



# COMSOL Multiphysics®

*Release Notes*



VERSION 4.4



# COMSOL Multiphysics Release Notes

© 1998–2013 COMSOL

Protected by U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,219,373; and 8,457,932. Patents pending.

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

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

Version:

November 2013

COMSOL 4.4

## Contact Information

Visit the Contact COMSOL page at [www.comsol.com/contact](http://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](http://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](http://www.comsol.com/support/case).

Other useful links include:

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

Part number: CM010001

# Release Notes

COMSOL® version 4.4 introduces all-new COMSOL Desktop environments, including a user interface for Windows with ribbon functionality. There are also many new functions and additions to the COMSOL product suite. These *Release Notes* provide information regarding new functionality in existing products.

# COMSOL Multiphysics

## *New Product in Version 4.4*

---

The Mixer Module complements the CFD Module to provide engineers and scientists with the necessary simulation tools for predicting and understanding the interactions between fluid flow and chemical processes in mixers and stirred vessels.

The Mixer Module, which requires a CFD Module license, includes the Rotating Machinery, Non-Isothermal Flow and Rotating Machinery, Reacting Flow interfaces. The added functionality extends the CFD Module in terms of turbulence modeling for the Rotating Machinery interfaces. It also includes free-surface features to capture the displacement of the liquid-air interface induced by the bulk motion in the domain, by the walls, and by the rotating shaft.

The model includes capabilities for analyzing frozen-rotor and time-dependent flows in two- and three-dimensional spaces. For a so-called frozen-rotor flow, the topology relative to the rotating reference frame is fixed (“frozen”). When the flow field is, or can be approximated to be, of this type, the computational time (CPU time) can be substantially reduced using the Frozen Rotor study type.

## *Major New Functionality in Version 4.4*

---

### **NEW COMSOL DESKTOP FUNCTIONALITY**

For use on the Windows® platform, the COMSOL 4.4 release introduces a ribbon layout, a style familiar to Microsoft® Office users and integrated into many other software designs. The ribbon-style layout is intuitive and makes it easy to locate similar and frequently used features. For the Linux® and Mac platforms, COMSOL 4.4 includes extended toolbars that provide almost identical single-click access to most functionality in the software.

The new Desktop layout highlights the most used functionality; menu items are grouped with similar tasks and actions. The features are still available when you right-click a node in the model tree to open the context menu—there are often several ways to complete the same task. As before, the available options in any context menu or ribbon tab are based on where you are in the model, what stage of the solving process you are at, and whether it is relevant to the model being built.

## SINGLE-CLICK SELECTIONS

It is much easier to assign geometric entities (domains, boundaries, edges, and points) to physics features with the new single-click selection behavior available in the Graphics window. Simply click (left-click) to add a selection to a feature; click again to deselect it. Geometric entities are highlighted when you hover over them with the mouse.

There is also a preselection mode, which you enter by setting the **Active** button to OFF in the settings windows for nodes that contain a selection list. There you can pick geometric entities (boundaries, for example), and then add a physics feature from the toolbar (a new boundary condition, for example, which then uses the selection that you picked).

To select overlapping objects (such as interior boundaries), use the scroll wheel (mouse wheel) to move the highlighting from the closest overlapping entity forward and back by rolling the wheel forward and backward (if you use COMSOL on a computer with a touchpad instead of a mouse, use its equivalent to the wheel; for example, moving two adjoining fingers up or down the touchpad). If you can move the mouse wheel in small distinct increments, each such increment moves the selection to the next or previous entity that you can reach. Alternatively, use the up arrow and down arrow keys to select the next or previous entity, respectively.

You can choose to use the selection mechanisms and highlighting that was available in earlier versions of COMSOL by selecting the **Use classic selections** check box on the **User interface** page in the **Preferences** dialog box (this will take effect after you restart COMSOL).

## NEW MULTIPHYSICS COUPLING METHODS

There are different approaches to using the new multiphysics coupling feature. It is either predefined when you add a specific physics interface, or it is automatically added when COMSOL recognizes a logical coupling inherent in the model design. When it is added as part of a predefined coupling, the coupling features are also included under a new Multiphysics node. Otherwise, all relevant coupling features are available from the context menu. Additional functionality is based on which add-on modules your license includes. For this version of COMSOL Multiphysics, and depending on the add-on module, some multiphysics interfaces are not yet converted to the predefined multiphysics coupling approach.

Predefined multiphysics interfaces provide you with a quick entry point for common multiphysics applications. You can create the same couplings using any of the other methods for multiphysics modeling, and you can continue to add, modify, disable, and remove physics in a model where you start by using one of the predefined multiphysics

interfaces. Add physics sequentially is another useful approach as you can verify that each type of physics or equation gives the expected results before adding more complexity to the model with other physics or coupling fields.

You can turn multiphysics on and off (that is, enable and disable features), giving you more flexibility to test and observe multiphysics effects.

The following multiphysics phenomena are available as Multiphysics subnodes for thermal stress and electromagnetic heating applications:

- Joule Heating
- Induction Heating (requires the AC/DC Module)
- Microwave Heating (requires the RF Module)
- Laser Heating (requires the Wave Optics Module)
- Thermal Stress (requires the Structural Mechanics or MEMS Module)
- Joule Heating and Thermal Expansion (requires the Structural Mechanics or MEMS Module)
- Thermoelectric Effect (requires the Heat Transfer Module)

### *General New Functionality*

---

#### **HIGHLIGHTING AND PICKING GEOMETRIC ENTITIES**

When selecting geometric entities, the entities are highlighted when you hover over them with the mouse. With the default graphics preference set to optimize for quality and OpenGL rendering (this can be set in the **Graphics and Plot Windows** section of the **Preferences** dialog box), the highlighting also creates a “glowing effect” so that it extends a bit outside of the highlighted objects, as long as your graphics card supports this. Previous versions of COMSOL used a default setting to optimize the graphics for performance instead.

#### **IMPROVED HELP SYSTEM**

- The context help takes you directly to the relevant section in the documentation for the selected node or window.

- A Sticky Help button makes it possible to “lock” the Help windows contents instead of automatically updating it when you click another node in the model tree. In previous this option was only available as a preference setting.
- Powerful search functionality: It is now possible to search for combinations of words, one word but not another word, and so on, including wildcard characters and searching for words that are almost identical to the one you enter.

### **MASS PROPERTIES**

Computation of mass properties is now available. When you add a Mass Properties node and provide an expression or value for the density, COMSOL defines variables for and computes the following mass properties: volume, mass, center of mass, moment of inertia, and principal moment of inertia.

### **NEW FUNCTIONS, OPERATORS, AND UNITS**

- New mathematical function: `erfinv`, the inverse error function.
- New summation operator: `sum`, which makes it easy to compute the sum of an indexed expression.
- New `emetric` operator: computes the square of the length of a global vector computed in the mesh element’s own metric.
- New unit: kilopond per square inch, `ksi`, is now available as a pressure unit.
- The following mathematical functions are now generally available and can be used and evaluated in parameter and variable definitions: `acosh`, `acoth`, `acsch`, `asech`, `asinh`, `atanh`, `besselj`, `bessely`, `besseli`, `besselk`, `erf`, `gamma`, and `psi`.

### **LOCALIZED INSTALLER**

The COMSOL Installer now supports the same local languages as the COMSOL Desktop. The language that you select during installation becomes the default language in the COMSOL Desktop.

### **STORAGE OF LICENSES USED**

If you block the use of certain licensed products in the **Licenses** dialog box, the new settings are stored for future COMSOL sessions. See [New Functionality in Studies and Solvers](#) below for information about the new `-checklicense` argument for COMSOL batch commands.

### **GEOMETRY SUBSEQUENCES**

New, powerful geometry subsequences make it possible to create your own geometric primitives. A geometry subsequence call corresponds to a subroutine call in a programming language. In other words, a geometry subsequence is a geometry sequence with a set of numerical input arguments and a set of geometry objects as output. You can view them as a user-defined geometry primitives. In the geometry sequence, you can call the subsequence repeatedly, or create instances of it, using a unique set of input argument values for each call. Calls can also be nested.

### **IMPROVED WORK PLANES**

There is now improved functionality for work planes:

- New *Transformed* work plane type that translates and rotates another work plane.
- You can position the result of a Subsequence Call node by matching work planes in 3D.
- Improved possibilities to control the work plane's coordinate system (that is, the position of the origin and the direction of the  $x_w$ -axis).
- For work planes of the *Quick* and *Face parallel* types, you can specify the plane offset distance by specifying a vertex.
- It is now possible to rotate *Circle perpendicular* work planes.
- A new *Unite Objects* sections includes a *Unite objects* check box, selected by default, and a tolerance setting for uniting 2D objects in the work plane. It is recommended to form the union of all objects in the 2D work plane sequence before extruding them into 3D.

### **PROGRAMMING OF GEOMETRY SEQUENCES**

For programming the geometry subsequences and for general use in the geometry sequences, *If*, *Else If*, *Else*, and *End If* statements are available to control the creation of a geometry depending on some conditions.

### **CUMULATIVE SELECTIONS**

A cumulative selection is a selection in the geometry sequence that is a union of contributions from several selections. Cumulative selections are especially useful for constructing a selection that has different definitions in different branches of an *If* statement. There is no node in the tree that corresponds to the cumulative selection. For example, you can create a cumulative selection that adds the boundaries from

several geometry objects in a geometry sequence, and then refer to such cumulative selections when defining materials and physics.

#### **ADDITIONAL NEW GEOMETRY FUNCTIONALITY**

- It is now possible to sweep several faces in a single Sweep feature, giving a single output object. Sweeping several faces this way improves usability.
- A new **Create cross-sectional faces** check box (active by default) in the Sweep node makes it possible to use the sweep operation for creating cross-sectional faces between the sweep sections. Such cross-sectional faces can be useful, for example, for a swept mesh where you want to specify the mesh distribution for each section of the sweep.
- A new **Split disconnected pairs** check box in the **Form Assembly** node makes it possible to split a boundary pair with disconnected sets of source and destination boundaries into several pairs. This can be useful if, for example, you want to define some of them as contact pairs and some of them as identity pairs.

#### **IMPROVED SWEPT MESHES**

For the swept mesher, a new method for the destination mesh generation is to project the source onto the destination. This method is used automatically when needed (using the default option to determine a suitable method), and it can be useful for creating swept meshes in very thin domains.

#### **EXPORT OF MESHES ON NASTRAN FORMAT**

2D and 3D meshes can now be exported on NASTRAN format (.nas files).

#### *New Functionality for Physics Interfaces*

---

For the PDE interfaces, you can add one or more **Excluded Points**, **Excluded Edges**, or **Excluded Surfaces** subnodes to a **Constraint**, **Dirichlet Boundary Condition**, or **Pointwise Constraint** node. This is done in order to exclude all or some of the surrounding points, edges, or surfaces from a constraint that acts on the edge, boundary, or domain inside of the excluded geometric entities. Excluding a constraint on a surrounding edge can be useful, for example, when you want to avoid the constraint's effects on the physics of an adjacent boundary.

### **IMPROVED PARAMETRIC SWEEP AND PARAMETRIC SOLVER**

Any scalar input to your model can be treated as a parameter that can be solved for over a range of values. COMSOL provides two different algorithms for solving for a range of parameters: the Parametric Sweep and the Parametric Solver. The user interface for these algorithms has been updated, and additional options have been added to the parametric solver. For transparency, efficiency, and flexibility you now have explicit control of when to run the Continuation algorithm with the Parametric Solver. Continuation can now be run for any single parameter in a multiparameter sweep. This makes it possible, for example, to run Continuation together with Load Cases. The Parametric Sweep and Parametric Solver functionalities in COMSOL provide several powerful options tailored to a range of different modeling needs. The following is a summary of the functionality in version 4.4, including features also available in earlier versions:

- **Parametric Sweep:** This is a parametric variation using either a Parametric Job or a Parametric Solver. The choice of computing a Parametric Sweep as a Parametric Job or using the Parametric Solver is largely automated, which makes it possible to use the Parametric Sweep functionality for the most common modeling tasks without needing to choose whether a Parametric Job or Parametric Solver is used. This functionality can be combined with almost any study step (Stationary, Time-Dependent, Eigenvalue) as well as a Solver Sequence containing multiple study steps. The parametric sweep can solve for any global parameter in the model, including those that affect the geometry, mesh, functions, and probes. You can use this functionality when trying out different dimensions, performing a mesh refinement study, solving for different load cases, and so on. In the general case, such as when sweeping for a geometric dimension, a Parametric Sweep is realized as a Parametric Job, with settings available under the Job Configuration node (this requires enabling Advanced Study Options). For special cases—such as a parameter variation where there is exactly one Stationary study step in the study and the parameters are not of geometry, mesh, or function type—then the parameter variation is instead realized through the Parametric Solver. A Frequency Domain study is a special type of Parametric Sweep that always invokes the Parametric Solver. In earlier versions, the Frequency Domain study used the Parametric Solver with Continuation with respect to the frequency parameter, whereas in version 4.4 it uses a Plain Sweep (see below). For typically linear frequency domain simulations, the

Plain Sweep in the frequency parameter is more efficient than the previously used Continuation method.

- **Parametric Solver:** This functionality is available for Stationary, Frequency Domain, and Time Dependent study steps. The Parametric Solver is a set of parametric variation methods, and it is used to vary boundary conditions, domain, and material property values. The Parametric Solver does not use the Job functionality from the Job Configurations node; thus, the Parametric Solver is not applicable for varying parameters used to define the geometry, mesh, functions, and probes. You can control the Parametric Solver settings from the Parametric node under Stationary Solver or the Time Parametric node under Time-Dependent Solver in the solver sequence. You can also control them from the settings for studies under Study Extensions.

The parametric solver provides additional functionality for solving nonlinear problems. When solving a nonlinear stationary problem, the choice of initial condition can strongly affect the convergence rate toward the solution or even the possibility of finding a solution at all. There are three different parametric sweep variants:

- **Continuation:** The parametric solver, by default, uses the previous solutions as initial conditions for the next stationary solution step. If the solver is unable to find a solution for a specified value in the range of parameters, the solver will backtrack and take a smaller step within the range of the parameters specified. This algorithm is also known as a continuation method, and the user interface gives you control over how the solver steps through the range of parameters. The solution for the previous continuation step is always used as the starting guess for the next step. The continuation method is used to reach convergence for highly nonlinear models where the degree of nonlinearity is “ramped up” using a continuation parameter. Either a tangent or a constant predictor can be used. If the solver is unable to find a solution for a requested parameter step, the solver either terminates or continues to the next requested parameter value. The time parametric solver does not support continuation.
- **Plain Sweep:** When parameters do not vary with the continuation algorithm, it is possible to choose whether the previous solution should be used as the starting guess or not. When reusing the solution, a simpler form of continuation is obtained. This is the method used in the previous version (4.3b) for multiparameter sweeps. A plain sweep, without using the previous solution as a

starting guess, is similar to a Parametric Sweep but has less overhead and is faster when applicable.

- Auxiliary Sweep: This option is used for single-parameter or multiparameter sweeps with the Parametric Solver and gives the user control of which variables should be solved for using the continuation or plain sweep methods. You can run continuation for at most one parameter. For a Stationary study, an Auxiliary Sweep controls all parameters that are swept for. For a Frequency Domain sweep, the Parametric Solver is used to vary the frequency at the Study node level, and you can optionally extend with other parameter variations using the settings for Auxiliary Sweep, which are available in the settings for the Frequency Domain study. Auxiliary Sweep is also available for a Time Dependent study, but with no Continuation option.
- The parametric sweep and the parametric solver can be combined within a single study. There is a computational advantage to using the parametric solver, and the software automatically calls the parametric solver algorithm when possible, even if you are using the Parametric Sweep interface. The Parametric Sweep can also be used on a cluster system, when using the Floating Network License type, to distribute the computational load.

The Parametric Sweep study step now behaves more consistently across stationary sweeps that are realized using the Parametric solver and sweeps that are realized using a Parametric Job. The Parametric Sweep study step realizes stationary parametric sweeps using a Parametric solver by default, if the problem and parameters allow it, which is more efficient than using a Parametric Job.

