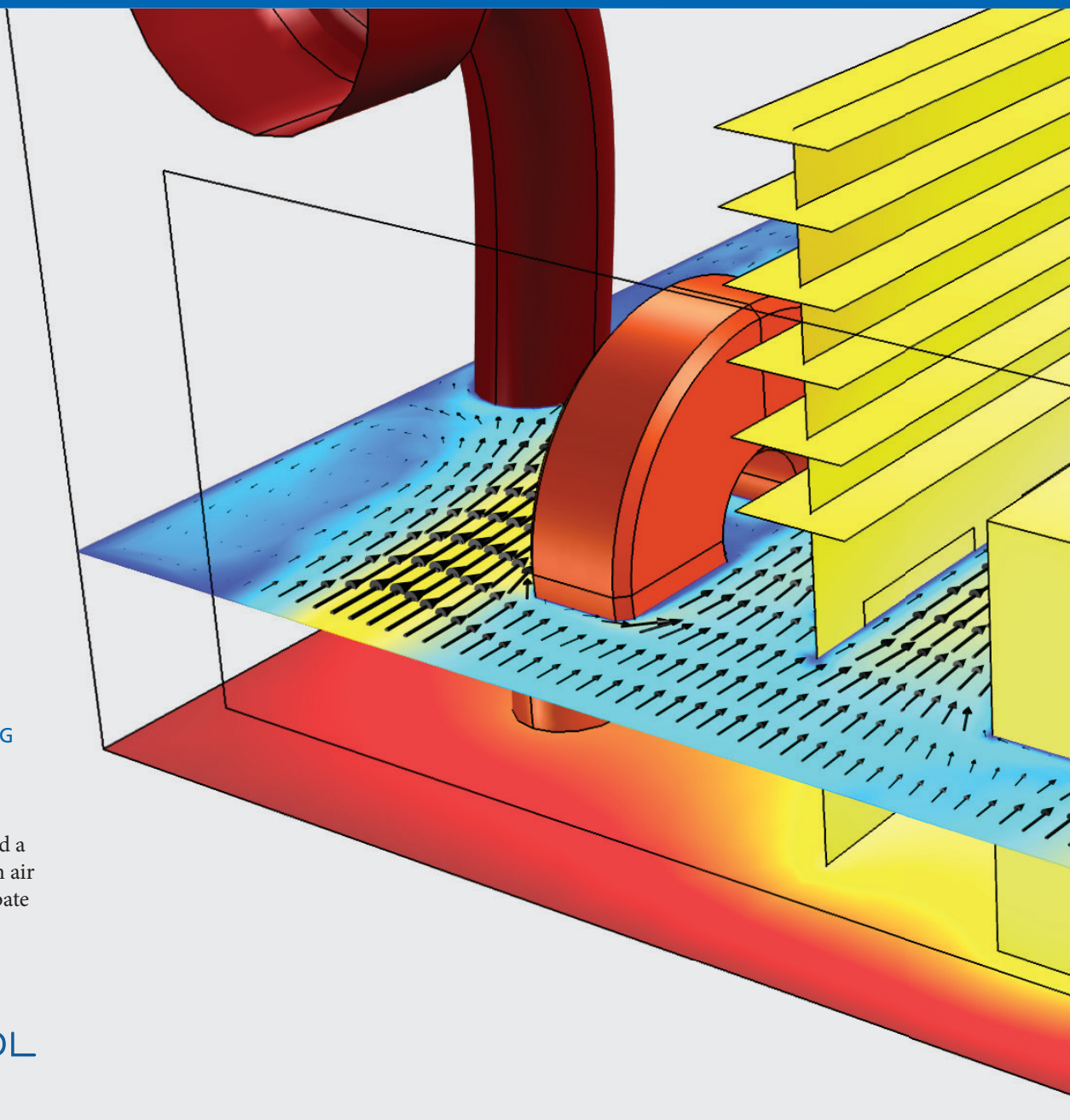
CONCLUDING REMARKS

This concludes our guide to the advanced postprocessing tools in COMSOL. This was by no means an exhaustive walk-through, nor did we name every operation or type of plot available. However, we hope that this guide was a good read for you and will enable you to better comprehend, refine, and share your results.

Being able to easily verify, interpret, validate, and implement design changes from your model makes the engineering process more straightforward and also more enjoyable. Ultimately these tools are meant to aid you in making design decisions, showcasing your findings, and addressing changes to your product. As you delve further into the flexible and powerful postprocessing tools of COMSOL Multiphysics, we hope you will find that your device designs, your research, and your ability to meet customers' needs reach new horizons.



HELMHOLTZ COIL
Simulation results show the uniform magnetic field between an identical pair of circular coils where current flows in the same direction in each coil.



POWER SUPPLY COOLING
Simulation of the thermal behavior of an enclosed computer power supply unit. An extracting fan and a perforated grille induce an air flow in the enclosure to abate internal heating.

COMSOL