

# COMSOL Multiphysics

Release Notes

# COMSOL Multiphysics Release Notes

© 1998–2015 COMSOL

Protected by U.S. Patents listed on [www.comsol.com/patents](http://www.comsol.com/patents), and U.S. Patents 7,519,518; 7,596,474; 7,623,991; 8,219,373; 8,457,932; 8,954,302; 9,098,106; and 9,146,652. Patents pending.

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

Support for implementation of the ODB++<sup>®</sup> Format was provided by Mentor Graphics Corporation pursuant to the ODB++ Solutions Development Partnership General Terms and Conditions. ODB++ is a trademark of Mentor Graphics Corporation.

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

Version: COMSOL 5.2

## 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/product-download](http://www.comsol.com/product-download)
- Product Updates: [www.comsol.com/support/updates](http://www.comsol.com/support/updates)
- COMSOL Blog: [www.comsol.com/blogs](http://www.comsol.com/blogs)
- Discussion Forum: [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<sup>®</sup> version 5.2 includes an extended and improved Application Builder, a more powerful COMSOL Server<sup>™</sup>, and updates and improvements for all COMSOL products. These *Release Notes* provide information regarding new functionality in version 5.2 for all COMSOL products.

# COMSOL Multiphysics

## *General New Functionality*

---

### **IMPROVED INSTALLER**

The COMSOL Installer now includes better support for installing and configuring the license manager software.

### **RELEASE OF LICENSE AT NEW/OPEN**

The default behavior in 5.2 is that any used COMSOL product licenses are released when you use New or Open to start working on a new application. The behavior in earlier versions—that you need to restart COMSOL Multiphysics—is also available as a preference setting.

## *New Functionality in the Application Builder*

---

### **GENERAL NEW FUNCTIONALITY**

- It is now possible to add custom toolbar items to the graphics form object toolbar.
- The Model Code window is now called Editor Tools. It can also generate form objects in addition to generating code for Methods.
- There is now support for adding Check Box and Combo Box form objects for some Model Data Access sources in the New Form Wizard and Editor Tools window.
- There is now extended support for Model Data Access so that you can also use array- and matrix-valued data, reference properties, and special settings that use indexed access to data of higher dimensions. Examples of the latter are time and parameter settings in results features and positions in geometry features.
- Improved copy-paste operation between applications. The following nodes now support copy-paste: Methods, Utility methods, External libraries, File declaration, Choice list declaration, Menu item, Menu divider, Menu, Ribbon section, and Ribbon tab.
- Plot-while-solving (including probe plots) now works from runnable applications. Enable it as usual in the study step settings, and ensure that the plot group you are plotting is set as the source for a graphics form object.
- Graphics form objects can now be updated while a command sequence or method is running.

- Declarations can be used in arguments of command sequence commands.

#### **IMPROVEMENTS TO THE FORM EDITOR**

- Multiple form references are now allowed, so that you can use the same form in different tabs, for instance.
- Form Collections can now change between a tiled and a tabbed look.
- The Form Editor now supports specifying the margins that are added between borders of the cells in the grid and Form Objects added to those cells.
- Toggle buttons are now available as a new form object type. A Toggle button acts as two buttons in one, with one action (command sequence) triggered when the button is selected, and another action triggered when the button is deselected.
- It is now possible to control the order of sections or panes in Form Collection objects using Move Up and Move Down buttons to rearrange the selected panes.
- The Data Display object can now be used to show arrays or matrices with LaTeX syntax. The source of a data display can be array declarations, matrix declarations, or array and matrix settings from Model Data Access such as material properties.
- The Card Stack form object now supports two type of cards: *local cards* (which work as cards do in earlier versions) and *card references*, which are new and act as Form References by directly referencing an existing form and displaying that form when the activating value is set).
- Images of 3D scenes in reports are now connected to application Graphics objects with the same source or the same geometry.
- You can now switch views from the graphics toolbar. There are also new methods for switching the view, resetting the view, and for retrieving and setting view properties such as the camera position.

#### **IMPROVEMENTS TO THE METHOD EDITOR AND APPLICATION LANGUAGE**

- The Method Editor can now indent and whitespace-format code. This happens automatically when you press the TAB key. Unless disabled in preferences, it also happens when the keyboard focus leaves a method editor window.
- A `print` command and a `printGraphics` method are available for printing graphics from command sequences and methods.
- The methods `long getLastComputationTime()`, `String getExpectedComputationTime()`, and `setExpectedComputationTime(String time)` have been added to the public API. The first one is available for the Model

object and Study objects, while the other two are only available for the Model object.

- A new `formattedTime` method converts a time in milliseconds to a string representation.
- New utility method `matrixSize()` to compute the size of a 2D matrix.
- Vectorized versions of `insert`, `append`, and `remove` for 1D and 2D arrays have been added to the application language.
- A new `sleep` method has been added that makes the caller sleep for a specified time.
- A new `setWebPageSource(...)` method allows changing the source of a Web Page form object.
- New built-in functions `getScreenHeight` and `getScreenWidth` can be used for retrieving the size of the primary screen (or of the browser window if Web Client is used).
- New application language methods `deleteFile` and `copyFile` make it possible to delete and copy files on the server.
- The `fileOpen` method is now available as a new built-in command in the command sequence tree.

For details and a full list of methods and language elements, see the *Introduction to Application Builder*.

## *New Functionality in COMSOL Server*

---

### **PRELAUNCHING OF APPLICATIONS**

To allow faster launching of applications, application processes are prelaunched on the server. Additionally, a number of applications can be preloaded in these processes. Typically the default settings work for most cases.

- **Number of recently run applications to preload:** Default: 1. Preloads this number of the most recently run applications.
- **Default preloaded applications:** Default: none. The selected applications are explicitly preloaded. If there is overlap with the most recently run applications, the overlapping applications are still only loaded once.

### **ALLOWING RECONNECTING TO RUNNING APPLICATIONS**

A new preference setting, **Allow reconnecting to running applications**, allows users to reconnect to previously launched applications and come back where they left off. Disconnecting and reconnecting is currently not possible while solving, for example.

### **LAUNCHING APPLICATION AT LOGIN**

A new preference setting, **Launch application on login**, is available but is off by default. If an application is selected here, it is automatically started when a nonadministrative user logs in. Only administrators can see the full Application Library.

### **LAUNCHING COMSOL CLIENT FROM WEB PAGE**

You can now launch an application directly in COMSOL Client through the COMSOL Server interface by clicking on the menu and selecting **Launch in COMSOL Client**.

### **IMPROVED FILTERING AND SORTING**

You can now filter and sort applications by favorite status, application visibility (Public, All groups, Private), or by individual groups.

### **CREATING DESKTOP SHORTCUTS**

You can now create shortcuts to quickly access your most frequently used applications from your desktop. First, open up an application's detailed view by clicking on its name or image. From there, you have the option to create a shortcut link that opens the application either in COMSOL Client or in your web browser.

### **NEW COMSOL COMMANDS**

The following COMSOL command is now available for COMSOL Server:

- `comsolserver / comsol server`

You can also use the options `-prefdir`, `-forcegcc`, `-forcecomsolgcc`, `-np`, `-tmpdir`, `-autosave`, `-recoverydir`, `-ipv6`, `-c`, and `-comsolinifile`.

### *New and Improved General Functionality in COMSOL Multiphysics*

- It is now possible to interface to external material functions written in C.
- The behavior when the license server connection is lost has been improved so that it is now possible to save the current COMOL project and also to reconnect to the license server.
- Searching in the Application Libraries window is now faster.

### **BETTER CONTROL OF OUTPUT SELECTIONS FROM GEOMETRIES**

To get better control over the output selections that are generated from a geometry feature, the following new settings are available:

- The **Selections of Resulting Entities** sections in the geometry features now contain the following:
  - A **Contribute to** list to which you select the cumulative selection that the result of this feature contributes to. The default is **None**. You can contribute to a cumulative selection even if you do not create a selection.
  - A **New** button to the right of the **Contribute to** list.
  - A **Resulting objects selection** check box, cleared by default. When you select it, a selection will be created for each applicable geometry entity level (object, domain, boundary, edge, point) for use within this geometry sequence.
  - A **Show in physics** list (enabled when the **Resulting objects selection** check box is selected) with the following options: **All levels**, **Domain selection**, **Boundary selection**, **Edge selection**, **Point selection**, and **Off** (but only applicable options appear). If you choose something other than **Off**, corresponding selections are created for the finalized geometry and will be available in applicable settings in the physics nodes and also for materials and meshes, for example. The default is typically **Domain selection** to use the selection to define a material, material property, or domain source, for example.
- If the feature is in a part, the **Show in physics** list is replaced with a **Show in instances** list, and the created selections appear in the selection tables of Part Instance nodes.
- For Part Instance nodes, the output selection tables now include the following columns:
  - **Keep**, containing a check box, cleared by default. When this is selected, the selection is kept for use within the geometry sequence.
  - **Physics** or **Instances**, containing a check box that is selected by default. This check box is enabled when the **Keep** check box is selected. Clear it if you do not want the selection to appear when defining physics, materials, and so on.



## NEW GEOMETRY FUNCTIONALITY

- New Partition Domain operation in 2D and 3D for partitioning of domains by implicitly defined partitioning curves or surfaces.
- Improved snapping in 2D with separate Snap to Grid and Snap to Geometry buttons in the Geometry toolbar.

## NEW MESHING ALGORITHM

A new, more robust free meshing algorithm is available as an alternative to the previous Delaunay algorithm. In COMSOL Multiphysics 5.2, this new, alternative Delaunay algorithm is used if the previous Delaunay algorithm fails when trying to respect the geometry boundaries. This fallback mechanism is the default as the **Automatic** setting in the **Method** list, available in the new **Tessellation** section in the Settings window for Free Tetrahedral nodes. You can also choose **Delaunay** or **Delaunay (alternative)** to always use the previous or alternative Delaunay algorithm, respectively.

## MESH PARTS AND MESH IMPORT

The import of external meshes, for use as geometry or mesh, now uses Mesh Parts, which are reusable global meshing sequences where the imported mesh (from an STL, VRML, NASTRAN, or MPHBIN/MPHTXT file) can be prepared for modeling with partitioning operations, for example. The mesh parts make it easier to use imported meshes as a geometry in COMSOL Multiphysics simulations. You add mesh parts under **Global Definitions>Mesh Parts**, and you can then refer to them from an Import node in a geometry sequence.

Additionally, a new option, **Detect adjacent fillet faces**, has been added to the settings for the Import (when using the **Detect faces** as the **Boundary partitioning** setting) and Detect Faces nodes in 3D mesh import sequences. Detecting fillets in imported meshes should help to create a natural partitioning and make it easier to create a geometry from the imported mesh.

## DIAGNOSTICS FOR INVERTED ELEMENTS IN DEFORMED GEOMETRIES

If inverted mesh elements occur when modeling a deformed geometry, you now get an error message pointing out inverted elements in the deformed mesh. Also, the problematic deformed mesh is saved (for visualization) in a **Problematic Deformed Mesh** node.

## *New Operators, Functions, and Definitions*

---

- The molar or molarity unit  $M = \text{mol/l}$  is now available as [M]. SI prefixes are supported.
- New operators: `fsensimag`, for evaluating a functional sensitivity with respect to the imaginary part of a control variable, and `residual` (see the section below).

## *New Functionality in Studies and Solvers*

---

- You can now choose to store only a subset of all DOFs in the solution, thereby saving storage by excluding parts of the solution that may not be of interest.
- Improved support for FFT:
  - Predefined window functions such as Hamming, Hanning (Hann), Blackman, and Tukey.
  - Input time for a forward FFT is now defined using a start time and an end time. The number of interpolated input solutions is derived from a user-defined maximum output frequency and is reported in the solver log.
  - A new option to not store negative frequencies for real input removes redundant information in complex output data for a forward FFT.
  - Now using the FFT library from Intel's MKL.
- Two new efficient Rung-Kutta time-stepping solvers are now available in addition to the Dormand-Prince method: RK34 and Cash-Karp 5.
- You now have access to the latest assembled residual using the `residual` operator, which you can plot while solving, for example, to illustrate the convergence process and find problematic areas.
- The number of eigenvalues or eigenfrequencies and the shift around which to search for them are now controlled by the physics, which provide suitable values. You can select check boxes for both of these properties to specify them manually.
- The settings for the initial values of variables solved for has been improved. There is now a **Settings** list, from which you can choose to use physics-controlled or user-controlled initial values.
- The **Store solution for all intermediate study steps** check box has been added to the main Study node's settings. Select it to store solutions for all intermediate study steps, regardless of which study steps the COMSOL software determines to store, which may be only the solution from the last study step.

- The Clear Solutions operation to clear all solutions in the study is now available from the main Study node's context menu.
- The identifiers for solution nodes in the solver sequence in the Model Builder tree now appear in the Model Builder tree. The corresponding identifiers also appear for Solution data sets under Results, to make visible which data sets contain a specific solution without opening any Settings windows.
- If you enable the Advanced Study Options, a **Distribute parametric sweep** check box is now available in the Settings windows for the study steps, so that you can activate a distributed parametric sweep without first creating a Cluster Computing study.

### *New and Improved Results and Visualization Functionality*

---

- A new Annotation plot type is available for adding annotations to any 3D or 2D plot group. You can add text, including evaluated expressions, at any position in the plot.
- The Far Field plots are now available in COMSOL Multiphysics without any additional license requirements. Also, for 3D far-field plots, it is now possible to control the threshold level to provide a more dynamic visualization.
- You can now choose the color for plot titles and color scale values in plots.
- Filled histograms in 1D histogram plots are now available. For discrete outputs, you can choose Solid as the type of histogram to create filled histogram plots.
- Filters can now cut mesh elements to provide smoother edges and better-looking plots where a Filter subnode is used to only include part of the geometry. The new **Smooth** option is the default setting in the **Element nodes to fulfill expression** list.
- Support is now included for engineering format for numerical values in color legends, axes, and tables.
- The arrow base can now be centered in arrow plots.
- The 3D grid box has been improved:
  - Intermediate labels are removed when they overlap with neighboring labels.
  - The **Settings** window for **Component>Definitions>View>Camera** now contains a **Grid** section at the bottom. This section contains the same settings as the corresponding section in the **Axis** node in 2D.
- It is now possible to hide the 2D axis.
- In the AC/DC Module, RF Module, and Plasma Module, Smith plots are now available in a Smith Plot Group node where you can add reflection, impedance, admittance, and table graphs. Smith plots are useful for displaying and relating

complex-valued S-parameters (reflection coefficients) to input impedances or admittances of antennas, transmission lines, and other network components.

- In the Acoustics Module, octave band plots are now available as Octave Band Plot nodes in 1D plot groups.
- You can now suppress small real or imaginary parts in the output of results data, for example, on the **General** page in the **Preferences** dialog box.
- In data and plot export, you can now concatenate several data sets by appending data to the same file. Under **Advanced**, select **Append** from the **If the file exists** list to concatenate a data set to an existing data file.
- In data export, there is also a **Recover** list, where you can add numerical recovery within domains or everywhere for the data export.
- It is now possible to add the unit to the legends in 1D Global plots.
- Using settings under **Floquet periodicity** in the **Advanced** section, you can now incorporate phase changes from Floquet periodicity in Array data sets.
- For easier navigation in reports, internal cross references are now hyperlinks. Hyperlinking can be disabled for printing.

### *New Applications in Version 5.2*

---

- The Helical Static Mixer application is a new app, with which you can simulate a helical static mixer with variable number of blades and dimensions. A concentration field is also included in the analysis in order to compute the extent of the contact between two streams injected into the static mixer through half circle-shaped inlets.
- The Transmission Line Calculator application, previously available in the AC/DC Module, is now included in the COMSOL Multiphysics Application Library. This app calculates the transmission line parameters, R, L, G, and C; the characteristic impedance; and the propagation constant for some common transmission line types.

### *General Backward Compatibility Considerations*

---

COMSOL 5.2 is backward compatible with COMSOL versions 4.0–5.1. To open models created using COMSOL version 3.5a in version 5.2, follow these steps:

- 1 Open and then save the COMSOL 3.5a mode using any of the COMSOL versions 4.0–4.4.
- 2 Open the model saved in versions 4.0–4.4 in COMSOL 5.2.

## *Backward Compatibility with Version 5.1*

---

### **CREATING SELECTIONS**

In version 5.2, the `selresult` property replaces the `createselection` property. `createselection` is still supported for backward compatibility.

### **SELECTIONS IN PART INSTANCES**

For backward compatibility for selections in part instances, there is a **Keep noncontributing selections** check box, cleared by default, in the **Selection Settings** section of the Settings window for Part Instance nodes. If you select the **Keep noncontributing selections** check box, the **Keep column** is disabled, and the selection is kept if the **Contribute to** value is **None**. In the COMSOL API, the default is that the **Keep noncontributing selections** setting is active.

### **MESH PARTS**

For backward compatibility regarding STL/VRML import, and for COMSOL API compatibility, the old user interface for mesh import (with a specified file name and import properties) and the corresponding properties are available in the COMSOL API but not in the COMSOL Desktop, unless it is the active option. This means that the old user interface can only be reached if the filename is set in combination with the STL import type (the API type `stlvrml`), which cannot be done from the new user interface.

### **DORMAND-PRINCE 5 TIME-STEPPING SOLVER**

The Dormand-Prince 5 Runge-Kutta solver in 5.1 did not use field norm scaling in version 5.1 when estimating errors. This means that the errors of, for instance, an ODE might become very small when solved together with a field with many DOFs, and the time steps taken might then be too large. In version 5.2, the Runge-Kutta solver uses field norm scaling, which means that old models using the Dormand-Prince 5 might need tighter tolerances to produce results in earlier versions that are similar to those in 5.2.