The following changes have been implemented for such Parametric solvers to make them more consistent with Parametric Jobs:

- A plain parametric sweep is always performed, instead of using the continuation algorithm.
- Each parameter step uses the initial values as an initial guess for the solver, instead of reusing the converged solution from the previous parameter step.

Simulations that require reusing the converged solutions or that need the continuation algorithm must now use the Auxiliary sweep functionality of the Stationary study step,

in the Study Extensions section. The Parametric Sweep study step can no longer be used in this case. The following settings should be made for such auxiliary sweeps:

- To use the continuation algorithm, select the parameter with the **Run continuation for** setting. The continuation algorithm always reuses the converged solutions for the selected continuation parameter.
- To further control how solutions are reused between the parameter steps, use the new **Reuse solution for previous step** setting, which can be **No** (the default), **Yes**, or **Auto**. The **Auto** setting analyzes the parameter stepping in a multiparameter sweep where continuation is not used and reuses the previous solution when it is beneficial.

For backward compatibility, see [Changes to Parametric Sweeps](#) below.

There is also new fallback functionality for the case when the solver does not converge for a certain parameter combination. This functionality is structurally similar to the Job Parametric Sweep. The new control mechanism is called **On error**, which has the options **Stop** and **Store empty solution**, and can be found in the Parametric node settings. The default is **Stop** (backward compatible). For the option **Store empty solution**, an empty solution (with NaNs) is stored and the solver continues with the next parameter combination. This is useful, for example, when doing frequency sweeps in which the frequencies of interest are so close to an eigenvalue that the solver does not converge. In these cases the new option allows you to avoid cumbersome restarts and also produce useful output. For results processing, any returned converged solutions can still be used. When this option is used together with continuation, the continuation sweep is not restarted after the failing parameter, and an empty solution is stored for all requested values after the failing one. This can be useful for multiparameter sweeps where, instead of the solver terminating, the next continuation sweep is started.

Distributed auxiliary sweeps on cluster now support distributing parameter sweeps over multiple parameters with continuation in one of the parameters. The substeps of the continuation sweeps are not distributed.

#### **TIME UNITS IN TIME-DEPENDENT STUDIES**

Time units can now be specified for time-dependent studies so that you can specify time steps and show results using, for example, minutes or days instead of seconds. The time unit setting propagates to Results and is used as the new default.

## OTHER NEW AND IMPROVED SOLVER FUNCTIONALITY

- For the eigenvalue solver, a new **Eigenvalue search method around shift** list makes it possible to search for eigenvalues that are **Closest in absolute value** (this is the default, and the only option in earlier versions) but also for eigenvalues with the **Larger real part**, **Smaller real part**, **Larger imaginary part**, or **Smaller imaginary part**.
- The relative error estimate (**LinErr**) and the relative residual (**LinRes**) are now printed in the solver log for direct linear solvers when **Check error estimate** is set to **Yes** or **Automatic**.
- From the **Programming** submenu for solver sequences, you can select **For** to add two nodes to the sequence: a **For** node and an **End For** node. Position those nodes in a solver sequence to create a for loop. These for loops can be useful, for example, for solving particle-field interactions in particle tracing by iterating between a stationary and a time-dependent solver.
- For COMSOL batch commands (`comsolbatch` and `comsolclusterbatch`), you can print the license requirements for a Model MPH-file using the argument `-checklicense <filename>`.

## *New and Improved Results and Visualization Functionality*

---

- An improved list of predefined quantities for plotting and general postprocessing, including a general search field at the top where you can enter a filter text to find predefined quantities that matches the filter text.
- *Filter* subnodes are now available for *Max/Min*-type plots, so that the maximum and minimum values can be filtered to reflect, for example, the maximum and minimum values of only certain domains in the geometry.
- For *Mesh* plots, you can turn the color legend off, and you can also reverse the color table for tuning, for example, the visualization of the mesh quality.
- For *Contour* plots that use filled contours, there is now an option to fill or exclude surfaces outside of the contour levels.

## *Backward Compatibility With Version 4.3b*

---

### MODEL NODES ARE NOW COMPONENT NODES

The Model nodes (as they were called in previous versions), which contained separate model components in a model file, are called **Component** nodes in version 4.4.

### CHANGES TO PARAMETRIC SWEEPS

Old models that use stationary parametric sweeps are loaded with the **Reuse solution for previous step** list set to **Yes**. The **Run continuation for** list is set to the parameter used, unless the continuation algorithm would not have been used for this model in previous versions (for example, if multiple parameters are used or if the parameter list is not monotonous).

### CHANGES TO THE PHYSICS SELECTION IN STUDY SETTINGS

The names of the states of a physics in the physics tree, which you can modify under **Physics and Variables Selection** in the study steps' settings windows, have changed:

- **Provide Degrees of Freedom** is now called **Disable in Solvers**.
- **Disable** is now called **Disable in Model**.

### REVISED FORMULATION FOR LAMINAR INFLOW AND LAMINAR OUTFLOW

The formulations of the laminar inflow and laminar outflow conditions have been corrected. The modified formulation will give a more accurate mass flux. Some models may now produce a slightly different flow field.

Laminar inflow and laminar outflow are available in the fluid flow physics interfaces in the following modules:

- Batteries and Fuel Cells Module
- CFD Module
- Corrosion Module
- Electrochemistry Module
- Electrodeposition Module
- Heat Transfer Module
- Microfluidics Module
- Plasma Module
- Subsurface Flow Module

### NEW DEFAULT FOR GRAPHICS OPTIMIZATION

Under **Graphics and Plot Windows** in the **Preferences** dialog box, the default settings in the **Optimize for** list is now **Quality** instead of **Performance**.

## VELOCITY/ACCELERATION INTEGRATION VARIABLE

For the Solid Mechanics interface (and all related multiphysics interfaces) and the Truss interface, a help variable `u0` (velocity integration variable) is used in **Prescribed Velocity** and **Prescribed Acceleration** features for Time Dependent study types. This variable computes the displacement for each point where the condition is prescribed. It has been changed to use the full feature scope in order to avoid collisions in cases where several such features exist within the same model component. As a result, when opening and running any old model that uses such features together with a segregated solver, an error message appears, stating that not all dependent variables occur in at least one of the segregated solver steps. The relevant action is to manually add the velocity integration variable to the segregated step containing the corresponding displacement field. Alternatively, you can regenerate any affected solver sequence.

## MATH LIBRARIES ON AMD PROCESSORS

MKL is now the default math library on AMD processors. Switching to the ACML math library for AMD processors might improve performance in some cases.

### *Backward Compatibility With Version 4.3a*

---

## ERROR ESTIMATION CHANGES IN THE SOLVERS

The **Automatic** method for **Check error estimate** for direct linear solvers and for **Validate error estimate** has changed. For nonlinear problems and for time-dependent problems the underlying nonlinear solver does not accept terminating for a linear solution step that does not fulfill the error estimate (unless the step size is very small). The motivation for this change is that the old method can lead to premature termination of the nonlinear solution process, which in some cases introduces a large error for the computed solution.

This change can cause the solvers to take more nonlinear steps for stationary problems and more time steps for time-dependent problems, and it may also lead to convergence problems. For such cases, use the **No** method to obtain the old behavior. However, doing so can hide numerical problems and potentially lead to large numerical errors.

## TERMINATION CRITERION FOR STATIONARY SOLVERS

The default termination criterion for stationary solvers has changed. In 4.3a the settings corresponded to **Solution**; now the default is **Solution or residual**. This change in default termination criterion might affect models created in earlier versions of COMSOL if you regenerate the solver sequence in 4.4.

### **BACKWARD EULER INITIALIZATION TIME STEP**

A new setting in the **Advanced** section of the settings window for the **Time-Dependent Solver**, called **Fraction of initial step for Backward Euler**, provides an option for entering a dimensionless quantity that determines the size of the time step for the backward Euler method (in terms of the initial step). This value can improve the accuracy of the initialization step but can also affect the start-up of some models. The default value is 0.001 (this differs from earlier versions, which used a value of 1). When opening models created in version 4.3a or earlier, the value for this fraction is set to 1 to maintain compatibility with those versions.

### **LOADING EXTERNAL PHYSICS BUILDER JAR FILES**

External physics builder JAR archives compiled with earlier versions of COMSOL include a manifest file that contains a reference to the CDO library, which has been removed. To load such JAR files in version 4.4, you must first do one of the following:

- Delete the META-INF/MANIFEST.MF file in the archive source on the file system, and then recompile the JAR archive.
- Manually remove the line with `org.eclipse.emf.cdo` in the META-INF/MANIFEST.MF file in the JAR archive. You can do so directly in a file archive manager such as 7-Zip or similar.

### **HIGHLIGHTING GEOMETRY OBJECTS WHEN DRAWING IN 2D**

When you have drawn one geometry object on top of another object, toggling of the highlighting of these objects occurs when you click several times. It is important that you click without moving the cursor in a position where the objects overlap in order to toggle. For example, if you draw a circle (C1) and then draw another smaller circle (C2) inside of C1, then a first click on C2 may highlight C1, while a second click, without moving the cursor, highlights C2. In previous versions, you would move the cursor after the first click in order to highlight C2.

### *Backward Compatibility With Version 4.3*

---

### **NEW TERMINOLOGY FOR CONSTRAINT TYPES**

The following constraint types have new names in version 4.4:

- *Bidirectional, symmetric* is now *Apply reaction terms on: All physics (symmetric)*.
- *Unidirectional* is now *Apply reaction terms on: Individual dependent variables*.

### **WEAK CONSTRAINTS UPDATE FOR THE LAMINAR FLOW INTERFACE**

The weak constraint formulations for the following boundary conditions has been updated:

- Symmetry
- The Slip option in the Wall feature

These boundary conditions are now formulated using the same set of Lagrange multipliers as all the other boundary conditions in the Laminar Flow interface. The Lagrange multiplier `un_1m` has been removed.

Models saved in version 4.3 will include `un_1m` until the model is re-solved. In some cases, occurrences of `un_1m` in the solver sequence must be replaced manually. This is the case if `un_1m` was the only Lagrange multiplier component in a segregated group or the only Lagrange multiplier component of a Vanka smoother. Alternatively, you can generate a new automatic solver sequence. Models saved in versions earlier than 4.3 must either be re-solved in version 4.4 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 4.4.

### **AUTOMATICALLY CREATED IDENTITY/CONTACT PAIRS**

The pairs created by the Form Assembly feature in the geometry can differ from their form in 4.3 for certain 3D and 2D geometries. The pairs should now be the same for the CAD and COMSOL representations.

# AC/DC Module

## *New Functionality in Version 4.4*

---

### **NONLINEAR MAGNETIC MATERIALS DATABASE**

A database of 165 ferromagnetic and ferrimagnetic materials is now included in the AC/DC Module. The database contains BH-curves and HB-curves, enabling the use of material properties in the magnetic fields formulations. The curve data is densely sampled and has been processed to eliminate hysteresis effects. Outside of the range of experimental data, linear extrapolation is used for maximal numerical stability.

### **NEW USER INTERFACES FOR INDUCTION HEATING**

The workflow for setting up simulations with induction heating has been significantly improved with the introduction of a dedicated Multiphysics node in the Model Builder. The new interfaces are appropriate when each of the constitutive physics can be modeled separately. Because the electrical time scales of a typical inductive process are on the order of thousands of cycles per second, whereas the temperature fields vary on the order of seconds, it is appropriate to model the electrical problem in the frequency domain, and the thermal problem in the time domain, or stationary domain. The new Inductive Heating interface brings up interfaces for computing the induced currents and losses via the Magnetic Fields interface and the temperature rise via the Heat Transfer interface. It also adds a Multiphysics node, which keeps track of the couplings between the physics. The magnetic fields and the heat transfer problems can also be solved separately.

# Acoustics Module

## *New Functionality in Version 4.4*

---

### **AEROACOUSTICS WITH THE LINEARIZED EULER EQUATIONS**

The new Linearized Euler physics interfaces compute the acoustic variations to pressure, velocity, and density for a given background mean flow. They solve for the linearized Euler equations, including the energy equation, with the assumptions that the background flow is an ideal gas (or is well approximated by an ideal gas) and that there are no thermal or viscous losses. The Linearized Euler physics interfaces are available for time domain, frequency domain, and eigenfrequency studies. Application examples include analyzing the propagation of noise from jet engines, modeling the attenuation properties of mufflers in the presence of non-isothermal flow, and the study of gas flow meters. These are all situations where a gas background flow influences the propagation of acoustic waves in the fluid.

### **NEW STRUCTURE FOR PRESSURE ACOUSTICS FLUID MODELS**

The fluid models for pressure acoustics are now organized into Pressure Acoustics, Poroacoustics, and Narrow Region Acoustics. The Dipole and Monopole domain sources are now available under the More menu.

#### *Poroacoustics*

For poroacoustics, the fluid models are now given names conforming with industry standards: Delany-Bazley-Miki and Johnson-Champoux-Allard. In addition, the defaults and organization of parameters have been streamlined.

#### *Narrow Region Acoustics*

For Narrow Region Acoustics, two options are available: Wide duct approximation and Very narrow circular duct. In addition, the defaults and organization of parameters have been streamlined.

## *New Models in Version 4.4*

---

- *Sedan Interior Acoustics*: This is a model of the acoustics inside a sedan—that is, inside a typical hard-top family car. The model sets up sources at loudspeaker locations as well as impedance conditions to model soft absorbing surfaces (seats and carpet). The model shows plots of the pressure, sound pressure level, and

intensity inside the car. The frequency response at given points inside the cabin is also determined.

### *Backward Compatibility With Version 4.3a and Older*

---

- The old style perfectly matched layer (PML) that is located under the physics node will be discontinued in the next version of COMSOL. From the next version on only the PMLs defined under the Definitions node are supported.
- In order for the old style perfectly matched layer (PML) to be backward compatible, the PML feature must be placed after any domain Monopole source, domain Dipole source, or Background Pressure Field features.

### *Backward Compatibility With Version 4.3*

---

- The symbols for the acoustics monopole ( $Q_m$ ; was  $Q$ ) and dipole ( $\mathbf{q}_d$ ; was  $\mathbf{q}$ ) volume sources in pressure acoustics have changed.
- The default value for the Typical wave speed property in the Acoustic-Piezoelectric Interaction, Frequency Domain interface has changed to 343 m/s.

# Batteries & Fuel Cells Module

## *New and Improved Functionality in Version 4.4*

---

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

### **POINT AND LINE MASS SOURCES**

A point source can be used to simulate a source distributed over a very small volume. While it can be applied to points in 3D or on the symmetry axis in axisymmetric models, its actual effect is distributed throughout the close vicinity of the point. The size of the distribution depends on the mesh and strength of the source—a finer mesh spreads the source over a smaller region but results in a more extreme pressure value. A line source in 3D and 2D axisymmetric models represents a source emanating from a tube with a very small cross-sectional area. Line sources can be added to lines in 3D, to the symmetry axis in 2D axisymmetric models, and to points in 2D, for which they represent the cross section of a tube of very small area.

Point and line mass sources for fluid flow are included as contributions to the continuity equation. This functionality has been added to the Reacting Flow in Porous Media, Diluted Species interface.

## *Backward Compatibility With Version 4.3*

---

### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

### **REACTING FLOW INTERFACE NAME CHANGE**

The Reacting Flow interfaces have been renamed Reacting Flow in Porous Media. Opening a model using either the Reacting Flow, Diluted Species or the Reacting Flow, Concentrated Species interface, the interface is renamed Reacting Flow in Porous Media.

## **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Brinkman Equations
- Free and Porous Media Flow

See [CFD Module](#) for details.

## **MOBILITY**

In physics user interfaces modeling migration (Transport of Diluted Species; Transport of Concentrated Species; Reacting Flow in Porous Media, Diluted Species; Reacting Flow in Porous Media, Concentrated Species; Tertiary Current Distribution; and Nernst-Planck) anisotropic mobility is now supported, and the mobility can be set to follow the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

# CAD Import Module and LiveLink Products for CAD

*New Functionality in the CAD Import Module and LiveLink Products for CAD in Version 4.4*

---

## **GEOMETRY REPRESENTATION**

When the **Geometry representation** list in the **Geometry** node's settings window is set to COMSOL kernel, the geometry operations that require the CAD Import Module are no longer visible. When the software suggests to change the geometry representation, a dialog box appears where you can confirm this.

