



## **User's Manual Version 3.0**

**Last Updated: January 5, 2026**

**Contributors:**

- Dr. Steve Maas ([steve.maas@utah.edu](mailto:steve.maas@utah.edu))
- Dr. Gerard Ateshian ([ateshian@columbia.edu](mailto:ateshian@columbia.edu))
- Dr. Jeff Weiss ([jeff.weiss@utah.edu](mailto:jeff.weiss@utah.edu))
- Michael Herron ([michaelrossherron@gmail.com](mailto:michaelrossherron@gmail.com))

**Contact Address:**

Weiss Lab, University of Utah <https://mrl.sci.utah.edu>

Ateshian Lab, Columbia University <https://mbl.me.columbia.edu>

**Website**

FEBio: <http://febio.org>

**Forum**

<https://forums.febio.org/>

**Acknowledgments**

Development of the FEBio project is supported in part by a grant from the U.S. National Institutes of Health (R01GM083925).



# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Overview of FEBioStudio . . . . .	9
1.2	About this document . . . . .	9
<b>2</b>	<b>Getting Started</b>	<b>10</b>
2.1	Starting FEBio Studio . . . . .	10
2.2	Creating a New Model . . . . .	10
2.3	Opening a Model . . . . .	11
2.4	Exploring a Model . . . . .	11
2.4.1	Navigating the Graphics View . . . . .	11
2.4.2	Selecting objects . . . . .	13
2.4.3	Transforming an object . . . . .	13
2.5	Solving the Model . . . . .	13
2.6	Loading the Results . . . . .	14
<b>3</b>	<b>The FEBio Studio Environment</b>	<b>15</b>
3.1	The Graphical User Interface . . . . .	15
3.1.1	Overview . . . . .	15
3.1.2	Navigating the GUI . . . . .	15
3.2	The Menu Bar . . . . .	16
3.2.1	The File Menu . . . . .	17
3.2.2	The Edit Menu . . . . .	18
3.2.3	The Physics Menu . . . . .	18
3.2.4	The FEBio Menu . . . . .	19
3.2.5	The Record Menu . . . . .	20
3.2.6	The Tools Menu . . . . .	20
3.2.7	The View Menu . . . . .	21
3.2.8	The Help Menu . . . . .	22
3.3	The Main Tool Bar . . . . .	22
3.4	The Build Tool Bar . . . . .	23
3.5	The Font Toolbar . . . . .	24
3.6	The Graphics View . . . . .	24
3.7	The Graphics Toolbar . . . . .	25
3.8	The Model Viewer . . . . .	25
3.8.1	The Search Panel . . . . .	28
3.8.2	The Model Viewer Panes . . . . .	28
3.8.3	Editing Selections . . . . .	29

3.8.4 The Appearance Editor . . . . .	29
3.9 The Build Panel . . . . .	31
3.9.1 The Create Panel . . . . .	31
3.9.2 The Edit Panel . . . . .	32
3.9.3 The Mesh Panel . . . . .	32
3.9.4 The Tools Panel . . . . .	32
3.10 The Repository Panel . . . . .	35
3.10.1 Connecting to the Repository . . . . .	35
3.10.2 Browsing and Downloading . . . . .	36
3.10.3 Managing Your Uploads . . . . .	37
3.10.4 The Upload Permission Request Dialog . . . . .	38
3.10.5 The Upload/Modify Wizard . . . . .	38
3.10.5.1 Adding or Editing Project Details . . . . .	39
3.10.5.2 The Add Publication Dialog . . . . .	40
3.10.5.3 Adding or Removing Files . . . . .	41
3.11 The Project Panel . . . . .	42
3.12 The Curve Editor . . . . .	43
3.13 The Mesh Inspector . . . . .	46
3.14 The Measure Tool . . . . .	46
3.15 FEBio Studio Settings . . . . .	48
3.15.1 Background Settings . . . . .	49
3.15.2 Camera Settings . . . . .	50
3.15.3 Colormap Settings . . . . .	50
3.15.4 Display Settings . . . . .	50
3.15.5 FEBio Settings . . . . .	51
3.15.6 Lighting Settings . . . . .	51
3.15.7 Palette Settings . . . . .	52
3.15.8 Physics Settings . . . . .	52
3.15.9 Post Settings . . . . .	53
3.15.10 Selection Settings . . . . .	53
3.15.11 UI Settings . . . . .	53
3.15.12 Units Settings . . . . .	53
3.15.13 Model Repository Settings . . . . .	54
3.15.14 Auto Update Settings . . . . .	54
<b>4 Creating, Loading, and Saving Models</b> . . . . .	<b>55</b>
4.1 Starting a new model . . . . .	55
4.2 Loading a model . . . . .	55
4.3 Saving a model . . . . .	56
<b>5 Creating and Editing Geometry</b> . . . . .	<b>57</b>
5.1 Creating Geometry . . . . .	57
5.2 Importing Geometry . . . . .	57
5.3 Editing Geometry . . . . .	58
5.3.1 Editing Primitives . . . . .	58
5.3.2 Editing Editable Surfaces . . . . .	58
5.3.2.1 Auto Partition . . . . .	58
5.3.2.2 Decimate . . . . .	58

5.3.2.3	Edge Collapse . . . . .	59
5.3.2.4	Fix Mesh . . . . .	59
5.3.2.5	Flip Edges . . . . .	59
5.3.2.6	MMG Remesh . . . . .	59
5.3.2.7	Partition . . . . .	60
5.3.2.8	Project Curve . . . . .	60
5.3.2.9	Refine . . . . .	60
5.3.2.10	Smooth . . . . .	60
5.3.2.11	Weld Nodes . . . . .	60
5.4	Creating and Editing a Mesh . . . . .	61
5.4.1	Meshing Primitives . . . . .	61
5.4.2	Meshing CAD Geometry . . . . .	61
5.4.3	Meshing Editable Surfaces . . . . .	62
5.4.4	Editable Meshes . . . . .	62
5.4.4.1	Add Node . . . . .	63
5.4.4.2	Align . . . . .	63
5.4.4.3	Auto Partition . . . . .	63
5.4.4.4	Boundary Layer . . . . .	63
5.4.4.5	Convert Mesh . . . . .	64
5.4.4.6	Create shells from faces . . . . .	64
5.4.4.7	Detach Elements . . . . .	64
5.4.4.8	Discard Mesh . . . . .	64
5.4.4.9	Extrude Faces . . . . .	64
5.4.4.10	Fix Mesh . . . . .	64
5.4.4.11	Inflate . . . . .	64
5.4.4.12	Invert . . . . .	65
5.4.4.13	Mirror . . . . .	65
5.4.4.14	MMG Remesh . . . . .	65
5.4.4.15	Partition . . . . .	65
5.4.4.16	Rebuild Mesh . . . . .	66
5.4.4.17	Refine Mesh . . . . .	66
5.4.4.18	Revolve Faces . . . . .	66
5.4.4.19	Rezone . . . . .	66
5.4.4.20	Set Axis . . . . .	66
5.4.4.21	Set Axis from curvature . . . . .	67
5.4.4.22	Set Fibers . . . . .	67
5.4.4.23	Shell Thickness . . . . .	68
5.4.4.24	Smooth . . . . .	68
5.4.4.25	TetGen . . . . .	68
5.4.4.26	Weld Nodes . . . . .	68
6	Generating and Editing Mesh Data . . . . .	69
6.1	Adding Mesh Data . . . . .	69
6.2	FEBio Mesh Data Generators . . . . .	70
6.3	The Scalar Field Tool . . . . .	72
6.4	The Fiber Generator Tool . . . . .	73
6.5	Visualizing Mesh Data . . . . .	75

<b>CONTENTS</b>	<b>5</b>
<b>7 Materials</b>	<b>76</b>
7.1 Adding materials . . . . .	76
7.2 Setting material parameters . . . . .	76
7.3 Assigning materials . . . . .	76
7.4 Creating a Solute Table . . . . .	77
7.5 Creating a Solid-Bound Molecule Table . . . . .	77
7.6 Adding Chemical Reactions . . . . .	77
<b>8 Boundary Conditions and Loads</b>	<b>79</b>
8.1 Boundary Conditions . . . . .	79
8.2 Surface Loads . . . . .	80
8.3 Initial Conditions . . . . .	80
8.4 Assigning Boundary Conditions . . . . .	81
<b>9 Contact and Constraints</b>	<b>83</b>
9.1 Contact . . . . .	83
9.2 Nonlinear constraints . . . . .	84
9.2.1 Surface Constraints . . . . .	85
9.2.2 Body Constraints . . . . .	85
9.2.3 General Constraints . . . . .	85
<b>10 Rigid Bodies</b>	<b>86</b>
10.1 Rigid Body Constraints . . . . .	86
10.2 Rigid Body Initial Conditions . . . . .	86
10.3 Rigid Body Loads . . . . .	87
10.4 Rigid Connectors . . . . .	87
10.5 The Rigid Boundary Condition . . . . .	89
<b>11 Defining Analysis Steps</b>	<b>90</b>
11.1 The Initial Step . . . . .	90
11.2 Adding an Analysis Step . . . . .	90
<b>12 Configuring Output</b>	<b>92</b>
12.1 FEBio Logfile . . . . .	92
12.2 FEBio Plotfile . . . . .	93
<b>13 Running FEBio from FEBioStudio</b>	<b>95</b>
13.1 Running FEBio . . . . .	95
13.2 FEBio Launch Configurations . . . . .	97
13.3 The FEBio Monitor . . . . .	99
13.4 Running Batch Jobs . . . . .	102
13.4.1 Create a new batch job . . . . .	102
13.4.2 Open or Save a batch file . . . . .	102
13.4.3 The Batch Job View . . . . .	103
13.4.4 Running the batch job . . . . .	104
13.5 FEBio Job Reports . . . . .	105

<b>14 The Post Environment</b>	<b>106</b>
14.1 The Post Environment UI . . . . .	106
14.2 The Post Menu . . . . .	107
14.3 The Post Toolbar . . . . .	108
14.4 The Graphics View . . . . .	110
14.4.1 Elements of the GV . . . . .	110
14.4.2 Customizing the GV . . . . .	110
14.4.2.1 Selecting and moving widgets . . . . .	111
14.4.2.2 Setting the GV widget's properties . . . . .	111
14.4.2.3 Adding GV Widgets . . . . .	112
14.4.2.4 Deleting GV Widgets . . . . .	112
<b>15 Saving Graphics</b>	<b>113</b>
15.1 The Capture Frame . . . . .	113
15.2 Taking a snapshot . . . . .	113
15.3 Using the Ray Tracer . . . . .	113
15.4 Recording an animation . . . . .	115
15.5 Camera Control . . . . .	115
15.5.1 Basic Camera control . . . . .	116
15.5.2 Element tracking . . . . .	116
15.5.3 Camera key-framing . . . . .	116
<b>16 The Post Panel</b>	<b>117</b>
16.1 The View Tab . . . . .	117
16.2 The Material Tab . . . . .	117
16.3 The Data Tab . . . . .	118
16.3.1 Adding data from a text file . . . . .	119
16.3.2 Adding data via an equation . . . . .	120
16.3.3 Filtering data . . . . .	120
16.3.4 Exporting Data . . . . .	121
16.4 The State Tab . . . . .	121
<b>17 Post Processing</b>	<b>123</b>
17.1 Properties of the Model . . . . .	123
17.2 Displacement Map . . . . .	124
17.3 Color Map . . . . .	124
17.4 Plane Cuts . . . . .	125
17.5 Mirror Plane . . . . .	127
17.6 Vector Plot . . . . .	127
17.7 Isosurface plot . . . . .	128
17.8 Slice plot . . . . .	128
17.9 Tensor plot . . . . .	129
17.10 Streamline Plot . . . . .	129
17.11 Particle Flow Plot . . . . .	130
17.12 Additional Windows . . . . .	131
17.12.1 Summary Window . . . . .	131
17.12.2 Graph Window . . . . .	132
17.12.3 Graph Tools . . . . .	133