## *Backward Compatibility with Version 5.0*

---

### **MESH IMPORT**

The method used to automatically partition the boundary of imported meshes in 3D has been improved. If you have an existing model, you can work with it without being affected. However, if you click the **Import** button of the **Import** feature in the meshing

sequence, the file is read again and the new partitioning method is used. The exception is if the **Import** feature had **Boundary partitioning** set to **Manual**. In this case, the modified parameter values are preserved under the **Feature detection** setting, which uses the same algorithm as in version 5.0.

COMSOL tries to map the old selection on boundaries to the new boundaries, but it is not always possible to do accurately when new faces have appeared or old faces have disappeared. You may have to manually review and update boundary, edge, and point selections after reimporting the mesh.

If you have a Java<sup>®</sup> or MATLAB<sup>®</sup> program that imports meshes, the number of geometric entities may have changed compared to older versions.

### **SECURITY SETTINGS**

In version 5.2, the **Allow external process and libraries** check box on the Security page in the Preferences dialog box is cleared by default to not allow applications to start external processes on the computer. The default setting in 5.0 was set to allow such external processes.

### **DISPLAY OF MAX/MIN MARKERS IN PLOTS**

The display of max/min markers in plots is now off by default to make plots fit better when using a small graphics canvas. In previous versions, the display of max/min markers in plots was on by default.

### **MERGED MPH-FILE FORMAT**

The MPHAPP-file and MPH-file formats have been merged since version 5.1, and all application files use the .mph file suffix. You can still open MPHAPP-files created in version 5.0.

## *Backward Compatibility with Version 4.4*

---

### **COMSOL SERVER**

The COMSOL Server from version 4.4 is now called the COMSOL Multiphysics Server.

### **MESH IMPORT**

An edge in an imported mesh that has a common start and end vertex, or that lacks start and end vertices, is now split into two edges with distinct start and end vertices.

For meshes where this happens, the numbering of all geometric entities may change when the mesh is rebuilt.

For MPH-files created in earlier versions, selections are automatically updated with the new entity numbers.

### LOCAL COORDINATE VALUES

For model components created in version 5.0 and onward, the variable names `xi1`, `xi2`, and `xi3` are reserved for the predefined local coordinate variables. To access the previous behavior (that local coordinate variables do not exist), you can use the API method `model.modelNode(<tag>).defineLocalCoord(false)`.

### TRANSPORT OF DILUTED SPECIES

Running Java<sup>®</sup>-files from previous versions may fail due to the new default name for the Transport of Diluted Species interface. This can be avoided by adding a command that specifies the identifier in accordance with the name of the interface. For example, when creating an interface using the old name (`chds`):

```
model.physics().create("chds", "DilutedSpecies", "geom1",
    new String[][]{{"c"}});
```

Add the following line to specify the identifier accordingly:

```
model.physics("chds").identifier("chds");
```

The Java<sup>®</sup> API syntax for creating and accessing vectors and tensors in the Transport of Diluted Species interface has changed as well as the syntax for setting physics properties. This does not affect MPH-files. See the backward compatibility notes for the Chemical Reaction Engineering Module for additional information that also applies to the Transport of Diluted Species interface.

### DOCUMENTATION

The *COMSOL Multiphysics Programming Reference Manual* replaces the *COMSOL API for use with Java<sup>®</sup> Reference Manual*.

### *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 5.2.

### 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 gives 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 & 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 is 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 an 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 termination 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 5.2.

### **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 5.2, 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 this 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 5.2:

- *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 have 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 5.2 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.2.

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

The pairs created by the Form Assembly feature in the geometry can differ from their forms 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 5.2*

---

### **EFFECTIVE NONLINEAR MAGNETIC CONSTITUTIVE RELATIONS FOR FREQUENCY DOMAIN STUDIES**

There is now support for effective magnetic constitutive relations, in the form of effective HB or BH curves. This functionality can be used to model nonlinear magnetic materials, such as saturable metals, in a frequency domain study by approximating them with an effective inhomogeneous linear material. You can use the formulation to compute the approximate (first-order harmonic) response of the nonlinear material subject to time-harmonic excitation, avoiding the computational cost of a full transient analysis.

### **EXTERNAL NONLINEAR MAGNETIC MATERIALS**

The Magnetic Fields physics interfaces can now use external material models implemented as user-defined dynamic libraries, increasing the flexibility and the modeling capabilities of the physics interfaces. This functionality is built into the External Material feature and makes it possible to define custom advanced material models such as hysteresis models, state-dependent models, or even models using a different discretization scheme.

## *New Applications in Version 5.2*

---

### **EFFECTIVE NONLINEAR MAGNETIC CURVES CALCULATOR**

Companion to the new Effective Nonlinear Constitutive Relations formulation described above, this utility application can be used to compute the effective HB and BH curves for a material given the interpolation data for the BH or HB curve. The computed effective curves are calculated using one of two different energy methods defined by the user. The resulting effective material properties can be exported as a COMSOL Material Library file.

### **TOUCHSCREEN SIMULATOR**

This application computes the capacitance matrix of a touchscreen in the presence of a human finger phantom, where the position and orientation of the finger is controlled via input parameters. Intended as a tool for early proof-of-concept in capacitive touch

device development, the resulting capacitance matrix is evaluated as well as the electric field norm.

### **MAGNETIC PROSPECTING**

With this application, you can simulate the effect of a deposit of magnetic ore (such as magnetite or hematite) on the geomagnetic field in the region. Users can import heightmap images or Digital Elevation Model (DEM) files to define the orography of the region and can download the geomagnetic field data for the specified location from the Internet. The computation generates a plot of the magnetic anomaly on the Earth's surface and numerical data on the expected anomaly at specified measurement locations in the region.

### **INDUCTION HEATING OF A STEEL BILLET**

This application simulates an induction heating system for the reheating of steel billets. Specifically, it simulates a billet traveling at constant speed through AC coils, heating up due to the induced eddy currents. The geometry of the system, including the number of coils, their placement, and the billet cross section, is parameterized to make those parameters available as inputs. The app contains preset billet cross sections, and additional cross sections can be imported as geometry parts from other COMSOL Multiphysics files. The temperature range in the billet and the power balance of the system are presented after the computation as analytical results, together with visual plots of the electric current density and the temperature field.

### *Backward Compatibility with Version 5.0 and older*

---

### **COIL CURRENT CALCULATION**

When models saved in previous versions are opened in version 5.1, the Coil Current Calculation study steps are migrated to Coil Geometry Analysis. Old solutions stored in the model can still be postprocessed, and the solver sequence will be regenerated automatically (with a Stationary solver) the first time the study is solved.

The Coil Geometry Analysis study step (formerly Coil Current Calculation) will now solve for all the coils in the active interfaces. To solve only for specific coils (specified using the `CoilName` property), set the `SpecifyCoil` property to 1.

The default solver sequence generated by the Coil Geometry Analysis is different in version 5.1. Code that accesses specific solver features in the generated solver sequence may need to be reviewed.

## **OTHER COIL IMPROVEMENTS**

Harmonic Perturbation subnodes under coil features (Single-Turn Coil, Multi-Turn Coil) are now global features, so the call to the create method should use the appropriate space dimension (-1):

```
model.physics("mf").feature("stcd1").create("hp1",  
    "CoilHarmonicPerturbation", -1);
```

Global features have no selections, so code that accesses the selection of the Harmonic Perturbation features may need to be reviewed.

Some of the improvements in the 3D Multi-Turn Coil features may require a review of existing code that uses the COMSOL API.

Subnodes required to set up the coil features are now added automatically. Existing code that uses the coil features may need to be updated.

The parameters `eCoil` and `length` have been moved from the Multi-Turn Coil features (boundary and domain) to the new subfeature `UserDefinedCoilGeometry`.

## **OTHER IMPROVEMENTS**

New functionality introduced in version 5.1 is disabled by default when opening models created in previous versions:

- New boundary conditions for Gauge Fixing features
- Accurate coil voltage calculation

This functionality can be enabled using the appropriate inputs in the Settings window. Refer to the documentation for the individual features for more details.

# Acoustics Module

*New Functionality in Version 5.2*

---

## **OCTAVE BAND PLOTS**

A new dedicated acoustics plot, called Octave Band Plot, for representing frequency domain transfer functions, responses, sensitivity curves, insertion and transmission loss. The plot has several built-in acoustics-specific features such as predefined weighting (Z-, A-, C-, and user-defined weighting) and the possibility to plot the desired response function in octaves, 1/3 octaves, or as a continuous curve. The band option corresponds to plotting the average or integrated value of, for example, the squared pressure over a given frequency band defined by the mid-frequency and the bandwidth. The data input to an octave band plot is a frequency-domain solution; for example, it can be the acoustic pressure resulting from a Frequency Domain study or a parametric frequency sweep. The plot will automatically plot a given expression type on the dB scale; this simplifies postprocessing as it is no longer necessary to define the expression as a variable. The input to geometric entity of the plot can be global, point, edge, boundary, or domain. On the latter three, an average is automatically performed. This means that, for example, an average power at an inlet is easy to define and plot.

## **DISSIPATED ENERGY VARIABLES IN PRESSURE ACOUSTICS**

The dissipated power density for all fluid models has been introduced for Pressure Acoustics, Poroacoustics, and Narrow Region Acoustics. The variable is called `acpr.Q_pw` and is located under the **Heating and losses** section in the **Replace Expressions** menu when working with results. The expression is valid in the plane wave limit for traveling waves. This predefined variable is used in the Focused Ultrasound Induced Heating in Tissue Phantom application where dissipated acoustic energy heats a tissue phantom.

## **NORMAL VELOCITY AND NORMAL DISPLACEMENT BOUNDARY CONDITIONS IN PRESSURE ACOUSTICS**

In the Pressure Acoustics interfaces, the normal acceleration boundary condition is now supplemented by two new boundary conditions for prescribing a normal velocity or a normal displacement. These conditions simplify the modeling procedure when modeling sources in acoustics. The source in the Generic 711 Coupler—An Occluded Ear-Canal Simulator application, for example, is defined via a normal displacement.

## **UPDATED INTENSITY VARIABLES IN ALL ACOUSTICS INTERFACES**

The intensity variables have been updated in all acoustics interfaces. The variables are now consistent across physics and across study types. Intensity is defined in the frequency domain (averaged values over one period) and the so-called instantaneous intensity is defined in the time domain. The intensity variables in the Thermoacoustics and Linearized Navier-Stokes interfaces now also include the viscous stress contributions. The variables are available for postprocessing when clicking the **Replace Expression** button.

## **MINOR UPDATES AND BUG FIXES**

Several minor updates and bug fixes are also available with this release:

- PARDISO solver used in Thermoacoustics and Linearized Navier-Stokes now uses the Multithreaded forward and backward solve option by default. This setting provides minor speedup in general, with the most noticeable effect in eigenfrequency problems.
- Model inputs are updated in Pipe Acoustics.
- An entropy variable, `ta.s_entropy`, is now available in Thermoacoustics for use in postprocessing.
- The user-defined band structure behavior is updated in the Acoustic Diffusion Equation interface.
- The dissipated thermal and viscous power density variable has been updated in Thermoacoustics.
- The slip condition is updated in Thermoacoustics, Frequency Domain to avoid locking on curved boundaries.
- The slip condition is updated in Linearized Navier-Stokes to avoid locking on curved boundaries.

## *New Applications in Version 5.2*

---

### **ABSORPTIVE MUFFLER DESIGNER**

This is an application for studying and designing a simple resonant muffler with a porous lining. The analysis includes both inductive and resistive damping in the muffler. This application enables the user to study the results of modifying the dimensions of the muffler, the ambient working conditions, and the material properties of the porous liner. The application is based on the Absorptive Muffler model located in the Automotive folder in the Acoustics Module.



### **ONE-FAMILY HOUSE ACOUSTICS ANALYZER**

The One-Family House Acoustics Analyzer is used to assess noise propagation in coupled rooms inside a two-story family house. The house consists of 10 rooms. The application will determine the sound pressure level (SPL) distribution in the house given a number of sources interactively placed in the house. The application illustrates a classical room acoustics problem where engineers or architects want to determine the noise environment in a system of coupled rooms like homes, office spaces, or workshops. This is, for example, important when verifying that an acoustic environment complies with noise and work regulations. An application, inspired by the one presented here, will allow users to get on-site and test different noise source scenarios, different wall insulation characteristics, and compare with live measurements. All in all, such an application allows for a faster characterization and solution. The application lets you place, remove, and define multiple acoustic sources in different rooms of the house in order to determine the resulting sound pressure level distribution. The acoustics are modeled using the Acoustic Diffusion Equation physics interface, which is fast and efficient at determining SPL distributions.

### **ORGAN PIPE DESIGN**

Working with acoustics, designing instruments or other sound transducing devices it can be interesting to not only look at a simulated frequency response but also listen to the sound generated by the device. This is auralization of the acoustics of the device. This demonstration application shows how this can be achieved using a simple model of an organ pipe. Changing the organ pipe design will, for example, result in a different pitch, which can be auralized. The organ pipe is modeled using the Pipe Acoustics, Frequency Domain interface. The application allows users to analyze how the first fundamental resonance frequency varies with pipe properties such as radius and wall thickness, and with the ambient pressure and temperature. The full frequency response is found including the fundamental frequency and the harmonics. Using a Java method, the results analysis is extended beyond the built-in functionality. The application will detect and find the location and amplitude of all harmonics in the response. It will also allow users to listen to the sound and pitch of a given organ design.

### **ACOUSTIC REFLECTION ANALYZER FOR A WATER-SEDIMENT INTERFACE**

This application analyzes the reflection and absorption coefficients of plane acoustic waves, scattering off a water-sediment interface, at different frequencies and angles of incidence. Moreover, the random-incidence absorption coefficients are determined for the studied frequencies. The ability of the Poroelastic Waves interface to model the

coupled acoustic and elastic waves in any porous substance (Biot's theory) is used to describe the water-sediment system. The application is based on the Acoustic Reflections Off a Water-Sediment Interface model located in the Underwater Acoustics folder.

#### **NEW MODELS IN THE APPLICATION GALLERY**

Four new applications have been added to the Application Gallery found on [www.comsol.com](http://www.comsol.com):

- **Vibrating Plate in a 2D Viscous Parallel Plate Flow:** This is a small 2D demonstration model that couples the linearized Navier-Stokes Frequency Domain, the Solid Mechanics, and the Creeping Flow physics interfaces to model the vibrations of a plate located in a 2D viscous parallel plate flow.
- **Apartment Acoustics Analyzed using the Acoustic Diffusion Equation:** The sound distribution from a TV in a two-room apartment is computed. The simulation demonstrates the use of the Acoustic Diffusion Equation interface to get a quick and simple estimate of the local sound pressure level. For increased accuracy, an analytical expression for the direct sound is added in the living room.
- **Acoustic-Solid Interaction with two Perfectly Matched Layers (PMLs):** This small tutorial model shows how to set up a model with both a solid mechanics and a pressure acoustics domain including two perfectly matched layers (PMLs), one for each physics.
- **Shape Optimization of a Tweeter Waveguide:** This application illustrates how to use the optimization capabilities in COMSOL Multiphysics to automatically develop novel designs satisfying critical design constraints. The model optimizes a simple speaker geometry. Examples of constraints could include the radius of the loudspeaker or a desired minimum achievable sound-pressure level.

#### *Backward Compatibility with Version 5.1*

---

- Plane Wave Radiation has been removed from Linearized Potential Flow, Frequency Domain and Linearized Potential Flow, Transient in 1D axisymmetry.
- A new Lagrange multiplier variable (`ta_lm_slip`) has been added to the slip conditions in Thermoacoustics, Frequency Domain. If you run a model with a segregated solver you will need to add this variable to the group containing the velocity degree of freedom. You can also regenerate the default solver. In the Java API, you need to add `comp1_ta_lm_slip` to the declaration of the segregated step, for example, adding:

```
model.sol("sol1").feature("s1").feature("se1").feature("ss1").  
set("segvar", new String[]{"comp1_p", "comp1_u", "comp1_T",  
"comp1_ta_lm_slip"});
```

- The *Waveguide end impedance* option has been removed from 2D, 1D axisymmetric, and 1D as it had no physical meaning in these space dimensions.

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

---

- The old perfectly matched layer (PML) node 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 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 volume sources 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 Functionality in Version 5.2*

---

## **COMPENSATE FOR TUBE VOLUME IN ELECTROLYTE IN THE CURRENT DISTRIBUTION ON EDGES, BEM INTERFACE**

By using this new feature, the effect of the volume of the tubes is now considered in the electrolyte charge transfer equations.

## **NAME CHANGES OF ELECTROLYTE-ELECTRODE BOUNDARY INTERFACE AND ELECTROLYTE-ELECTRODE DOMAIN INTERFACE NODES**

The new names are Electrode Surface and Internal Electrode Surface, respectively.

## **WEAK CONSTRAINTS NOW USED BY DEFAULT IN PRIMARY CURRENT DISTRIBUTION BY THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

To improve accuracy the default constraint setting has been change to use Weak Constraints in Primary Current Distributions.

The change may cause Java API backward compatibility issues with regards to slightly different results or issues with the convergence of certain iterative solvers.