There is a new preference on the **Geometry** page in the **Preferences** dialog box: The **When opening an existing model** list under **Geometry representation**. By default, it is set to **Use model's kernel**. If you change it to **Convert to COMSOL kernel**, you will get a possibility to rebuild the geometry using the COMSOL kernel when opening a model made with the CAD Import Module. The corresponding API methods are `ModelUtil.setOpenGeometryKernel` and `ModelUtil.getOpenGeometryKernel`.

The **Geometry** preference **Switch kernel automatically** and the corresponding API methods `ModelUtil.setAutoSwitchGeometryKernel` and `ModelUtil.getAutoSwitchGeometryKernel` available in earlier versions have been removed.

## **CAD IMPORT MODULE GEOMETRY KERNEL UPGRADE**

The CAD Import Module and the LiveLink™ products for CAD utilizes the Parasolid® geometry kernel from Siemens PLM for solid modeling operations, geometry repair, and defeaturing. (Without these products, a COMSOL-native geometry modeling kernel is used.) The CAD Import Module released with COMSOL 4.4 includes an upgraded version of the Parasolid kernel and as a result a number of stability issues have been fixed, which makes import of CAD models and solid operations more robust.

## *New Functionality in LiveLink™ for Inventor® in Version 4.4*

---

### **IMPROVED SYNCHRONIZATION**

Synchronizing the geometry between Inventor and COMSOL now also includes the synchronization of material selections. Selections that contain synchronized geometry objects (bodies) are created in COMSOL based on the material definitions from the CAD design. The selections get their names from the material name in Inventor. Use these selections as input for geometry features requiring object selections or for any model definitions, physics, or material settings requiring domain selections. The LiveLink node contains a table with a list of the synchronized selections.

## *New Functionality in LiveLink™ for SolidWorks® in Version 4.4*

---

### **IMPROVED SYNCHRONIZATION**

Expanding on the functionality that synchronizes selections based on material assignments to the CAD design in SolidWorks, the LiveLink™ interface now adds support for user-defined selections. In the added COMSOL Selections interface in SolidWorks, you can define selections that are synchronized to the COMSOL model. You can choose to synchronize selections for bodies, faces, edges, or points, which become selections in the model when the design is synchronized with the COMSOL Desktop. Setting up a model becomes more efficient as you can also create selections from features of the Model Builder or from components of an assembly.

# CFD Module

## *New Functionality in Version 4.4*

---

### **WALL ROUGHNESS FOR TURBULENT FLOW**

For modeling surface roughness of walls in turbulent flow, there are now two roughness models available: Sand Roughness and Generic Roughness. The Wall roughness feature modifies the turbulence wall functions and is available for the k-epsilon and k-omega turbulence models in the CFD Module. The Sand Roughness model is commonly used in engineering applications and introduces a single parameter for the *equivalent sand roughness height*. The Generic Roughness model is more general and has, in addition to the roughness height, a *roughness parameter*, which you can use to model other types of roughness. The default value for the roughness parameter corresponds to that for sand roughness.

Wall functions for rough walls have been implemented for the following physics interfaces:

- Single-Phase Flow
  - Turbulent Flow, k-epsilon
  - Turbulent Flow, k-omega
- Single-Phase Flow, Rotating Machinery
  - Turbulent Flow, k-epsilon
  - Turbulent Flow, k-omega
- Bubbly Flow, Turbulent Bubbly Flow
- Mixture Model, Turbulent Flow
- Turbulent Two-Phase Flow, Level Set
- Turbulent Two-Phase Flow, Phase Field
- Fluid-Structure Interaction interface with a turbulence model selected

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## DRAG MODEL FOR NON-SPHERICAL PARTICLES

In addition to the previously available drag models, Schiller-Naumann, Hadamard-Rybczynski, and Gidaspow, there is now a new *Haider-Levenspiel drag model* for nonspherical particles. This new drag model is available for the following physics interfaces:

- Mixture Model
- Euler-Euler Model
- Particle Tracing for Fluid Flow (requires the Particle Tracing Module)

The settings are somewhat different for each physics interface. The drag model requires the sphericity,  $S_p$ , which is a measure of how spherical a particle is. For a spherical particle  $S_p = 1$ , while nonspherical particles have  $S_p < 1$ . Nonspherical particles typically result in higher drag than spherical particles.

## POINT AND LINE MASS SOURCES

A point source can be used to simulate a source distributed over a very small volume. While it can be applied to points in 3D or on the symmetry axis in axisymmetric models, its actual effect is distributed throughout the close vicinity of the point. The size of the distribution depends on the mesh and strength of the source—a finer mesh spreads the source over a smaller region but results in a more extreme pressure value. A line source in 3D and 2D axisymmetric models represents a source emanating from a tube with a very small cross-sectional area. Line sources can be added to lines in 3D, to the symmetry axis in 2D axisymmetric models, and to points in 2D, for which they represent the cross section of a tube of very small area.

Point and line mass sources for fluid flow are included as contributions to the continuity equation. This functionality has been added to the following physics interfaces for fluid flow:

- Single-Phase Flow
- Brinkman Equations
- Free and Porous Media Flow
- Reacting Flow in Porous Media, Diluted Species (requires the Batteries & Fuel Cells Module, CFD Module, or Chemical Reaction Engineering Module)
- Two-Phase Flow (requires the CFD Module or the Microfluidics Module)
- Rotating Machinery, Fluid Flow (requires the CFD Module or the Mixer Module)

- Fluid-Structure Interaction (requires the Structural Mechanics Module or the MEMS Module)
- Two-Phase Flow, Moving Mesh (requires the Microfluidics Module)

#### **OUTLET BOUNDARY CONDITION**

The Outlet boundary condition for fluid flow has been revised for improved mass conservation as well as faster and more robust convergence. The new Outlet feature has only one Pressure option, which corresponds to the Normal Stress option from earlier versions. In addition to specifying the pressure, there are two new options for specifying a normal flow and suppressing backflow:

- Normal flow prescribes a zero tangential velocity at the outlet. This condition can be expected if the outlet represents a straight pipe or channel. However, it is not selected by default because the flow can be disturbed upstream of the outlet, potentially altering the solution significantly.
- Suppressing backflow reduces the tendency for fluid to enter the domain from the outside. It does not completely prevent backflow, and in the case that backflow occurs, this option locally decreases the specified pressure. Controlling backflow is important when combining fluid flow with other transport equations, such as mass and heat transport. If the flow reverses, the outlet boundary condition for the transport equations is no longer valid; this can lead to convergence problems or nonphysical solutions. The Suppress backflow option is therefore selected by default.

The Outlet boundary condition has been revised for the new version in the following physics interfaces in the CFD Module:

- Single-Phase Flow
- Brinkman Equations
- Free and Porous Media Flow
- Two-Phase Flow
- Level-Set
- Phase-Field
- Non-Isothermal Flow and Conjugate Heat Transfer
- Reacting Flow

- Reacting Flow in Porous Media
  - Diluted Species
  - Concentrated Species
- Rotating Machinery, Single-Phase Flow
- Fluid-Structure Interaction (requires the Structural Mechanics Module or the MEMS Module)

This change is valid for laminar flow, Stokes flow, and turbulent flow when applicable. The Outlet boundary condition from previous versions still exists but has been excluded from the physics context menu. Models created in previous versions retain the old Outlet feature, but adding a new Outlet node gives them the new functionality.

### *Backward Compatibility With Version 4.3b*

---

#### **BUBBLE-INDUCED TURBULENCE IN BUBBLY FLOW**

Equation terms accounting for bubble-induced turbulence in the Turbulent Bubbly Flow interface are now added correctly. As a result, models with Turbulent Bubbly Flow now show higher levels of turbulence.

### *Backward Compatibility With Version 4.3a*

---

#### **MIXTURE MODEL EQUATION FORMULATION**

The Mixture Model equation formulation has been revised to increase the accuracy and facilitate the solution of a number of problem setups. As a consequence of the revision, old models can return slightly different results in the new version than they did in the previous versions. As a part of the revision, a penalty diffusion on the dispersed phase has been deactivated because after the other revisions, it made the equation system too nonlinear and therefore more difficult to make it converge. However, you can still activate the penalty diffusion in the Advanced section of the Mixture Model settings.

### *Backward Compatibility With Version 4.3*

---

#### **FLUID-STRUCTURE INTERACTION**

The Fluid-Structure Interaction (FSI) multiphysics interface has been updated. The separate `vWall` field is no longer required and has been removed. FSI models from 4.3

and earlier versions that include Time Dependent study steps will be affected in the following way:

- Model files for Java will fail. Any reference to the `vWall` field must be removed.
- Models older than 4.3 (that is, 4.2a and older) must either clear their time-dependent solutions when opened in 4.4. An alternative procedure is to:
  - Open the model in version 4.3.
  - Right-click any study with a time-dependent study step and select *Update Solution*.
  - Save the model.
  - Open the model in version 4.4.

FSI models with only stationary study steps will not be affected. Note that `vWall` will still be available as a variable. Hence, references to `fsi.vWall` made in, for example, another physics still work.

#### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

#### **REACTING FLOW INTERFACE NAME CHANGE**

The Reacting Flow interfaces have been renamed Reacting Flow in Porous Media. If you open a model using either the Reacting Flow, Diluted Species or the Reacting Flow, Concentrated Species interface, the interface is renamed Reacting Flow in Porous Media.

#### **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for the following boundary conditions in the following interfaces has been updated:

*Laminar Flow and Creeping Flow*

- Symmetry
- The Slip boundary condition in the Wall feature

*Turbulent Flow,  $k$ - $\varepsilon$  and Turbulent Flow,  $k$ - $\omega$*

- Symmetry
- The Wall functions boundary condition in the Wall feature

- The Moving wall (wall functions) boundary condition in the Wall feature
- The Slip boundary condition in the Wall feature

*Turbulent Flow, low-Re  $k$ - $\varepsilon$  and Turbulent Flow, Spalart-Allmaras*

- Symmetry
- The Slip option in the Wall feature

*Rotating Machinery*

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)
- Rotating Wall (turbulent flow only)

*Bubbly Flow*

- Symmetry
- The Slip boundary condition for the liquid phase in the Wall feature
- The Wall functions boundary condition for the liquid phase in the Wall feature (turbulent flow only)

*Mixture Model*

- Symmetry
- The Slip boundary condition for the mixture in the Wall feature
- Wall functions boundary condition for the mixture in the Wall feature (turbulent flow only)

*Euler-Euler Model, Laminar Flow*

- Symmetry
- The Slip boundary condition for either phase in the Wall feature
- The Slip boundary condition for either phase in the Inlet feature

*Brinkman Equations and Free and Porous Media Flow*

- Symmetry
- The Slip boundary condition in the Wall feature

*Two-Phase Flow, Level Set*

- Symmetry

- The Slip boundary condition in the Wall feature
- The Wetted wall boundary condition in the Wall feature
- The Moving wetted wall boundary condition in the Wall feature
- The Wall function boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)

#### *Two-Phase Flow, Phase Field*

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall function boundary condition in the Wall feature (turbulent flow only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow only)

#### *Non-Isothermal Flow and Conjugate Heat Transfer*

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow  $k$ - $\epsilon$  and turbulent flow  $k$ - $\omega$  only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow  $k$ - $\epsilon$  and turbulent flow  $k$ - $\omega$  only)

#### *High Mach Number Flow*

- Symmetry
- The Slip boundary condition in the Wall feature
- The Wall functions boundary condition in the Wall feature (turbulent flow  $k$ - $\epsilon$  only)
- The Moving wall (wall functions) boundary condition in the Wall feature (turbulent flow  $k$ - $\epsilon$  only)

These boundary conditions are now formulated using the same set of Lagrange multipliers as all the other boundary conditions for the dependent velocity variables. The previously used Lagrange multiplier `un_1m` has been removed.

When you open models saved in version 4.3, they include `un_1m` until the model is re-solved. In some cases, occurrences of `un_1m` in the solver sequence must be replaced manually. This is the case if `un_1m` was the only Lagrange multiplier component in a segregated group or the only Lagrange multiplier component of a Vanka smoother. Alternatively, you can generate a new automatic solver sequence. Models saved in

versions previous to version 4.3 must either be re-solved in version 4.4 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 4.4.

Weak constraints for the Interior Wall feature are no longer available.

#### **REVISION OF THE TURBULENCE MODELS**

The formulations of some variables in the turbulence models have been revised in order to improve accuracy. Models using a turbulence model can display a different convergence behavior in version 4.4 than in version 4.3 and the result can differ slightly between the versions.

# Chemical Reaction Engineering Module

*New Functionality in Version 4.4*

---

## **SUPPORT FOR FLASH CALCULATIONS**

The Chemical Reaction Engineering Module now includes the ability to perform *flash calculations* as part of its Thermodynamics interface. A flash calculation determines the equilibrium state between phases in a system of one or several chemical species and phases, when the system has been subjected to a large decrease in pressure, usually caused by passing through a throttling device. Using inputs provided by CAPE-OPEN thermodynamic libraries, COMSOL can perform flash calculations of multicomponent mixtures and couple these calculations to other physics involved in your chemical process simulations.

## **GLOBAL QUANTITIES FOR FLOW AT BOUNDARIES**

A variety of averaged global quantities for inflow and outflow boundaries have been introduced and made accessible during postprocessing and to use in multiphysics couplings:

- The *total mass flow* and *average pressure* within Darcy's Law calculations of fluid flow
- The *total mass flow* at the outlet when modeling laminar flow
- The *cup-mixing temperature* when modeling non-isothermal flow
- The *averaged mass fraction* when modeling the transport of both dilute and concentrated species

## **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## **POINT AND LINE MASS SOURCES**

A point source can be used to simulate a source distributed over a very small volume. While it can be applied to points in 3D or on the symmetry axis in axisymmetric

models, its actual effect is distributed throughout the close vicinity of the point. The size of the distribution depends on the mesh and strength of the source—a finer mesh spreads the source over a smaller region but results in a more extreme pressure value. A line source in 3D and 2D axisymmetric models represents a source emanating from a tube with a very small cross-sectional area. Line sources can be added to lines in 3D, to the symmetry axis in 2D axisymmetric models, and to points in 2D, for which they represent the cross section of a tube of very small area.

Point and line mass sources for fluid flow are included as contributions to the continuity equation. This functionality has been added to the Reacting Flow in Porous Media, Diluted Species interface.

### *Backward Compatibility With Version 4.3*

---

#### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

#### **REACTING FLOW INTERFACE NAME CHANGE**

The Reacting Flow interfaces have been renamed Reacting Flow in Porous Media. If you open a model that uses either the Reacting Flow, Diluted Species or the Reacting Flow, Concentrated Species interface, the interface is renamed Reacting Flow in Porous Media.

#### **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Brinkman Equations
- Free and Porous Media Flow

See [CFD Module](#) for details.

#### **MOBILITY**

In physics user interfaces modeling migration (Transport of Diluted Species; Transport of Concentrated Species; Reacting Flow in Porous Media, Diluted Species; Reacting Flow in Porous Media, Concentrated Species; and Nernst-Planck Equations)

anisotropic mobility is now supported, and the mobility can be set to following the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

# Corrosion Module

## *New and Improved Functionality in Version 4.4*

---

### **PRIMARY CURRENT DENSITY DISTRIBUTION INTERFACES**

You can specify that you want to model primary current distribution in the physics interfaces for corrosion by choosing the new Corrosion, Primary interface. Previously, you had to specify this from the Corrosion, Secondary interface.

### **ACCURATE BOUNDARY FLUXES**

Two new variables,  $nI_s$  and  $nI_l$ , are now available for accurately computing the normal current density in the electrode and electrolyte phases, respectively.