17.12.4 Selecting mesh items . . . . .	133
17.12.5 Integration Tool . . . . .	134
17.13 Post Session Files . . . . .	135
17.13.1 PSF File Structure . . . . .	135
17.13.1.1 model . . . . .	135
17.13.1.2 material . . . . .	136
17.13.1.3 datafield . . . . .	136
17.13.1.4 plot . . . . .	136
17.13.2 PSF Examples . . . . .	136
<b>18 Visualizing 3D Image Data</b> . . . . .	<b>138</b>
18.1 Loading 3D image data . . . . .	138
18.1.1 Supported Image Types . . . . .	138
18.2 Image GUI Environment . . . . .	139
18.2.1 Properties Box . . . . .	139
18.2.2 3D Image Settings Panel . . . . .	139
18.2.3 Slice View . . . . .	140
18.2.4 Slice Sequence View . . . . .	140
18.3 Filters . . . . .	141
18.3.1 Adding, Removing, and Editing Filters . . . . .	141
18.3.2 Available Filters . . . . .	141
18.4 Other 3D Image Visualization Options . . . . .	141
18.4.1 Image Slicer . . . . .	142
18.4.2 Volume Renderer . . . . .	142
18.4.3 Image Isosurface . . . . .	142
18.5 Other 3D Image Tools . . . . .	143
18.5.1 Image Map Tool . . . . .	143
18.5.2 Fiber ODF Analysis . . . . .	147
18.5.2.1 Running a Fiber ODF Analysis . . . . .	147
18.5.2.2 The Fiber ODF Analysis UI . . . . .	151
18.5.2.3 The Fiber ODF Constitutive Model . . . . .	152
<b>19 Using Python in FEBio Studio</b> . . . . .	<b>154</b>
19.1 The Embedded Python Interpreter . . . . .	154
19.2 The External Python Module . . . . .	155
<b>20 Using and Creating FEBio Plugins</b> . . . . .	<b>156</b>
20.1 The Plugin Repository . . . . .	156
20.2 Creating FEBio Plugins . . . . .	157
20.2.1 Prerequisites . . . . .	157
20.2.2 The FEBio Plugin Wizard . . . . .	158
20.2.3 Building the plugin . . . . .	160
20.2.4 Plugin Caveats . . . . .	161
<b>A Mesh Import Formats</b> . . . . .	<b>162</b>
A.1 FEBio . . . . .	162
A.2 NIKE3D . . . . .	162
A.3 HyperMesh ASCII . . . . .	163

A.4 ABAQUS . . . . .	163
A.5 LSDYNA keyword . . . . .	164
A.6 ANSYS . . . . .	164
A.7 DXF . . . . .	164
A.8 Hypersurface ASCII . . . . .	165
A.9 GMsh . . . . .	165
A.10 BYU . . . . .	165
A.11 VTK . . . . .	165
A.12 STEP . . . . .	166
A.13 BREP . . . . .	166
A.14 IDEAS . . . . .	166
A.15 NASTRAN . . . . .	167
A.16 MESH . . . . .	167
A.17 TETGEN . . . . .	167
A.18 RAW . . . . .	167
A.19 COMSOL . . . . .	167
A.20 PLY . . . . .	168
<b>B Standard Data Fields</b>	<b>169</b>
<b>Bibliography</b>	<b>170</b>

# Chapter 1

## Introduction

### 1.1 Overview of FEBioStudio

FEBio Studio is an integrated environment for the finite element modeling program FEBio. Although it offers a variety of functionality, its primary function is to set up the boundary and loading conditions, material properties, and analysis parameters for finite element analysis for the software FEBio. The main features of FEBio Studio include:

- User-friendly UI that greatly facilitates the FE modeling process
- Primitive mesh generation, e.g. boxes, cylinders, spheres, etc.
- Tetrahedral mesh generation and remeshing via TetGen, NetGen, and MMG
- Mesh editing on element and sub-element level
- Supports various FE file formats, including FEBio, Abaqus, Ansys, Comsol, LSDYNA
- Supports various mesh file formats, including STL, VTK, BYU.
- Supports most of FEBio's modeling capabilities, including boundary conditions, surface and body loads, contact interfaces, analysis types, and more.
- Mesh can also be exported to various file formats, including LSDYNA, VTK, and more.
- FEBio model can be run from within the GUI or exported to file.
- FEBio plot files, which contain the analysis results, can be visualized directly in FEBioStudio.
- 3D Image visualization and processing.

### 1.2 About this document

This document is the User's Manual to FEBio Studio. Although FEBio Studio has been designed for FEBio, it does not describe the FEBio features in much detail. A more in-depth description of the FEBio features can be found in the *FEBio User's Manual* and the *FEBio Theory Manual*. These manuals and other helpful material can be found on the febio.org website. See the [FEBio Knowledgebase](#) in particular.

# Chapter 2

## Getting Started

This chapter gives a brief tour of FEBio Studio. The reader will be introduced to the graphical user interface (GUI) and some of the different components of FEBio Studio.

### 2.1 Starting FEBio Studio

When FEBio Studio starts, the *Welcome Page* is shown.

The Welcome Page displays some quick access links that allow you to get started quickly. It is divided in the following sections.

- **Start:** Shows links to several **File** menu items, including opening files or creating new models. Clicking a link here will access the corresponding menu item.
- **Recent:** This will show a list of recently created models and projects. Clicking a link here will open the corresponding model or project file.
- **Online Resources:** Some links to online FEBio resources, such as the FEBio Knowledgebase, where you can find tutorials on FEBio Studio. Clicking a link here will open the corresponding page in a web browser.

### 2.2 Creating a New Model

A new model can be started either by clicking the *New Model...* link on the Welcome page, or selecting the **File** → **New Model...** menu item. A dialog box will be shown next, where you can select a model template and name. Model templates are discussed in more detail in Chapter 3, but the main purpose is to configure the UI so that only relevant features are shown. For instance, if the Structural Mechanics template is chosen, the Physics menu will only show options that can be applied to a mechanics model. After selecting a template, and optionally choosing a name for the new model, click OK to close this dialog and return to FEBio Studio. The UI will look something like the figure below.

The *Main menu* lists all the available menu items. The *Main Toolbar*, located directly below the menu, offers an alternative way to invoke some of these menu items. The *Model Viewer* shows an overview of the components of the model. The *Graphics View* covers the largest part of the GUI and shows a 3D view of the model. The *Graphics Toolbar* is located below the Graphics View and displays some information regarding the Graphics View. Additional panels may be shown, such as

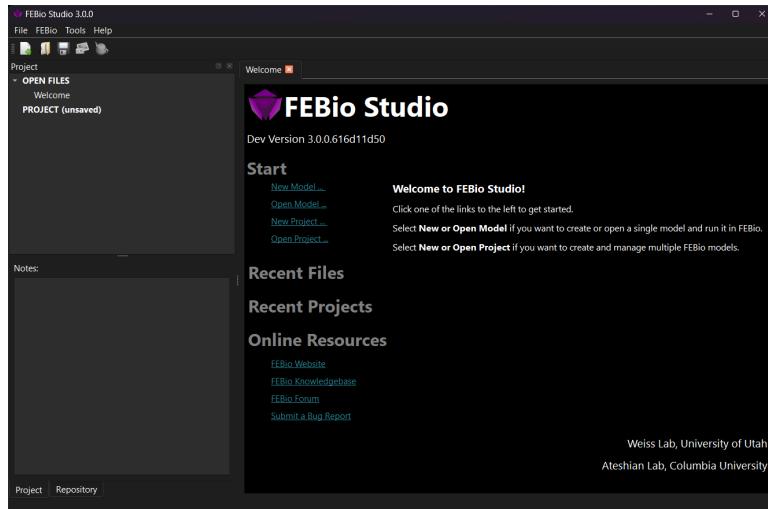


Figure 2.1: The Welcome Page when FEBio Studio opens for the first time.

the Log panel or the Build panel. Most panels can be detached, dragged and dropped anywhere around the main window's borders. Panels can also be stacked by dropping a panel on top of another one. A tab bar will then appear below the panel. If a panel is not visible, either click the tab below stacked panels or access it via the menu **View** → **Windows**. The various panels and other UI components are discussed in more detail in Chapter 3 .

## 2.3 Opening a Model

You can open a model from the **File** → **Open Model File ...** menu. You can also open recent models from the list shown on the Welcome Page. Clicking a link in the Recent section will open the model. FEBio Studio also supports drag-and-drop for some file formats. For instance, you can drag a model file from Windows Explorer and drop it on FEBio Studio to open it.

Open the file *example.fsm* from the *Examples* folder of your FEBioStudio installation folder. You can do this by selecting the **File** → **Open Model File ...** menu item. A standard file open dialog box appears. Locate the file, select it and click on *Open*. The UI will now look something like Figure 2.3.

## 2.4 Exploring a Model

### 2.4.1 Navigating the Graphics View

To rotate the view, click anywhere in an open area of the graphics view with the left mouse button and hold it down. By dragging the mouse left and right you can rotate the view to the left or to the right. Similarly, by dragging the mouse up and down you rotate the model up and down. Panning works similarly as rotating, except you use the middle mouse button instead of the left. Hold down the middle mouse button and drag the mouse to pan the view. You can zoom in or out by holding down the right mouse button. Drag the mouse up to zoom in and move it down to zoom out. See section 3.1.2 for an overview of the different methods of changing the view.

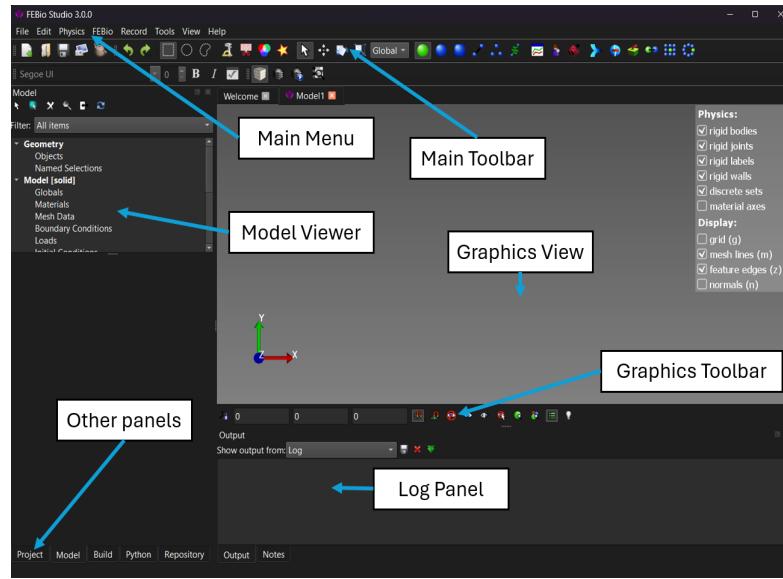


Figure 2.2: The main components of the FEBioStudio GUI. The Main Menu provides access to most of FEBioStudio's functionality. The Main toolbar offers shortcuts for some commonly used features. The Graphics View shows a 3D rendering of the model. The Build panel shows all the geometry creation and editing tools. The Model Viewer displays a hierarchical overview of the finite element model.

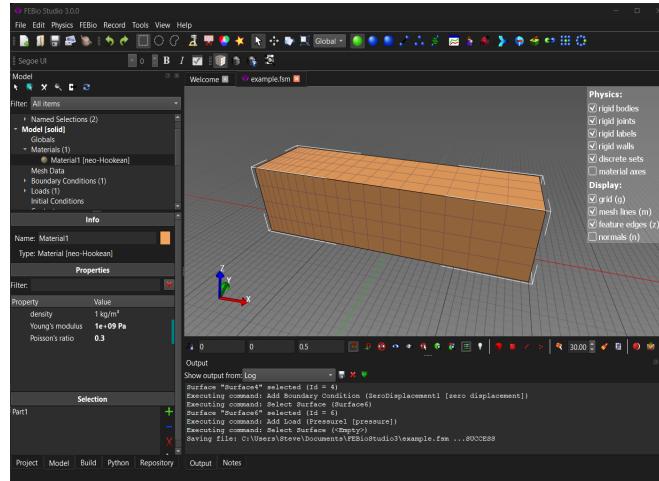


Figure 2.3: The UI after opening the example.fsm file. The Model Viewer is now populated with the model components.