## **STOICHIOMETRY SECTION IN THE DOUBLE LAYER CAPACITY AND POROUS DOUBLE LAYER CAPACITY NODES**

A **Stoichiometry** section has been added to the Double Layer Capacity and Porous Double Layer Capacity nodes in the Tertiary Current Distribution, Nernst-Planck, and Electroanalysis interfaces. Use the stoichiometry coefficients to control how ions flow into and out from the double layer. The unused stoichiometry field input in the (Porous) Electrode Reaction node has also been removed.

## **CYCLIC VOLTAMMETRY NOW INCORPORATES EXPLICIT EVENTS AT THE VERTEX POTENTIALS**

By the use of explicit events at the vertex potentials the accuracy of cyclic voltammetry in the Electroanalysis interface has been improved. Old models will open with the settings turned off. To turn them on, you need to enable **Show Advance Physics Options** in the Model Builder window and click the corresponding check box under **Advanced Settings** in the Electrode Surface node.

The change may cause Java API backward compatibility issues with regards to slightly different results.

#### **BOUNDARY ELECTRIC POTENTIAL INITIAL VALUE CAN NOW BE SET IN THE ELECTRODE CURRENT AND THE ELECTROLYTE CURRENT NODES**

You can now set the initial value for the boundary potential in the Electrode Current and Electrolyte Current nodes. By changing the initial value, convergence properties of certain models can be improved.

#### **NEW HARMONIC PERTURBATION SUBNODES FOR THE ELECTROLYTE CURRENT AND ELECTRODE CURRENT BOUNDARY NODES, ALONG WITH DEFAULT NYQUIST PLOTS**

Harmonic perturbation (used in AC Impedance studies) is now supported by the Electrolyte Current and the Electrode Current nodes.

Default Nyquist plots are also added by default when using these nodes.

#### **MATERIAL LIBRARY UPDATE OF BATTERY ELECTROLYTE CONDUCTIVITIES**

The conductivity of battery electrolytes has been set to  $10^{-6}$  at zero electrolyte salt concentrations. The change improves accuracy and convergence for battery models running at very high current densities during prolonged times.

Note that old models will not be affected by the change. You need to remove and add the electrolyte material to your model again in order to apply new material data.

#### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE FLOW COUPLING MULTIPHYSICS NODE**

The Multiphysics Flow Coupling node now supports Darcy's Law interface as Source, and the Transport of Concentrated Species, Electroanalysis and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility since the Source or Destination may now have to be explicitly specified.

#### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE TEMPERATURE COUPLING MULTIPHYSICS NODE**

The Multiphysics Temperature Coupling node now supports Darcy's Law; Transport of Concentrated Species; Primary Current Distribution; Secondary Current

Distribution; Tertiary Current Distribution, Nernst-Planck; and the Lead-Acid Battery interfaces as Destination.

The change may have implications on the Java API backward compatibility since the Destination may now have to be explicitly specified.

### *New and Updated Applications in Version 5.2*

---

#### **NEW DIFFUSE DOUBLE LAYER APPLICATION**

A new `diffuse_double_layer` application has been added to the application library.

#### **UPDATED AND REVISED APPLICATIONS**

- The `ac_fuel_cell` now uses the new Harmonic Perturbation subnode to Electrode Current node.

### *Backward Compatibility with Previous Versions*

---

#### **TRANSPORT OF CONCENTRATED SPECIES INTERFACE**

See [Chemical Reaction Engineering Module](#).

#### **SURFACE REACTIONS INTERFACE**

See [Chemical Reaction Engineering Module](#).

# CAD Import Module, Design Module, and LiveLink Products for CAD

## *New Functionality in the CAD Import Module, Design Module, and LiveLink Products for CAD in Version 5.2*

---

The CAD file import functionality included with these products has been extended to support new versions for some of the supported file formats (see under Read from File, CAD, on [www.comsol.com/products/specifications/cad](http://www.comsol.com/products/specifications/cad)).

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

The CAD Import Module, the Design Module, and the LiveLink™ products for CAD utilize 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 5.2 includes an upgraded version of the Parasolid kernel and as a result, a number of stability issues have been fixed, which makes the import of CAD models and solid operations more robust.

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

---

The LiveLink™ interface add-in now loads much faster when starting Autodesk® Inventor®.

## *New Functionality in LiveLink™ for Solid Edge®*

---

The LiveLink™ interface now supports Solid Edge® ST8.

### **BACKWARD COMPATIBILITY WITH VERSION 5.0**

The default value for the `keepfree` property of the `LiveLinkSolidEdge` function is now set to `on`. Previously the default was set to `off`.

## *New Functionality in LiveLink™ for SOLIDWORKS®*

---

- The LiveLink™ interface now supports SOLIDWORKS® 2016.
- Global variables defined with units in SOLIDWORKS can now be synchronized.

**BACKWARD COMPATIBILITY WITH VERSION 5.0**

The default value for the `keepfree` property of the `LiveLinkSOLIDWORKS` function is now set to `on`. Previously the default was set to `off`.



# CFD Module

## *New Functionality in Version 5.2*

---

### **THREE-PHASE FLOW, PHASE FIELD INTERFACE**

The new *Three-Phase Flow, Phase Field* interface can be used to model the flow and interaction of three different, immiscible fluids when the exact positions of the interfaces separating the fluids are of interest, also known as separated flow with surface tracking. The fluid-fluid interfaces are tracked using a ternary phase field formulation that accounts for differences in the fluids' densities and viscosities, and the formulation includes the effects of surface tension. The phase-field method can handle moving contact lines on no-slip boundaries. The movement of the fluid-fluid interfaces is determined by the minimization of free energy.

### **TERNARY PHASE FIELD**

The corresponding *Ternary Phase Field* interface, used to track moving interfaces between three immiscible phases in the CFD Module and the Microfluidics Module, is also found as a standalone Mathematics interface.

### **ROTATING MACHINERY FLUID FLOW INTERFACES: TURBULENT FLOW, ALGEBRAIC YPLUS AND TURBULENT FLOW, L-VEL**

Two new physics interfaces have been added in the Rotating Machinery Fluid Flow branch: the *Turbulent Flow, Algebraic  $\gamma$ Plus* interface and the *Turbulent Flow, L-VEL* interface. For these algebraic turbulence models, the turbulent viscosity is determined using two different extensions of the logarithmic wall law, and the local Reynolds number based on the distance to the nearest wall. Algebraic turbulence models are computationally cheaper and more robust but generally less accurate than transport equation turbulence models, such as the  $k$ - $\epsilon$  and  $k$ - $\omega$  turbulence models.

### **STATIONARY FREE SURFACE**

This new feature in the Rotating Machinery Fluid Flow interfaces estimates the deformation of a free surface due to the combined effects of fluid flow and volume forces (such as gravity) when solving a quasi-steady flow using the Frozen Rotor study type. An average pressure is applied on the selected boundary in the fluid-flow computation, and the surface elevation is evaluated from the resulting pressure variations on the boundary in a postprocessing study step.

### **INKJET**

The Inkjet application demonstrates how to model the flow of a droplet produced by an inkjet nozzle, such as one in a printer. An ink droplet is ejected from the nozzle and travels through air until it hits the target. The fluid flow is incompressible, and the movement of the interface between the droplet and the ambient air is modeled using the level-set method. The user can specify liquid properties in designated input fields, and the geometry—including nozzle, reservoir, compartment, and outlet—is fully parameterized. In addition, the user can control the injection pulse profile by varying the pulse interval and smoothing time.

### **NACA AIRFOIL OPTIMIZATION**

The lift and drag coefficients of a fully parametrized NACA airfoil can be computed in this application. It incorporates an Optimization interface to find the wing geometry parameters that maximize the lift-to-drag ratio. The flow regime, laminar or turbulent, is detected automatically based on the user-defined Reynolds number. Low Reynolds number simulations are performed with the Laminar Flow interface. Higher Reynolds number simulations use the Spalart-Allmaras turbulence model, which has been specifically developed for airfoil design simulations.

### **WATER TREATMENT BASIN**

This application runs simulations of a water treatment basin used for the mixing and reacting of an added chemical species (for example, chlorine). The supported geometries are rectangular basins with baffles attached to the sides of the basin in vertical or horizontal configurations, and the inlet and outlet channels can have a circular or rectangular cross section. The graphical results show the flow field and the concentration field of the added chemical species given a first-order chemical reaction, and the numerical results give the space time, half life, and pressure drop.

### *Backward Compatibility with Version 5.0*

---

### **EULER-EULER MODEL, LAMINAR FLOW**

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity,  $\mu_D$ , by the variable `ee.phidPos`.

## TRANSPORT OF CONCENTRATED SPECIES

See [Chemical Reaction Engineering Module](#).

### *Backward Compatibility with Version 4.4*

---

#### **INLET FEATURE**

The old inlet features still exist, but have been excluded from the physics context menu. Old models retain old Inlet features, but adding a new Inlet feature comes with the new functionality. Old Java files create Inlet features of the old types. The Normal stress condition can still be prescribed on inlets using an Open Boundary or a Boundary Stress feature.

#### **NEW NAMES FOR MASS TRANSPORT INTERFACES**

Running Model Java-files from previous versions may fail due to the new default name in the Transport of Diluted Species and Transport of Concentrated Species interfaces. This can be avoided by adding a command that specifies the identifier in accordance with the Name of the interface. For example, when creating a Transport of Concentrated Species interface using the old name (chcs):

```
model.physics().create("chcs", "ConcentratedSpecies", "geom1",  
    new String[][]{{"wO2", "wH2O", "wN2"}});
```

Add the following line to specify the identifier accordingly:

```
model.physics("chcs").identifier("chcs");
```

#### **WALL DISTANCE EQUATION**

The old Continuity node is still included in the interface but is excluded from the feature list. Hence, old models retain their old Continuity node when opened in version 5.0 and old Model Java-files are fully backward compatible.

#### **EULER-EULER MODEL, LAMINAR FLOW**

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity,  $\mu_D$ , by the variable `ee.phidPos`.

### *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.

#### **EULER-EULER MODEL, LAMINAR FLOW**

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity,  $\mu_D$ , by the variable `ee.phidPos`.

### *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 previous versions. As 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 physics interface settings.

#### **EULER-EULER MODEL, LAMINAR FLOW**

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in version 5.1 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity,  $\mu_D$ , by the variable `ee.phidPos`.

#### **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 ways:

- 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 clear their time-dependent solutions when opened in 5.2. 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 5.2.

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  $-\mathbf{Q}_{br} \cdot \mathbf{u}/\epsilon_p^2$  in the right hand side previously lacked the factor  $1/\epsilon_p^2$ , where  $\epsilon_p$  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 have 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
- The 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 prior to version 4.3 must either be re-solved in version 5.0 for postprocessing, or opened and re-saved in version 4.3 before being opened in version 5.0.

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 5.2 than in version 4.3 and the results can differ slightly between the versions.

#### **EULER-EULER MODEL, LAMINAR FLOW**

The results from models using the Euler-Euler Model, Laminar Flow interface may differ in between versions 4.3 and 5.2 due to the more general formulation for the viscous stress in the momentum equation for the dispersed phase. The behavior in the old formulation may be reproduced by dividing the dispersed phase viscosity,  $\mu_D$ , by the variable `ee.phidPos`.



# Chemical Reaction Engineering Module

*New Functionality in Version 5.2*

---

## **NEW CATALYTIC PARTICLE SHAPES IN REACTIVE PELLET BEDS**

In the Reactive Pellet Bed feature, you can now specify non-spherical catalytic pellet shapes. Cylinder shapes, flakes, and user-defined pellet shapes are now available as options.

## **EDITABLE SPECIES REACTION RATES IN SPECIES NODES, REACTION ENGINEERING AND CHEMISTRY INTERFACES**

It is now easy to change the definition of the reaction rate of a species directly in the Species node. The entered expression overrides the definition set up by the Reaction feature, where the rate is defined by the stoichiometry of the reactions in which the species participates.

## **IMPROVED EQUILIBRIUM REACTION FUNCTIONALITY, REACTION ENGINEERING INTERFACE**

For equilibrium reactions, the **Suppress negative concentrations** check box has been introduced to aid the computation of equilibrium systems. A selected check box ensures that no negative values of concentrations are accepted as solutions to the equilibrium condition.

## **MORE VERSATILE IMPORT OF EXPERIMENTAL DATA FOR PARAMETER ESTIMATION, REACTION ENGINEERING INTERFACE**

The Experiment feature can now convert the units of the imported data used for Parameter Estimation. The importance of each data type during the optimization can also be controlled with the weight functionality.

## **MOLAR AND MILLIMOLAR ADDED AS CONCENTRATION UNIT**

M (moles/l) and mM (mmoles/l) were added as units to visualize concentration, in addition to the SI unit moles/m<sup>3</sup>.

### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE FLOW COUPLING MULTIPHYSICS NODE**

The Multiphysics Flow Coupling node now supports Darcy's Law interface as Source, and the Transport of Concentrated Species and Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Source or Destination may now have to be explicitly specified.

### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE TEMPERATURE COUPLING MULTIPHYSICS NODE**

The Multiphysics Temperature Coupling node now supports Darcy's Law, Transport of Concentrated Species, and the Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Destination may now have to be explicitly specified.

## *New and Updated Applications in Version 5.2*

---

### **MEMBRANE DIALYSIS APPLICATION**

This new application computes the contaminant concentration in a blood stream that undergoes dialysis. The contaminant removal within a single hollow fiber can be investigated by varying the fiber geometry and mass transport properties.

### **LIQUID CHROMATOGRAPHY APPLICATION**

A new runnable app for studying the separation of two chemical components in a high-performance liquid chromatography (HPLC) column.

### **FERMENTATION IN BEER BREWING**

The application models the fermentation process in 0D with the Reaction Engineering interface. Additionally, it models the process in a sphero-conical tank geometry with mass transfer and heat transfer, and natural convection accounted for. These two model setups enable in-depth investigation of several parameters affecting the final alcohol content and the taste of the beer.

## **REACTION ENGINEERING AND CHEMISTRY INTERFACES**

The Java API syntax for creating surface species has changed. Running Java files from previous versions may fail due to the new default name index **ads** for surface species and **s** for solids in the *Reaction Engineering* and *Chemistry* interfaces. The surface species names and solid species names need to be changed accordingly in any Java code:

- Change any species index (s) to (ads)
- Change any species index (S) to (s)

The following shows an example of the needed code change:

Version 5.0 and earlier:

```
model.physics("re").feature("rch1").set("formula", "A+B(S)=>C+D(s)");
model.physics("re").feature("spec1").set("specName", "As(s)");
model.physics("chem").feature("rch1").set("formula", "A+B(S)=>C+D(s)");
model.physics("chem").feature("spec1").set("specName", "As(s)");
```

Change this to

Version 5.2:

```
model.physics("re").feature("rch1").set("formula", "A+B(s)=>C+D(ads)");
model.physics("re").feature("spec1").set("specName", "As(ads)");
model.physics("chem").feature("rch1").set("formula", "A+B(s)=>C+D(ads)");
model.physics("chem").feature("spec1").set("specName", "As(ads)");
```

## **TRANSPORT OF CONCENTRATED SPECIES INTERFACE**

### *Updated Internal Variable Names*

The internal variable names of vector and tensor components have changed in the *Transport of Concentrated Species* interface. In most cases, users will not be affected by this change. When opening a model file created in a previous version, the previous variable definitions remain. However, when updating or re-computing the solution, new variable names will be generated. This means that if any of these variable names have been used in a user-defined expression or in a plot, they need to be updated manually as described below.

In version 5.1, the spatial direction indicator string (x, xy and so on) has been moved to the end of the variable name. This applies to all vector and tensor variables generated by the interface.

For example, the diffusive flux vector components have changed from (5.0 and earlier):

```
tcs.dfluxx_w1, tcs.dfluxy_w1, tcs.dfluxz_w1
```

to (5.2)

```
tcs.dflux_w1x, tcs.dflux_w1y, tcs.dflux_w1z
```

and the components of the diffusion tensor, defined by the Fick's Law diffusion model have been changed from (5.0 and earlier):

```
tcs.Dfxx_w1, tcs.Dfxy_w1, tcs.Dfxz_w1, tcs.Dfyx_w1, ...
```

to (5.2)

```
tcs.Df_w1xx, tcs.Df_w1xy, tcs.Df_w1xz, tcs.Df_w1yx, ...
```

### Java API Syntax Changes

The Java API syntax for setting physics properties has been changed in the *Transport of Concentrated Species* interface. Now the property name corresponds to the name of the GUI section containing the property. In order to run Java files from previous versions, apply the syntax changes as given in the examples below:

Version 5.0 and earlier:

```
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",  
"MaxwellStefan", 0);  
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",  
"MixtureAveraged", 0);  
model.physics("tcs").prop("DiffusionModel").setIndex("DiffusionModel",  
"FicksLaw", 0);  
model.physics("tcs").prop("Convection").setIndex("Convection", "0", 0);  
model.physics("tcs").prop("Migration").setIndex("Migration", "0", 0);  
model.physics("tcs").prop("Regularization").setIndex("Regularization", "1",  
0);
```

For version 5.2, change this to:

```
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",  
"MaxwellStefan");  
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",  
"MixtureAveraged");  
model.physics("tcs").prop("TransportMechanism").set("DiffusionModel",  
"FicksLaw");  
model.physics("tcs").prop("TransportMechanism").set("Convection", false);  
model.physics("tcs").prop("TransportMechanism").set("Migration", false);  
model.physics("tcs").prop("AdvancedSettings").set("Regularization", "0");
```

## SURFACE REACTIONS INTERFACE

### *Updated Internal Variable Names*

The same pattern for vector and tensor variable name changes as in [Transport of Concentrated Species Interface](#), described above, holds for the Surface Reactions interface in version 5.1.