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## *New Model in Version 4.4*

---

### **ATMOSPHERIC CORROSION**

Metallic structures are susceptible to atmospheric corrosion when they are exposed to humid air containing salt particles. This model simulates atmospheric galvanic corrosion of an aluminum alloy in contact with steel. The electrolyte film thickness depends on the relative humidity of the surrounding air and the salt load density of NaCl crystals on the metal surface. The Secondary Current Distribution interface is used in the model with parametric sweeps for relative humidity and salt load density. The limiting current density for the oxygen reduction reaction is evaluated using empirical expressions for diffusivity and the solubility of  $O_2$  in the electrolyte solution.

#### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

#### **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Brinkman Equations
- Free and Porous Media Flow

See [CFD Module](#) for details.

#### **MOBILITY**

In physics user interfaces modeling migration (Transport of Diluted Species; Tertiary Current Distribution, Nernst-Planck; and Corrosion, Tertiary Nernst-Planck) anisotropic mobility is now supported, and the mobility can be set to follow the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

# ECAD Import Module

*New Functionality in Version 4.4*

---

## **ODB++ IMPORT**

By implementing import of the ODB++ format, the functionality of the ECAD Import Module is expanded to include support for one of the most popular formats for transferring printed circuit board (PCB) data. Using this new import capability you can now extract geometrical data from an ODB++ file and use it to create a geometry of the PCB for simulation in COMSOL. Geometry import in the ECAD Import Module now supports the additional file extensions: .zip, .tar, .tgz, .gz, and .Z for the ODB++ file format.

# Electrochemistry Module

## *New and Improved Functionality in Version 4.4*

---

### **ACCURATE BOUNDARY FLUXES**

Two new variables,  $nIs$  and  $nIl$ , are now available for accurately computing the normal current density in the electrode and electrolyte phases, respectively.

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## *New Model in Version 4.4*

---

### **ELECTROCHEMICAL GLUCOSE SENSOR**

Electrochemical glucose sensors use amperometric methods to measure the concentration of glucose in a sample. This example models the diffusion of glucose and ferri/ferrocyanide redox mediators in a unit cell of electrolyte above an interdigitated electrode. The sensor gives a linear response over a suitable range of concentrations. The Electroanalysis interface is used to couple the chemical species transport to the electrolysis at the working and counter electrodes, and the glucose is oxidized by the glucose oxidase enzyme in the solution according to Michaelis-Menten kinetics.

# Electrodeposition Module

## *New and Improved User Interfaces and Features in 4.4*

---

### **PRIMARY CURRENT DENSITY DISTRIBUTION INTERFACES**

You can specify that you want to model primary current distribution in the physics interfaces for corrosion by choosing the new Electrodeposition, Primary interface. Previously, you had to specify this from the Electrodeposition, Secondary interface.

### **ACCURATE BOUNDARY FLUXES**

Two new variables,  $nI_s$  and  $nI_l$ , are now available for accurately computing the normal current density in the electrode and electrolyte phases, respectively.

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## *New Model in Version 4.4*

---

### **ELECTROCOATING ON A CAR DOOR**

This model simulates the primary current distribution of the electrocoating of a car door. The distribution of the deposited paint's thickness becomes more uniform due to the paint's high resistivity. A varying film resistance together with a constant electrolyte conductivity are used to describe the charge transport in the electrolyte.

## *Backward Compatibility With Version 4.3*

---

### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

## **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Brinkman Equations
- Free and Porous Media Flow

See [CFD Module](#) for details.

## **MOBILITY**

In physics user interfaces modeling migration (Transport of Diluted Species; Tertiary Current Distribution, Nernst-Planck; and Electrodeposition, Tertiary Nernst-Planck) anisotropic mobility is now supported, and the mobility can be set to follow the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

# Fatigue Module

## *New Functionality in Version 4.4*

---

### **THERMAL FATIGUE**

The Fatigue Module has added functionality for thermal fatigue through two families of fatigue models: one predicts fatigue based on the inelastic strains, and the other predicts fatigue based on the dissipated energy. Both models are also suitable for low-cycle fatigue prediction in ductile materials.

#### *Energy-Based Fatigue Models*

The energy-based fatigue option provides fatigue models that are based on energy dissipation. Two models are available:

- Morrow
- Darveaux

The Morrow model uses pointwise fatigue life evaluation, while the Darveaux model computes fatigue life based on volume-averaged energy dissipation. The Darveaux model is only available at the domain level, while the Morrow model is available on all dimensional levels. The volume average of the Darveaux model can be evaluated in two ways: for the Individual domains option, each individual geometric domain is evaluated separately; for the Entire selection option, the volume average is evaluated over all geometric domains simultaneously. Because the Darveaux model separates the total life into crack initiation and crack propagation, it is possible to evaluate the number of cycles necessary for each event.

In both these fatigue models, different types of energy change can be evaluated. The following are predefined:

- Creep dissipation density
- Plastic dissipation density
- Total dissipation density
- User defined

The first three options require that the evaluated material be modeled with nonlinear materials, and that the calculation of energy dissipation is enabled using the Advanced Physics Options. The User defined option allows you to specify a custom energy density variable and use it in one of the above models. This can be done by combining

existing energy variables or by defining new energy variables based on equations using one of the Mathematics interfaces for PDEs and ODEs.

#### *Coffin-Manson Type Strain-Based Fatigue Models*

The family of strain-based models has been extended with a model based on the Coffin-Manson relation. This model is frequently used for low-cycle fatigue evaluation. It has been modified so that different types of inelastic strains can be used in the Coffin-Manson relation. The following strain types are available:

- Effective creep strain
- Effective plastic strain
- User defined

The User defined option allows you to evaluate all the other strains defined in any of the structural interfaces or to evaluate a customized strain expression based on equations using one of the Mathematics interfaces for PDEs and ODEs. This makes it possible to evaluate different shear and normal strain components. It is even possible to evaluate different creep contributions, such as secondary creep, when simulating fatigue. The original Coffin-Manson relation is obtained by selecting Effective plastic strain as the Inelastic strain option.

#### *New Model in Version 4.4*

---

#### **THERMAL FATIGUE IN A SOLDER JOINT OF A SURFACE MOUNT RESISTOR**

A new tutorial model shows a surface mount resistor subjected to accelerated thermal cycling. A cycled temperature change of 50 °C takes place over a two-minute period and is followed by a dwell of three minutes. Thermal stresses are introduced by the differences in thermal expansion in different parts of the assembly. The solder joint that connects the resistor to the printed circuit board is the weakest link in the assembly. It responds nonlinearly to changes in both temperature and time, and is modeled using the Garofalo creep material model. To assess the structural integrity of the component, a fatigue analysis is made based on creep strain and dissipated energy. Several cycles of heating and cooling are simulated, followed by a fatigue study.

# Geomechanics Module

*New Functionality in Version 4.4*

---

## **PLASTICITY HARDENING DATA FROM MATERIAL**

There is now a From material selection option for the Hardening function list in the Plasticity node. This makes it easier to build your own material libraries with elastoplastic material properties.

## **DISSIPATED PLASTIC ENERGY DENSITY**

Dissipated plastic energy can now be computed for Creep, Plasticity, and Viscoelasticity, which are all available as subnodes for a Linear Elastic Material node. However, doing such a computation adds one extra degree of freedom to the solving process, which requires extra computational resources. You can control whether to compute the dissipated plastic energy or not by enabling and disabling it in the Energy Dissipation section in the Linear Elastic Material and Hyperelastic Material settings windows. This section is only displayed if Advanced Physics Options is enabled.

# Heat Transfer Module

*New and Improved Functionality in Version 4.4*

---

## **NEW METHODS FOR RADIATION IN PARTICIPATING MEDIA**

Two new fast and memory-efficient methods are available for **Radiation in Participating Media**:

- Rosseland approximation
- P1 approximation

These are approximate methods and are not as accurate or as general as the discrete ordinate method, which has been available since earlier versions of the software. Yet they solve simulations with radiation in participating media much faster. The Rosseland approximation is only available with the full **Heat Transfer in Solids** and **Fluids** interfaces, not in the radiation-only interface for participating media. Both methods are also available for 2D axisymmetric geometries. As a matter of comparison, the verification model *Radiative Heat Transfer in Finite Cylindrical Media* solves in a couple of seconds using the P1 approximation method—as opposed to more than two hours with the discrete ordinate method.

## **HEAT TRANSFER IN BIOLOGICAL TISSUE WITH DAMAGE INTEGRAL ANALYSIS**

Tissue necrosis (permanent damage or death of living tissue) occurs if either of the following conditions is met:

- Too much thermal energy has been absorbed
- A critical temperature has been exceeded (typically boiling)

The Damaged Integral Analysis is utilized in medical treatment and surgical methods based on tissue heating. Thermal energy absorption is often modeled by so-called damage integrals. The Biological Tissue interface in the Heat Transfer Module includes two forms of damage integral: *Temperature threshold* and *Energy absorption*.

The *Temperature threshold* form is a simple integrated inequality representing how long tissue has been above a certain temperature. User-defined parameters include

**Damage temperature, Damage time, and Necrosis temperature.** In this case, tissue necrosis is assumed to occur under the following circumstances:

- When the tissue temperature exceeds a given damage temperature for more than a certain time period
- Instantly after the tissue temperature exceeds the necrosis temperature

The energy absorption form uses an Arrhenius type expression to directly estimate absorbed energy. User-defined parameters include **Frequency factor** and **Activation energy** for the integrated Arrhenius equation.

The material properties of the damaged tissue take into account the influence of tissue damage. The conductivity and the effective heat capacity (density multiplied by heat capacity at constant pressure) are modified with respect to the volume fraction of necrotic tissue. Six new generic biomaterials are available in the material library shipped with the Heat Transfer Module: Bone, Fat, Liver, Muscle, Prostate, and Skin.

The new physics interface for heat transfer in biological tissue is available for Heat Transfer in Solids as well as for any multiphysics combination where this physics interface participates, including the following:

- Joule Heating
- Induction Heating
- Microwave Heating
- Laser Heating
- Thermal Stress
- Joule Heating and Thermal Expansion

Two models, Microwave Cancer Therapy and Tumor Ablation, which are available in the Model Library of the Heat Transfer Module, have been updated with the new damage integral analysis.

#### **THE THERMOELECTRIC EFFECT MULTIPHYSICS NODE**

The new Thermoelectric Effect multiphysics interface is available in the Heat Transfer Module. It is a multiphysics combination of the Electric Currents and Heat Transfer in Solids interfaces. Choosing the interface from the Model Wizard will subsequently add a dedicated Multiphysics node in the Model Builder to allow you to control the couplings between the individual physics. Alternatively, you can start with the individual physics, such as the Electric Currents interface, and then add the Heat

Transfer in Solids interface later on, which will automatically also add the Multiphysics node.

Two thermoelectric materials have been added to the Material Library: bismuth telluride and lead telluride.

#### **NEW METHODS AND VARIABLES FOR HEAT AND ENERGY BALANCES**

The mathematical formulations for heat transport computations have been enhanced. This has resulted in revised variables for heat and energy balances. In addition, heat fluxes on boundaries can now be computed with increased accuracy.

##### *Variables for Boundary Fluxes*

For boundary fluxes, the following variables now provide the accurate value of fluxes when they are available:

- `ndflux`: normal convective heat flux;
- `ntflux`: normal total heat flux;
- `nteflux`: normal total energy flux.

These boundary flux variables are available in all physics interfaces for Heat Transfer and all multiphysics interfaces that include Heat Transfer. The new method is active by default, but can be switched off by clearing the **Compute boundary fluxes** check box in the **Discretization** section of the physics interfaces for heat transfer. To display the **Discretization** section, enable it from the **Show** menu in the Model Builder toolbar. If this check box is not selected, then the computations of fluxes on boundaries are performed by extrapolating values from within neighboring finite elements. This was the method used in COMSOL 4.3b and earlier versions.

##### *Variables for Global Heat and Energy Balances*

Energy balance is now easier and faster to check through the introduction of new global variables. Evaluating these scalar quantities replaces the need for integrating all the contributions to the energy balance over domains, boundaries, edges, and points.

The following global variables have been added to Heat Transfer in Solids, Heat Transfer in Fluids, Heat Transfer in Porous Media, Heat Transfer with Phase Change, and Heat Transfer in Biological Tissue interfaces:

- `dEiInt`: total accumulated heat power
- `dEi0Int`: total accumulated energy power
- `ntfluxInt`: total net heat power
- `ntefluxInt`: total net energy power

- QInt: total heat source
- WInt: total work source
- WnsInt: total fluid losses

The following global variables have been added to most heat transfer boundary conditions:

- Tave: weighted average temperature
- ntfluxInt: total net heat power
- ntefluxInt: total net energy power
- ntfluxInt\_u: total net heat power, upside
- ntefluxInt\_u: total net energy power, upside
- ntfluxInt\_d: total net heat power, downside
- ntefluxInt\_d: total net energy power, downside

#### **LINE AND POINT HEAT SOURCES ON AXIS OF SYMMETRY**

For 2D axisymmetric models, you can now define line and point heat sources on the axis of symmetry. The previous point heat source has been replaced by a line heat source applicable at points and now provides a **Total Line Power** option. The **Line Heat Source on Axis** is applicable on the symmetry axis only. The **Line Heat Source** feature is applicable at points and represents a line revolved about the axis of symmetry. This feature is not applicable on the symmetry axis. The **Point Heat Source on Axis** feature is a point source that is applicable only at points on the symmetry axis. In 2D, the **Point Heat Source** has been replaced by a **Line Heat Source** that can be applied at points. It now provides a **Total Line Power** option through the selection of thickness and multiple points, representing lines.

#### **HEAT TRANSFER IN POROUS MEDIA**

##### *Coordinate Systems for Anisotropic Porous Media*

For Heat Transfer in Porous Media, it is now possible to choose any coordinate system from the **Definitions** node. This is useful when defining heat transfer in anisotropic materials where the thermal conductivity varies with the direction.

##### *Better Usability: Heat Transfer in Porous Media With a List of Fluid Materials*

You can now easily define multiple porous materials using material selections, and then link the fluid material property to another material from the domain material list. This avoids multiple instances of the Heat Transfer in Porous Media feature.

## HEAT TRANSFER IN THIN SHELLS

The Heat Transfer in Thin Shells interface has several important updates. It has been combined with the Deformed Geometry interface for time-dependent studies. This is useful when, for example, modeling a growing shell where matter is added during the simulation. The names of features in Heat Transfer in Thin Shells have been consolidated. Heat sources and heat fluxes are now called *Heat Source* and *Heat Flux* and come with dedicated icons that indicate which dimension they apply to. Heat Transfer in Thin Shells is now also available for 2D and 2D axisymmetric models.

## FAN, INTERIOR FAN, GRILLE, SCREEN, AND VACUUM PUMP BOUNDARY CONDITIONS

The boundary conditions Fan, Interior Fan, Grille, Screen, and Vacuum Pump are now available in both CFD Module and Heat Transfer Module.

## ACCURATE FLUXES UPDATED

The specific accurate flux variables have been removed. The standard variables now evaluate:

- With accurate flux method when the **Compute boundary fluxes** is selected in the **Discretization** section on the interface.
- With the extrapolated method when it is unselected.

The variables with the `_acc` suffix can be replaced by the following variables:

- Normal conductive heat flux:
  - `ndflux_acc` -> `ndflux` (at all boundaries)
  - `ndflux_acc_u` -> `ndflux_u` (at interior boundaries only)
  - `ndflux_acc_d` -> `ndflux_d` (at interior boundaries only)
- Normal total heat flux:
  - `ntflux_acc` -> `ntflux` (at all boundaries)
  - `ntflux_acc_u` -> `ntflux_u` (at interior boundaries only)
  - `ntflux_acc_d` -> `ntflux_d` (at interior boundaries only)
- Normal total energy flux:
  - `nteflux_acc` -> `nteflux` (at all boundaries)
  - `nteflux_acc_u` -> `nteflux_u` (at interior boundaries only)
  - `nteflux_acc_d` -> `nteflux_d` (at interior boundaries only)

These variables are available for Heat Transfer interfaces and their multiphysics combinations, which include Non Isothermal Flow, High Mach Number Flow, and Slip Flow.

The previous variables names, with a `_acc` at the end, remain available but are considered obsolete and will be removed in future versions.