## 2.4.2 Selecting objects

Selection is another important aspect for interacting with the model. Selection requires a selection context, which tells FEBioStudio what type of entity you would like to select. The current selection context is highlighted on the main toolbar and can be changed by clicking on the corresponding selection button. For instance, to select an object, first click the “Select Object” button on the main toolbar (i.e. the button with the green ball). With the button selected, you can now select any object in the model by clicking on it with the left mouse button. Selecting another object will automatically unselect the first selected object. To add an object to the current selection, hold down the *shift* key when you select the object. Similarly, you can remove an object from the current selection by holding down the *ctrl* key when clicking on the object.

To select several objects with one motion, first choose a selection method from the main toolbar. There are three selection methods: rectangle, circle, and free form. For example, select the rectangle selection method. Now, hold down the *shift* key and then click and hold down the left mouse button on an empty part of the screen. With the *shift* key and left mouse button down, drag the mouse to draw a rectangle around the objects you wish to select. Once you let go of the left mouse button, all objects that are inside or intersect the rectangle will be selected. You can also use this procedure to unselect several objects at once. Just repeat the procedure but hold down the *ctrl* key instead of the *shift* key. The other selection methods work similarly.

## 2.4.3 Transforming an object

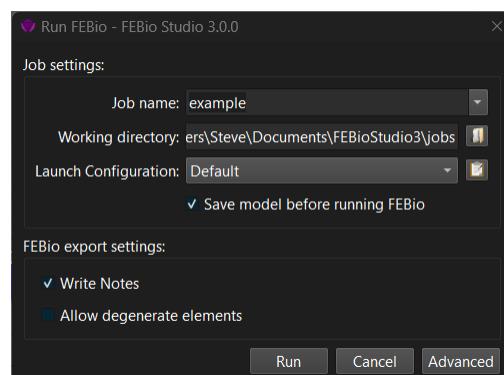
Once an object is selected you can transform it, which means you can translate, rotate, or scale it. To apply a particular transformation, make sure one or more objects are selected. Then click on one of the transform buttons on the toolbar. For example, to translate an object click on the translate toolbar button (i.e. the button with the crossed arrows). A triad appears on the selected object(s). The triad is positioned at the current pivot point, which is the center from which the transformation is applied. The triad consists of three colored arms and three colored planes. When you move the mouse over one of the arms or planes, it becomes highlighted. A highlighted arm or plane can be selected and when you drag the mouse, you can move the object in the corresponding direction.

For more precise modeling you can also enter the transformation numerically using the *Transform* dialog which is accessible from the **Edit** menu (**Edit → Transform ...**).

If you accidentally moved an object, you can undo your action by selecting the **Edit → Undo** menu (shortcut *ctrl+z*). The **Edit → Redo** menu option allows you to redo your last undone action. Note that you have a virtually unlimited undo-redo stack so you rarely have to worry about making mistakes. However, for large models the undo stack may consume a lot of memory. The undo stack is cleared each time you save your model.

## 2.5 Solving the Model

The installation of FEBio Studio comes with the latest version of FEBio and you can run the model in FEBio directly from the FEBio Studio UI. After you save the model, you can run it from the menu **FEBio → Run ...**, or by clicking the corresponding icon on the main toolbar.



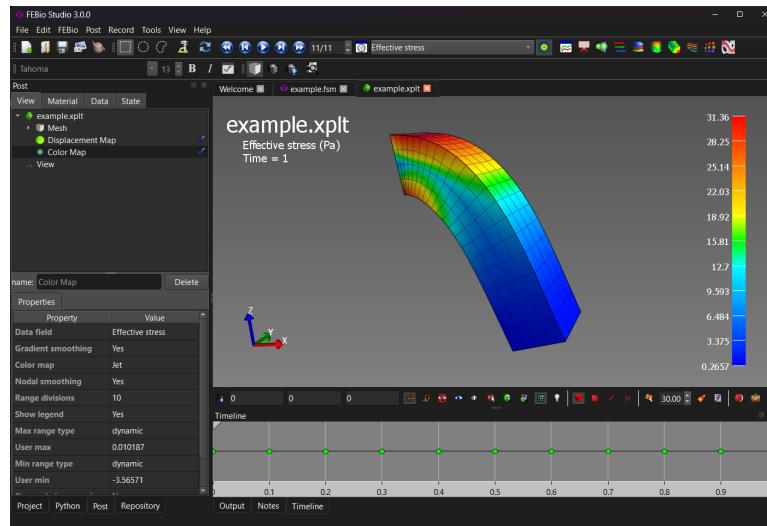


Figure 2.5: FEBio Studio’s UI when a plot file, the results file of an FEBio analysis, is loaded.

The Run FEBio dialog box will appear. When clicking the Run button, FEBio Studio will write the FEBio model input file (usually, this file will have the job’s name with the .feb file extension) and then start FEBio, passing the input file to it. Depending on the selected launch configuration, FEBio will run in the same process (default), or in a separate process. Users can create special launch configuration that will run FEBio on a remote server.

The output generated by FEBio is displayed on the Log panel. See [13](#) for more details on running febio models from within FEBio Studio.

When FEBio is done, the results are stored in a “plot file”. The file name is usually the job’s name and the .xplt file extension.

When running an FEBio model, a new item is added to the model tree. This item will appear under Jobs and has the job’s name. When the model is running in FEBio, this item will show some progress and status information. After the run is completed, the results can be loaded via this item.

## 2.6 Loading the Results

When running a model in FEBio, a new item is added to the model tree with the job’s name. When FEBio completed you can open the results by double-clicking the item in the model tree. This will load the job’s plot file and open it in FEBio Studio.

When you open a plot file in FEBio Studio you will notice that the UI changes. The Model and Build panels will disappear and the Post panel will appear. The Build toolbar is also replaced by the Post toolbar. The Post panel has somewhat the same purpose as the Model panel and displays a tree structure of the contents of the plot file. It also contains some additional tabs that provide additional information on the results and will be discussed in more detail in Chapter [14.1](#).

# Chapter 3

# The FEBio Studio Environment

This chapter provides an in-depth description of FEBio Studio's graphical user interface and how to interact with it. FEBio Studio configures its UI depending on whether you are building a model (the *Build* configuration), or whether you are analyzing a model's result (the *Post* configuration). In this chapter, the *Build* configuration is discussed as well as the UI features that are common in both configurations.

## 3.1 The Graphical User Interface

### 3.1.1 Overview

FEBio Studio has a powerful graphical user interface (GUI) that offers an intuitive approach to setting up a finite element problem. It has several components to it, and in order to make optimal use of FEBio Studio it is important that the user is familiar with GUI. Chapter 2 guides the user through the most important skills needed to manipulate the GUI. In this chapter we provide a more in-depth discussion of it.

Figure 3.1 shows the FEBio Studio GUI and many of its most important components. The *Main Menu* bar gives access to most features such as file I/O, editing selections, setting up the physics, customizing the view and much more. The *Main Toolbar* provides an alternative way to invoke some of the most commonly used menu commands. It also has some buttons that affect the way the user interacts with the model. The *Model* panel shows a hierarchical overview of the model and its components. The *Graphics View* displays a 3D view of the model. The *Status bar* at the bottom of the screen displays some information regarding what FEBio Studio is doing. The *Graphics Toolbar* shows additional tools for interacting with objects and selections in the Graphics View. The *Build panel* has several child panels that are used to select the current active working context. The *Create panel* allows the user to create geometry, the *Edit panel* lists several options to select and edit the geometry. The *Mesh panel* collects all parameters that affect mesh generation. Finally the *Tools panel* offers some alternative editing tools.

### 3.1.2 Navigating the GUI

Navigating the GUI is done using the mouse. By holding down one of the three mouse buttons and dragging the mouse, the viewing position can be changed. The view can be rotated, panned (i.e. translated), or zoomed. The following table gives an overview of the different methods to change the view.

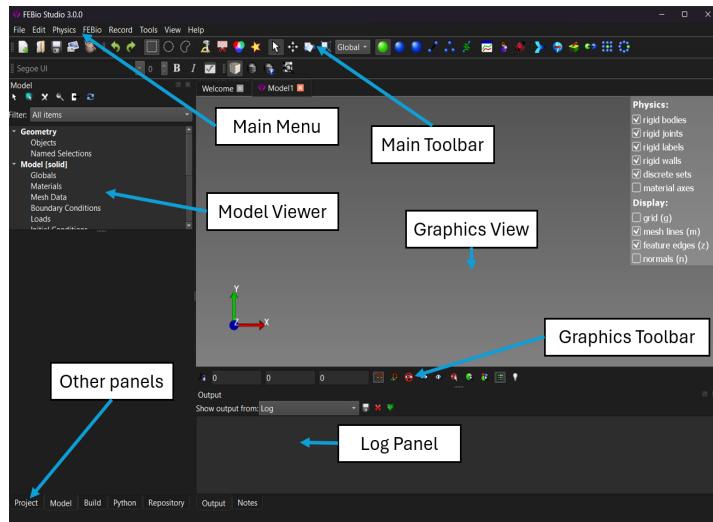


Figure 3.1: FEBio Studio's Graphical User Interface.

Action	Standard Method	Alternative Method
Rotate	left mouse button	
Rotate in-plane	left mouse button + Alt key	
Panning	middle mouse button	right mouse button + Alt key
Zooming	right mouse button	

Table 3.1: Overview of methods to change the view.

In the following sections, the different components of the menu bar will be explored.

## 3.2 The Menu Bar

The *Menu Bar* found at the top of the window gives access to the following menus:

- **File:** Open files, save files, export or import files, etc.
- **Edit:** Undo/Redo feature, options to edit the selection.
- **Physics:** Define boundary conditions, contact interfaces, materials , etc.
- **FEBio:** Run FEBio models
- **Record:** create video recordings of the Graphics View
- **Tools:** provides access to a set of useful tools, including the Settings dialog.
- **View:** modify view settings and access additional windows.
- **Help:** Access to the help options and About box.

A more detailed explanation of the available menu items follows.

### 3.2.1 The File Menu

The File menu offers the following menu items:

- **New Model:** Start a new FEBio Studio model. This will open the Model Template dialog where you can select a template and set the model's name.
- **New Project:** Create a new FEBio Studio project.
- **Open Model File:** Open a FEBio Studio model from file.
- **Open Recent:** Open a recently opened model file.
- **Open Project:** Opens a project file (but not the models in the project).
- **Open Recent Project:** Provides a list of recently opened projects.
- **Import Geometry:** Import a surface or volume mesh into the current model.
- **Import recent Geometry:** Shows a list of recently accessed mesh files.
- **Close all:** Close all the currently opened models.
- **Save:** Save the current FEBio Studio model to file.
- **Save as:** Save the current FEBio Studio model under a different name.
- **Save all:** Saves all the currently opened models.
- **Save Project as:** Save the current project to file.
- **Export FE Model:** Export the current model to a FE file format.
- **Export Geometry:** Export the mesh of the selected geometry to a surface or volume mesh file format.
- **Import Project Archive:** Opens a project from a project archive.
- **Export Project Archive:** Saves the current open project to a project archive.
- **Import Image:** Import image data into the current model.
- **Batch convert:** Convert a list of files from one format to another. (For example, converting between an older to a newer FEBio file format.)
- **Batch Run:** Allows users to set up a batch run, which will run a list of FEBio models.
- **Exit:** Exits the application.

### 3.2.2 The Edit Menu

The Edit menu offers the following menu items:

- **Undo:** undo the last operation
- **Redo:** redo the last undone operation
- **Change log:** opens a dialog showing all the changes that are made to the model.
- **Invert selection:** Invert the current selection
- **Clear selection:** Clear the current selection
- **Delete selection:** Delete the current selection
- **Name selection:** name the current selection
- **Hide selection:** hide the current selection
- **Hide unselected:** hides the unselected items.
- **Unhide all:** Shows all the hidden objects.
- **Toggle Visibility:** Hides visible objects and parts and shows hidden ones.
- **More selection options:** Access to some additional options for working with selections.
- **Find:** Locate nodes, elements, etc., based on their IDs.
- **Transform:** Modifies the current selections position, rotation, and scale.
- **Collapse Transform:** Applies the selected object's transformation to its mesh and resets its transform.
- **Clone object:** create a copy of the currently selected object.
- **Merge objects:** Merges all the selected objects into a single editable mesh.
- **Copy Object:** create a copy of an object that can be pasted into a different model.
- **Paste Object:** pastes the copied object into the current model
- **Purge:** removes all physics from the model (materials, BC's, loads, etc.)