### *Java API Syntax Changes*

The Java API syntax for setting the diffusion tensor of surface species, the molar mass and density of the bulk species, and initial values have been changed in the *Surface Reactions* interface.

### **Setting Diffusion Tensor**

The surface species name needs to be included in the first argument of the set method. In order to run Java files from previous versions, apply the syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("sp1").set("D_0", new String[]{"Dcs1", "0", "0", "0", "Dcs1", "0", "0", "0", "Dcs1"});
model.physics("sr").feature("sp1").set("D_1", new String[]{"Dcs2", "0", "0", "0", "Dcs2", "0", "0", "0", "Dcs2"});
```

For version 5.2, change this to:

```
model.physics("sr").feature("sp1").set("D_cs1", new String[]{"Dcs1", "0", "0", "0", "Dcs1", "0", "0", "0", "Dcs1"});
model.physics("sr").feature("sp1").set("D_cs2", new String[]{"Dcs2", "0", "0", "0", "Dcs2", "0", "0", "0", "Dcs2"});
```

### **Setting Molar Mass and Density**

The bulk species name needs to be included in the first argument of the `setIndex` method. Apply the Java API syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("sp1").set("M_0", "Mcb1");
model.physics("sr").feature("sp1").set("M_1", "Mcb2");

model.physics("sr").feature("sp1").set("rho_0", "rhocb1");
model.physics("sr").feature("sp1").set("rho_1", "rhocb2");
```

For version 5.2, change this to:

```
model.physics("sr").feature("sp1").setIndex("M_cb1", "Mcb1", 0);
model.physics("sr").feature("sp1").setIndex("M_cb2", "Mcb2", 0);
```

```
model.physics("sr").feature("sp1").setIndex("rho_cb1", "rhocb1", 0);
model.physics("sr").feature("sp1").setIndex("rho_cb2", "rhocb2", 0);
```

### Setting Initial Values

Apply the Java API syntax changes as in the example below:

Version 5.0 and earlier:

```
model.physics("sr").feature("init1").set("cs1", "cs10");
model.physics("sr").feature("init1").set("cs2", "cs20");
model.physics("sr").feature("init1").set("cb1", "cb10");
model.physics("sr").feature("init1").set("cb2", "cb20");
```

For version 5.2, change this to:

```
model.physics("sr").feature("init1").setIndex("initcs", "cs10", 0);
model.physics("sr").feature("init1").setIndex("initcs", "cs20", 1);
model.physics("sr").feature("init1").setIndex("initcb", "cb10", 0);
model.physics("sr").feature("init1").setIndex("initcb", "cb20", 1);
```

# Corrosion Module

*New Functionality in Version 5.2*

---

## **COMPENSATE FOR TUBE VOLUME IN ELECTROLYTES IN THE CURRENT DISTRIBUTION ON EDGES, BEM INTERFACE**

By using this new feature, the effect of the volume of the tubes is now considered in the electrolyte charge transfer equations.

## **NAME CHANGES OF ELECTROLYTE-ELECTRODE BOUNDARY INTERFACE AND ELECTROLYTE-ELECTRODE DOMAIN INTERFACE NODES**

The new names are Electrode Surface and Internal Electrode Surface, respectively.

## **WEAK CONSTRAINTS NOW USED BY DEFAULT IN PRIMARY CURRENT DISTRIBUTION BY THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

To improve accuracy the default constraint setting has been change to use Weak Constraints in Primary Current Distributions.

The change may cause Java API backward compatibility issues with regards to slightly different results or issues with the convergence of certain iterative solvers.

## **STOICHIOMETRY SECTION IN THE DOUBLE LAYER CAPACITY AND POROUS DOUBLE LAYER CAPACITY NODES**

A **Stoichiometry** section has been added to the Double Layer Capacity and Porous Double Layer Capacity nodes in the Tertiary Current Distribution, Nernst-Planck, and Electroanalysis interfaces. Use the stoichiometry coefficients to control how ions flow into and out from the double layer. The unused stoichiometry field input in the (Porous) Electrode Reaction node has also been removed.

## **CYCLIC VOLTAMMETRY NOW INCORPORATES EXPLICIT EVENTS AT THE VERTEX POTENTIALS**

By the use of explicit events at the vertex potentials the accuracy of cyclic voltammetry in the Electroanalysis interface has been improved. Old models will open with the settings turned off. To turn them on, you need to enable **Show Advance Physics Options** in the Model Builder window and click the corresponding check box under **Advanced Settings** in the Electrode Surface node.

The change may cause Java API backward compatibility issues with regards to slightly different results.

#### **INITIAL VALUES FOR BOUNDARY ELECTRIC POTENTIAL CAN NOW BE SET IN THE ELECTRODE CURRENT AND THE ELECTROLYTE CURRENT NODES**

You can now set the initial value for the boundary potential in the Electrode Current and Electrolyte Current nodes. By changing the initial value, convergence properties of certain models can be improved.

#### **NEW HARMONIC PERTURBATION SUBNODES FOR THE ELECTROLYTE CURRENT AND ELECTRODE CURRENT BOUNDARY NODES, ALONG WITH DEFAULT NYQUIST PLOTS**

Harmonic perturbation (used in AC Impedance studies) is now supported by the Electrolyte Current and the Electrode Current nodes.

Default Nyquist plots are also added by default when using these nodes.

#### **THIN INSULATING LAYER IN THE PRIMARY, SECONDARY, AND TERTIARY CURRENT DISTRIBUTION INTERFACES**

The Thin Insulating Layer node can be used to model a thin insulating sheet, located on an interior boundary in an electrolyte domain. The node can be used as an alternative to drawing the actual insulating domain in the model geometry, which may significantly reduce meshing time, especially in 3D models. Thin insulating sheets are commonly inserted in the electrolyte in various types of electrochemical cells. For example, they can be used for optimizing the current distribution in a corrosion protection application or for optimizing the local deposition rate in a deposition bath.

#### **DISSOLVING-DEPOSITING SPECIES SECTION IN THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

Use dissolving-depositing species in the Electrode Surface and Internal Electrode Surface nodes in any Current Distribution interface to calculate the accumulated surface concentration changes of dissolving or depositing processes. An arbitrary number of species can be added. The dissolving-depositing species can also be used to calculate the resulting boundary deformation velocity. The deformation velocity can be used to couple the current distribution interface to the Deformed Geometry interface (see below).

You specify the reaction rates in the **Stoichiometry** section in the Electrode Reaction subnodes.



### **REDESIGNED CORROSION, DEFORMED GEOMETRY INTERFACES**

The Corrosion, Deformed Geometry interfaces have been redesigned as predefined multiphysics interfaces. Upon choosing a Corrosion, Deformed Geometry interface in the Select Physics menu, the corresponding Current Distribution interface and a Deformed Geometry interface are added to the model, together with a Non-Deforming Boundary and a Deforming Electrode Surface Multiphysics node. The automatically added Deforming Electrode Surface Multiphysics node is used to couple the dissolution/deposition rate of an Electrode Surface node (see the Dissolving-Depositing Species Section above) in the Current Distribution interface to the boundary velocity in the Deformed Geometry interface.

In addition, the moving boundary smoothing in 3D now also acts on the intersecting edges between deforming and non-deforming boundaries.

Models created prior to 5.2 using a Corrosion interface will not be affected by the change.

### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE FLOW COUPLING MULTIPHYSICS NODE**

The Flow Coupling multiphysics node now supports Darcy's Law interface as Source, and the Transport of Concentrated Species, Electroanalysis and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Source or Destination may now have to be explicitly specified.

### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE TEMPERATURE COUPLING MULTIPHYSICS NODE**

The Temperature Coupling multiphysics node now supports Darcy's Law; Transport of Concentrated Species; Primary Current Distribution; Secondary Current Distribution; and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Destination may now have to be explicitly specified.

## *New and Updated Applications in Version 5.2*

---

### **NEW DIFFUSE DOUBLE LAYER APPLICATION**

A new simpler `diffuse_double_layer` application has been added to the application library.

The old diffuse double layer application has been renamed `diffuse_double_layer_with_charge_tran`.

### **UPDATED AND REVISED APPLICATIONS**

- The `ship_hull` application has been revised with updated boundary conditions for more accurate ICCP control and updated plots.
- The `oil_platform` application has been revised with regards to the plotting of potentials.

## *Backward Compatibility with Earlier Versions*

---

### **SURFACE REACTIONS INTERFACE**

See [Chemical Reaction Engineering Module](#).

# ECAD Import Module

## *New Functionality in Version 5.2*

---

- The ECAD Import Module now generates selections for imported layers to enable a more efficient simulation setup. When importing, for example, ODB++ archives and GDS-II files, the selections will have easily recognizable names that are generated from the layer names. In the Import Settings window, you can rename the layers or choose to keep the names found in the file.
- The length unit of a selected ECAD file is now displayed, and a new check box either enables or disables the update of the geometry sequence length unit to the detected unit in the file.
- For GDS import, arc recognition is improved to work when interior boundaries are deleted even when the arc segments are originally not located on the same polygon.

## *Backward Compatibility with Version 5.0*

---

- The default value for the `grouping` property of the `Import` function is now `layer`. Previously the default was `all`.
- For ODB++@ and ODB++(X) files (the `type` property is `odb`) new rules apply for the initialization of the `importlayer` property of the `Import` function. Now only layers of type *Metal* and *Dielectric* are initialized with the string on in the `importlayer` string array. Previously this also included layers of type *Drill*.

# Electrochemistry Module

*New Functionality in Version 5.2*

---

## **COMPENSATE FOR TUBE VOLUME IN ELECTROLYTES IN THE CURRENT DISTRIBUTION ON EDGES, BEM INTERFACE**

By using this new feature, the effect of the volume of the tubes is now considered in the electrolyte charge transfer equations.

## **NAME CHANGES OF ELECTROLYTE-ELECTRODE BOUNDARY INTERFACE AND ELECTROLYTE-ELECTRODE DOMAIN INTERFACE NODES**

The new names are Electrode Surface and Internal Electrode Surface, respectively.

## **WEAK CONSTRAINTS NOW USED BY DEFAULT IN PRIMARY CURRENT DISTRIBUTION BY THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

To improve accuracy the default constraint setting has been change to use Weak Constraints in Primary Current Distributions.

The change may cause Java API backward compatibility issues with regards to slightly different results or issues with the convergence of certain iterative solvers.

## **STOICHIOMETRY SECTION IN THE DOUBLE LAYER CAPACITY AND POROUS DOUBLE LAYER CAPACITY NODES**

A **Stoichiometry** section has been added to the Double Layer Capacity and Porous Double Layer Capacity nodes in the Tertiary Current Distribution, Nernst-Planck, and Electroanalysis interfaces. Use the stoichiometry coefficients to control how ions flow into and out from the double layer. The unused stoichiometry field input in the (Porous) Electrode Reaction node has also been removed.

## **CYCLIC VOLTAMMETRY NOW INCORPORATES EXPLICIT EVENTS AT THE VERTEX POTENTIALS**

By the use of explicit events at the vertex potentials the accuracy of cyclic voltammetry in the Electroanalysis interface has been improved. Old models will open with the settings turned off. To turn them on, you need to enable **Show Advance Physics Options** in the Model Builder window and click the corresponding check box under **Advanced Settings** in the Electrode Surface node.

The change may cause Java API backward compatibility issues with regards to slightly different results.

#### **INITIAL VALUES FOR BOUNDARY ELECTRIC POTENTIAL CAN NOW BE SET IN THE ELECTRODE CURRENT AND THE ELECTROLYTE CURRENT NODES**

You can now set the initial value for the boundary potential in the Electrode Current and Electrolyte Current nodes. By changing the initial value, convergence properties of certain models can be improved.

#### **NEW HARMONIC PERTURBATION SUBNODES FOR THE ELECTROLYTE CURRENT AND ELECTRODE CURRENT BOUNDARY NODES, ALONG WITH DEFAULT NYQUIST PLOTS**

Harmonic perturbation (used in AC Impedance studies) is now supported by the Electrolyte Current and the Electrode Current nodes.

Default Nyquist plots are also added by default when using these nodes.

#### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE FLOW COUPLING MULTIPHYSICS NODE**

The Flow Coupling multiphysics node now supports Darcy's Law interface as Source, and the Transport of Concentrated Species, Electroanalysis and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Source or Destination may now have to be explicitly specified.

#### **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE TEMPERATURE COUPLING MULTIPHYSICS NODE**

The Temperature Coupling multiphysics node now supports Darcy's Law; Transport of Concentrated Species; Primary Current Distribution; Secondary Current Distribution; and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Destination may now have to be explicitly specified.

## *New Applications in Version 5.2*

---

### **NEW DIFFUSE DOUBLE LAYER APPLICATION**

A new simpler `diffuse_double_layer` application has been added to the application library.

The old diffuse double layer application has been renamed `diffuse_double_layer_with_charge_tran`.

## *Backward Compatibility with Previous Versions*

---

### **SURFACE REACTIONS INTERFACE**

See [Chemical Reaction Engineering Module](#).

# Electrodeposition Module

*New Functionality in Version 5.2*

---

## **COMPENSATE FOR TUBE VOLUME IN ELECTROLYTES IN THE CURRENT DISTRIBUTION ON EDGES, BEM INTERFACE**

By using this new feature, the effect of the volume of the tubes is now considered in the electrolyte charge transfer equations.

## **NAME CHANGES OF ELECTROLYTE-ELECTRODE BOUNDARY INTERFACE AND ELECTROLYTE-ELECTRODE DOMAIN INTERFACE NODES**

The new names are Electrode Surface and Internal Electrode Surface, respectively.

## **WEAK CONSTRAINTS NOW USED BY DEFAULT IN PRIMARY CURRENT DISTRIBUTION BY THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

To improve accuracy the default constraint setting has been change to use Weak Constraints in Primary Current Distributions.

The change may cause Java API backward compatibility issues with regards to slightly different results or issues with the convergence of certain iterative solvers.

## **STOICHIOMETRY SECTION IN THE DOUBLE LAYER CAPACITY AND POROUS DOUBLE LAYER CAPACITY NODES**

A **Stoichiometry** section has been added to the Double Layer Capacity and Porous Double Layer Capacity nodes in the Tertiary Current Distribution, Nernst-Planck, and Electroanalysis interfaces. Use the stoichiometry coefficients to control how ions flow into and out from the double layer. The unused stoichiometry field input in the (Porous) Electrode Reaction node has also been removed.

## **CYCLIC VOLTAMMETRY NOW INCORPORATES EXPLICIT EVENTS AT THE VERTEX POTENTIALS**

By the use of explicit events at the vertex potentials the accuracy of cyclic voltammetry in the Electroanalysis interface has been improved. Old models will open with the settings turned off. To turn them on, you need to enable **Show Advance Physics Options** in the Model Builder window and click the corresponding check box under **Advanced Settings** in the Electrode Surface node.

The change may cause Java API backward compatibility issues with regards to slightly different results.

#### **INITIAL VALUES FOR BOUNDARY ELECTRIC POTENTIAL CAN NOW BE SET IN THE ELECTRODE CURRENT AND THE ELECTROLYTE CURRENT NODES**

You can now set the initial value for the boundary potential in the Electrode Current and Electrolyte Current nodes. By changing the initial value, convergence properties of certain models can be improved.

#### **NEW HARMONIC PERTURBATION SUBNODES FOR THE ELECTROLYTE CURRENT AND ELECTRODE CURRENT BOUNDARY NODES, ALONG WITH DEFAULT NYQUIST PLOTS**

Harmonic perturbation (used in AC Impedance studies) is now supported by the Electrolyte Current and the Electrode Current nodes.

Default Nyquist plots are also added by default when using these nodes.

#### **THIN INSULATING LAYER IN THE PRIMARY, SECONDARY, AND TERTIARY CURRENT DISTRIBUTION INTERFACES**

The Thin Insulating Layer node can be used to model a thin insulating sheet, located on an interior boundary in an electrolyte domain. The node can be used as an alternative to drawing the actual insulating domain in the model geometry, which may significantly reduce meshing time, especially in 3D models. Thin insulating sheets are commonly inserted in the electrolyte in various types of electrochemical cells. For example, they can be used for optimizing the current distribution in a corrosion protection application or for optimizing the local deposition rate in a deposition bath.

#### **DISSOLVING-DEPOSITING SPECIES SECTION IN THE ELECTRODE SURFACE AND INTERNAL ELECTRODE SURFACE NODES**

Use dissolving-depositing species in the Electrode Surface and Internal Electrode Surface nodes in any Current Distribution interface to calculate the accumulated surface concentration changes of dissolving or depositing processes. An arbitrary number of species can be added. The dissolving-depositing species can also be used to calculate the resulting boundary deformation velocity. The deformation velocity can be used to couple the current distribution interface to the Deformed Geometry interface (see below).

You specify the reaction rates in the **Stoichiometry** section in the Electrode Reaction subnodes.



## **REDESIGNED ELECTRODEPOSITION, DEFORMED GEOMETRY INTERFACES**

The Electrodeposition, Deformed Geometry interfaces have been redesigned as predefined multiphysics interfaces. Upon choosing an Electrodeposition, Deformed Geometry interface in the Select Physics menu, the corresponding Current Distribution interface and a Deformed Geometry interface are added to the model, together with a Non-Deforming Boundary and a Deforming Electrode Surface Multiphysics node. The automatically added Deforming Electrode Surface Multiphysics node is used to couple the dissolution/deposition rate of an Electrode Surface node (see the Dissolving-Depositing Species Section above) in the Current Distribution interface to the boundary velocity in the Deformed Geometry interface.