MPH-files that are opened in COMSOL Multiphysics 4.4 automatically use updated variable names. However, if evaluating these variables is needed, it is necessary to update the solution once. After that, no additional manipulation is needed.

For Model files for Java using COMSOL Multiphysics API, an manual update of variables names is recommended. Even though former variable names are still handled in the present version, future versions will not support them.

### **INTERIOR FLUXES UPDATED**

Issues in the interior flux variable definitions have been fixed.

The heat fluxes `ncflux_u`, `ncflux_d`, `ntr1flux_u`, `ntr1flux_d`, `nteflux_u`, and `nteflux_d`, defined on interior boundaries, now use the upside and downside normal vectors `un` and `dn` instead of `n`.

The interior conductive heat fluxes `ndflux_u` and `ndflux_d` are now `-uflux_spatial(T)` and `-dflux_spatial(T)`, so that they return outgoing upside flux and outgoing downside flux.

### **RADIATION IN PARTICIPATING MEDIA**

`qw`, `qin` and `qout` have been replaced by `qr_net`, `qr_in`, and `qr_out`, which are defined by:

- $qr\_in = -qin$ , is always  $> 0$
- $qr\_out = qout$ , is always  $> 0$
- $qr\_net = qr\_in - qr\_out = -qw$ , is positive when the domain receives more radiation than it emits.

For backward compatibility the older variables, `qw`, `qin`, and `qout` remain available.

**THERMAL PERFORMANCE OF WINDOWS, FOLLOWING NORM ISO 1077-2:2012**

These benchmark models reproduce the ten test cases from the ISO 10077-2:2012 standard for thermal performance of windows. Thermal performance is evaluated through thermal conductance and transmittance, and results are validated against published data.

**DISK-STACK HEAT SINK**

This model shows the cooling effects of a disk-stack heat sink on an electronic component. The heat sink consists of several thin aluminum disks stacked around a central hollow column. Such a configuration allows for the air cooling of the surfaces of aluminum fins.

**THERMAL EFFECTS OF THE SUN AS AN EXTERNAL HEAT RADIATION SOURCE**

This model, containing a beach umbrella and two styrofoam coolers, simulates the thermal effects of the sun using an external heat radiation source. The simulation runs from 10 a.m. to 4 p.m. During part of the day, the umbrella protects the box from the sun irradiation. This model uses the external radiative heat source feature with the solar position option. The sun's position and shadow effects are automatically updated during the simulation.

**COOLING AND SOLIDIFICATION OF METAL MODEL**

This example shows a continuous casting process using the Heat Transfer with Phase Change and Surface-to-Ambient radiation interfaces. Liquid metal is poured into a mold of uniform cross section. The outside of the mold is cooled and the metal solidifies as it flows through it. When the metal leaves the mold, it is completely solidified on the outside, but is still liquid inside. The metal will continue to cool and eventually solidify completely, at which point it can be cut into sections. This model does not compute the flow field of the liquid metal; it is assumed that the velocity of the metal is constant throughout. The phase transition from molten to solid state is modeled using a temperature-dependent specific heat. Techniques for reaching convergence and selecting a proper mesh are addressed in this highly nonlinear model.

## **THERMOELECTRIC LEG MODEL**

This model of a thermoelectric leg shows Peltier cooling. It is a verification model that demonstrates how to use the new Thermoelectric Effect multiphysics interface and reproduces results available in literature.

### *Backward Compatibility With Version 4.3b*

---

## **INTERIOR FLUXES UPDATED**

Because the interior flux variable definitions have been revised, models that use interior fluxes in definition, physics interfaces, or postprocessing show different results in version 4.4 than in earlier versions.

## **IMPROVED ENERGY BALANCE**

Changes introduced in the equation formulation to enable better energy balance may slightly modify the results.

Among the changes, the reference enthalpy,  $H_{\text{Ref}}$ , the value of which is defined up to a constant, is now set to 0 J/kg at the reference temperature and pressure. This changes the absolute values of the enthalpy and several variables based on it. In particular, the value of the convective heat flux is changed. Even the sign may be changed in some cases. However, the quantity of interest, the net convective heat flux, is identical because the shift induced by the change of  $H_{\text{Ref}}$  applies for the enthalpy definition at the inlet as well as at the outlet.

### *Backward Compatibility With Version 4.3a*

---

## **HEAT TRANSFER IN POROUS MEDIA**

Old models using the Porous feature will be supported in COMSOL Multiphysics interface. It is however recommended to replace this feature by Heat Transfer in Porous Media (see [Obsolete Features and Variables](#)). For a model using the API, it will be needed to update the model to use Heat Transfer in Porous Media.

## **SURFACE TO SURFACE RADIATION**

The surface radiosity, previously named  $J$ , is now named  $ht.J$  by default (assuming that  $ht$  is the physics interface tag).

Models created using versions 4.3 and 4.3a that contain surface-to-surface radiation support the old syntax in multiphysics interfaces. It is, however, recommended that

you update the models to use the new syntax. Models created in a version before 4.2a need to be opened in version 4.3 or 4.3a and resaved before being opened in 4.4.

For a model using the API, you must update the model to use the new syntax.

#### **RADIATION IN PARTICIPATING MEDIA**

The radiative intensity variables names, previously named `I1` and so on, became `ht.I1` (when there are less than 10 discrete directions) or `ht.I01` (when there are 10 or more discrete directions) by default.

Models created in versions 4.3 and 4.3a that include radiation in participating media support the old syntax in multiphysics interfaces. It is, however, recommended that you update the models to use the new syntax. Models created in a version before 4.2a need to be opened in version 4.3 or 4.3a and resaved before being reopened in 4.4. In addition, for all versions, it may be necessary to regenerate the default solver to get the correct solver settings.

The S2 quadrature in 2D has been replaced by the LSE symmetric quadrature. It has been updated so that it coincides with the other quadratures. LSE symmetric quadratures were already used for S4, S6, and S8 in 2D and for all 3D quadratures. This quadrature update can result in significant changes in 2D models using S2 quadrature. However, the theoretical error is not different and the 2D and 3D implementations are now consistent.

#### *Backward Compatibility With Version 4.3*

---

#### **THE DEFAULT MODEL LIST HAS BEEN REMOVED**

Heat transfer user interfaces no longer have the default model list. This list was used to change the default feature for heat transfer models. You can obtain a similar result by adding the corresponding domain feature immediately below the default feature and setting its domain selection to *All domains*.

Any model files for Java that modify the default model value require a manual update.

#### **IMPROVED STABILIZATION OF HEAT TRANSFER IN SOLIDS**

The streamline diffusion stabilization for *Heat Transfer in Solids* and *Biological Tissue* features has been improved. It now accounts for contributions from linear source terms from the *Heat Source*, *Out-of-Plane Convective Cooling*, *Out-of-Plane Radiation*, and *Out-of-Plane Heat Flux* features. This improves the robustness of the

convergence when these contributions are large. This change may modify the convergence behavior of existing models.

#### **FRAME SELECTION IN DISCRETIZATION SECTION**

The *Frame type* list, which was previously available when moving mesh was detected, has been removed. The frame type is now controlled by the features.

#### **UPDATE OF FEATURE VARIABLE NAMES**

In order to avoid name conflicts between feature variable names some of them have been renamed. This change concerns contributive features. Because they are contributing, it is possible to have similar features active on the same boundary. To make it possible to distinguish the variables from each feature, the feature scope has been introduced in the variable name so that variable names now use a prefix. These changes affect the following feature variables:

- Heat flux
- Out-of-plane heat flux
- Convective cooling
- Out of plane convective cooling
- Boundary heat source
- Heat source
- Line heat source
- Point heat source
- Edge/Point heat flux
- Electrochemical reaction heat flux
- Reaction heat flux

For example, in a *Boundary Heat Source* feature, `ht.surf` has been renamed into `ht.bhs1.surf` (assuming that `ht` is the physics interface tag). In the *Heat Flux* feature, the variable that was previously named `ht.q0_hf1` is now `ht.hf1.q0`.

Any model files for Java that use the old variable names in expressions (such as expressions used for plotting or data evaluation that include such old variable names) need to be updated manually.

#### **NEW DEFAULT FLUID FEATURES AND OPAQUE SUBFEATURE**

All Model MPH-files created in version 4.3 are converted to appear and behave the same way as in 4.4.

Due to these new default features, model files for Java can be simplified. In addition, model files for Java that add the *Fluid* feature or the *Opaque* subfeature with the default tag require a manual update to avoid duplicate tag conflicts.

#### **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW USER INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Turbulent Flow,  $k$ - $\epsilon$
- Turbulent Flow, low-Re  $k$ - $\epsilon$
- Non-Isothermal Flow
- Conjugate Heat Transfer

See the CFD Module Release Notes for details.

Weak constraints for the Interior Wall feature are no longer available.

#### **REVISION TO THE TURBULENCE MODELS**

The formulations of some variables in the turbulence models have been revised in order to improve accuracy. Simulations using a turbulence model can display a different convergence behavior in version 4.4 compared to version 4.3, and the numerical result can differ slightly between the versions.

#### *Obsolete Features and Variables*

---

The following features are still supported when models created in older versions are loaded in version 4.4. However, they will be removed in future versions. Hence, you should replace them by the new corresponding feature.

- **Porous Matrix** is replaced by **Heat Transfer in Porous Media**.
- **Infinite Element** domain feature should be removed and **Infinite Element Domain** should be declared in the **Definitions** node.
- **Electrochemical Heat Flux** is replaced by **Boundary Heat Source**.
- **Reaction Heat Flux** is replaced by **Boundary Heat Source**.
- **Heat Transfer in Participating Media** is replaced by **Heat Transfer with Radiation in Participating Media**.

The table below contains a list of variables that are available but obsolete. They should each be replaced by the new corresponding expressions because they will be removed in future versions.

TABLE I-1: OBSOLETE VARIABLES AND CORRESPONDING EXPRESSIONS

<b>OBSOLETE VARIABLE</b>	<b>CORRESPONDING EXPRESSION</b>
qin	-qr_in
qout	qr_out
qw	-qr_net
ndflux_acc	ndflux
ndflux_acc_u	ndflux_u
ndflux_acc_d	ndflux_d
ntflux_acc	ntflux
ntflux_acc_u	ntflux_u
ntflux_acc_d	ntflux_d
nteflux_acc	nteflux
nteflux_acc_u	nteflux_u
nteflux_acc_d	nteflux_d

# MEMS Module

## *New Functionality in Version 4.4*

---

### **IMPROVED WORKFLOW FOR THERMAL STRESS AND JOULE HEATING**

The new Multiphysics nodes improve modeling workflow, making it possible to progressively increase the complexity of the system being modeled. For instance, you can now begin by solving a simple thermal problem and then add the structural effects and thermal stress coupling. It is still possible to add the thermal and structural effects simultaneously using the Thermal Stress physics option in the Model Wizard; this option automatically adds the Heat Transfer in Solids and Solid Mechanics interfaces together with the appropriate Multiphysics nodes for multiphysics couplings.

This same functionality has also been included in the Joule Heating and Thermal Expansion interface. Once again, the contributing physics interfaces can be added one at a time, and their couplings managed from the Multiphysics node. Selecting the Joule Heating and Thermal Expansion interface in the Model Wizard sets up the Heat Transfer in Solids, Solid Mechanics, and Electric Currents interfaces together with the Multiphysics node. This approach makes it straightforward to activate or deactivate the contributing interfaces. Consequently, you can choose to solve the same model for each physics individually, for a combination of multiphysics couplings, or for all three at the same time.

### **GRAVITY, CENTRIFUGAL, CORIOLIS, AND EULER FORCES**

Mass forces and loads such as gravity, centrifugal forces, Coriolis forces, and Euler forces can now be added using two new options: Gravity and Rotating Frames. These options make it easy to define loads that act on all objects having mass—that is, domains with mass density, point masses, added mass, rigid connectors with mass, and so on. The forces and loads are added from the domain level even though they may be automatically applied to features at boundaries, edges, and points.

The Rotating Frame feature includes all types of fictitious forces occurring in a rotating system. By default, Centrifugal force and Spin-softening are included.

Because Added Mass can be used to describe load effects that are not true structural masses, sometimes the contribution from Added Mass is not wanted. The option to include or exclude the contribution is controlled in a new section called Frame Acceleration Forces.

## **NEW ALGORITHM FOR DETECTING WHETHER SOURCE AND DESTINATION ARE IN CONTACT**

The detection of contact on a *contact pair* is now based on the structural mechanics interface results, which improves the accuracy of the contact definition. Consequently, in any pair feature, use a contact pair only when the component contains an active structural mechanics contact pair feature. In other cases, use an *identity pair*. Only one contact pair is supported for each boundary pair feature.

### *New Model in Version 4.4*

---

## **RF MEMS SWITCH**

This model analyzes an RF MEMS switch consisting of a thin micromechanical bridge suspended over a dielectric layer. A DC voltage greater than the pull-in voltage is applied across the switch, causing the bridge to collapse onto the dielectric layer and increasing the capacitance of the device. A penalty-based contact force is implemented to model the contact forces as the bridge comes into contact with the dielectric. The dielectric itself is represented by a spatially varying function for the dielectric constant between the two terminals.

### *Backward Compatibility With Version 4.3b*

---

For models using contact, plot expressions are not updated when you open an old model where the pair names have been edited manually. Also, opening an old model where a pair name has been manually edited might affect the solver configuration. It is recommended that you regenerate a new default solver configuration.

# Microfluidics Module

## *New Functionality in Version 4.4*

---

### **REVISED FORMULATION FOR LAMINAR INFLOW AND LAMINAR OUTFLOW**

The formulation of the laminar inflow and laminar outflow conditions has been corrected. The modified formulation will give a more accurate mass flux. Some models may produce a slightly different flow field. Laminar inflow and laminar outflow are available in the fluid flow physics interfaces.

### **OUTLET BOUNDARY CONDITION**

The Outlet boundary condition for fluid flow has been revised for improved mass conservation as well as faster and more robust convergence. The new Outlet feature has only one Pressure option, which corresponds to the Normal Stress option of earlier versions. In addition to specifying the pressure, there are two new options for specifying a normal flow and suppressing backflow:

- Normal flow prescribes a tangential velocity of zero at the outlet. This condition can be expected if the outlet represents a straight pipe or channel. However, it is not selected by default because the flow can be disturbed upstream of the outlet, potentially altering the solution significantly.
- Suppressing backflow reduces the tendency for fluid to enter the domain from the outside. It does not completely prevent backflow, and in the case that backflow occurs, this option locally decreases the specified pressure. Controlling backflow is important when combining fluid flow with other transport equations, such as mass and heat transport. If the flow reverses, the outlet boundary condition for the transport equations is no longer valid; this can lead to convergence problems or nonphysical solutions. The Suppress backflow option is therefore selected by default.

The Outlet boundary condition has been revised for the new version in the following physics interfaces in the Microfluidics Module:

- Two-Phase Flow, Moving Mesh
- Slip Flow

This change is valid for laminar flow, Stokes flow, and turbulent flow when applicable. The Outlet boundary condition from previous versions still exists but has been

excluded from the physics context menu. Models created in previous versions retain the old Outlet feature, but new Outlet nodes that are added have the new functionality.

#### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

#### **POINT AND LINE MASS SOURCES**

A point source can be used to simulate a source distributed over a very small volume. While it can be applied to points in 3D or on the symmetry axis in axisymmetric models, its actual effect is distributed throughout the close vicinity of the point. The size of the distribution depends on the mesh and strength of the source—a finer mesh spreads the source over a smaller region but results in a more extreme pressure value. A line source in 3D and 2D axisymmetric models represents a source emanating from a tube with a very small cross-sectional area. Line sources can be added to lines in 3D, to the symmetry axis in 2D axisymmetric models, and to points in 2D, for which they represent the cross section of a tube of very small area.