### 3.2.3 The Physics Menu

The Physics menu offers the following menu items:

- **Assign current selection:** Assigns the current selection to the item that is selected in the model tree.
- **Add Nodal BC:** Apply a boundary condition to a selection of nodes.
- **Add Surface BC:** Apply a special boundary condition to a surface.
- **Add linear constraint:** Apply a linear constraint boundary condition.

- **Add Nodal Load:** Apply a load on selected nodes.
- **Add Surface Load:** Apply a boundary load to a surface.
- **Add Body Load:** Apply a body load.
- **Add Initial condition:** Define an initial condition for the selection.
- **Add Contact:** Set up different contact conditions between the objects in the model.
- **Add Surface Constraint:** Add a nonlinear constraint to a surface.
- **Add Body Constraint:** Add a nonlinear constraint to a part.
- **Add a general constraint:** *Add a nonlinear constraint.*
- **Add Rigid Constraint:** Define a constraint to a rigid body.
- **Add Rigid Initial Condition:** Add an initial condition to a rigid body.
- **Add Rigid Load:** Add a load to a rigid body.
- **Add Rigid Connector:** Define a rigid connector between two rigid bodies.
- **Add Material:** create a new material and manage material libraries.
- **Add Analysis Step:** define a new analysis step.
- **Solute Table:** Define all solutes for this analysis.
- **Solid-bound Molecule Table:** Define all the solid-bound molecules of this analysis.
- **Chemical Reaction Editor:** Define all the chemical reactions of this analysis.
- **Membrane Reaction Editor:** Define special membrane reactions for the current model.
- **Edit Physics Modules:** Change the settings of the active model. This allows users to activate or deactivate the different FEBio physics modules.

### 3.2.4 The FEBio Menu

- **Run FEBio:** Run the active model in FEBio.
- **Stop FEBio:** Stop the current FEBio run.
- **Run FEBio Monitor:** Runs the FEBio model using the FEBio Monitor.
- **Create Optimization Study:** Opens wizard that walks users through generating an FEBio optimization input file.
- **Generate Tangent Diagnostic:** Generates a tangent diagnostics file, which can be used to debug material implementations.
- **FEBio Info:** Brings up a dialog box that lists all the available features pulled from FEBio.
- **Plugins Repository:** Access the plugin repository that allows users to download and install FEBio plugins in FEBio Studio.
- **Create FEBio Plugins:** Start a wizard that will help users generate an FEBio plugin template.

### 3.2.5 The Record Menu

The Record menu gives the user access to the screen recording capabilities of FEBio Studio.

- **New...** : selects a new target file for the recording.
- **Start**: starts recording the GV and stores the frames to a file.
- **Pause**: pauses the current recording.
- **Stop**: stops the current recording and closes the target file.

### 3.2.6 The Tools Menu

The *Tools* menu offers the following menu items:

- **Curve Editor**: Show the Curve Editor tool
- **Mesh Inspector**: Activate the Mesh Inspector tool.
- **Mesh Diagnostics**: Opens the mesh diagnostics tool, which allows users to run a diagnostic on the mesh of the currently selected object.
- **Unit Converter**: Convert quantities between different units.
- **Elasticity Converter**: convert between parameter for defining elastic materials.
- **Material Test**: A tool that can be used to run simple scenarios (e.g. uni-axial testing) on one of the materials in the model.
- **Distribution Visualizer**: This tool can be used to visualize the continuous fiber distribution options.
- **Kinemat**: The Kinemat tool can be used to apply kinematics data to a model. To use it, first select the model file, which currently has to be in the .k (LSDYNA keyword) format. Then, select the kinematics file. This text file should define for each time step a line with a comma separated list of 4x4 transformation matrices, one for each material in the model. The matrices have to be entered in row-major order. (The last row is currently ignored and can be zero.) Finally, enter the range and stride of the rows of the kine file that will be read in. When Apply is pressed the Kinemat tool will apply the transformation matrices to all the nodes of the models, generating a state for each row of the kine file.
- **Plot Mix**: This tool can be used to combine several plot files into one. A file list can be defined via the “Add file” button. When Load is pressed, the tool will grab the last state from each plot file and use it to define a time step.
- **Python Editor**: Opens an editor that can be used to write and execute Python scripts.
- **Settings**: Open a dialog box that allows you to edit FEBio Studio’s settings.

### 3.2.7 The View Menu

The *View* menu offers the following menu items:

- **Undo view change**: undo the last view change
- **Redo view change**: redo the last undone view change
- **Show GV Context menu**: Toggles the quick-menu box displayed on the graphics view.
- **Zoom to Selection**: Zooms in on the current selection.
- **Orthographic Projection**: Toggles between orthographic or perspective projection mode.
- **Show Normals**: Toggles surface normal on or off of the active object.
- **Show capture frame**: toggles the *capture frame*, which is the part of the screen that will be captured during recording on screenshots.
- **Toggle Overlay**: hides all GV items that are rendered on the overlay texture.
- **Show Grid**: toggle the grid in the GV on or off.
- **Show mesh lines**: toggle the mesh lines in the GV on or off.
- **Show Edge lines**: toggle the rendering of the crease edges on or off.
- **Color smoothing**: toggles between smooth and discrete contouring. (only used on Post side.)
- **Toggle Render Mode**: toggles between solid and wireframe render mode.
- **Show fibers**: Brings up the fiber visualization dialog box, which allows visualization of material fibers onto the mesh.
- **Toggle Material axes**: toggles the option to render the material's local axes.
- **Show discrete sets**: Toggles whether discrete element sets (e.g. springs) will be shown or not.
- **Toggle Tag Info**: Toggles between the different tag info levels (none, item label, item and connected nodes)
- **3D cursor to selection**: Place the 3D cursor on the center of the current selection.
- **Track selection**: when active, keeps the camera focused on the current selection. (only used on Post side.)
- **Toggle Lighting**: Turns lighting (i.e. shading) on or off.
- **Toggle select connected**: toggles selection mode between selecting a single item or all items connected to the selected item.
- **Toggle FPS**: toggles rendering stats.
- **Standard Views**: Select one of the standard views from the submenu.

- **Save viewpoint:** Save the current camera view (only used by Post side)
- **Previous viewpoint:** Returns camera to the previous view point. (only used by Post side)
- **Next viewpoint:** Returns camera to the next view point (only used by Post side)
- **Sync all views:** Applies the camera view from the current model to all the open models.
- **Windows:** Toggles visibility of the FEBio Studio panels.

### 3.2.8 The Help Menu

The Help menu offers the following menu items.

- **Check for updates:** see if any updates are available.
- **Show Welcome Page:** Makes the Welcome page active.
- **FEBio Website:** Opens the FEBio website ([febio.org](http://febio.org)) in an external web browser.
- **FEBio Knowledgebase:** Opens the FEBio Knowledgebase website in an external browser.  
The FEBio Knowledgebase contains user manuals, tutorials, and more information for working with FEBio and FEBio Studio.
- **FEBio User Manual:** Open the online FEBio User Manual in an external browser.
- **FEBio Theory Manual:** Open the online FEBio Theory Manual in an external browser.
- **FEBio Studio Manual:** Open the online FEBio Studio Manual in an external browser.
- **FEBio Forums:** Opens the FEBio forum website in an external web browser.
- **FEBio Publications:** Opens a web page with a list of publications that have used FEBio.
- **Submit a bug report:** Allows users to generate and submit bug reports that will be posted to the FEBioStudio GitHub issues page.
- **About FEBio Studio:** Displays the FEBio Studio About Box.

## 3.3 The Main Tool Bar

Some of the menu items can also be accessed through the Main Tool bar. The Main Tool bar offers the following options.



Start a new FEBio Studio model.



Open a saved FEBio Studio model.



Save the current FEBioStudio model.

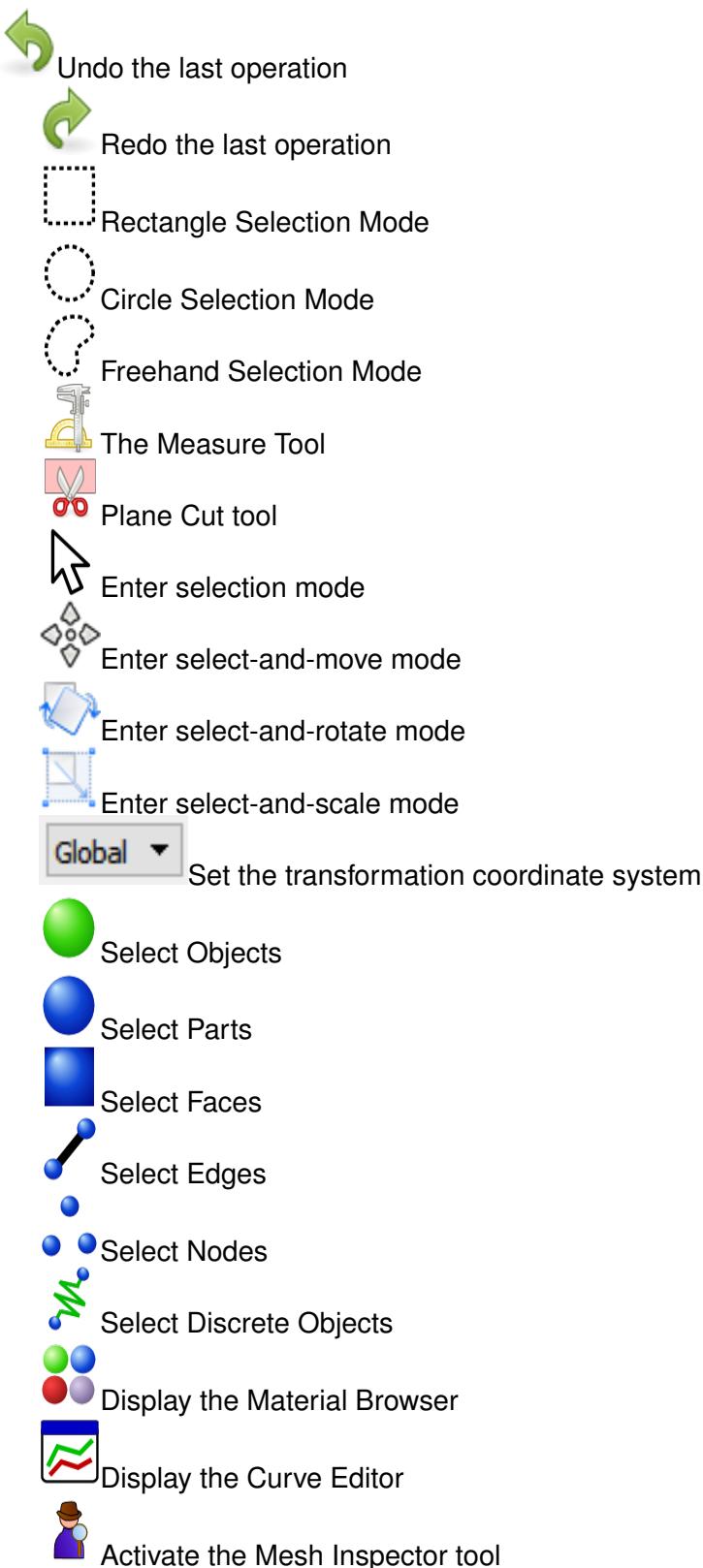


Saves a screenshot to file.



Creates a ray-traced rendering of the active model.

### 3.4 The Build Tool Bar



-  Opens the Run FEBio dialog box.
-  Activates Merge tool
-  Activate Detach Elements tool
-  Active Extract Faces tool
-  Clone Object
-  Clone Grid Object
-  Clone Revolve Object

## 3.5 The Font Toolbar

The font toolbar allows users to easily modify the font of the selected graphics widget.

-  Make the font **bold**
-  Make the font *italic*
-  Open the widget properties dialog box

## 3.6 The Graphics View

The Graphics View shows a 3D view of the current model. The user can adjust the view in several ways. The fastest way of maneuvering through the view is by using the mouse. The different mouse buttons invoke different commands depending on the current *mode* of the Graphics View. There are four buttons on the Toolbar that control this mode, namely the *select*, *move*, *rotate* and *scale* buttons. The current mode is indicated by the button that is highlighted.