In addition, the moving boundary smoothing in 3D now also acts on the intersecting edges between deforming and non-deforming boundaries.

Models created prior to 5.2 using an Electrodeposition interface will not be affected by the change.

## **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE FLOW COUPLING MULTIPHYSICS NODE**

The Flow Coupling multiphysics node now supports Darcy's Law interface as Source, and the Transport of Concentrated Species, Electroanalysis and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Source or Destination may now have to be explicitly specified.

## **EXTENDED PHYSICS INTERFACE SUPPORT FOR THE TEMPERATURE COUPLING MULTIPHYSICS NODE**

The Temperature Coupling multiphysics node now supports Darcy's Law; Transport of Concentrated Species; Primary Current Distribution; Secondary Current Distribution; and the Tertiary Current Distribution, Nernst-Planck interfaces as Destination.

The change may have implications on the Java API backward compatibility because the Destination may now have to be explicitly specified.

### *New and Updated Applications in Version 5.2*

---

## **DIFFUSE DOUBLE LAYER APPLICATION**

A new `diffuse_double` layer application has been added to the application library.

#### **UPDATED AND REVISED APPLICATIONS**

- Updated the number of active species in the `cu_electroless_deposition` application.

#### *Backward Compatibility with Previous Versions*

---

#### **SURFACE REACTIONS INTERFACE**

See [Chemical Reaction Engineering Module](#).

# Geomechanics Module

*New Functionality in Version 5.2*

---

## **IMPROVED FORMULATION FOR SMALL STRAIN PLASTICITY WITH GEOMETRIC NONLINEARITY**

When **Small plastic strains** is selected as the plasticity model in the **Plasticity** node, and the **Include geometric nonlinearity** check box is selected in the settings for the study, the Cauchy stress tensor is now used to evaluate the yield function and plastic potential. In previous versions the second Piola-Kirchhoff stress tensor was used instead. This change will give significant improvements in accuracy for moderate to large strains. As a result, you might experience changes in stresses and plastic strains for old models having significant plastic strains when this combination of settings is used.

# Heat Transfer Module

*New and Improved Functionality in Version 5.2*

---

## **SYMMETRY PLANE FOR SURFACE-TO-SURFACE RADIATION**

The new **Symmetry Plane for Surface-to-Surface Radiation** global feature is available in 2D, 2D axisymmetric, and 3D models when a **Heat Transfer with Surface-to-Surface Radiation** interface or a **Surface-to-Surface Radiation** interface is active. It is used to take advantage of plane symmetries to reduce the mesh for a more efficient computation of the view factor. Depending on the dimension, the user interface provides different inputs to define the symmetry plane through control points.

## **TEMPERATURE ON EXTERIOR SIDE OF THIN LAYERS**

The new **External Temperature** subfeature is available on external boundaries under the feature **Thin Layer**, when the **Layer type** is set to **Resistive** or **General**. It is used to specify the temperature on the exterior side of the layer.

## **TEMPERATURE ON EDGES OF THIN FILMS AND FRACTURES**

New **Temperature** subfeatures are available under the features **Thin Film** and **Fracture**. They are used to specify the temperature on a set of lines (geometrical edges in 3D models or geometrical points in 2D and 2D axisymmetric models) that represent thin boundaries of a thin film or a fracture.

## **LINE HEAT FLUX ON BOUNDARIES OF THIN FILMS AND FRACTURES**

New **Line Heat Flux** subfeatures are available under the features **Thin Film** and **Fracture**. They are used to add a heat flux across boundaries of a thin film or a fracture. Their settings provide different options for the definition of the flux: **General inward heat flux**, **Inward heat flux**, and for 3D models, **Overall heat transfer rate**.

## **SURFACE-TO-AMBIENT RADIATION ON BOUNDARIES OF THIN FILMS AND FRACTURES**

New **Surface-to-Ambient Radiation** subfeatures are available under the features **Thin Film** and **Fracture**. They are used to add surface-to-ambient radiation on a set of lines (geometrical edges in 3D models or geometrical points in 2D and 2D axisymmetric models) that represent thin boundaries of a thin film or a fracture. Their settings provide inputs for the **Ambient temperature** and the **Surface emissivity**, which are used for the definition of the net inward heat flux.

## HEAT SOURCE IN THIN FILMS AND FRACTURES

New **Heat Source** subfeatures are available under the features **Thin Film** and **Fracture**. They are used to add an internal heat source within a thin film or a fracture. Their settings provide two options for the definition of the heat source: **General Source** and **Overall heat transfer rate**.

## ADDITIONAL CORRELATION FOR HEAT TRANSFER COEFFICIENTS

A convective heat transfer coefficient correlation has been added to the heat transfer coefficients library. This correlation corresponds to external flow induced by natural convection around a vertical thin cylinder. Such a heat transfer coefficient can be used to reduce the simulation cost when the model configuration corresponds to one of the predefined correlations. In that case, the flow computation and the heat convection in the fluid are replaced by a heat flux boundary condition on the solid boundaries using the corresponding predefined correlation.

## MISCELLANEOUS

- The formulation of the **Thermal dispersion** subfeature contribution has been updated to improve the convergence when the heat transfer and the flow equations are solved together.
- The equations of **Thin Layer** and **Thin Layered Shell** features have been updated to support fully anisotropic thermal conductivity.
- The **Thin Layer** feature with the **General** option—which defines the layer as an extra dimension—is now compatible with moving frame deformations. When a **Moving Mesh** interface or structural mechanics physics interface is active for example, the material's thermal conductivity and density properties are updated to account for the deformation effects.
- The **Thin Layered Shell** feature available in the **Heat Transfer in Thin Shells** interface has been updated in the same way, to be compatible with moving frame deformations.
- The detection of moving frames in the heat transfer interfaces has been enhanced. Unnecessary updates of the view factor have been removed for efficiency when the physics interface responsible for a frame deformation is not solved. This improvement also benefits the solver suggestions, leading to better computational performances.
- In the **Thin Layered Shell** feature, the initialization of the shape function  $T_{xdim}$ —the extra dimension temperature variable—has been updated to match the

temperature's initial value in the adjacent domain. For interior boundaries,  $T_{xdim}$  is initialized as the average of adjacent initial temperatures.

- The weak equation of the **Fracture** feature has been updated to use the fluid heat capacity instead of the effective heat capacity in the convective term.
- The automatic substitution mechanism of the temperature  $T$  by the variable representing the temperature on the appropriate side has been revised. This mechanism is designed to compute the temperature at the appropriate location when it is used to define material properties or black body radiative intensity. It is now only active when the default settings are used. When a model input is defined using the **User defined** option, no substitution is made. In this case the user should call the appropriate variable name to use the temperature on a particular side. For example,  $ht.Tu$  should be used to compute the temperature on the upside. See [Boundary Wall Temperature](#) in the *Heat Transfer Module User's Guide* to get the definition of all temperature variables available on boundaries.
- The solver settings for the heat transfer models have been updated. SOR is used instead of SOR Line as the presmoothen in the multigrid solver for large models, unless the model contains a **Non-Isothermal Flow** coupling node. A lower limit of zero has been added for the temperature when a segregated solver is used and the temperature is in kelvin.
- The default opacity properties of domains adjacent to boundaries where a **Surface-to-Surface Radiation** interface is active have been updated. These domains are set as transparent by default, and can be turned to opaque using the **Opaque** feature.
- The definition of the velocity residual in time-dependent study steps with the **Non-Isothermal Flow** multiphysics coupling feature has been fixed to include the time-dependent term.

### *New Applications in Version 5.2*

---

#### **FLASH METHOD**

The flash method is one of the most widely used methods for measuring the thermal diffusivity of a given sample material. It consists of delivering a heat pulse on one side of a thin piece of material to be tested. The temperature elevation on the other side is then monitored and contains all the information to recover the thermal diffusivity.

Knowing the density and heat capacity at constant pressure, the thermal conductivity can finally be deduced. This experiment protocol is reproduced in a 2D axisymmetric model with a time-dependent study computing the temperature elevation for various operating conditions, sample dimensions, and Gaussian laser pulse profiles.

## **FINNED PIPE**

Finned pipes are used for coolers, heaters, and mainly in heat exchangers to increase thermal performance. They come in different sizes and designs depending on the application and requirements. This app calculates the thermal performance of a pipe filled with water and cooled or heated by the surrounding air. The model makes use of periodicity conditions to be able to compute only a small section of the pipe. Various geometric configurations are available for the outer fins (disc-stacked blades, circular grooves, helical blade, helical grooves, or none) and for the inner fins (none or straight grooves). Other parameters such as the pipe dimensions and the air flow conditions can also be set.

## **THERMOELECTRIC COOLER**

Thermoelectric coolers are widely used for electronics cooling in different application areas, ranging from consumer products to spacecraft. Due to this, countless configurations exist. This app calculates the thermal performance of single-stage thermoelectric coolers of different sizes and with different thermocouple materials for different operating conditions. It outputs common performance plots that are used to describe the thermoelectric cooler characteristics. The underlying 3D model uses the equations for the thermoelectric effect for thermocouples and the heat conduction equation for electrical conductors and ceramics.

*Updated Application in Version 5.2*

---

## **GEOHERMAL HEAT PUMP**

An app with a user interface has been built from the existing application. It analyzes the performance of ground-coupled heat exchangers for different specifications (depth, pattern, pipe configuration, and heating conditions), temperature conditions, and the soil's thermal conductivity and temperature gradient.

The model couples heat transfer in solids with heat transfer in pipes. With the help of the **Events** interface, the heater is turned off if the daily heat demand is achieved and is turned on again after 24 hours.

The temperature at the pipe's outlet can be controlled and compared to the minimum temperature required in the heat exchanger specifications.

*Backward Compatibility with Version 5.1*

---

## **CHANGES IN VARIABLE NAMES**

The names of heat capacity variables have changed in several features for consistency:

- In the **Bioheat** feature, `ht.cb` has become `ht.cp_b`.
- In the **Thin Layer** feature, `ht.cs` has become `ht.cp_s`.
- In the **Thin Rod** feature, `ht.c1` has become `ht.cp_1`.
- In the **Thin Layered Shell** feature, `htsh.cs1` has become `htsh.cp_s1`, and so on.
- In the **Heat Transfer in Porous Media** feature, `ht.c_pp` has become `ht.cp_p`, `ht.c_pp1` has become `ht.cp_p1`, and so on.
- In the **Immobile Fluids** feature, `ht.c_pg1` has become `ht.cp_g1`, and so on.
- In the **Fracture** feature, `ht.c_pfr` has become `ht.cp_fr`.

The names of thermal resistance and absolute thermal resistance variables have changed in the **Isothermal Domain Interface** feature:

- `ht.idi1.Rs` has become `ht.idi1.R_t`
- `ht.idi1.Rsth` has become `ht.idi1.R_tth`

The name of the fluid viscosity variable has changed in the **Local Thermal Non-Equilibrium** multiphysics coupling: `1tne1.mu_f` has become `1tne1.mu`.

### *Backward Compatibility with Version 5.0*

---

#### **OBSOLETE FEATURES**

A number of features are now obsolete because they have been merged or replaced by a more general one. However, when models from COMSOL 5.0 or before contain these features, some of the obsolete features are visible in version 5.1 with a warning sign notifying that the feature is now obsolete (see [Table 1-1](#)). Some features have been completely removed (see [Table 1-2](#)). A model containing any of them can be opened in 5.1, but the feature will be missing.

Updating the models is recommended. Use the new feature and remove the obsolete one to make sure that the model will remain compatible with future versions of COMSOL Multiphysics.



The table below summarizes the changes:

TABLE I-1: OBSOLETE FEATURES THAT ARE OBSOLETE BUT THAN CAN BE LOADED IN 5.1

OBSOLETE FEATURE	REPLACE IN 5.1 BY
<b>Convective Heat Flux</b>	<b>Heat Flux</b> with the <b>Convective heat flux</b> option selected
<b>Surface-to-Ambient Radiation</b>	<b>Diffuse Surface</b> with <b>Include surface-to-surface radiation</b> deselected
<b>Surface-to-Surface Radiation</b>	<b>Diffuse Surface</b> with <b>Include surface-to-surface radiation</b> selected
<b>Highly Conductive Layer</b>	<b>Thin Layer</b> with <b>Layer type</b> set to <b>Conductive</b>
<b>Thin Thermally Resistive Layer</b>	<b>Thin Layer</b> with <b>Layer type</b> set to <b>Resistive</b>

TABLE I-2: OBSOLETE FEATURES THAT ARE NO LONGER AVAILABLE

OBSOLETE FEATURE	REPLACE IN 5.1 BY
<b>Infinite Element</b>	<b>Infinite Element</b> is no longer defined in the physic interface. Infinite elements are defined in the definitions section instead.
<b>Electrochemical Heat flux</b>	<b>Boundary Heat Source</b>
<b>Reaction Heat Flux</b>	<b>Boundary Heat Source</b>
<b>Heat Transfer in Participating Media</b>	<b>Heat Transfer in Solids</b> contributing to <b>Radiation in Participating Media</b>
<b>Porous Matrix</b>	<b>Heat Transfer in Porous Media</b>

Note that it is not possible to add any of the obsolete features from the COMSOL Desktop in version 5.1; only the new features are available.

#### OBSOLETE POSTPROCESSING VARIABLES DELETED IN 5.1

List of deleted postprocessing variables: `ndflux_acc`, `ndflux_acc_u`, `ndflux_acc_d`, `ntflux_acc`, `ntflux_acc_u`, `ntflux_acc_d`, `nteflux_acc`, `nteflux_acc_u`, `nteflux_acc_d`, `qout`, `qout_u`, `qout_d`, `qin`, `qin_u`, `qin_d`, `qw`, `qw_u`, and `qw_d`.

#### CHANGES IN VARIABLE NAMES

The variable `ht.Tvar` used for non-isothermal flow wall functions becomes a fully scoped variable in the following features:

- **Heat Flux** (`ht.hf1.Tvar`), **Line Heat flux**, **Point Heat flux**, and **Out of plane heat flux**.
- **Heat Source** (`ht.hs1.Tvar`), **Boundary heat source**, **Line heat source**, and **Point heat source**.
- **Diffuse Surface** (`ht.ds1.Tvar`).

The fully scoped variables  $P_{tot}$ ,  $P_{btot}$ ,  $P_{ltot}$ , and  $P_{stot}$  become  $P_0$ ,  $P_b$ ,  $P_l$ , and  $P_s$ , respectively, in the following features:

- **Heat Flux**
- **Inflow Heat Flux**
- **Line Heat Flux**
- **Heat Source**
- **Boundary Heat Source**
- **Line Heat Source**
- **Line Heat Source (Thin Rod subfeature)**
- **Line Heat Source on Axis**
- **Layer Heat Source (Thin Layer subfeature)**

In the **Thermal Contact** feature:

- $Q_{fric}$ ,  $Q_{fricPow}$ , and  $Q_{rate}$  become  $Q_b$ ,  $P_b$ , and  $Q_{bpart}$ , respectively
- $Q_{fric0}$ ,  $Q_{rate0}$ ,  $E_{contact0}$ ,  $h_{rad0}$ ,  $h_{gap0}$ , and  $h_{constr0}$  are removed and migrated to existing variables  $Q_b$ ,  $Q_{bpart}$ ,  $E_{contact}$ ,  $h_{rad}$ ,  $h_{gap}$ , and  $h_{constr}$ , respectively, in old models.

The **Change Effective Thickness** feature from the **Heat Transfer in Thin Shells** physics interface merges  $htsh.de$  into the existing variable  $htsh.ds$ .

The **Heat Transfer in Thin Shells** physics interface also merges the property  $de$  and variable  $htsh.de$  into existing property  $ds$  and variable  $htsh.ds$ . An API command such as `model.physics("htsh").prop("de").set("de", [...])` now does not have any effect.

### *Backward Compatibility with Version 4.4*

---

#### **OUT-OF-PLANE HEAT TRANSFER UPDATE**

The Out-of-plane heat transfer property, which was available in 2D and 1D geometries, has been removed as of COMSOL Multiphysics 5.0. Now all geometries are considered 3D geometries and thickness or area parameters are always available.

#### *API Users*

For the following features:

- **Heat Source**
- **Heat Flux**

- **Boundary Heat Source**
- **Layer Heat Source**
- **Inflow Heat Flux**

With the **Overall heat transfer rate** option (previously named **Total power** or **Total heat flux**), the thickness parameters (`dz_entr` and `Ac_input`) have been removed.

Compatibility with previous code is maintained for most of the cases. However, three configurations require manual updates of the code:

- If you specified the thickness parameter twice (there were two instances of specifying the parameter).
- If you specified first the thickness and in a second command the total power value.
- If you first specified the thickness and then activated the out-of-plane heat transfer option.

For these 3 cases you need to rewrite this part of the Java code by doing the following.

For a heat source example (2D and 1D axisymmetric geometries):

```
Ptot_50 = Ptot_44_value*ht.d/dz_entr_44_value,
```

where `Ptot_44_value` is the total power value used in previous versions, `Ptot_50` is the total power value used since COMSOL Multiphysics 5.0, and `dz_entr_44_value` is the value used in previous versions. In 1D geometries, the following update should be performed: `Ptot_50 = Ptot_44_value*ht.d/Ac_input_44_value`, where `Ac_input_44_value` is the value used in previous versions.