Point and line mass sources for fluid flow are included as contributions to the continuity equation. This functionality has been added to the following physics interfaces for fluid flow in the Microfluidics Module:

- Two-Phase Flow
- Two-Phase Flow, Moving Mesh (requires the Microfluidics Module)

#### *New Model in Version 4.4*

---

#### **MICROCHANNEL DISPERSION OPTIMIZATION**

This new example model shows how to optimize the shape of a curved microchannel to improve the performance of a chemical species detector downstream of the curve. The shape of the channel is defined as a set of Bézier curves, which depend on five optimization parameters. The model computes the values of these parameters to minimize the difference in the time it takes for the fluid moving along the inner and outer walls of the channel to reach the detector. The model uses the new gradient-free optimization solver for *bound optimization by quadratic approximation* (BOBYQA), a trust region gradient-free optimization solver written by Professor M.J.D Powell (Cambridge).

### *Backward Compatibility With Version 4.3a*

---

The Free Molecular Flow and Transitional Flow user interfaces now form the Molecular Flow Module. A Molecular Flow Module license is required to use these interfaces.

### *Backward Compatibility With Version 4.3*

---

#### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u} / \varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

#### **MOBILITY**

In physics user interfaces modeling migration (Transport of Diluted Species) anisotropic mobility is now supported, and the mobility can be set to follow the Nernst-Einstein relation. The default setting when making new models has been changed to use the Nernst-Einstein relation. Model files for Java generated prior to 4.3b using Migration will have to be modified manually to account for these changes.

# Mixer Module

## *Introduction*

---

The Mixer Module is a new add-on module to the CFD Module.

The Mixer Module is designed for modeling mixers with rotating parts, which are used in many industrial processes in the fine chemicals, consumer products, pharmaceutical, and food industries. These mixers are usually used for multiple purposes and often in batch processes.

## *Key Functionality*

---

The Mixer Module adds the following key functionality on top of the CFD Module:

- Rotating Machinery, Non-Isothermal Flow interface
- Rotating Machinery, Reacting Flow interface
- Rotating Machinery, Turbulent flow interface with k-omega and low-Re k-epsilon turbulence models.
- Free surface computation

# Molecular Flow Module

*New Model in Version 4.4*

---

## **MONTE CARLO RECONSTRUCTION OF NUMBER DENSITY**

It is possible to model molecular flows using a particle-based approach with the Particle Tracing Module. A new model has been added that compares the computed number density in an S-bend geometry. This model uses a particle-based approach and the Free Molecular Flow physics interface. While the results are in good agreement, the particle-based approach introduces statistical noise and takes more than 100 times longer to solve. This shows the advantage of the angular coefficient method in the Molecular Flow Module over the particle-based approach.

# Multibody Dynamics Module

*New Functionality in Version 4.4*

---

## **FIXED JOINT, DISTANCE JOINT, AND UNIVERSAL JOINT**

Three additional joint types have been added to the Multibody Dynamics interface: Fixed Joint, Distance Joint, and Universal Joint. They are available under More Joints in the Joints menu. These new joint types differ from the ones already available in that they are more abstract and do not have any subnodes. Below is a summary of the new joint types:

- A Fixed Joint welds two parts together.
- A Distance Joint is similar to a rigid link with ball joints at the ends, but the distance may change because it is possible to set a variable in the Extension field.
- A Universal Joint is also known as a Cardan joint.

## **JOINTS WITH FRICTION**

It is now possible to add friction to the Prismatic, Hinge, Cylindrical, Screw, Planar, and Ball joint types. It is only possible to include friction in multibody dynamics models using time-dependent studies.

# Nonlinear Structural Materials Module

*New Functionality in Version 4.4*

---

## **PLASTICITY HARDENING DATA FROM MATERIAL**

There is now a From material selection option for the Hardening function list in the Plasticity node. This makes it easier to build your own material libraries with materials that have elastoplastic properties.

## **DISSIPATED PLASTIC ENERGY DENSITY**

Dissipated plastic energy can now be computed for Creep, Plasticity, and Viscoelasticity, which are all available as subnodes for a Linear Elastic Material node. However, doing such a computation adds one extra degree of freedom to the solving process, which requires extra computational resources. You can control whether you want to compute the dissipated plastic energy or not. It can be enabled and disabled in the Energy Dissipation section in the Linear Elastic Material and Hyperelastic Material settings windows. This section is only displayed if Advanced Physics Options is enabled.

# Optimization Module

*New Functionality in Version 4.4*

---

## **NEW OPTIMIZATION SOLVERS**

Two new optimization solvers expand the scope of the Optimization Module. One of the solvers (BOBYQA) is a gradient-free method and can be applied to a great variety of optimization problems, including those that vary one or more geometry dimensions of a CAD model. The other solver (MMA) requires the derivatives to be computed and is more limited in its scope but, when applicable, converges much faster.

*Gradient-Free Optimization Solver: Bound Optimization by Quadratic Approximation*

The new *bound optimization by quadratic approximation* (BOBYQA) method is a trust region gradient-free optimization solver. The method uses an iteratively constructed quadratic approximation of the objective function that is valid in a region around the current iterate—the trust region. This solver is very efficient in that it requires fewer objective function evaluations than early-generation gradient-free optimization solvers. The method supports simple bounds but not general constraints, and it is expected to outperform the Nelder-Mead and coordinate search methods as the number of control variables grow.

The following gradient-free methods are available in COMSOL version 4.4:

- Coordinate search
- Monte Carlo
- Nelder-Mead
- BOBYQA

You can access these optimization methods from the Optimization study type. Control parameters are not limited to geometric dimensions but can represent nearly any quantity in a model, including parameters controlling the mesh.

*Gradient-Based Optimization Solver: Method of Moving Asymptotes (MMA)*

The *method of moving asymptotes* (MMA) is a gradient-based optimization solver written by Professor K. Svanberg at the Royal Institute of Technology in Stockholm, Sweden. It is designed with topology optimization in mind. The method is called GCMMA in the literature and is available in the Optimization Module under the name MMA.

The following gradient-based methods are available in COMSOL version 4.4:

- SNOPT
- MMA
- Levenberg-Marquardt

### *Backward Compatibility With Version 4.3a*

---

The Optimization check box in the Stationary and Time Dependent study steps have been removed. Instead, use the Optimization study step. The Optimization check box does appear in models created in earlier versions of COMSOL where you have selected that check box.

# Particle Tracing Module

*New Functionality in Version 4.4*

---

## **EFFICIENT CALCULATION OF PARTICLE-FIELD AND FLUID-PARTICLE INTERACTIONS**

A new approach to modeling particle-field and fluid-particle interactions is now available. In this approach, the particle trajectories are computed using a time-dependent study step, and the fields in the surrounding medium are computed using a stationary study step. These two steps are repeated until a self-consistent solution for the particle trajectories and the surrounding fields is reached. This procedure greatly reduces the number of model particles needed to model systems operating under steady-state conditions, like charged particle beams. The new approach makes it easy to quantify the amount of beam spreading occurring due to its self potential.

In the settings for the Charged Particle Tracing interface, changing the Release Type to Static causes all release features to provide a specified charged particle current. Similarly, in the Particle Tracing for Fluid Flow interface, changing the Release Type to Static causes all release features to provide a specified mass flow rate. The Particle Field Interaction or Fluid Particle Interaction feature then computes the space charge or force density exerted by the particles.

New solver nodes are available to compute the self-consistent interaction between particles and fields. The addition of the For and End For nodes to a Solver sequence allows a part of the sequence to run in a continuous loop. This approach allows the particle trajectories to be computed with a time-dependent solver and the fields with a stationary solver.

## **RELEASING PARTICLES IN A CONE**

It is now possible to specify the initial velocities of released particles in a cone with a user-specified angle between 0 and 180 degrees. Below is an overview of the new options and actions available:

- When using the Release from Grid or Release features, a new option, **Constant speed, cone** is available in the **Initial velocity** settings.
- You can specify the initial speed of the particles, the direction of the cone axis, and the cone angle.

- By releasing particles in a cone, it is now much easier to create models involving jets or sprays of incoming particles.
- The **Constant speed, cone** setting can be thought of as a generalization of the **Constant speed, hemispherical** and **Constant speed, spherical** settings; the latter two corresponding to the special cases of a 90-degree cone and a 180-degree cone, respectively.

#### **ELASTIC COLLISION COUNTERS**

It is now possible to count the number of elastic collisions between a model particle and background gas particles simply by selecting a check box. The following changes have been made:

- When the Collision Model is set to Monte Carlo in the settings for the Elastic Collision Force feature, a new section called Collision Statistics becomes available.
- In the Collision Statistics section, clicking the Count Collisions check box introduces a new degree of freedom for each particle, which is increased by one every time an elastic collision occurs.
- The variable introduced by the Count Collisions check box applies only to a specific Elastic Collision Force feature; this makes it possible to separately count the collisions of a model particle with several different background species.

#### **NEW DRAG MODEL: HAIDER-LEVENSPIEL**

A new option, Haider-Levenspiel, is available to compute the drag force for nonspherical particles. The mathematical model employed is similar to the Schiller-Naumann option, except the sphericity of the particles is taken into account. Nonspherical particles typically result in higher drag than spherical particles.

#### **REINITIALIZATION OF AUXILIARY DEPENDENT VARIABLES**

The Velocity Reinitialization and Elastic Collision Force features can now reinitialize auxiliary dependent variables whenever a velocity reinitialization takes place. In the settings for the Velocity Reinitialization feature, a new section, New Value of Auxiliary Dependent Variables, is available. Reinitialization can be activated or deactivated separately for each variable. The New Value of Auxiliary Dependent Variables section is also available in the settings for the Elastic Collision Force feature when the Collision Model is set to Monte Carlo. Use this section to reinitialize auxiliary variables every time a collision occurs.

In 2D and 2D axisymmetric models, the out-of-plane velocity component can be reinitialized as well. As a consequence, when the Collision Model is set to Monte Carlo

in 2D and 2D axisymmetric geometries, the results are now just as accurate as full 3D models.

#### **STATISTICS FOR EACH RELEASE FEATURE**

The total number of particles released by a given release feature is now available as a variable to use in equations and evaluate during results processing. This provides a convenient means of tracking the number of particles released from each feature, even when using a mesh-based initial position.

#### **MINIMUM, MAXIMUM, AND AVERAGE OPERATORS**

Minimum, maximum, and average operators are now available for the particles. These operators allow you to use the following in equations, as stop conditions, or during results processing:

- Minimum value of a variable, evaluated over all particles
- Maximum value of a variable, evaluated over all particles
- Average value of a variable, evaluated over all particles

These variables typically evolve over time and can be used, for example, in stop conditions to terminate the simulation when the average kinetic energy of the particles reaches some threshold. They have been added for the Newtonian and Lagrangian formulations.

### *New Models in Version 4.4*

---

#### **ELECTRON BEAM DIVERGENCE**

This new model simulates the propagation of charged particle beams at high currents where the space charge force generated by the beam significantly affects the trajectories of the charged particles. Perturbations to these trajectories, in turn, affect the space charge distribution. The Charged Particle Tracing interface includes an iterative procedure to efficiently compute the strongly coupled particle trajectories and electric field for systems operating under steady-state conditions. Such a procedure reduces the required number of model particles by several orders of magnitude, compared to methods based on explicit modeling of Coulomb interactions between the beam particles. A mesh refinement study confirms that the solution agrees with the analytical expression for the shape of a nonrelativistic, paraxial beam envelope.

This model requires the Particle Tracing Module and the AC/DC Module.

### **ION DRIFT VELOCITY BENCHMARK**

The drift velocity of Ar<sup>+</sup> is calculated using a Monte Carlo simulation in which the elastic collisions of argon ions with ambient neutrals are modeled explicitly. This model uses data from energy-dependent collision cross-section experiments. The average ion velocity values are consistent with the experimental data over a wide range of reduced electric field magnitudes. This agreement suggests that Monte Carlo simulations of elastic collisions between particles can be applied to a wide variety of devices.

### **ION FUNNEL**

This model investigates the focusing effect of an electrodynamic ion funnel. Because of their ability to operate at high background gas pressures, ion funnels are often used to improve the sensitivity of devices such as ion mobility spectrometers and mass spectrometers through coupling. The model uses a Monte Carlo collision setting to simulate the interaction of ions with the neutral background gas.

This model requires the Particle Tracing Module and the AC/DC Module.

# Plasma Module

*New Functionality in Version 4.4*

---

## **THERMAL DIFFUSION OF ELECTRONS**

The thermal diffusion of electrons contributes to electron current density, and this phenomenon can now be included in plasma simulations. Thermal diffusion is available as a property in the following physics interfaces: Capacitively Coupled Plasma, DC Discharge, Drift Diffusion, Inductively Coupled Plasma, and Microwave Plasma.

The additional contribution to the current density only makes a difference when the electron diffusivity is nonconstant (that is, a function of the electron temperature).

This option is only available for the finite element formulation.

# RF Module

*New Functionality in Version 4.4*

---

## **TRANSITION BOUNDARY CONDITION FOR HIGH CONDUCTIVITY**

The Transition boundary condition formulation has been improved to handle the case of an interior boundary that has very high material conductivity. You can use this boundary condition to model a layer of metal that is much thinner than any of the other model dimensions.

## **INTERIOR PORT BOUNDARY CONDITIONS**

When modeling electromagnetic wave sources in the RF Module, port boundary conditions are usually set at the exterior boundaries of a model to represent a source that is located outside the modeling space. However, sometimes it is more convenient to place the source inside the modeling domain. The new Slit Port introduces the capability to place a source at an interior boundary. This source can be a *domain-backed* or *PEC-backed* slit port:

The PEC-backed slit port introduces two boundary conditions at an interior boundary. On one side of the boundary, the PEC condition is applied; on the other side, you can use any of the regular port boundary conditions to excite a field propagating away from the boundary. The direction in which the field propagates away from the boundary is specified by the Port Orientation.

The domain-backed slit port, on the other hand, is a transparent boundary. It can excite a wave propagating away from the boundary, and any wave incident upon the domain-backed port passes through unimpeded. The domain-backed slit port is also useful for modeling periodic problems. When modeling structures that have many higher-order diffraction orders, such as gratings, you must account for each diffracted order with a separate port boundary condition. For 3D structures, diffraction can even occur in multiple planes. However, sometimes you may only want to know the bulk transmittance and reflectance of a periodic structure. In this case, you can use a domain-backed slit port. The slit port can insert an incident plane wave, entering at any angle, such that any wave reflected back toward the port passes through and into a PML placed behind it. The PML absorbs all of the higher-order modes simultaneously.

## **DEPOSITED PORT POWER**

In many microwave heating applications, it is desirable to control the amount of power that is deposited into the model. When you specify the deposited power, a feedback condition is added to the model and the applied power is adjusted so that the desired power is deposited within the model. This functionality has applications in biomedical RF heating, plasma modeling, and other areas.

## **NUMERIC PORT BOUNDARY MODE ANALYSIS WITH IMPEDANCE BOUNDARY CONDITIONS**

The Numeric Port boundary condition is used to compute the fields at a boundary to a waveguide where the field distributions cannot be computed analytically (such as rectangular, coaxial, or circular ports). These numeric port calculations can now consider the impedance boundary condition. The impedance boundary condition accounts for the effect of lossy walls, instead of assuming that the walls are perfect electric conductors. They can also account for the periodic boundary condition.

## **GAUSSIAN BEAM BACKGROUND FIELD**

The scattered field formulation is used to compute the scattering of electromagnetic fields off of an object. Typically, a uniform plane wave is specified as the background field, but the new Gaussian beam background field makes it possible to specify a Gaussian beam, propagating along one of the axis directions, of a specified beam waist and focal point. The polarization of the beam can also be specified.

## **NEW USER INTERFACES FOR MICROWAVE HEATING**

The new multiphysics functionality introduced in COMSOL 4.4 has made it easier to set up simulations involving microwave and RF heating. A dedicated Multiphysics node is added when the Microwave Heating multiphysics interface is chosen from the Model Wizard, along with the appropriate Electromagnetic Waves interface and the Heat Transfer in Solids interface.

The new multiphysics approach makes it possible to model the constitutive physics separately to understand a model's reactions to the contributing physics before considering the effects in a coupled multiphysics problem. This approach is also appropriate for managing the Study sequence when solving for electromagnetic waves in the frequency domain followed by the heat transfer in the time domain or stationary domain. Under the Multiphysics node you can keep track of the microwave heat sources on domains and boundaries as well as temperature nonlinearities in all of the material properties.