Regardless of the Graphics view mode, the view can be rotated by dragging the mouse button outside any geometry while holding down the left mouse button. A similar action but with the right mouse button down allows the user to zoom the view. The user can pan the current view by holding down the middle mouse button. Objects and items can be selected in all four Graphics view modes. To select an object or item (e.g. part, surface, edge), bring the mouse cursor over the object or item, and click on it with the left mouse button. To add objects or items to the current selection, hold down the shift button while clicking on the object or item. To add several objects or items, drag the mouse while holding down the shift button and left mouse button. To deselect objects or items, the same operations can be performed, only this time, hold down the ctrl button instead of the shift button. To deselect all objects or items at once, click in an empty area of the Graphics View. There are also several menu items that allow you to modify the current selection. See section 3.2.1 for more information.

By default, FEBio Studio renders the model using *perspective projection* in the Graphics View. By right-clicking with the mouse button a popup menu shows up under the cursor that allows the

user to toggle between perspective and orthographic projection. In addition, this menu provides some buttons that allows the user to quickly orient the geometry along one of the three coordinate axes.

When the Graphics View is in one of three transform modes (move, rotate or scale) a triad will appear in the center of the selection. When moving the mouse over the triad, one of its arms may highlight, indicating that this arm can be selected. To apply a transformation to the current selection, bring the mouse cursor over one of the triad arms so that it highlights. Next, drag the mouse button while holding down the left mouse button. Releasing the left mouse button will let go of the arm and finish the transformation.

## 3.7 The Graphics Toolbar

The *Graphics Toolbar* can be found at the bottom of the Graphics View and provides the user with some information regarding the Graphics View and some additional tools that affect the current selection.



This toggles the *pivot lock* mode. The pivot is the point that serves as the origin for transformations. Usually it is calculated automatically based on the current selection, but by clicking this button, the user can lock the pivot and edit it coordinates manually in the next three panels.



When the *snap-to-grid* button is checked, a selection can only be translated in increments of the current grid spacing.



This option allows users to position the 3D cursor at nodes of the geometry.



This toggles all visible objects and parts. (Or elements when in mesh selection model.)



Zoom to the current selection.



Zoom to the extents of the current model.



Toggle Mesh Lines

## 3.8 The Model Viewer

The *Model Viewer* shows a hierarchical overview of all the components of the model and their interdependencies. The model data is organized in separate categories, depending on the purpose of the data.

- **Geometry:** Contains a list of all the objects and the named selections in the model. The named selections can be used to easily refer to a collection of parts, surfaces, edges, or nodes and are used for defining boundary conditions, loads, contact interfaces and other model components that are defined on the geometry.
- **Globals:** Displays the global model parameters.
- **Materials:** Lists all the materials defined for the model.
- **Mesh Data:** This item contains all the mesh data maps and generators.

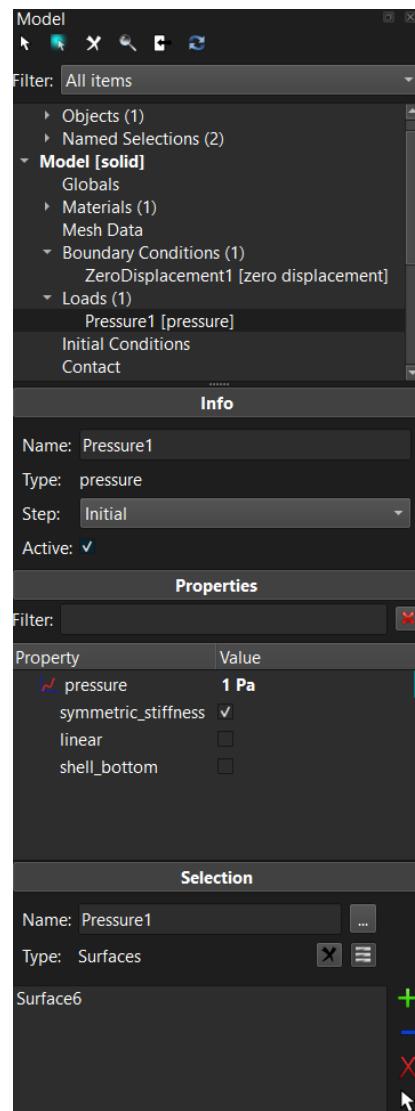


Figure 3.2: The Model Viewer shows a hierarchical overview of the model components.

- **Boundary Conditions:** Lists all the boundary conditions defined for the model and for all analysis steps.
- **Loads:** Lists all the loads that are defined for the model and for all analysis steps.
- **Initial Conditions:** Lists all the initial conditions for the model and for all analysis steps.
- **Contact:** Lists all the contact interfaces that are defined for the model and for all analysis steps.
- **Constraints:** Lists all the nonlinear constraints defined for the model and for all analysis steps.
- **Rigid:** contains a list of all the rigid body constraints, connectors, loads, and initial conditions.
- **Discrete:** lists all the discrete element sets (e.g. springs) defined on the model
- **Mesh Adaptors:** Lists all the mesh adaptors used for this model. Mesh adaptors allows the mesh to be modified as part of the solution. (e.g. adaptive mesh refinement)
- **Steps:** Lists all the steps that are defined for this model. Each step will be listed as a sub-item of this item. Each sub-item has additional items that list all the components that will be active only during that particular step.
- **Load controllers:** Lists all the load controllers in use by this model. Load controllers allows model parameter to be dependent on (simulation) time.
- **Output:** Can be used to define the field variables that need to be output to the FEBio plot file.

Many of the FEBio Studio features can also be accessed by right-clicking on an item in the Model Editor. A popup menu will appear that lists the available options for that particular item. For example, you can add a new material by right-clicking on the Materials item and selecting **Add Material ...** from the popup menu.

At the top of the Model Editor additional buttons are located that allow access to the following features.



For model components that have associated geometry items, this button selects these items in the Graphics View.



Enables automatic highlighting. When this is enabled, selecting an item in the model tree will automatically highlight the corresponding selection in the Graphics View.



Delete the currently selected items in the Model Viewer.



Activate the Search panel, which provides an alternative method for selecting items in the Model Viewer.



Clicking this button will synchronize the selection between the Graphics View and the Model Viewer. This only works for geometry items.



Clicking this button will refresh the contents of the model tree, making sure all selections are up-to-date and all components are validated.

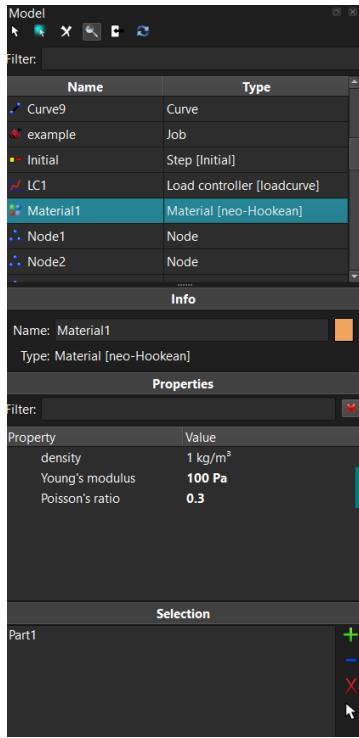


Figure 3.3: The Model Viewer with the Search Panel active.

### 3.8.1 The Search Panel

The Search panel is an alternative method for inspecting model components. When active, users can enter a search filter at the top. All items with a name that contains the filter will be displayed in the box below the filter. Users can select items here and the properties will be displayed below. Users can also right-click on items and select an option from the popup menu. Double-clicking on an item will toggle back to the tree view with that item selected.

### 3.8.2 The Model Viewer Panes

Below the Model Viewer you will see several panes that provide information about the item that is active in the Model Viewer's tree view. The particular panes that are shown depend on the selected item.

- *Object*: Shows the name and type of the active item. For physics components this also shows the step in which that component is active.
- *Properties*: This panel displays the editable parameters of the item that is selected in the Model Viewer.
- *Selection*: This panel is shown for all components that can have a selection assigned to it, such as materials, named selections, boundary conditions, and loads. Contact interfaces will have two selection panels, one for the primary and one for the secondary surface. Detailed instructions on how to edit the selection can be found in section 3.8.3.

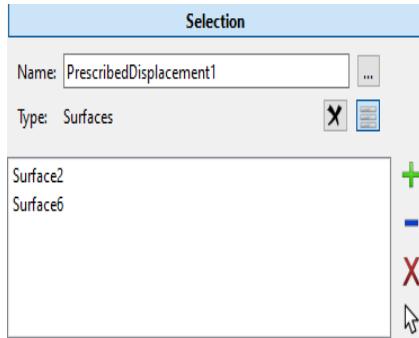


Figure 3.4: The selection pane allows users to edit the geometry selections assigned to certain model components.

Most properties can be edited directly from the properties panel. However, some properties cannot be edited that way. In that case, clicking on the property name may bring up an additional widget that provides the controls to edit the property. This is the case, for instance, for properties that represent parameterized or discretized curves. (e.g. force-displacement curve for nonlinear springs.) Clicking the property name will then bring up a widget that allows users to customize the curve.

### 3.8.3 Editing Selections

Some components of the model, such as materials, named selections, and boundary conditions, need to be applied to a selection of the geometry. These components will have a selection box in the Model Viewer to which items of the model can be assigned to. For instance, for materials, it will show the parts of the model to which the material is assigned.

The list that is displayed in the selection box can be edited using the buttons located on the right side of the selection box. Note that you can select the items in the selection box.

This button adds the current selection in the Graphics View to the list. Note that if the list was not empty, then the selection will be added if it has the same type as the items already in the list. For instance, if you applied a boundary condition to a surface, you can only add (or remove) surfaces from its list.

This button will remove the current selection in the Graphics View from the list.

This button removes the items selected in the list from the box.

This button selects the items that are selected in the box also selected in the Graphics View.

### 3.8.4 The Appearance Editor

In FEBio Studio, the user can change the visual appearance of objects and parts. The main tool to do this is the **Appearance Editor**. The Appearance Editor is opened by clicking the appearance editor button. This button is shown in the properties list of model tree items for which the appearance can be set. At the moment, this is shown when an object or a material in the model tree is selected. The button itself shows a small colored sphere that reflects the appearance of the corresponding model tree item.