#### *Flux Variables*

The flux variables are always given in  $W/m^2$  (also for 2D or 1D geometries). To obtain the corresponding flux variables in  $W/m$  on a boundary in 2D for example, the flux variables have to be multiplied by the thickness `ht.d`.

The following variables have been updated following this convention:

- Conductive heat flux, `ht.dflux`
- Convective heat flux, `ht.cflux`
- Translational heat flux, `ht.tr1flux`
- Total energy flux, `ht.teflux`
- Total heat flux, `ht.tflux`
- Boundary convective heat flux, `ht.chflux`
- Radiative heat flux, `ht.rflux`

All flux variables now have the following sign convention: positive heat flux corresponds to heating, whereas negative heat flux corresponds to cooling. The variables `chflux` and `rflux` may have opposite signs compared to previous versions depending on the context where they are defined.

#### **INFLOW HEAT FLUX IS ONLY APPLICABLE ON NON-SOLID DOMAINS**

The **Inflow Heat Flux** feature is no longer applicable on boundaries adjacent to the following domain features: **Heat Transfer in Solids**, **Biological Tissue**, and **Isothermal Domain**. This restriction avoids ill-posed configurations due to the null velocity field in these features. This may change model results if this boundary condition was previously applied to a boundary adjacent to a non-fluid domain.

#### **HEAT TRANSFER WITH PHASE CHANGE NOW USES A VOLUMETRIC FORMULATION**

The variables `ht.thetai` now represent volume fractions (they were mass fractions before). This implies changes in the definition of density and heat capacity at constant pressure close to the phase change interface. Far from the phase change interface the density and heat capacity are unchanged.

#### *Backward Compatibility with Version 4.3b*

---

#### **INTERIOR FLUXES UPDATED**

Because the interior flux variable definitions have been revised, models that use interior fluxes in definitions, physics interfaces, or postprocessing show different results in versions 5.0 and later 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.

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

Old models using the Porous feature will be supported in the COMSOL Multiphysics interface. However, it is recommended to replace this feature with Heat Transfer in Porous Media (see [Obsolete Features and Variables](#)). For a model using the API, it will be necessary 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 5.0 or a later version.

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 fewer 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 5.0 or later version. 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 the  $S2$  quadrature. However, the theoretical error is not different and the 2D and 3D implementations are now consistent.

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

Heat transfer physics 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 a 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 MPH-files created in version 4.3 are converted to appear and behave the same way as in 5.0 or a later version.

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 versions 5.0 and later compared to version 4.3, and the numerical result can differ slightly between the versions.

## Obsolete Features and Variables

---

The following features have been removed:

- **Porous Matrix** (replaced by **Heat Transfer in Porous Media**)
- **Infinite Element** (replaced by **Infinite Element Domain** declared in the **Definitions** node)
- **Electrochemical Heat Flux** (replaced by **Boundary Heat Source**)
- **Reaction Heat Flux** (replaced by **Boundary Heat Source**)
- **Heat Transfer in Participating Media** (replaced by **Heat Transfer with Radiation in Participating Media**)

When models created in older versions are loaded in version 5.2, these features are removed from the model tree and the model can still be opened. The new corresponding feature can be added to replace the removed one. In the particular case of a feature defined as a default feature of an interface in the model, the model cannot be opened.

The table below contains a list of variables that have been removed and replaced by new corresponding expressions.

TABLE 1-3: REMOVED VARIABLES AND CORRESPONDING EXPRESSIONS

REMOVED VARIABLE	CORRESPONDING EXPRESSION
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 5.2*

---

### **MATERIAL MODELS FROM EXTERNALLY PROGRAMMED LIBRARIES**

A new way to specify user-defined material models has been added. You can now access external material functions, written in C or some other programming language, which have been compiled into a shared library. This makes it possible to program your own material models and also to distribute such models as an add-on.

The external library is accessed through an **External Material** feature under **Global Definitions>Materials**. The external library can either completely define the stress-strain relation or only return an inelastic strain contribution to available material models. Two new features are available in the Solid Mechanics interface to accommodate this: the **External Stress-Strain Relation** material model, and the **External Strain** subnode under the **Linear Elastic Material**.

## *New Application in Version 5.2*

---

### **MEMS PRESSURE SENSOR DRIFT DUE TO HYGROSCOPIC SWELLING**

For protection, electronic devices are often overmolded with an epoxy mold compound (EMC). Due to hygroscopic swelling, the epoxy might endure dimensional changes that can affect the performance of the device. In this application, a parameterized pressure sensor can be analyzed by changing a number of geometrical parameters, as well as the material properties. The app results show the stability of the sensor readings due to moisture diffusion in the epoxy. The Transport of Diluted Species, Solid Mechanics, and Shell interfaces are combined in the application.

## *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 5.2*

---

### **THREE-PHASE FLOW, PHASE FIELD INTERFACE**

The new *Three-Phase Flow, Phase Field* interface can be used to model the flow and interaction of three different, immiscible fluids when the exact positions of the interfaces separating the fluids are of interest, also known as separated flow with surface tracking. The fluid-fluid interfaces are tracked using a ternary phase field formulation that accounts for differences in the fluids' densities and viscosities, and the formulation includes the effects of surface tension. The phase-field method can handle moving contact lines on no-slip boundaries. The movement of the fluid-fluid interfaces is determined by minimization of free energy.

### **TERNARY PHASE FIELD**

The corresponding *Ternary Phase Field* interface, used to track moving interfaces between three immiscible phases in the CFD Module and the Microfluidics Module, is also found as a standalone Mathematics interface.

## *New Application in Version 5.2*

---

### **INKJET**

The Inkjet application demonstrates how to model the flow of a droplet produced by an inkjet nozzle, such as one in a printer. An ink droplet is ejected from the nozzle and travels through air until it hits the target. The fluid flow is incompressible, and the movement of the interface between the droplet and the ambient air is modeled using the level-set method. The user can specify liquid properties in designated input fields, and the geometry—including nozzle, reservoir, compartment, and outlet—is fully parameterized. In addition, the user can control the injection pulse profile by varying the pulse interval and smoothing time.

## *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.

**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  $-Q_{br} \cdot \mathbf{u} / \epsilon_p^2$  on the right hand side previously lacked the factor  $1/\epsilon_p^2$ , where  $\epsilon_p$  is the porosity.

**MOBILITY**

In physics interfaces modeling migration (for example, 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 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

## *New Functionality in Version 5.2*

---

### **ROTATING MACHINERY FLUID FLOW INTERFACES: TURBULENT FLOW, ALGEBRAIC YPLUS AND TURBULENT FLOW, L-VEL**

Two new physics interfaces have been added in the Rotating Machinery Fluid Flow branch: the *Turbulent Flow, Algebraic  $\gamma$ Plus* interface and the *Turbulent Flow, L-VEL* interface. For these algebraic turbulence models, the turbulent viscosity is determined using two different extensions of the logarithmic wall law, and the local Reynolds number based on the distance to the nearest wall. Algebraic turbulence models are computationally cheaper and more robust but generally less accurate than transport equation turbulence models, such as the  $k$ - $\varepsilon$  and  $k$ - $\omega$  turbulence models.

## *New Application in Version 5.2*

---

### **MIXER**

This application is intended for the simulation of mixers equipped with axial or radial impellers. With a given geometry, the application computes the efficiency of the mixing process, where the measure of efficiency is the eddy diffusivity, from which the time scale of mixing of chemical species can be estimated. The app user can select eleven different types of impellers, each with its own distinctive combination of features, such as blade cuts and folds. Three different types of vessel are available: dish bottom, flat bottom, and cone bottom vessels, and the vessels can also be equipped with baffles.

# Molecular Flow Module

## *New Application in Version 5.2*

---

### **CHARGE EXCHANGE CELL SIMULATOR**

A charge exchange cell consists of a region of gas at an elevated pressure within a vacuum chamber. When an ion beam interacts with the higher-density gas, the ions undergo charge exchange reactions with the gas, creating energetic neutral particles. It is likely that only a fraction of the beam ions will undergo charge exchange reactions, so in order to neutralize the beam, a pair of charged deflecting plates are positioned outside the cell. In this way, an energetic neutral source can be produced.

This application simulates the interaction of a proton beam with a charge exchange cell containing neutral argon. User inputs include several geometric parameters for the gas cell and vacuum chamber, the beam properties, and the properties of the charged plates that are used to deflect the remaining ions. The app computes the efficiency of the charge exchange cell, measured as the fraction of ions that are neutralized, and records statistics about the different types of collisions that occur.

## *Backward Compatibility with Version 5.0 and Earlier*

---

Due to the fact that multiple species are now supported, some variable names have changed. In general, variables will now have `_G` appended to their names, where `G` is the name specified for the **Incident molecular fluxes** in the **Dependent Variables** section. So, if a model is updated and resolved, the expressions used in results should be changed as follows:

- The number density, `fmf.N` should be changed to `fmf.N_G`.
- The pressure should be changed from `p` to `fmf.p_G`.
- The reconstructed number density, `fmf.N` should be changed to `fmf.ndr1.Nr_G`, where `ndr1` is the tag of the **Number Density Reconstruction** feature.
- Any other variables can be updated using the **Replace Expression** button available during results processing.

In addition, new variables exist for the total pressure, number density, and so on:

- The total number density, `fmf.ntot`.

- The total pressure  $f_{mf} \cdot p_{tot}$ .
- The total incident molecular flow,  $f_{mf} \cdot G_{tot}$ .

# Multibody Dynamics Module

## *New Functionality in Version 5.2*

---

### **IMPROVED SOLVER SETTINGS FOR FRICTION**

The default solver settings when **Friction** is used in the Multibody Dynamics interface have been significantly improved.

### **PRINCIPAL MOMENTS OF INERTIA FOR RIGID DOMAINS**

The principal moments of inertia for a Rigid Domain feature are now available as variables that can be directly displayed during postprocessing.

### **RIGID DOMAIN RESULTS IN FREQUENCY DOMAIN**

All results in a Rigid Domain, including phase information, are now available in a Frequency Domain study.

## *New Applications in Version 5.2*

---

### **RIGID BODY SYSTEM SIMULATOR**

In some cases, it can be more convenient to build a model using input data structured in tables than by working directly in the Model Builder tree. In this app, a multibody dynamics simulation of an assembly of rigid bodies can be set up by pure tabular input; the tables can even be prepared off-line and read directly into the app either individually or in the form of an input file. In this way, data is given for the geometry in terms of nodes and their connectivities, and the rigid bodies in terms of mass properties, joint properties, loads, and constraints. Transient as well as stationary parametric analyses can be performed to obtain configuration plots, trajectory plots, joint motions, and joint forces.

### **CENTRIFUGAL GOVERNOR SIMULATOR**

A centrifugal governor is a classical angular speed regulation device. In this app, a parameterized centrifugal governor can be analyzed with respect to steady state, transient response, and natural frequencies.

# Nonlinear Structural Materials Module

## *New Functionality in Version 5.2*

---

### **IMPROVED FORMULATION FOR SMALL STRAIN PLASTICITY WITH GEOMETRIC NONLINEARITY**

When **Small plastic strains** is selected as the plasticity model in the **Plasticity** node, and the **Include geometric nonlinearity** check box is selected in the settings for the study, the Cauchy stress tensor is now used to evaluate the yield function and plastic potential. In previous versions the second Piola-Kirchhoff stress tensor was used instead. This change will give significant improvements in accuracy for moderate to large strains. As a result, you might experience changes in stresses and plastic strains for old models having significant plastic strains when this combination of settings is used.

## *New Applications in Version 5.2*

---

### **STRESS ANALYSIS OF A PRESSURE VESSEL**

In this app, a pressure vessel is analyzed in order check the allowed internal pressure. Some yielding can be allowed, and the plasticity model can be orthotropic to accommodate for effects introduced by plate rolling. Inputs include the main cylinder dimensions of the vessel, the geometry of the head, computed automatically for a chosen standard head type, design parameters, and material properties. The results give the analytic yielded volume fraction and the graphical stress, plastic strain, yielded volume and volume fraction, and local directions.



# Optimization Module

## *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 5.2*

---

## **RELEASE FROM EDGES AND POINTS**

The Release from Edge and Release from Point nodes can be used to release particles from edges and points in a geometry, respectively. When releasing particles along an edge, the particle positions can be mesh-based, weighted by a user-defined density function, or uniformly distributed along the edge length.

## **IMPROVED DENSITY-BASED RELEASE**

The release features that initialize particle position according to a density function have new settings that can make them much more accurate. It is now possible to specify a **Release distribution accuracy order** and **Position refinement factor** in the settings for the Release, Inlet, and Particle Beam nodes as well as the new Release from Edge node. The improvement in accuracy is most noticeable when the underlying mesh is very coarse or the particle density varies significantly across different mesh elements.

## **CHARGE EXCHANGE COLLISIONS**

Two new collision types can now be added to the Collisions node: Resonant Charge Exchange and Nonresonant Charge Exchange.

The Resonant Charge Exchange node is used when energetic ions undergo charge exchange reactions with ambient neutral atoms of the same element or molecules of the same substance. The Nonresonant Charge Exchange feature is used when the ionized and neutral species are of different elements or substances. In both cases, after the collision, it is possible to continue tracking the ionized species, the neutral species, or both.

## **PARTICLE BEAM IMPROVEMENTS**

New options are available in the Particle Beam feature to make the transverse position and velocity distributions easier to specify. This makes it much easier to release beams with phase space ellipses of a certain size, shape, and orientation. The Equation Display has been improved and augmented with images to provide a better indication of what the various options do.

## **PARTICLE COUNTERS**

A Particle Counter feature is a domain or boundary feature that provides information about particles arriving on a set of selected domains or surfaces from a release feature. Such quantities include the number of particles transmitted, the transmission probability, transmitted current, mass flow rate, and so on. This feature provides convenient results expressions that can be used in the Filters node of the Particle Trajectories plot, which allows only the particles that reach the particle counter selection to be visualized.

## **PARTICLE-MATTER INTERACTIONS**

You can now model the interaction of energetic ions with solid matter using the dedicated Particle-Matter Interactions feature. This feature supports two subfeatures for different types of interactions:

- *Ionization loss* is used to model the continuous loss of energy as ions interact with electrons in the target material.
- *Nuclear stopping* is used to model the deflection of energetic ions by target nuclei.

## **MAGNETIC FORCE USING EARTH'S MAGNETIC FIELD**

The Magnetic Force feature now has a new option to use data for the Earth's magnetic field. It is possible to specify locations in the world either based on particle coordinates, entering latitude and longitude, or by selecting one of several built-in city locations.

## **BUILT-IN DOMAIN ACCUMULATORS**

The Space Charge Density Calculation feature is now available for the Charged Particle Tracing interface. This feature computes the space charge density of particles in a domain. It is the single-physics equivalent of the Electric Particle-Field Interaction Multiphysics node. Similarly, a Volume Force Calculation feature is now available for the Particle Tracing for Fluid Flow interface.

## **SUPPORT FOR DIELECTROPHORESIS WITH NESTED SHELLS**

It is now possible to add a Dielectrophoretic Force to particles with multiple thin shells. To do so, add two or more Shell subnodes to the Dielectrophoretic Force node.

#### **CHARGE EXCHANGE CELL SIMULATOR**

A charge exchange cell consists of a region of gas at an elevated pressure within a vacuum chamber. When an ion beam interacts with the higher-density gas, the ions undergo charge exchange reactions with the gas, creating energetic neutral particles. It is likely that only a fraction of the beam ions will undergo charge exchange reactions, so in order to neutralize the beam a pair of charged deflecting plates are positioned outside the cell. In this way, an energetic neutral source can be produced.

This application simulates the interaction of a proton beam with a charge exchange cell containing neutral Argon. User inputs include several geometric parameters for the gas cell and vacuum chamber, the beam properties, and the properties of the charged plates that are used to deflect the remaining ions. The app computes the efficiency of the charge exchange cell, measured as the fraction of ions that are neutralized, and records statistics about the different types of collisions that occur.

#### **LAMINAR STATIC PARTICLE MIXER DESIGNER**

In static mixers, a fluid is pumped through a pipe containing stationary mixing blades. This mixing technique is well suited for laminar flow mixing because it generates only small pressure losses in this flow regime. However, the performance of the mixer can vary greatly depending on its geometry.

This application computes the fluid velocity and pressure field in a static mixer and then the trajectories of particles that are carried by the fluid. Because the particles have mass, they do not follow the fluid velocity streamlines exactly, causing some particles to hit the mixing blades. This app computes the transmission probability of particles in the mixer. It also evaluates the index of dispersion, which is a measurement of the uniformity with which different species of particle are mixed together.

#### **ION RANGE BENCHMARK**

This simple benchmark model confirms that when modeling a combination of ionization losses and nuclear scattering, the computed average range of protons in silicon agrees with experimental data. The average range is computed for initial proton energies from 1 keV to 100 MeV.

#### **SENSITIVE HIGH-RESOLUTION ION MICROPROBE (SHRIMP)**

The SHRIMP can be used to transmit ions of a given initial energy and specified charge-to-mass ratio by subjecting an incoming beam to appropriately tuned electric

and magnetic forces. The beam is first sent through a curved sector with a radial electric force, then through a second curved sector with a uniform magnetic flux density. The Particle Counter feature is used to visualize the transmitted beam and to compute the transmission probability of the ions.