## **ADDITIONAL ANTENNA VARIABLES**

It is now possible to extract the *antenna gain* (in linear and dB scale), the *axial ratio* (in linear and dB scale), and the *far-field variables* in terms of theta and phi, elevation, and azimuthal angles.

### *Backward Compatibility With Version 4.3a*

---

The following models have been moved from the RF Module Model Library to the Wave Optics Module Model Library:

- Beam Splitter
- Photonic Crystal
- Scattering on Substrate
- Optical Scattering Off of a Gold Nanosphere
- Dielectric Slab Waveguide
- Fabry-Perot Cavity

# Semiconductor Module

*New Functionality in Version 4.4*

---

## **HETEROJUNCTION BOUNDARY CONDITION**

A *heterojunction boundary condition* is made available for interior boundaries by default. It determines the conditions for continuity of the normal component of the electric field and currents for homojunctions and heterojunctions. Two heterojunction models are defined:

- The continuous quasi-Fermi model (the default), which ensures current continuity by forcing both sides of the junction to have equal quasi-Fermi energies.
- The thermionic emission model, which defines the thermionic currents generated by the potential barrier created by the junction of dissimilar materials.

## **SUPPORT FOR SMALL-SIGNAL ANALYSIS**

The Semiconductor interface now supports the small-signal analysis, frequency domain study type. This study type enables the calculation of the AC device response, and therefore the computation of quantities such as the output conductance and the transconductance.

## **IMPACT IONIZATION**

In regions where the electric field perpendicular to the direction of current flow is high, electrons and holes are generated by *impact ionization*, which the Semiconductor Module now supports. This makes it possible to model avalanche effects in photodiodes and avalanche breakdown in MOSFETs. Initially, the current-voltage relation is linear (the ohmic region). As the drain-source voltage increases, the extracted current begins to saturate (the saturation region). As the drain-source voltage is further increased, the relation enters the breakdown region, where the current increases exponentially for small increases in the applied voltage. This effect is due to impact ionization.

## **FIELD-DEPENDENT MOBILITY MODELS**

Field-dependent mobility models are now available. Such mobility models allow for the decrease in the mobility of the holes and electrons according to an empirical model. There are two field-dependent mobility models available: the Caughey-Thomas model and the Lombardi surface model. The Caughey-Thomas model is generally applicable

for all types of semiconductor devices. The mobility of the electrons and holes is reduced with the increase of the component of the electric field parallel to the current flow. In contrast to the constant mobility case, this behavior inhibits current flow. The Lombardi surface model is applicable for modeling variations in mobility in the vicinity of surfaces—for example, under the gate of a MOSFET.

#### **INCOMPLETE IONIZATION**

The Dopant Ionization settings include a new option for *incomplete ionization*. At low temperatures in silicon and at room temperature for wide bandgap semiconductors, not all of the donors and acceptors are ionized. In these cases, the ionization of the donors and acceptors must be computed as a function of temperature. The dopant ionization is a function of the donor and acceptor energies and their corresponding degeneracy factors. A user-defined option is also available and makes it possible to specify the ionization ratio directly as a function.

#### **FINITE ELEMENT BASED LOGARITHMIC FORMULATION**

Because of the high degree of nonlinearity inherent in the drift-diffusion equations, the electron and hole number densities can span 10 orders of magnitude over a very small distance. This can create numerical instabilities when using the finite element method, for example, for negative concentrations. One way of handling this from a numerical point of view is to solve for the logarithm of the electron and hole number density. This formulation has been added to the Semiconductor interface as an additional discretization option.

#### **PHYSICS-BASED MESHING FOR SEMICONDUCTOR SIMULATIONS**

You can now use physics-based meshing in the Semiconductor interface. A very fine mesh is automatically generated for ohmic contact, thin insulator gate, and Schottky boundaries. This removes the need to manually create mesh sequences based on the settings in your model. The defaults have been carefully tuned for balancing accuracy and speed. Physics-based meshing is the new default and is recommended for all semiconductor models.

#### **METAL CONTACT BOUNDARY CONDITION**

The Metal Contact boundary condition includes the Schottky contact and Ohmic Contact boundary conditions from previous versions, now renamed the Ideal Schottky Contact and Ideal Ohmic Contact. This more general feature is a parent feature for both the Ideal Ohmic Contact (the case of a highly doped semiconductor with negligible barrier thickness) and the Ideal Schottky Contact (the case of a thermionic emission for large barrier thickness), which are ideal cases of metal contacts.

### **TERMINALS FOR GATES**

The Thin Insulator Gate boundary condition is now defined using terminals. Three terminal options are available:

- Voltage
- Charge
- Circuit

### **IMPROVED CIRCUIT COUPLING**

The Circuit option for the terminal type in the Metal Contact and the Thin Insulator Gate boundary conditions is now much easier to use. In the Electrical Circuits interface you can simply select the appropriate terminal in the External I vs. U, External U vs. I, or External I-Terminal feature. No additional equation contributions need to be added.

### **NEW MATERIALS FOR THE MATERIAL LIBRARY**

The material library in the Semiconductor Module now includes these materials:

- $\text{Al}(x)\text{Ga}(1-x)\text{As}$
- GaN (Wurtzite)
- GaN (Zinc Blende)
- GaP
- GaSb
- InAs
- InP
- InSb

### **CONTINUITY SETTINGS FOR DOPING AND FOR NONLINEAR FEATURES**

The new Continuity Settings enable ramping parameters to gradually introduce quantities into the equation system. For example, the doping concentration or the thermionic current can be slowly turned on, which makes it easier to solve strongly nonlinear models. To use this setting, you have to use the study extensions setting and ramp the continuation parameter as part of the intended study steps. The Continuation settings for introducing, for example, a thermionic current to the system provide three options:

- *No continuation.*

- *Use interface continuation parameter*, which links the continuation settings of the feature to an interface level continuation parameter (Cp) defined in the Semiconductor interface node. This makes it possible to ramp up multiple equation terms simultaneously.
- *User-defined*, which makes it possible for you to define a specific parameter for the doping continuation.

### **IMPROVED INITIAL VALUES SPECIFICATION**

There are now several ways to specify the initial conditions for the potential and concentration of electrons and holes. This added flexibility makes it easier to obtain a converged solution. The Initial Values options are:

- *Default*: Automatically chooses a proper Initial Value depending on the discretization method. For the finite volume method it is equivalent to the Equilibrium conditions, while for the finite element method it is equivalent to the Intrinsic concentrations.
- *Equilibrium conditions*: The field variables for electron concentration (N), hole concentration (P), and potential (V) are set to their calculated equilibrium values.
- *Equilibrium carriers only*: This has the same settings as Equilibrium conditions, but you can specify the initial value of the potential field (V).
- *Intrinsic concentrations*: The field variables for electron concentration (N) and hole concentration (P) have their initial values set to the intrinsic concentration. You can specify the potential field (V).
- *User-defined*: Three user inputs are displayed for each field variable: electron concentration (N), hole concentration (P), and potential (V).

These options maximize flexibility for models that require different initial values.

### **VARIABLES FOR COMPUTING CURRENTS AND CURRENT COMPONENTS**

The following new current variables for holes and electrons are available for display in the plot menu:

- Drift current density
- Diffusion current density
- Thermal diffusion current density

The norm and log of the norm of these quantities are also available.

# Structural Mechanics Module

*New Functionality in Version 4.4*

---

## **FAST CONTACT WITH THE PENALTY METHOD**

A new approximate *penalty contact method* is now available, which is more robust and converges faster than the standard augmented Lagrangian formulation. It avoids solving for the degrees of freedom associated with the contact pressure and negates the need for a special segregated solver. While expediting the solving process, the penalty formulation does not converge toward a zero gap distance between contact surfaces, and the formulation that estimates contact pressure is not as precise as the heavy-duty augmented Lagrangian formulation. The contact pressure and friction force formulations are independent of each other. The desired contact method can be selected in the new Contact Pressure Method and Tangential Force Method sections of the Contact settings window. For the penalty contact method, the penalty factor needs to be set by the user. For the normal pressure, it is also possible to define an offset. This means that the gap overclosure can be reduced if a good estimate for the contact pressure can be given. For the Friction feature, it is possible to inherit the penalty factor from the parent in cases where the normal contact pressure also uses the penalty method.

## **GRAVITY, CENTRIFUGAL, CORIOLIS, AND EULER FORCES**

You can now add mass forces and loads such as *gravity*, *centrifugal forces*, *Coriolis forces*, and *Euler forces* using two new options: Gravity and Rotating Frames. This makes it easy to define loads that act on all objects having mass: domains with mass density, point masses, added mass, rigid connectors with mass, and so on. The forces and loads are added from the domain level, even though they can be automatically applied to features at boundaries, edges, and points. Load cases are supported.

The Rotating Frame node includes all types of fictitious forces occurring in a rotating system. By default, Centrifugal force and Spin softening are included.

Because Added Mass can be used to describe load effects that are not true structural masses, sometimes the contribution from Added Mass is not wanted. The option to include or exclude the contribution is controlled in a new section called Frame Acceleration Forces.

## **NEW MULTIPHYSICS NODE FOR THERMAL STRESS**

COMSOL version 4.4 introduces Multiphysics nodes to facilitate the modeling of multiphysics applications such as thermal stresses. These nodes provide better control over modeling and allow you to increase workflow complexity by adding individual single-physics interfaces. Adding the Thermal Stress interface from the Model Builder now adds the Solid Mechanics and Heat Transfer in Solids interfaces along with the Multiphysics node, which is tailor-made for simulating the physics couplings involved in thermal stress. Alternatively, if you have already set up the Solid Mechanics interface and performed some modeling, you can then add the Heat Transfer in Solids interface. The Multiphysics node is then automatically added to the Model Builder.

The Joule Heating and Thermal Expansion interface also comes with a new Multiphysics node. Adding this interface directly from the Model Wizard adds the Electric Currents, Heat Transfer in Solids, and Solid Mechanics interfaces to the Model Builder together with the Multiphysics node. Alternatively, you can add the contributing physics interfaces sequentially to increase the model's complexity. Once the second physics interface has been added, the Multiphysics node shows up in the Model Builder. The possibilities for defining multiphysics couplings increases once the third physics interface is added. Various simulations can then be run on the model, including variations in the multiphysics couplings, as you can activate and deactivate these couplings on the fly through the Multiphysics node.

## **SOLID-SHELL-BEAM CONNECTIONS**

A suite of solid-shell-beam connections makes it much easier to set up models that mix solids, shells, and beams. The connections are available as several different options:

For 3D models:

- Connect a shell edge to a solid boundary
- Connect a shell boundary to a solid boundary (also known as “cladding”)

For 2D models:

- Connect a beam point to a solid boundary
- Connect a beam edge to a solid boundary

In all cases, the connection is created by adding two features, one in each physics interface:

- A Shell Connection or a Beam Connection in a Solid Mechanics interface to connect a solid to a shell or a beam.
- A Solid Connection in a Shell or Beam interface to connect a shell or beam to a solid.

#### **RIGID DOMAIN FEATURE**

A new feature called Rigid Domain has been added to the Solid Mechanics interface, replacing the Rigid Domain subnode under Rigid Connector. The Rigid Domain is available for the Solid Mechanics and Multibody Dynamics interfaces. It has several advantages when compared to the previously-available Rigid Domain subnode, including:

- It is a proper material model and overrides other material models such as Linear Elastic.
- It has its own degrees of freedom, which can be initialized with Initial Values.
- It has its own specialized constraints and loads boundary conditions in the form of subnodes.
- It is very easy to define a location using the centroid of selected surfaces, edges, or points to initialize, prescribe, or apply a load.
- It supports structural load boundary conditions such as gravity, rotating frame, body load, and spring foundation.
- It automatically eliminates non-applicable structural constraint boundary conditions.
- It creates automatic continuity with the neighboring material models.
- It has its own global postprocessing variables as well as domain postprocessing variables, similar to other material models.
- It is possible to plot results inside of Rigid Domains.

#### **TIMOSHENKO BEAMS**

The formulation of the beam element has been changed completely, so that incorporation of shear flexibility (so-called Timoshenko theory) can be taken into account. This is in addition to the previously-available Euler-Bernoulli beams. Timoshenko beams are used when the cross-sectional dimensions are large relative to the beam length but still thin enough for a beam approximation to be used. In the case

of Timoshenko theory, shear correction factors must be given in addition to the other cross section data.

#### **VISCOELASTICITY IMPROVEMENTS**

A new Viscoelasticity subnode is available for the Linear Elastic Materials nodes. This subnode allows for the seamless extension of linear elastic models with viscoelastic properties. The combination of the Linear Elastic Material and Viscoelasticity nodes replaces the Linear Viscoelastic Material feature available in earlier versions. With the new formulation, it is no longer necessary to use the viscoelastic initialization step in the Solver node.

Two new viscoelastic material models are available: the *Standard Linear Solid model* and the *Kelvin-Voigt model*, in addition to the Generalized Maxwell Model, which was available in previous versions.

The Static stiffness property allows you to choose between Long-term or Instantaneous stiffness in stationary analyses.

The Thermal Effects section now includes two new formulations to prescribe the time-shift for thermorheologically simple solids: Arrhenius and User defined shifts. This is in addition to the already-available Williams-Landel-Ferry (WLF) shift function.

#### **NEW ALGORITHM FOR DETECTING WHETHER SOURCE AND DESTINATION ARE IN CONTACT**

The detection of contact on a *contact pair* is now based on the structural mechanics interface results, which improves the accuracy of the contact definition. Consequently, in any pair feature, use a contact pair only when the component contains an active structural mechanics contact pair feature. In other cases, use an *identity pair*. Only one contact pair is supported for each boundary pair feature.

#### *Updated Model in Version 4.4*

---

#### **ROTATING BLADE**

This rotordynamics tutorial model has been updated and now uses the built-in Rotating Frame loads rather than the expressions for Body Load.

### *Backward Compatibility With Version 4.3b*

---

- If an old model using the Beam interface is opened, the Euler-Bernoulli formulation is used. The Beam Formulation selection is still shown but cannot be changed from Euler-Bernoulli. If Advanced Physics Options is enabled, then the new section **Backward Compatibility** is shown. If you clear the **Use pre 4.4 formulation** check box, then the new formulation is used. Doing so enables the use of Timoshenko beams, but you must manually handle solver settings like segregation and scaling. This legacy option does not support beams mixed with solids or shells in cases where the same names for degrees of freedom were used in both physics interfaces.
- For models using contact, plot expressions are not updated when you open an old model where the pair names have been edited manually. Also, opening an old model where a pair name has been manually edited might affect the solver configuration. It is recommended that you regenerate a new default solver configuration.

### *Backward Compatibility With Version 4.3a*

---

- In the Solid Mechanics physics (SolidMechanics), the property `ControlsSpatialFrame` no longer exists, and COMSOL now always assumes that the displacements control the spatial frame. Remove all lines setting the value of `ControlsSpatialFrame`. See Support Knowledge Base article 1156.
- When opening old models that include contact, the penalty factor control is set to **User defined** and the contact normal penalty factor to the expression used in the model.

# Subsurface Flow Module

## *New Functionality in Version 4.4*

---

### **PERMEABILITY TENSOR FOR THE BRINKMAN EQUATIONS**

For porous media flow, the Brinkman equations extend the well-known Darcy's law. New in version 4.4 is support for an anisotropic permeability tensor. Different domains can have different anisotropic materials and the tensor components can even vary spatially.

## *Backward Compatibility With Version 4.3*

---

### **BRINKMAN EQUATIONS AND FREE AND POROUS MEDIA FLOW**

The momentum equations in the Brinkman Equations interface and the Free and Porous Media Flow interface have been corrected. The term  $-Q_{br} \cdot \mathbf{u}/\varepsilon_p^2$  in the right hand side previously lacked the factor  $1/\varepsilon_p^2$ , where  $\varepsilon_p$  is the porosity.