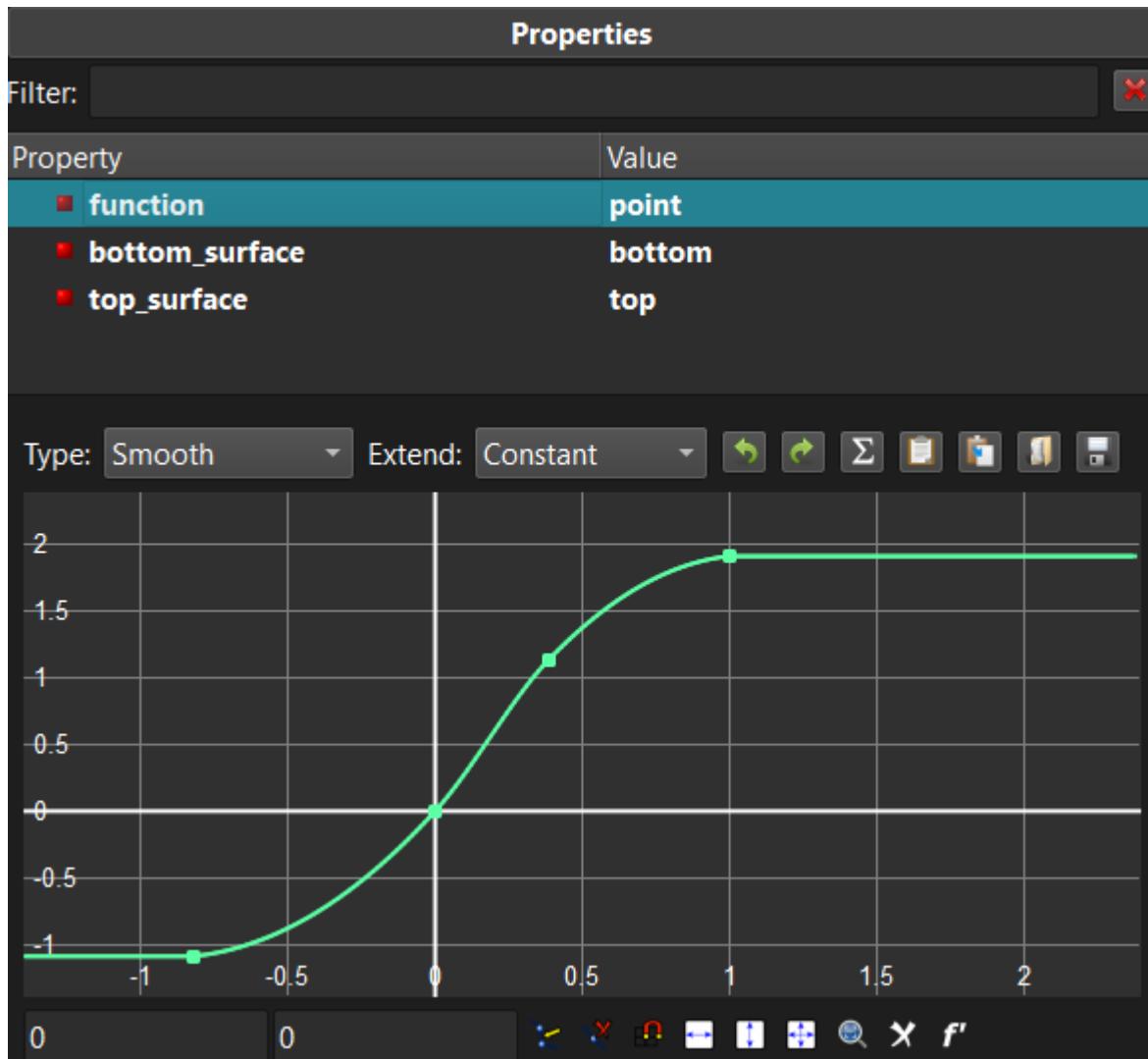


Figure 3.5: Sometimes selecting the property's name may bring up an additional widget that allows users to customize the property. For instance, in the image above a curve is created using a special curve-editing widget.

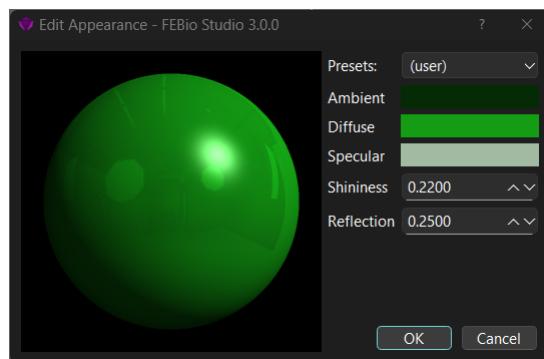


Figure 3.6: Using the Appearance Editor, the user can change the visual appearance of objects and parts.

Using the Appearance Editor, the user can change several settings that affect the visual appearance of objects and parts in the model.

The shading model that is used to color items in FEBio Studio is the classical Phong shading model. This model composes the color as the sum of three separate color contributions:

$$\text{color} = 0.2 \times \text{ambient} + \text{diffuse} \times \cos(N, L) + \text{specular} \times (\cos(H, L))^n$$

The *ambient* color is a constant color that is added to the final color. The *diffuse* color is the main color and is weighted by the direction cosine between the light's direction (*L*) and the local normal (*N*). Areas where the normal is directed towards the light will be bright, and areas that are pointing away from the light will be darker. The *specular* color adds a highlight, i.e. a very bright area, to the overall color, and is weighted by the direction cosine between the halfway vector (*H*), a vector between the light and the view direction, and the light's direction (*L*). To control the size of the specular highlight, the direction cosine is exponentiated to the power *n*, often called the specular exponent, and the higher the exponent, the smaller the highlight. It typically ranges between 0 and 128. In FEBio Studio, however, the specular exponent cannot be set directly. Instead, the *shininess* parameter is used and ranges between 0 and 1.

In addition, users can set the *reflection* parameter to add a reflective effect to the appearance. This will only work if the environment map is set and enabled, which can be done in the FEBio Studio Settings dialog (under the Lighting settings.)

In summary, the following parameters can be set in the Appearance Editor.

**Presets** Choose from a list of common materials, such as various metals, plastics, and biological materials.

**Ambient** Sets the constant ambient color of the material.

**Diffuse** Sets the diffuse color of the material

**Specular** Sets the specular color of the material.

**Shininess** Sets the specular exponent of the specular highlight indirectly and ranges between 0 and 1.

**Reflection** Sets how much the material reflects the environment map and ranges between 0 and 1. (Requires enabling the environment map.)

## 3.9 The Build Panel

The Build panel allows users access to the geometry and mesh editing tools in FEBio Studio. It consists of several child panels, each offering a different set of model editing tools.

### 3.9.1 The Create Panel

The Create panel is used to create geometry. FEBio Studio offers a basic set of mesh generation features; you can create *primitive* geometries, such as boxes, cylinders, spheres and so on. By activating the *Create* panel a list of buttons is displayed. By clicking on one of the buttons, the creation parameters appear for a particular primitive. Once you have entered the parameters you can click the *Create* button to add the primitive to your model. See section 5.1 for more details on how to create geometry.

### 3.9.2 The Edit Panel

The *Edit* panel allows you to modify the object creation parameters. By modifying these parameters, the user can change the geometry at any time. Note that if the geometry is already meshed, changing the object's parameters may invalidate this mesh as well as any selections associated with the mesh. See section 4.3 for more details on how to edit geometry.

### 3.9.3 The Mesh Panel

The *Mesh* panel allows the user to set the meshing parameters of the geometry. By modifying these parameters, the user can control the number and distribution of elements in the mesh. The available parameters depend on the selected geometry. For *Editable Meshes* this panel will show a set of tools that allow the mesh to be modified. Exercise caution when editing meshes in a model that already has boundary conditions, loads, etc., assigned. If any of the selections assigned to these model components depend directly on the mesh, problems may arise if the mesh is changed in any way.

### 3.9.4 The Tools Panel

This panel defines a few specialized tools that can be useful for very specific tasks.

**Scalar Field** With this tool, users can generate scalar data maps on a mesh. These maps can then be used for defining heterogeneous material parameters or other spatially varying model parameters. (See 6.3 for more details.)

**Edit Data Field** This tool allows users to assign values to elements for an existing data map. To use the tool, first select the elements for which you wish to assign a value. Then, choose the data field and the new value, and click **Apply**.

**Plane cut** With this tool, the user can cut a mesh in two parts along a user-defined plane. (Only works for triangle meshes.)

**Fiber Generator** This tool can be used to generate a vector field on a mesh. The vector field can be stored in a data map, which can then be used to define e.g. material fiber orientations. (See 6.4 for more details.)

**Import Springs** This tool imports a collection of line elements from a text or VTK file and produces either discrete (i.e. spring) elements, or linear truss elements. This tool expects that the user has an object selected (which has to be an editable mesh) to which the new nodes and line elements will be added. When adding new nodes, first the tool will check if a node already exists at that location (within the user-defined tolerance) and if so, pick the existing node. If no node exists at the location, a new node will be added to the object. The “Check for intersection” option will check if the new line elements intersect with the existing mesh, and if so, will truncate the element at the intersection point.

**Quadric Fit**

This tool fits a quadric surface to the selected faces. Quadrics include ellipsoids, cylinders and cones. The general equation fitted to the data is

$$\begin{aligned} F(X, Y, Z) = & C_0 X^2 + C_1 Y^2 + C_2 Z^2 \\ & + C_3 YZ + C_4 ZX + C_5 XY \\ & + C_6 X + C_7 Y + C_8 Z + C_9 = 0 \end{aligned}$$

This equation may be rewritten in matrix form as

$$F(X, Y, Z) = [X \ Y \ Z] \begin{bmatrix} C_0 & C_5 & C_4 \\ C_5 & C_1 & C_3 \\ C_4 & C_3 & C_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + [C_6 \ C_7 \ C_8] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + C_9 = 0.$$

Once the coefficients  $C_0 - C_9$  are obtained, the principal axes of the quadric may be found by evaluating the eigenvalues  $(c_0, c_1, c_2)$  and eigenvectors  $(v_0, v_1, v_2)$  of the  $3 \times 3$  coefficient matrix,

$$\begin{bmatrix} C_0 & C_5 & C_4 \\ C_5 & C_1 & C_3 \\ C_4 & C_3 & C_2 \end{bmatrix} = [Q] \begin{bmatrix} c_0 & 0 & 0 \\ 0 & c_1 & 0 \\ 0 & 0 & c_2 \end{bmatrix} [Q]^T,$$

where

$$[Q] = \begin{bmatrix} v_{0x} & v_{1x} & v_{2x} \\ v_{0y} & v_{1y} & v_{2y} \\ v_{0z} & v_{1z} & v_{2z} \end{bmatrix}.$$

Then, in the basis of these eigenvectors, the equation of the quadric may be rewritten as

$$F(x, y, z) = [x \ y \ z] \begin{bmatrix} c_0 & 0 & 0 \\ 0 & c_1 & 0 \\ 0 & 0 & c_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + [c_6 \ c_7 \ c_9] \begin{bmatrix} x \\ y \\ z \end{bmatrix} + c_9 = 0,$$

where

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [Q]^T \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \begin{bmatrix} c_6 \\ c_7 \\ c_8 \end{bmatrix} = [Q]^T \begin{bmatrix} C_6 \\ C_7 \\ C_8 \end{bmatrix}, \quad c_9 = C_9.$$

The different types of quadric surfaces are described on Wikipedia (<https://en.wikipedia.org/wiki/Quadric>). To reduce the above equation to one of the forms presented there, we find the origin  $(x_0, y_0, z_0)$  of the quadric and consider three cases:

1. If  $c_0 \neq 0$  and  $c_1 \neq 0$  and  $c_2 \neq 0$ ,

$$F(x, y, z) = c_0 (x - x_0)^2 + c_1 (y - y_0)^2 + c_2 (z - z_0)^2 + \kappa = 0$$

where

$$\begin{aligned} x_0 &= -\frac{c_6}{2c_0} & y_0 &= -\frac{c_7}{2c_1} & z_0 &= -\frac{c_8}{2c_2} \\ \kappa &= c_9 - c_0 x_0^2 - c_1 y_0^2 - c_2 z_0^2 \end{aligned}$$

2. If  $c_0 \neq 0$  and  $c_1 \neq 0$  and  $c_2 = 0$ ,

$$F(x, y, z) = c_0 (x - x_0)^2 + c_1 (y - y_0)^2 + c_8 z + \kappa = 0$$

where

$$\begin{aligned} x_0 &= -\frac{c_6}{2c_0} & y_0 &= -\frac{c_7}{2c_1} & z_0 &= 0 \\ \kappa &= c_9 - c_0 x_0^2 - c_1 y_0^2 \end{aligned}$$

3. If  $c_0 \neq 0$  and  $c_1 = 0$  and  $c_2 = 0$ ,

$$F(x, y, z) = c_0(x - x_0)^2 + c_7y + c_8z + \kappa = 0$$

where

$$\begin{aligned}x_0 &= -\frac{c_6}{2c_0} & y_0 &= 0 & z_0 &= 0 \\ \kappa &= c_9 - c_0x_0^2\end{aligned}$$

When  $\kappa \neq 0$ ,  $F(x, y, z) = 0$  may be divided across by  $|\kappa|$ .