### *Backward Compatibility with Version 5.2*

---

#### **RENAMED PHYSICS INTERFACE SETTINGS**

The **Release type** list in the settings windows for the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces in earlier versions has been renamed **Particle release specification**. For the Charged Particle Tracing interface the options **Static** and **Transient** have been renamed to **Specify current** and **Specify release times** (the default), respectively. For the Particle Tracing for Fluid Flow interface the options **Static** and **Transient** have been renamed to **Specify mass flow rate** and **Specify release times** (the default), respectively.

### *Backward Compatibility with Version 5.0 and Earlier*

---

#### **OLD NODES FOR PARTICLE-FIELD AND FLUID-PARTICLE INTERACTIONS ARE OBSOLETE**

The old **Particle-Field Interaction** and **Fluid-Particle Interaction** nodes, which could be added directly to the Charged Particle Tracing and Particle Tracing for Fluid Flow interfaces, respectively, are now obsolete. They will be removed in a future version. Simulations of particle-field and fluid-particle interactions should instead use the dedicated Multiphysics nodes.

#### **ELASTIC COLLISION FORCE NODE HAS BEEN REPLACED**

If a model created in Version 5.0 or earlier containing the **Elastic Collision Force** node is opened in Version 5.1, this node will be replaced. If the **Collision model** in the old **Elastic Collision Force** node was set to **Monte Carlo**, a **Collisions** node with the **Elastic** subnode will be created. If instead the **Collision model** was set to **Friction**, the **Friction Force** subnode will be added.

# Pipe Flow Module

## *New Functionality in Version 5.2*

---

### **REVISED FORMULATION OF THE PIPE FLOW INTERFACE**

The Pipe Flow interface has been amended with an inertial term. This means that gradually diverging or converging pipe sections will account for a pressure change due to the velocity change (Bernoulli effect). Also, sudden expansions and contractions will account for the Bernoulli effect properly. The stabilization and element shape functions have also been changed to handle compressible flows for  $Ma < 0.3$  properly. Models created in previous versions will be opened using the old interface with a notification. For new models, only the new interface is available in the physics interface list.

## *New Applications in Version 5.2*

---

### **GROUND HEAT RECOVERY APPLICATION**

The performance of a ground heat exchanger is analyzed. The app covers different designs and specifications of the heat exchanger at different ambient conditions. It calculates the temperature distribution, heat production, and heater state.

### **ORGAN PIPE DESIGN**

Working with acoustics, designing instruments or other sound transducing devices it can be interesting to not only look at a simulated frequency response but also listen to the sound generated by the device. This is auralization of the acoustics of the device. This demonstration application shows how this can be achieved using a simple model of an organ pipe. Changing the organ pipe design will, for example, result in a different pitch, which can be auralized. The organ pipe is modeled using the Pipe Acoustics, Frequency Domain interface. The application allows users to analyze how the first fundamental resonance frequency varies with pipe properties such as radius and wall thickness, and with the ambient pressure and temperature. The full frequency response is found including the fundamental frequency and the harmonics. Using a Java method, the results analysis is extended beyond the built-in functionality. The application will detect and find the location and amplitude of all harmonics in the response. It will also allow users to listen to the sound and pitch of a given organ design.

# Plasma Module

## *New Functionality in Version 5.2*

---

### **IMPROVED STABILITY OF THE FINITE VOLUME DISCRETIZATION**

The finite volume discretization has been improved so that numerical overflow errors no longer occur. This means that nonconvergence of models using the finite volume discretization should occur less often.

### **REDUCED ELECTRIC FIELDS STUDY NOW SUPPORTS UNITS**

The Reduced Electric Fields study now interprets units. The SI unit for the reduced electric field is  $V \cdot m^2$ , but it is possible to enter a value in Townsend, for example, 10[Td].

### **IMPROVED MATERIAL DATA FOR EQUILIBRIUM DISCHARGES**

Some outlying data points in the Equilibrium Discharge material data have been removed. This potentially makes models more stable when solving.

## *Backward Compatibility with Version 5.1*

---

Models using the Reduced Electric Fields study will interpret the unit in the edit field for the Reduced electric fields as Townsend. For example, when opening a model created in version 5.1, if the value for the Reduced electric fields was 10, this will now appear as 10[Td] in the study settings.

## *Backward Compatibility with Version 4.4*

---

Old models will be updated to use the new boundary conditions, but re-solving a model still gives the same results as before. The following changes to the model are made upon opening:

- If an Electric Potential feature is applied to boundaries adjacent to a Plasma Model feature, it is converted to a Metal Contact feature. The settings from the original model are retained. If an Electric Potential feature is applied to boundaries adjacent to a Charge Conservation feature, it remains the same.
- If an Electric Potential feature is applied to a combination of boundaries adjacent to a Plasma Model and Charge Conservation feature, a Metal Contact feature is

created for the boundaries adjacent to the Plasma Model feature, and the Electric Potential feature remains on boundaries adjacent to the Charge Conservation feature.

- Any External Surface Charge Accumulation boundary conditions are replaced by a new Dielectric Contact feature. The settings from the original model are retained. The External Surface Charge Accumulation boundary condition is now obsolete.



# Ray Optics Module

## *New Functionality in Version 5.2*

---

### **NEW PART: ASPHERIC LENS**

Aspheric lenses are often used as an alternative to spherical lenses to reduce monochromatic aberrations. An aspheric lens is now included in the Part Library for the Ray Optics Module. The depth or sag of the curved surface is defined using a standard aspheric formula in which the values of all coefficients are given as input parameters.

### **RELEASE RAYS FROM EDGES AND POINTS**

It is now possible to release rays from edges in 3D using the new Release from Edge feature, and to release rays from selected points using the new Release from Point feature. When releasing rays from edges, you can select any set of edges and release rays uniformly, based on the mesh, or with initial positions weighted according to a user-defined density function.

### **NEW OPTIONS FOR DENSITY-BASED RELEASE OF RAYS**

The release features that initialize ray position according to a density function have new settings that make them more accurate. You can now specify a Release distribution accuracy order and Position refinement factor in the settings for the Release, Inlet, and Illuminated Surface nodes as well as the new Release from Edge node. The improvement in accuracy is most noticeable when the underlying mesh is very coarse or the density expression varies significantly across different mesh elements.

### **DEPOSITED RAY POWER PERFORMANCE IMPROVEMENTS**

The Deposited Ray Power node on domains now uses the latest domain Accumulator technology, making it much faster and more accurate compared to older versions.

## *Backward Compatibility with Version 5.0*

---

### **RENAMED SETTINGS**

The Wall condition **Bounce** has been renamed to **Specular reflection**.

## CHANGES TO FREQUENCY-DEPENDENT MATERIAL PROPERTIES

The **Frequency-dependent refractive indices** check box has been renamed to **Allow frequency distributions at release features**.

Expressions involving the ray frequency must always be included within the `noenv()` operator. Some expressions for user-defined refractive indices may fail to be evaluated properly in version 5.2 unless this operator is applied.

## THIN DIELECTRIC FILM SETTINGS

In version 5.0, if the ray intensity was computed, the option to add **Thin Dielectric Film** nodes to a **Material Discontinuity** node was always available. In version 5.2, the Thin Dielectric Film subnode can only be added if **Add layers to surface** or **Add layers to surface, repeating** is selected from the **Thin dielectric films on boundary** list in the **Material Discontinuity** settings window. If a model created in version 5.0 is opened in version 5.2, and any **Thin Dielectric Film** nodes have been added, then **Add layers to surface** is selected from the **Thin dielectric films on boundary** list instead of the default **None**.

## CHANGES TO THE CIRCULAR WAVE RETARDER

The **Retardance** of the **Circular Wave Retarder** now indicates the rotation angle of the polarization ellipse of a ray that passes through the device. In version 5.0, the retardance corresponded to half of this rotation angle. The behavior of the **Circular Wave Retarder** is now consistent with the documentation. If a model created in version 5.0 is opened in version 5.2, the expression for the **Retardance** of any **Circular Wave Retarder** node is multiplied by 2.

## OBSOLETE DEPOSITED RAY POWER NODE

The **Deposited Ray Power** node for absorbing domains is obsolete and will be removed in future versions. For simulations in which rays generate heat due to absorption, the **Ray Heat Source** Multiphysics node should be used instead.

# RF Module

## *New Functionality in Version 5.2*

---

### **ENHANCED PHYSICS-CONTROLLED MESH HANDLING LOSSY MEDIA**

The physics-controlled mesh takes lossy electric and magnetic media and automatically scales the size by skin depth on the lossy domain boundaries. When **Resolve wave in lossy media** is selected, the outer boundaries of lossy media domains are meshed with a maximum mesh element size, in free space, given by the minimum value of half a skin depth plus 1/5 of the vacuum wavelength.

### **MULTI-ELEMENT UNIFORM LUMPED PORT**

A new lumped port type is included in COMSOL Multiphysics 5.2, the Multi-element uniform port for multi-excitation or -termination of, for example, a coplanar waveguide port or a differential port. The direction of the field in each subelement of the port is defined by the **Uniform Element** subnodes, where the direction between uniform element terminals,  $a_h$ , defines the electric potential polarity.

### **SMITH PLOT, CONVENTIONAL WAY TO PRESENT MATCHING PROPERTIES**

When an S-parameter plot is generated by default, a Smith plot will be automatically generated as well.

### **BOUNDARY MODE, FREQUENCY-STATIONARY AND BOUNDARY MODE, FREQUENCY-TRANSIENT STUDY SEQUENCES**

There are new study sequences in the Model Wizard for the Wave Optics Module and the RF Module multiphysics interfaces called Laser Heating and Microwave Heating, respectively. The Boundary Mode, Frequency-Stationary study sequence adds a Boundary Mode Analysis study step and a Frequency-Stationary study step, whereas the Boundary Mode, Frequency-Transient study sequence adds a Boundary Mode Analysis study step and a Frequency-Transient study step. The Boundary Mode Analysis study step is used for solving the mode field for numeric ports in the electromagnetic interfaces. The Frequency-Stationary and Frequency-Transient study steps couple stationary and transient analyses for the Heat Transfer in Solids interface with a frequency domain analysis for the Wave Optics and RF interfaces.

## **INITIAL VALUE SETTINGS FOR THE TRANSIENT SCATTERING BOUNDARY CONDITION**

In the Settings window for the Scattering boundary condition for time-dependent simulations, there is a new **Initial Values for Incident Wave** section for setting the initial values of the magnetic vector potential for the incident wave. Note that the section is initially collapsed by default. When the incident wave is defined by an electric field, you can specify the initial value for the magnetic vector potential for the incident wave. When the incident wave is defined by a magnetic field, the user can specify the initial value for the time derivative of the magnetic potential, in addition to the initial value for the magnetic vector potential. The new settings enable the user to define the exact wave form for the solved magnetic vector potential.

### *New Applications in Version 5.2*

---

## **FREQUENCY SELECTIVE SURFACE SIMULATOR**

Frequency selective surfaces (FSS) are periodic structures that generate a bandpass or a bandstop frequency response. This application simulates a user-specified periodic structure chosen from the built-in unit cell types. It provides five unit cell types popularly used in FSS simulations along with two predefined polarizations in one fixed direction of propagation that has normal incidence on the FSS. The analysis includes the reflection and transmission spectra, the electric field norm on the top surface of the unit cell, and the dB-scaled electric field norm shown on a vertical cut plane in the unit cell domain.

## **SLOT-COUPLED MICROSTRIP PATCH ANTENNA ARRAY SYNTHESIZER**

This application simulates a single slot-coupled microstrip patch antenna fabricated on a multilayered low temperature co-fired ceramic (LTCC) substrate. Results include the far-field radiation pattern of the antenna array and its directivity. The far-field radiation pattern is approximated by multiplying the array factor and the single antenna radiation pattern to perform an efficient far-field analysis without simulating a complicated full array model. Phased antenna array prototypes for 5G mobile networks can be evaluated with the default input frequency, 30 GHz. This app includes two view types, a Compact View and a Wide View, depending on how much space the user has to operate the app.

## **COPLANAR WAVEGUIDE (CPW) BANDPASS FILTER**

Coplanar waveguide (CPW) bandpass filters can be designed using interdigital capacitors (IDCs) and short-circuited stub inductors (SSIs). Such a filter can readily be

implemented on a GaAs wafer. The filter in this model is compact in relation to its resonant frequency and provides a relatively high Q-factor compared to capacitively coupled microstrip line model designs.

### *Backward Compatibility with Version 4.3a*

---

The following models have been moved from the RF Module Application Library to the Wave Optics Module Application 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 5.2*

---

## **EQUATION DISPLAY REDESIGN**

The equation display has been completely redesigned to give a clearer indication of the equations and boundary conditions that are being applied. Auxiliary relations for the boundary conditions have been moved to their own relevant section.

# Structural Mechanics Module

*New Functionality in Version 5.2*

---

## **MATERIAL MODELS FROM EXTERNALLY PROGRAMMED LIBRARIES**

A new way to specify user-defined material models has been added. You can now access external material functions written in C or some other programming language which have been compiled into a shared library. This makes it possible to program your own material models, and also to distribute such models as add-ons.

The external library is accessed through an **External Material** feature under **Global Definitions>Materials**. The external library can either completely define the stress-strain relation, or only return an inelastic strain contribution to available material models. Two new features are available in the Solid Mechanics interface to accommodate this: the **External Stress-Strain Relation** material model, and the **External Strain** subnode under the **Linear Elastic Material**.

## **CONTACT WITH SMALL RELATIVE DISPLACEMENTS**

A new simplified method for computing the distances within contact pairs has been introduced. It can be used when there is little sliding between the contacting surfaces, like in a shrink fit or bolted joint. In this method, the mapping between source and destination is computed only once, a strategy which leads to faster and more stable convergence. To use this method, set **Mapping method** to **Initial configuration** in the settings for the contact pair.

## **ADJUSTMENT OF THE INITIAL CONTACT GAP**

Sometimes, the finite element discretization of curved boundaries causes the initial distance between the two boundaries in a contact pair to have noticeable irregularities. This can now be compensated for through a built-in computation of the initial gap. This initial gap can then be subtracted in the subsequent analysis if you select the **Force zero initial gap** check box in the settings for **Contact** in the Solid Mechanics interface.

## **FULL MASS MATRIX IN ADDED MASS**

The **Added Mass** feature has been improved so that it is possible to enter a full mass matrix.

### **OPTIONAL INCLUSION OF THERMOELASTIC DAMPING**

In the **Thermal Expansion** multiphysics coupling, it is now possible to select whether the time derivative of the stresses should act as a heat source in the heat transfer problem or not. In previous versions, this thermoelastic damping effect was always active.

### **PRINCIPAL MOMENTS OF INERTIA FOR RIGID DOMAINS**

The principal moments of inertia for a Rigid Domain are now available as variables which can be directly displayed during postprocessing.

### **RIGID DOMAIN RESULTS IN FREQUENCY DOMAIN STUDIES**

All results in a Rigid Domain, including phase information, are now available in Frequency Domain studies.

## *New Applications in Version 5.2*

---

### **BEAM SECTION CALCULATOR**

Using this app, which is built upon the Beam Cross Section interface, it is possible to evaluate cross section data for a wide range of US and European standard beams. The computed data can then be transferred for use as input data in a Beam interface. Given a set of forces and moments acting on the section, you can also compute a detailed stress distribution.

An additional app is included for users with LiveLink™ *for* Excel®<sup>®</sup>, which has somewhat more functionality, such as the possibility to extend it with other beam standards.

### **INTERFERENCE FIT**

In an interference fit, two parts are joined by employing the contact pressure formed when a slightly oversized part is fitted into a hole. The stress distribution, contact pressure, and transferable force and moment in an interference fit between two circular pipes are computed in this app. You can vary the dimensions of the two pipes and the coefficient of friction to test different designs.

### **MEMS PRESSURE SENSOR DRIFT DUE TO HYGROSCOPIC SWELLING**

For protection, electronic devices are often overmolded with an epoxy mold compound (EMC). Due to hygroscopic swelling, the epoxy might endure dimensional changes that can affect the performance of the device. In this app, a parameterized pressure sensor can be analyzed by changing a number of geometrical parameters, as well as the material properties. The app results show the stability of the sensor readings



due to moisture diffusion in the epoxy. The Transport of Diluted Species, Solid Mechanics, and Shell interfaces are combined in the application.

#### **TRUSS BRIDGE DESIGNER**

This is an example of an application for studying a civil engineering structure, a Pratt truss bridge. The underlying model consists of a Shell interface for the roadway and a Beam interface for the supporting structure. With this app, you can modify all dimensions of the bridge, including the number of spans and the cross sections of all members. Additionally, you can configure the load with an arbitrary number of vehicles and a wind load. In addition to the static results, the lowest natural frequencies are computed.

#### **LINEAR BUCKLING OF A TRUSS TOWER**

In this app, users can compute the buckling load for unstayed and guyed truss towers. First, a user enters the tower dimensions, consisting of individual (longitudinal, transverse, and diagonal) members. Then, choose between the unstayed or the guyed option, which calls for data for the wires, including prestress. Finally, for material data, select structural steel, aluminum, or a user-defined material. After running the app, there are graphical and analytical results for the critical load.

#### **VISCOELASTIC STRUCTURAL DAMPER**

The dynamics properties of a viscoelastic damper are studied in this app. You can modify the viscoelastic material's properties and the loads at the attachment points. The primary result is the force-displacement relation at the attachment points. Alternate results include hysteresis plots and storage and loss moduli. You can even compare two different materials side by side.

#### **INTERFERENCE FIT CONNECTION IN A MOUNTAIN BIKE FORK**

This is a tutorial example showing how to model interference fits. The stresses caused by a shrink fit between the steering tube and the crown in a bike fork are computed.