### **WEAK CONSTRAINTS UPDATE FOR FLUID FLOW INTERFACES**

The weak constraint formulations for some boundary conditions in the following interfaces have been updated:

- Laminar Flow
- Brinkman Equations
- Free and Porous Media Flow

# Wave Optics Module

*New Functionality in Version 4.4*

---

## **LASER HEATING**

A new *Laser Heating* multiphysics interface has been introduced, which combines the Electromagnetic Waves, Beam Envelopes, and Heat Transfer in Solids interfaces. The Laser Heating multiphysics interface utilizes a new dedicated Multiphysics node in the Model Builder in a similar way to Joule Heating (COMSOL Multiphysics), Induction Heating (AC/DC Module), and Microwave Heating (RF Module). The beam envelope method formulation is appropriate for beams of light that have a slowly varying envelope, such as beams along an optical fiber. The Laser Heating multiphysics interface couples the electromagnetic losses with the heat transfer in solids. The temperature variation can be computed in time or under steady-state conditions. Material dependency on temperature can also be considered; for example, thermal and optical material properties can depend directly on temperature.

The Multiphysics node also provides much better control over modeling your multiphysics applications. This is epitomized by the Activation and Deactivation features within the Multiphysics node, which make it possible to model single-physics applications separately. Alternatively, combinations of two of the three contributing physics interfaces can be simulated using this feature.

## **USER-DEFINED PHASE FUNCTIONS FOR THE BEAM ENVELOPES FORMULATION**

The Beam Envelopes formulation can solve for the electromagnetic field envelope assuming that the propagation vector of the fields is approximately known everywhere in the domain. It is especially memory efficient for problems where the field envelope varies slowly relative to the wavelength and the direction of propagation is known. It is now possible to explicitly enter a phase function in different domains, which is useful if the beam is changing directions.

## **MATCHED BOUNDARY CONDITION**

The new *Matched boundary condition* in the Beam Envelopes formulation is perfectly transparent to a wave of known direction. Because the wave-vector is typically known at the boundaries when using the Beam Envelopes interface, this boundary condition

introduces less artificial reflections than the Scattering boundary condition and requires less memory than the Perfectly Matched Layer domain truncation.

#### **GAUSSIAN BEAM BACKGROUND FIELD**

The scattered field formulation is used to compute the scattering of electromagnetic fields off of an object. Typically, a uniform plane wave is specified as the background field, but the new Gaussian beam background field makes it possible to specify a Gaussian beam, propagating along one of the axis directions, of a specified beam waist and focal point. You can also specify the polarization of the beam.

#### **INTERIOR PORT BOUNDARY CONDITIONS**

When modeling wave sources, port boundary conditions are usually set at the exterior boundaries of a model to represent a source that is located outside the modeling space. However, sometimes it is more convenient to place the source inside the modeling domain. In the Electromagnetic Waves, Frequency Domain interface, the new Slit Port introduces the ability to place a source at an interior boundary. This source can be a *domain-backed* or *PEC-backed* slit port:

- The PEC-backed slit port introduces two boundary conditions at an interior boundary. On one side of the boundary, the PEC condition is applied; on the other side, you can use any of the regular port boundary conditions to excite a field propagating away from the boundary. The direction in which the field propagates is specified by the Port Orientation.
- The domain-backed slit port, on the other hand, is a transparent boundary. It can excite a wave propagating away from the boundary, and any wave incident upon the domain-backed port passes through unimpeded. The domain-backed slit port is also useful for modeling periodic problems. When modeling structures that have many higher-order diffraction orders, such as gratings, you must account for each diffracted order with a separate port boundary condition. For 3D structures, there can even be diffraction into multiple planes. However, sometimes you might not be interested in considering each diffracted order separately; you may only want to know the bulk transmittance and reflectance of a periodic structure. In this case, a domain-backed slit port can be used. The slit port can insert an incident plane wave, coming in at any angle, and any wave reflected back toward the port passes through and into a PML placed behind it. The PML absorbs all of the higher-order modes simultaneously.

### **NUMERIC PORT BOUNDARY MODE ANALYSIS IMPROVEMENTS**

The Numeric Port boundary condition is used to compute the fields at a boundary to a waveguide where the field distributions cannot be computed analytically (such as at rectangular, coaxial, or circular ports). These numeric port calculations can now consider the impedance boundary condition. The impedance boundary condition considers the effect of lossy walls, instead of assuming that the walls are perfect electric conductors. The periodic boundary condition can also be considered.

# Material Library

## *New Materials and Material Data in Version 4.4*

---

- There is new data for the following aerospace alloys: GTD111 DS, Rene N5, CMSX-4, CMSX-486, B-1900, PWA 1480, and In-738LC. This data was provided by Chromalloy Gas Turbine Corporation.
- The following materials and properties are now available: 2.25Cr-1Mo, A106 grade B, 9Cr-1Mo, A333 grade 6, A533 Gr B, A508 Cl3, A302 GrB, GRCo-84, 2025 Al, and several rare earth silicates.

# LiveLink for MATLAB®

## *New Functionality in COMSOL 4.4*

---

### **EXPORT PLOT DATA FOR DELAYED PLOTTING**

Exporting plot data to a data structure using the command `mphp1ot` is now supported for all plot types. This enables delayed plotting of the data, and the option to plot additional data together with the exported data structure.

### *Known Issues*

---

The option to plot on a server by starting a COMSOL server with the option `-graphics` is not supported for Mac.

# LiveLink for Excel®

*New Functionality in Version 4.4*

---

## **EXPORT OF FIELD-DEPENDENT MATERIAL PROPERTIES**

The export of material properties stored in an Excel file to a COMSOL material library now includes the export of field-dependent properties. This includes, for example, temperature-dependent properties and material properties such as the B-H curve.

## **PARAMETRIC SWEEPS**

You can now extract the parameter values for a parametric sweep to a range of cells in a worksheet. You can also edit the parameters and update the model with the new values.

## **CONNECTING TO A COMSOL SERVER**

The LiveLink™ for Excel® now allows computations to take place on a different computer running a COMSOL Server. This makes it possible, for example, to connect from Excel and the COMSOL Desktop to the COMSOL Server at the same time.

# The COMSOL API for use with Java®

## COMSOL 4.4 API Changes

---

### PHYSICS INTERFACES

The default Initial Value feature in the following interfaces have been extended

- Transport of Concentrated Species
- Reacting Flow in Porous Media (rfcs)
- Reacting Flow
- Rotating Machinery, Reacting Flow

Previously, only the initial mass fractions could be specified. Now the initial values can be specified in terms of mass fractions, mole fractions, molar concentrations, number densities, or densities. Opening an old Model MPH-file, *Mixture specification* is set to *Mass fractions*, and the mass fractions specified are entered in the respective text fields. In old model files for Java, commands specifying the initial value of a mass fraction return an error in version 4.4. To specify the initial mass fraction of a species w2, which is the second in the list of species (seen in the Dependent Variables section when clicking the interface), change the COMSOL Java command

```
model.physics("chcs").feature("init1").set("w2", 1, "0.1");
```

into

```
model.physics("chcs").feature("init1").set("w0", 2, "0.1");
```

- The Pressure Acoustics, Frequency Domain interface and the Boundary Mode Acoustics interface have a new default feature. If the default feature has been edited in a Java file, the following line should be added to the Java file to obtain the old behavior:

```
model.physics("acpr").feature().create("pam1",  
"PressureAcousticsModel").selection().all();
```

for Pressure Acoustics, Frequency Domain, or

```
model.physics("acbm").feature().create("pam1",  
"PressureAcousticsModel").selection().all();
```

for Boundary Mode Acoustics.

This line should be added directly after the line where the physics interface is created.

- For models using contact, the variable field name in the solver settings for the old contact variables is different than in earlier versions. For example:
  - In version 4.4: `mod1_solid_contact_old_p1`, `mod1_solid_cm_old_p1`.
  - In earlier versions: `mod1_solid_contact_p1_old`, `mod1_solid_cm_p1_old`.

### *COMSOL 4.3b API Changes*

---

#### Physics Interfaces

- The *Crosswind diffusion* formulation has been updated for all physics user interfaces supporting Crosswind diffusion. Models solved with the new formulation can give different results than models solved in version earlier than 4.3b. Java files can be modified to retain old crosswind formulations. Please contact support for details.
- The default settings have changed for the Background Pressure Field in the Pressure Acoustics interfaces. Add the following line to obtain the old behavior of this feature:
 

```
model.physics("acpr").feature("bpf1").set("c", 1, "acpr.c_c");
```
- In the Acoustics Module, the Far-Field Calculation feature is no longer available in 1D and 1D axisymmetry.
- The shape function property `border` has been deprecated and replaced by `order`.

### *COMSOL 4.3a API Changes*

---

#### **GENERAL API CHANGES**

- Class files compiled with COMSOL 4.3 or earlier need to be recompiled if they contain code that changes the contents of `model.selection()`.
- The `solutionintitle` property governs whether to show solution information in plot titles. But since solution information has never been shown for 1D plots, regardless of the content of the property, the property has been removed from 1D plot groups.

#### **FLUID FLOW INTERFACES API**

- Fluid flow features in the CFD Module and Microfluidics Module that previously created a scalar Lagrange multiplier field with default name `model.un_lm`. now create a vectorial Lagrange multiplier field with default name `model.u_lm`. The

default component names in 3D are `model.u_1m`, `model.v_1m` and `model.w_1m` respectively. Java files must be updated accordingly.

- Weak constraints for the Interior Wall feature are no longer available. Any references to its weak constraint parameter (`weakConstraints`) or Lagrange multipliers must be removed.
- The **Automatic** setting for the **Pseudo time stepping** property now sets the variable `<phtag>.locCFL` to the built-in variable `CFLCMP`, which in turn triggers a PID regulator via the automatic solver suggestions. Here, `<phtag>` is the physics interface tag. Java files where pseudo time stepping is active and have local CFL number set to **Automatic** must be modified by adding the command

```
model.physics(<tag>).prop("PseudoTimeProperty").set("CFLNumbExpr", 1, "Manual");
```

before calling a solver.

### COMSOL 4.3 API Changes

---

- In the Acoustics Module, the far-field variables `pfar` and `Lp_far` have new names with full scope. They are now referred to as `acpr.ffc1.pfar` and `acpr.ffc1.Lp_far`, respectively. Any model files for Java that use the old variable names in expressions (such as expressions used for plotting or evaluation that include such old variable names) require a manual update.
- The following methods

```
model.physics(<tag>).feature(<ftag>).params();
model.physics(<tag>).prop(propname).params();
```

are depreciated and replaced by the methods

```
model.physics(<tag>).feature(<ftag>).param();
model.physics(<tag>).prop(propname).param();
```

### COMSOL 4.2a API Changes

---

- Class files compiled with COMSOL 4.1 or lower version need to be recompiled.
- The far-field feature in the Electromagnetic Waves interface has changed from being a boundary feature with a boundary selection only, to a domain feature with a domain selection. It also has a subfeature—a boundary selection.
- The dependent variable associated with gauge fixing in the Magnetic Fields, Magnetic and Electric Fields, Electromagnetic Waves, and Transient Electromagnetic Waves interfaces is now unique to each interface. It is no longer

available in the model scope, for example, `mod1.psi`. Instead the gauge fixing field is only available in the interface scope, for example, as `mod1.mf.psi`.

- In the scattered field formulation in the Electromagnetic Waves interface, the scattered field is no longer available in the model scope (for example, `mod1.relEx`). Instead the scattered field is only available in the interface scope as, for example, `mod1.emw.relEx`.
- In the Solid Mechanics interfaces (`SolidMechanics`, `PiezoelectricDevices`, `AcousticStructure`, `Poroelasticity`, `ThermallyInducedStresses`, `JouleHeatingThermalExpansion`, `TransientAcousticStructure`, `FluidStructureInteraction`), tensors in local coordinate systems (`e1`, `ee1`, `S1`, `si`, and `ei`) have new names. The coordinates of the local system (for example `x1`, `x2`, `x3`) are no longer used. Instead 1, 2, and 3 are used together with double indexing for all components. As an example, `e1x2` is replaced by `e122` and `e1x2x3` is replaced by `e123`. The tensors `si` and `ei` are now called `S1i` and `ei1`.
- In the Darcy's Law interface and the Richards' Equation interface in the Subsurface Flow Module, the compressibility of fluid is now a material parameter and no longer has a default value. If the default value was used, you now have to set the value. The following example sets the permeability to the old default value:

```
model.physics("d1").feature("smm1").set(chif_mat,userdef);
model.physics("d1").feature("smm1").set(kappa,4e-10);
```
- In the Poroelasticity interface in the Subsurface Flow Module, the compressibility of fluid and the Biot-Willis coefficient are now material parameters and no longer have a default value. If the default value was used you now have to set the value. The following example sets the permeability to the old default value:

```
model.physics("d1").feature("smm1").set(chif_mat,userdef);
model.physics("d1").feature("smm1").set(kappa,4e-10);
```
- The Level Set and Phase Field interfaces now include the Initial Interface feature by default. If you have a model that was created in an earlier version of COMSOL, it will fail to create a feature with the same tag name.

## *COMSOL 4.1 API Changes*

---

The following changes were made to the COMSOL API between version 4.0a and 4.1:

- In version 4.0a the property `nonlin` in the stationary solver could have the values `auto`, `on`, `off`, and `linearized`. The default in most cases was `auto`, but in some cases, specifically in a frequency-domain study, the default was `linearized`. In version 4.1 the possible values are `auto`, `on`, `off`, and `linper`, with `auto` as default,

except in the *Frequency Domain, Linearized* study step where the default is `linper`. The value `linearized` can still be set, but this is treated as `auto`. If the problem is linear and the linearization point is zero, `auto` should give the same solution as `linearized`. For nonlinear problems where the linearization point is zero, the solution given by `auto` might be slightly different, and probably more accurate, while changing to `off` should give exactly the same solution as `linearized`. When there is a nonzero linearization point, it is relevant to use the `linper` option. This option works like `linearized` except that source terms that contribute to the linearized problem must be enclosed by the `linper` operator.

- In the Darcy's Law interface, Brinkman interface, Richards' Equation interface, and Poroelasticity interface (in the CFD Module and Subsurface Flow Module), the permeability and porosity are now material parameters and no longer have a default value. If the default values were used in a model file for Java, you now have to set the value. The following example sets the permeability to the old default value:

```
model.physics("d1").feature("d1m1").set(kappa_mat,userdef);
model.physics("d1").feature("d1m1").set(kappa,3e-11);
```

- The pressure variable in solid mechanics, typically `solid.pw`, now only gets allocated degrees of freedom for incompressible materials. If you have referred to it, for example, during segregation in the solver, the code will have to be changed to remove the field.
- The solution to eigenvalue and eigenfrequency problems may now appear in a different order.
- The Batteries & Fuel Cells Module's boundary feature, `BoundaryReactionCoefficient`, is tagged by `rc` instead of `brc` by default. This means that files that explicitly refer to the old default tag name must be modified to refer to the new tag name.

### *COMSOL 4.0a Java API Changes*

---

The following changes were made to the COMSOL API between version 4.0 and 4.0a:

- The load face parameter (`Fph`) of the *Phase* feature in `AcousticStructure`, `TransientAcousticStructure`, `ThermallyInducedStresses`, `SolidMechanics`, `JouleHeatingThermalExpansion`, `Poroelasticity`, and `FluidStructureInteraction` has changed interpretation from degrees to radians.
- The `physics.field()` of `Beam` and `Truss` are not backward compatible with 4.0. Scalar fields are now vector fields.

- The variables for strain and stress in the `Shell` interface have been renamed. Variable names that began with a lowercase `s` now begin with an uppercase `S`. For variables names that ended in lowercase `l` (`L`), the letter was moved to the third position. For example, `emXXl` is now `emlXX`.
- Force loads on solids and trusses have been moved from the material to the spatial frame. This implies that the index notation of some variables has changed from uppercase `XYZ` to lowercase `xyz`.
- The acoustics interfaces (`PressureAcoustics`, `TransientPressureAcoustics`, `AcousticStructure`, and `TransientAcousticStructure`) have been moved from the material to the spatial frame. This implies that index notation of vector variables has changed from uppercase `XYZ` to lowercase `xyz`.

This concludes the release notes for COMSOL Multiphysics version 4.4.