Once we find  $\{x_0, y_0, z_0\}$  in the basis of eigenvectors, we can convert these coordinates back to the reference coordinate system using

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = [Q] \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}.$$

The output of the quadric fit tool corresponds to the following values:

Quadric Type	Best guess to cases in <a href="https://en.wikipedia.org/wiki/Quadric">https://en.wikipedia.org/wiki/Quadric</a>
Center	$X_0, Y_0, Z_0$
A	max eigenvalue, $c_0 (\kappa = 0)$ or $c_0 /  \kappa  (\kappa \neq 0)$
B	mid eigenvalue, $c_1 (\kappa = 0)$ or $c_1 /  \kappa  (\kappa \neq 0)$
C	min eigenvalue, $c_2 (\kappa = 0)$ or $c_2 /  \kappa  (\kappa \neq 0)$
U	should be 0 for cases 1, 2, 3
V	should be 0 for cases 1 and 2, and $c_7 (\kappa = 0)$ or $c_7 /  \kappa  (\kappa \neq 0)$ for case 3
W	should be 0 for case 1, and $c_8 (\kappa = 0)$ or $c_8 /  \kappa  (\kappa \neq 0)$ for cases 2 and 3
c	evaluated as $\kappa$ or $\kappa /  \kappa $ , should be 0, 1, or -1
Axis 1	normalized eigenvector $v_1$ for max eigenvalue
Axis 2	normalized eigenvector $v_2$ for mid eigenvalue
Axis 3	normalized eigenvector $v_3$ for min eigenvalue

**ICP Registration** This tool implements the Iterative Closest Point registration method, which can be used to align two objects assuming an affine transformation.

## Image Map

This tool can be used to extract a scalar data map from a 3D image data. Before you can use this tool, you must import a 3D image (menu **File** → **Import Image**). Then, give the new data map a name and select the image model. The following options can be set to control how the data is extracted from the image.

- **Method:** The sampling method that will be used to extract the image data.
- **Project surface nodes inward:** When checked, surface nodes are evaluated by finding the closest non-zero image data, instead of evaluating the image data directly at the surface.
- **Normalize:** image data is normalized before sampling.
- **Filter:** When checked, the user can provide a transfer function that will be applied after sampling the image data.

**Discrete Element Network** This tool converts a surface mesh to a mesh composed of discrete elements.

### Select Near Plane

This tool can be used to select all nodes that are on or close to a plane defined by the following parameters.

- **Direction:** The orientation of the plane
- **Position:** The position (i.e. normal displacement) of the plane.
- **Threshold:** The max distance to the plane for a node to be included in the selection.
- **Add to selection:** Add the plane selection to the current selection, if any.

**Kinemat** The Kinemat tool allows users to setup an FEBio problem using only geometry and kinematics data. The geometry is assumed to be rigid bodies, and the kinemat data describes the motion of the rigid parts. FEBio Studio will read the data and construct a complete FEBio model: Rigid materials are assigned to each bone. The kinemat data is used to apply prescribed displacements and rotations for each rigid body. After import, the model can be saved and run immediately in FEBio.

## 3.10 The Repository Panel

This panel provides access to the online FEBio model repository. This repository is a way for FEBio users to easily access models created by other users, and to share models of their own from within FEBio Studio itself. The repository is not meant to be used as cloud storage for in-progress models, but as a means of sharing well-built, thoroughly-tested models with the FEBio community.

### 3.10.1 Connecting to the Repository

FEBio Studio does not connect to the repository automatically when launched. In order to connect to the repository, simply select the *Repository Panel*, and click the *Connect* button. If this is your first time connecting to the repository, a dialog will appear asking you to choose a location for FEBio Studio to store repository files (Figure 3.7). This location will be used to store a database containing metadata about the projects stored on the repository, along with any files that you download from the repository.



Figure 3.7: This dialog allows the user to set the location of the model repository files.

This setting can be changed at any time in the options (Section 3.15.13).

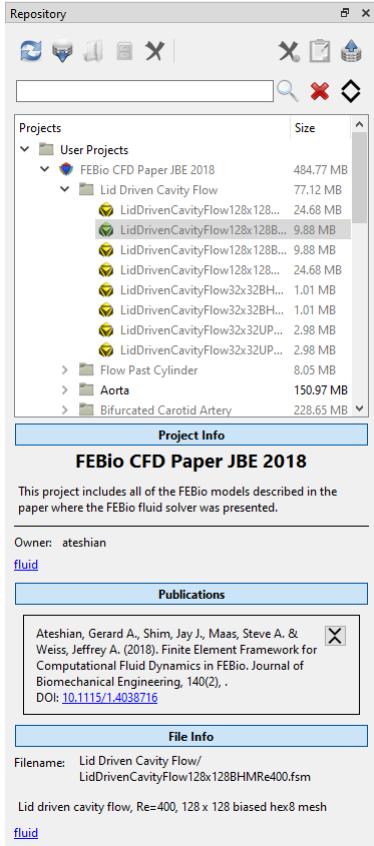


Figure 3.8: The Repository Panel allows you to browse and download files from the repository.

### 3.10.2 Browsing and Downloading

Once you have connected to the repository, you will be presented with the *Repository Panel* as seen in Figure 3.8.

The *Repository Panel* shows a list of all projects available on the repository organized into a tree structure. The top-level items are categories of projects, the second-level items are individual projects that are available for download, and all sub-items are part of a given project's file structure containing any arrangement of folders and files. If a copy of a project, file, or folder currently exists on your machine the name the item will be black, otherwise it will appear gray.

When a project is selected, information about the project will appear below the list of projects, including: the project name, the project's description, the owner of the project, and any tags associated with the project. If there are any publications associated with a project, they will be listed below the project info. Additionally, if a file from a project is selected, information about the selected file will be displayed, including: the file name, a description of the file (if available), and any tags associated with the file.

At the top of the *Repository Panel*, there are several buttons that allow for access to the panel's functions:



The *Refresh* button will connect to the model repository server, and re-download the project tree and metadata, effectively refreshing the panel.



The *Download* button download the currently selected item, whether that be an entire

project or an individual file.



The *Open* button will open the currently selected project or file. This button will only be enabled if the currently selected item has already been downloaded.



The *Open Containing Folder* button will open the folder containing the currently selected item in your system's default file browser. This button will only be enabled if the currently selected item has already been downloaded.



The *Delete* button will delete the local copy of the currently selected item from your file system. This button will only be enabled if the currently selected item has already been downloaded.

The functionality provided by these buttons can also be accessed by right-clicking an item in the project list, and selecting the appropriate item from the menu. Additionally, double-clicking a project or file will download the item, and automatically open the file when the download has finished. If the item has already been downloaded, double-clicking the item will open the local copy.

Below the panel's main toolbar is a second toolbar used to search the repository. To search the repository, enter one or more search terms into the text box, and either click the *Search* button next to the bar or hit return. All items not matching your search terms will be removed from the project tree.



The *Search* button removes all items from the project tree that do not match the search terms in the search box.



The *Clear* button clears the search box, and restores all items to the project tree.



The *Expand* button shows the advanced search options. When expanded, additional search fields are shown. Most of these advanced fields are drop-down boxes that allow you to select specific search terms for a particular type of model component. For example, clicking on the material field, will bring up a list of materials that you can select from. After clicking the search button, all files in the repository that use that particular material will be shown.

### 3.10.3 Managing Your Uploads

In order to upload projects to the repository, or to manage projects that you have already uploaded, you must first log in to the repository by clicking the *Upload* button, and entering your febio.org username and password into the dialog that appears.



If you are not currently logged in to the model repository, and you click this button, you will be prompted to log in using your febio.org username and password, in order to manage your own uploads. Once you have logged in, the repository will refresh, and a new category of projects will appear in the project list, called *My Projects*. Any projects owned by your account will now appear in this category.

If you do not have an febio.org account, you may sign up for one for free by visiting <https://febio.org/register/>.

If you currently have permission to upload projects to the repository, the *Upload* button will open the *Upload/Modify Wizard* (see Section 3.10.5) to allow you to upload a new project. If you do not currently have permission to upload projects, the *Upload* button will open the *Upload Permission Request Dialog* (see Section 3.10.4).



The *Delete From Repository* button will permanently delete the currently selected project from the model repository. This cannot be undone.

In order to upload to the repository, you must first request permission. The repository is not meant to be used as cloud storage for in-progress models, but as a means of sharing well-built, thoroughly-tested models with the FEBio community.

Please fill out this form, and our team will notify you when your account has been given permission to upload.

Email Address:

Organization: (e.g. University of Utah)

Why do you want to upload projects to the repository?

Figure 3.9: This dialog will allow you to request permission to upload to the repository.

 The *Modify Project* button will allow you to edit the currently selected project. It will open the *Upload/Modify Dialog* (see Section 3.10.5).

#### 3.10.4 The Upload Permission Request Dialog

The repository is meant to be a means whereby users may share well-built, thoroughly-tested models with the FEBio community. In order to ensure the quality of uploads to the repository, users are not given permission to upload projects by default, but must request it. If you do not currently have permission to upload, you may request permission by logging in to the repository and clicking the *Upload* button. Doing so will cause the *Upload Permission Request Dialog* to appear.

The fields in this dialog are meant to help our team determine if your account should be given permission to upload, so please be thorough in your responses. Once you fill out these fields and click *OK*, the request will sent to our team. You should receive an email at the specified address once you have been given permission to upload.

Please note that this is not an automated process. As such, allow for 1-2 business days for the request to be processed. If you do not receive a response in that time, please resend the request in the same manner.

#### 3.10.5 The Upload/Modify Wizard

The repository is meant to be a means whereby users may share well-built, thoroughly-tested models with the FEBio community. If you have a model or set of models that you believe meets these requirements, you may upload them as part of a project to the repository using this wizard.

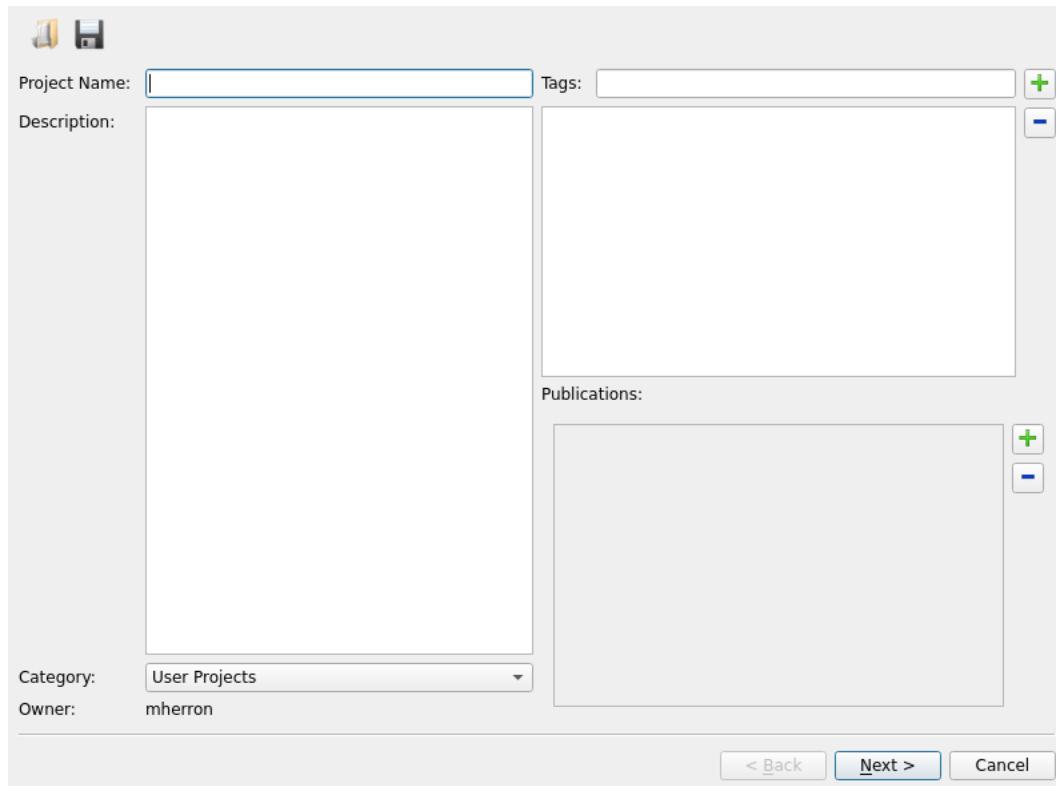


Figure 3.10: The first page of the Upload/Modify Wizard

### 3.10.5.1 Adding or Editing Project Details

The buttons on the top of this page provide access to various functions:



The *Save* button allows you to save all of the information in this dialog to a JSON file. This is useful when you are working on creating a large project that may need to be worked on across multiple sessions, or if you are simply worried about losing this information before the upload is complete.



The *Open* button allows you to load a previously saved JSON file that contains information about a project that you are preparing to upload. Please note that all information currently entered in the wizard will be cleared before the wizard loads the information in the file.