#### *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 this 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 mechanical 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*

---

- When using version 5.2, in the Solid Mechanics interface (SolidMechanics) the property `ControlsSpatialFrame` from version 4.3a no longer exists, and COMSOL Multiphysics 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 is set to the expression used in the model.

# Subsurface Flow Module

*Backward Compatibility with Version 4.4*

---

## **SOLUTE TRANSPORT**

The Solute Transport interface has been replaced by the new interface Transport of Diluted Species in Porous Media.

# Wave Optics Module

*New Functionality in Version 5.2*

---

## **ENHANCED PHYSICS-CONTROLLED MESH HANDLING OF LOSSY MEDIA**

The physics-controlled mesh takes lossy electric and magnetic media and provides a boundary layer mesh automatically scaled by the skin depth. When **Resolve wave in lossy media** is selected in the properties of the Electromagnetic Waves, Frequency Domain interface, the outer boundaries of a lossy media domain are processed with a boundary layer mesh approximately one skin depth thick. The two-layer mesh has a 1.2 stretching factor. As the first layer is half a skin depth thick, the second layer is 0.6 skin depth thick, making the total thickness of the boundary layer mesh 1.1 skin depths thick.

## **BOUNDARY MODE, FREQUENCY-STATIONARY AND BOUNDARY MODE, FREQUENCY-TRANSIENT STUDY SEQUENCES**

For the Laser Heating multiphysics interface there are new study sequences in the Model Wizard. The Boundary Mode, Frequency-Stationary study sequence adds a Boundary Mode Analysis study step and a Frequency-Stationary study step, whereas the Boundary Mode, Frequency-Transient study sequence adds a Boundary Mode Analysis study step and a Frequency-Transient study step. The Boundary Mode Analysis study step is used for solving for the mode field for numeric ports in the electromagnetic interfaces. The Frequency-Stationary and Frequency-Transient study steps couple a frequency-domain analysis for the electromagnetic interface with stationary and transient analyses, respectively, for the Heat Transfer in Solids interface.

## **INITIAL VALUE SETTINGS FOR THE TRANSIENT SCATTERING BOUNDARY CONDITION**

In the settings for the Scattering Boundary Condition for time-dependent simulations, there is a new section called **Initial Values for Incident Wave** for setting the initial values of the magnetic vector potential for the incident wave. This section is collapsed by default. When the incident wave is defined by an electric field, you can also specify the initial value for the magnetic vector potential for the incident wave. When the incident wave is defined by a magnetic field, you can also specify the initial value for the time derivative of the magnetic potential, in addition to the initial value for the magnetic vector potential.

## **NO SCATTERED FIELD SETTING FOR THE MATCHED BOUNDARY CONDITION**

To prevent spurious solutions to appear when using the Matched boundary condition with the Electromagnetic Waves, Beam Envelopes interface, a new setting has been introduced—the **No scattered field** check box. When this check box is selected, the scattered wave is constrained to be zero on the boundary that the feature is selected for. An example model using this new setting is the `brewster_interface` model.

### *New Application in Version 5.2*

---

## **FIBER SIMULATOR**

This application performs mode analysis on a concentric circular dielectric layer structure. Both step-index fibers and graded-index optical fibers can be simulated. The geometry and refractive index distribution, the mode fields, the mode-field diameter, the attenuation, the group delay, and the dispersion coefficient are visualized.

# Material Library

## *New Material Data in Version 5.2*

---

Data has been added for the following materials: Be<sub>3</sub>N<sub>2</sub>, Ca<sub>3</sub>N<sub>2</sub>, HfN, HfN, TiN, ZrN, HfC, NbC, Ta<sub>2</sub>O<sub>3</sub>, Sc<sub>2</sub>O<sub>3</sub>, Lu<sub>2</sub>O<sub>3</sub>, Tm<sub>2</sub>O<sub>3</sub>, Gd<sub>2</sub>O<sub>3</sub>, Sm<sub>2</sub>O<sub>3</sub>, CeO<sub>2</sub>, SrO, C<sub>5</sub>H<sub>12</sub>, Mo<sub>2</sub>C, WCIN-939, 26NiCrMoV14-5, 18Mn18Cr (ASTM A289), 2507 (S32750), S32053, ASTM B625 (N08031), UNS N08354, NAS NM15M, NAS NM17, Pyroceram 9606, Corning 1737, Corning Eagle 2000, Corning Eagle XG, Corning Lotus, Corning Lotus XT, and SiO<sub>2</sub>-TiO<sub>2</sub>.

# LiveLink for MATLAB

## *New Functionality in Version 5.2*

---

LiveLink™ for MATLAB® now supports MATLAB® R2015b.

The following sections list new and improved functionality in LiveLink™ for MATLAB®:

### **MPHPLOT FUNCTION**

The wrapper function `mphplot` has been enhanced in many ways to provide plots in MATLAB that look more like the plot created by COMSOL Multiphysics. This results in more accurate colors, line styles, symbols, and so on.

A new options 'normals' has been added to `mphplot`. It is 'on' by default and tells MATLAB if normals should be used when rendering surfaces in 3D. This is mostly noticeable when lighting is used or when displaying arrows.

A new option 'createplot' has been added. It is 'on' by default, but when set to 'off' it is possible to extract data from the model without actually creating a plot on the screen. The data can later be used to recreate the plot. This operation is mostly useful when working with servers that do not have a display but is also useful for users who wish to extract data from a plot.

The new plot types Octave Plots, Smith Plots, and Annotation Plots have been added.

### **MPHGETEXPRESSIONS FUNCTION**

The `mphgetexpressions` function now returns the value and unit of the evaluated expression using the model's default unit system.

### **MPHEVALGLOBALMATRIX FUNCTION**

New options 'outersolnum' and 'outerdataseries' have been added to the `mphevalglobalmatrix` function for improved support for models that use parametric sweeps.

### **MPHINPUTMATRIX FUNCTION**

The `mphinputmatrix` function now supports complex-valued matrices.

## **MPHXMESHINFO FUNCTION**

The `mphxmeshinfo` function now supports separate DOF coordinates for models that use extra dimensions.

### *Backward Compatibility with Version 5.0*

---

- The `mphint2` command: For data sets other than Solution, Particle, Cut Line, Cut Plane, Time Integral, Time Average, Surface, and Line, the integration order now corresponds to the element refinement. The default value is set to 4 instead of 1. This change can affect the solution in cases where the problem is not resolved well enough.
- The `mphapplicationlibraries` function replaces the `mphmodellibrary` function for showing the Application Libraries window.



# LiveLink for Excel

## *New Functionality in Version 5.2*

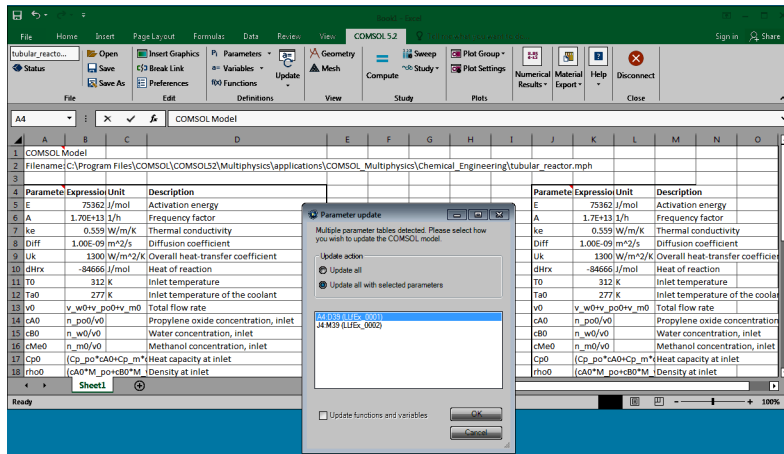
The following new functionality is now available in LiveLink™ for Excel®:

### **SUPPORTED VERSIONS**

Microsoft® Excel® 2016 has been added to the list of supported versions of Excel. Now Excel 2007, 2010, 2013, and 2016 are supported when using Windows. Loading and Saving of Excel files is supported on Linux and Mac OS X.

### **PARAMETER UPDATE DIALOG BOX**

A new dialog box for updating parameters and other settings in a model. This dialog box provides improved control of which parameters are updated when there are multiple sets of parameters in an Excel workbook.



# The COMSOL API for Use with Java<sup>®</sup>

## *COMSOL 5.1 API Changes*

---

### **ACOUSTICS MODULE CHANGES**

The ratio of specific heats is now a material parameter. If the default value was used in a model, add the following line to the Java file to obtain the old behavior:

```
model.physics("cpf").feature("cpf1").set("gamma_mat", "userdef");
```

### **ELECTROCHEMISTRY MODULES CHANGES**

There are changes for the Tertiary Current Distribution, Nernst-Planck interface. The Java API syntax for creating and accessing vectors and tensors has changed, as well as the syntax for setting physics properties. See the following examples:

- Specifying the diffusion tensor.

Old syntax:

```
model.physics("tcdee").feature("ice1").set("D_0", new  
String[]{"D1", "0", "0", "0", "D1", "0", "0", "0", "D1"});  
model.physics("tcdee").feature("ice1").set("D_1", new  
String[]{"D2", "0", "0", "0", "D2", "0", "0", "0", "D2"});  
model.physics("tcdee").feature("ice1").set("D_2", new  
String[]{"D3", "0", "0", "0", "D3", "0", "0", "0", "D3"});
```

New syntax (includes the species name in the first argument of the set method):

```
model.physics("tcdee").feature("ice1").set("D_c1", new  
String[]{"D1", "0", "0", "0", "D1", "0", "0", "0", "D1"});  
model.physics("tcdee").feature("ice1").set("D_c2", new  
String[]{"D2", "0", "0", "0", "D2", "0", "0", "0", "D2"});  
model.physics("tcdee").feature("ice1").set("D_c3", new  
String[]{"D3", "0", "0", "0", "D3", "0", "0", "0", "D3"});
```

- Accessing tensor components (in the definitions of other variables, for example).

Old syntax:

```
Dxx_c1, Dxy_c1, Dxz_c1...
```

New syntax (component index after the species name):

```
D_c1xx, D_c1xy, D_c1xz...
```

- Specifying the reaction rate vector.

Old syntax:

```

model.physics("tcdee").feature("react1").set("R_0", new
String[]{"R1"});
model.physics("tcdee").feature("react1").set("R_1", new
String[]{"R2"});
model.physics("tcdee").feature("react1").set("R_2", new
String[]{"R3"});

```

New syntax (includes the species name in the first argument of the set method):

```

model.physics("tcdee").feature("react1").set("R_c1", new
String[]{"R1"});
model.physics("tcdee").feature("react1").set("R_c2", new
String[]{"R2"});
model.physics("tcdee").feature("react1").set("R_c3", new
String[]{"R3"});

```

- Accessing vector components.

Old syntax:

```
tfluxx_c1, tfluxy_c1, tfluxz_c1
```

New syntax (component index after the species name):

```
tflux_c1x, tflux_c1y, tflux_c1z
```

- Specifying initial values:

Old syntax:

```

model.physics("tcdee").feature("init1").setIndex("c2", "c20", 0);
model.physics("tcdee").feature("init1").setIndex("c3", "c30", 0);
model.physics("tcdee").feature("init1").setIndex("V", "V0", 0);

```

New syntax:

```

model.physics("tcdee").feature("init1").setIndex("initc", "c20",
1);
model.physics("tcdee").feature("init1").setIndex("initc", "c30",
2);
model.physics("tcdee").feature("init1").setIndex("initphil",
"V0");

```

- Selecting properties.

Old syntax:

```

model.physics("tcdee").prop("Convection").set("Convection", 1,
"0");
model.physics("tcdee").prop("ConvectiveTerm").setIndex("Convectiv
eTerm", "noncons", 0);

```

New syntax (the property name corresponds to the section in the user interface):

```

model.physics("tcdee").prop("TransportMechanism").set("Convection
", 1, "0");

```

```
model.physics("tcdee").prop("AdvancedSettings").setIndex("ConvectiveTerm", "noncons", 0);
```

For information about API backward compatibility for the Surface Reactions interface, see the [Chemical Reaction Engineering Module](#) release notes.

#### ECAD IMPORT MODULE CHANGES

- The default value for the grouping property of the Import function is now set to layer. Previously the default was all.
- For ODB++® and ODB++(X) files (the type property is odb) new rules apply for the initialization of the importlayer property of the Import function. Now only layers of type *Metal* and *Dielectric* are initialized with the string on in the importlayer string array. Previously this also included layers of type *Drill*.

#### LIVELINK FOR SOLID EDGE CHANGES

The default value for the keepfree property of the LiveLinkSolidEdge function is now set to on. Previously the default was set to off.

#### LIVELINK FOR SOLIDWORKS CHANGES

The default value for the keepfree property of the LiveLinkSOLIDWORKS function is now set to on. Previously the default was set to off.

### COMSOL 5.0 API Changes

---

#### DOCUMENTATION

The *COMSOL Multiphysics Programming Reference Manual* replaces the *COMSOL API for use with Java® Reference Manual*.

#### DEPRECATED METHODS

The following methods are deprecated in COMSOL 5.0:

TABLE I-4: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
com.comsol.model.OptFeature.field()	OptFeature.comp()
com.comsol.model.OptFeature.field(String)	OptFeature.comp(String)
com.comsol.model.Group.identifier()	Group.paramName()
com.comsol.model.Material.identifier()	ModelEntity.tag()
com.comsol.model.MaterialModel.identifier()	ModelEntity.tag()

TABLE I-4: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
<code>com.comsol.model.ModelNode.identifier()</code>	<code>ModelEntity.tag()</code>
<code>com.comsol.model.physics.Physics.identifier()</code>	Use <code>tag()</code> instead.
<code>com.comsol.model.physics.MultiphysicsCoupling.identifier()</code>	<code>ModelEntity.tag()</code>
<code>com.comsol.model.Group.identifier(String)</code>	<code>Group.paramName(String)</code>
<code>com.comsol.model.Material.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.MaterialModel.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.ModelNode.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.physics.Physics.identifier(String)</code>	Use <code>tag(string)</code> instead.
<code>com.comsol.model.physics.MultiphysicsCoupling.identifier(String)</code>	<code>ModelEntity.tag(String)</code>
<code>com.comsol.model.ModelEntity.name()</code>	<code>ModelEntity.label()</code>
<code>com.comsol.model.ModelEntity.name(String)</code>	<code>ModelEntity.label(String)</code>
<code>com.comsol.model.ParameterEntity.set(String, int, double)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, double)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, double[])</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int[])</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, double)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, int)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, int, String)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.

TABLE I-4: DEPRECATED METHODS AND THEIR REPLACEMENTS

DEPRECATED METHOD	NEW METHOD
<code>com.comsol.model.ParameterEntity.set(String, int, String)</code>	Use the 0-based version ( <code>setIndex</code> ) instead.
<code>com.comsol.model.ParameterEntity.set(String, int, String[])</code>	Use the 0-based version ( <code>setIndex</code> ) instead.

## COMSOL 4.4 API Changes

### PHYSICS INTERFACES

- The default Initial Value features 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. When 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 5.0 and later. 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");
```

to

```
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 for Pressure Acoustics, Frequency Domain:

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

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 mechanical contact, the variable field name in the solver settings for the old contact variables is different than in earlier versions. For example:
  - In version 5.2: `mod1_solid_contact_old_p1`, `mod1_solid_cm_old_p1`.
  - In version earlier than 5.0: `mod1_solid_contact_p1_old`,  
`mod1_solid_cm_p1_old`.
- The parameter `form` in the property `EquationForm` can no longer be set to `ModeAnalysis` in 3D and 2D axisymmetric calculations for the Pressure Acoustics, Frequency Domain, and Pressure Acoustics, Transient interfaces. In these cases, the parameter `modeFreq` has been removed.

## *COMSOL 4.3b API Changes*

---

### **PHYSICS INTERFACES**

- The *Crosswind diffusion* formulation has been updated for all physics interfaces supporting Crosswind diffusion. Models solved with the new formulation can give different results than models solved in versions earlier than 4.3b. Java files can be modified to retain old crosswind formulations. Please contact COMSOL 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 axisymmetric models.
- 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 Multiphysics version 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_lm`, `model.v_lm`, and `model.w_lm`, 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 `<phitag>.locCFL` to the built-in variable `CFLCMP`, which in turn triggers a PID regulator via the automatic solver suggestions. Here, `<phitag>` is the physics interface tag. Java files where pseudo time stepping is active and the local CFL number is 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 deprecated 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 Multiphysics version 4.1 or earlier 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`, `eel`, `S1`, `si`, and `ei`) have new names. The coordinates of the local system (for example `x1`, `x2`, and `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 `Si1` and `ei1`.
- In the Darcy's Law interface and the Richards' Equation interface in the Subsurface Flow Module, fluid compressibility 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 fluid compressibility and the Biot-Willis coefficient are now material parameters and no

longer have default values. If the default values were 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 versions 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 default values. If the default values were used in a model file for Java, you now have to set them. 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 versions 4.0 and 4.0a:

- The units of the load face parameter (`Fph`) of the *Phase* feature in `AcousticStructure`, `TransientAcousticStructure`, `ThermallyInducedStresses`, `SolidMechanics`, `JouleHeatingThermalExpansion`, `Poroelasticity`, and `FluidStructureInteraction` have changed from degrees to radians.
- The `physics.field()` operator for `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 means 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 means that the index notation of vector variables has changed from uppercase `XYZ` to lowercase `xyz`.

This concludes the release notes for COMSOL Multiphysics version 5.2.