Each of the fields on the first page of the wizard must be filled out with appropriate information. If you are modifying an existing project, these fields will be filled out with the current project information.

- *Project Name*: Provide a unique, descriptive name for your project.
- *Description*: Provide a description of your project as a whole (descriptions of individual files can be given on the second page of the wizard). Please be thorough, providing at least a few sentences. A user should be able to understand the purpose of your project simply by reading the description.
- *Category*: By default, users are only given permission to upload to the User Projects category. Unless you have been given special permission, this field will not be editable.

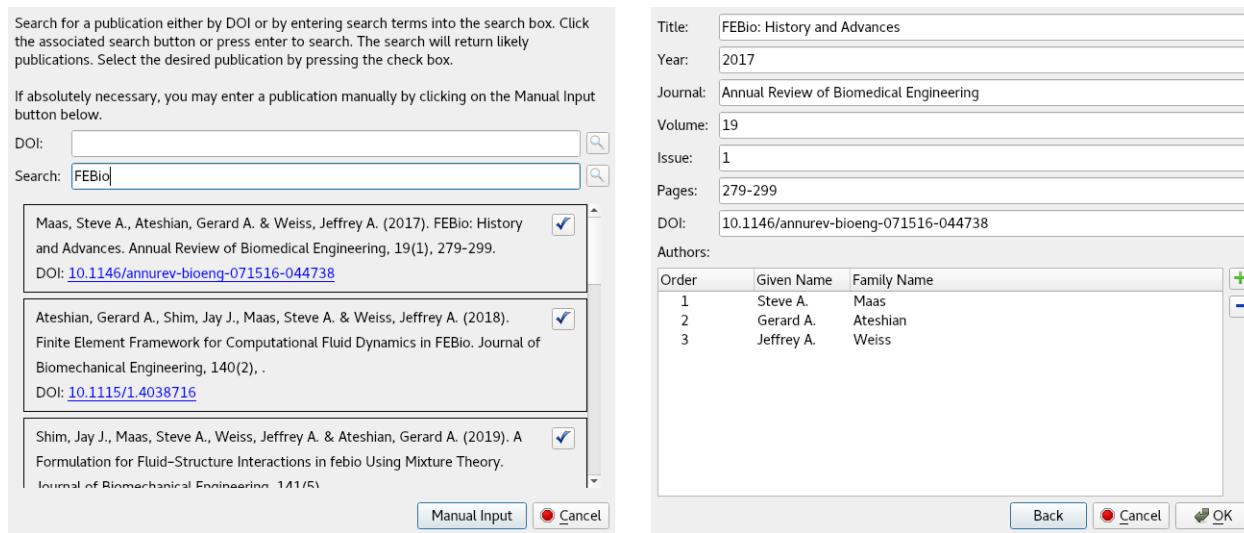


Figure 3.11: First and second pages of the dialog.

- **Owner:** Your username. This field is not editable.
- **Tags:** Provide at least one, but preferably several, tags to help categorize your project.
  - To add a tag, type into the field next to the *Tag* label, and click the plus button. This will add the tag to the box below. Suggestions for tags will appear in a drop-down menu as you begin to type.
  - To remove a tag, select it from the list in the box, and click the minus button.
- **Publications:** If one or more of your models is associated with one or more published papers, you may add details about these publications to your project.
  - To add a publication, click the plus button next to the box under the *Publications* label. This will open the *Add Publication Dialog* (see Section 3.10.5.2). Once added, the publication will appear in the box.
  - To remove a publication, check the checkbox next to the publication and click the minus button located next to the box.

### 3.10.5.2 The Add Publication Dialog

This dialog allows you to search for publications and add them to your project. You may either search by DOI (the recommended approach), or by search term. Once the dialog has found possible candidates based on your search criteria, it will display them in a list below the search boxes. To select one, click the blue checkbox button. This will take you to the second page of the dialog. On the second page you may review and edit the fields as necessary.

If absolutely necessary, you may enter all of the information of a publication manually by clicking the *Manual Input* button at the bottom of the dialog. Doing so will take you to the second page where you can manually enter the publication information.

- To add an author, click the plus button to the right of the list of authors.
- To remove an author, select the author(s) that you wish to remove and click the minus button.

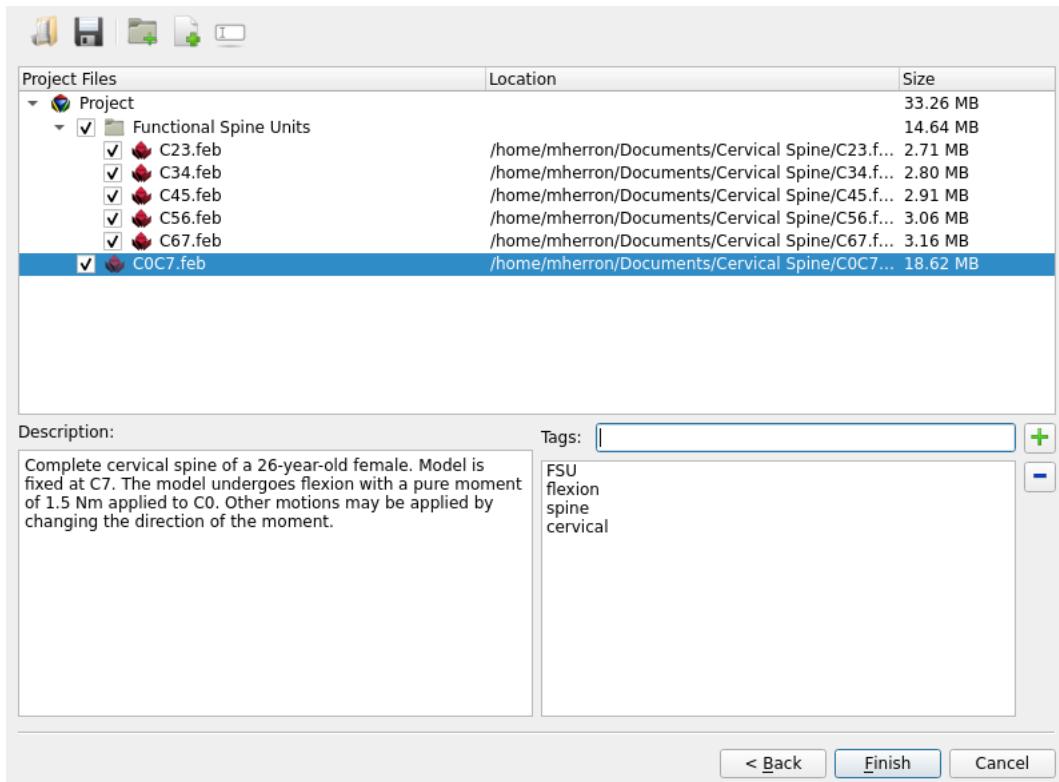


Figure 3.12: The first page of the Upload/Modify Wizard

- To rename an author, double-click on either their given or family name.
- To reorder the authors, drag and drop the authors. The *Order* field will be updated automatically.

Once all of the information on the second page looks correct, click the *OK* button.

### 3.10.5.3 Adding or Removing Files

The second page of the wizard is used to add or remove files, and add or edit metadata for them. If you are modifying an existing project, the project's current file structure will be displayed here.

The first column of the file tree contains name of the file or folder as it will appear in your project. The second column shows the location of the file. If the file is to be newly added to a project, its location on your system appears in this column. If the file is part of an existing project that you are modifying, its location relative to the model repository will be displayed in this column, starting with the text “{*Repository*}”. The third column shows the size of the file or folder.

You may organize the files in your project however you would like, with any arrangement of files, folders, and subfolders. It is best practice to avoid unnecessary subfolders, and to avoid extensive subfolder trees. Please note that empty folders will not be uploaded.

The buttons on the top of this page provide access to various functions:



The *Add Folder* button adds a new folder to your project. Upon adding a new folder, the name folder name will become editable. The folder will be added at the level of the currently selected file, or as a subfolder of the currently selected folder.



The *Add File* button opens a system file dialog which will allow you to select any number of files from your system to include in your project. The files will be added to the file tree at the level of the currently selected file, or inside of the currently selected folder.



The *Rename* button allows you to rename the currently selected file or folder. You can also rename a file or folder by double-clicking on the desired item.



The *Replace File* button only appears when you are modifying an existing project. It will allow you to replace a currently selected file that is already on the repository with another one on your machine. It will copy the name, description, and tags from the currently selected file to the newly added file. The primary purpose of this button is to allow you to easily replace an existing file with a more up-to-date version of that file.

Other functions of this page are described below:

- To reorder files or folders, drag and drop the files or folders to the desired location.
- To exclude previously added files or folders from the upload, uncheck the check box to the left of the file or folder names.
- To remove a file or folder from an existing project that you are modifying, uncheck the check box to the left of the file or folder names. Please note that unchecking a folder, will uncheck its contents, and will result in their deletion. If you simply want to move files out of a folder, then you must drag them out of the folder.
- To add or edit a file's description, select one or more files and then type a description into the *Description* box. The description of all selected files will be changed to what you type. Folders cannot be given descriptions.
- To add tags to a file, select one or more files, type a tag into the box next to the *Tag* label, then click the plus icon. This tag will be added to all selected files. Folders cannot be assigned tags.
- To remove tags from a file, select one or more files, select the tags you wish to remove from the box below the *Tag* label, then click the minus button. Only tags from the most recently selected file will appear in this list.

When you are happy with the state of your file structure, and your files have been given descriptions and tags, click the *Finish* button to finalize your project and initiate the upload.

### 3.11 The Project Panel

FEBio Studio offers a multi-document interface, where at any given time, multiple FEBio models, plot files, and other documents, can be open. Typically, all these open documents have some relation to each other. For instance, a single FEBio model and multiple plot files can be open at the same time, where each plot file shows the results of a parameter study. To help manage and keep track of documents that somehow belong together, the user can create an FEBio Studio project.

FEBio Studio will always maintain an active project in the background that keeps tracks of all the files that are currently open and have been opened in the current FEBio Studio session. The status of this project can be inspected on the Project panel. The Project panel is divided into two

panes. The top pane shows a tree view of the current files in the project. At the bottom of the Project panel is a Notes pane. Here, users can add notes and comments to the currently selected item in the project tree.

The project tree is split into two main branches:

**Open Files** shows all the files that are currently open in FEBio Studio. Double-clicking a file in the Open Files list will activate the corresponding tab.

**Project** shows all the files in the project.

Users can further customize the project in several ways. Right-clicking on the PROJECT item allows the user to do the following operations.

**Save Project as** Save the project to a file. (More on this below)

**Create Group** Create a new group (a new branch in the project tree) that can be used to group certain files together.

**Close Project** Close the current project and start a new empty project.

**Clear Project** Clears the current project. This may lead to loss of data since projects are automatically saved. If the current project is stored as a file, this will overwrite the file with an empty project.

**Add File** Add a file to the project. Note that any file can be added. It doesn't have to be a file that can actually be opened in FEBio Studio. For instance, you can keep a Word document in the project that contains additional project details.

**Import Folder** Adds files from a user-specified folder to the project. Only files with the following extensions will be imported: *fsm*, *fs2*, *feb*, *xplt*.

The active project can be saved to a file using either the menu **File** → **Save Project as ...** or by right-clicking the PROJECT item on the Project panel. Once a project is saved, FEBio Studio will automatically ensure that any changes to the project (e.g. by loading files) will be reflected in the file. There is no need to save the project again. The project file itself is a standard xml file and can be viewed and edited with any text editor.

## 3.12 The Curve Editor

The *Curve Editor* is accessed either from the **Tools** → **Curve Editor** menu or by pressing the corresponding button on the toolbar. It can also be accessed using the *F9* shortcut. The *Curve Editor* gives an overview of all the time-dependent parameters in the model. FEBioStudio allows the user to define the time dependency explicitly through the use of so-called *load controllers*. In an FEBio analysis, the load controller is evaluated at the simulated time and usually multiplies the value of the associated model parameter. The Curve Editor allows user to create load controllers, associate them with one or more model parameters, and edit the load controllers.

In the left panel, the model tree is presented, but only parameters that can be associated with a load controller will be listed here. Select a parameter here in order to see the load controller, if any, currently associated with this parameter. At the top of this panel, a Filter selection is available that allows you to show the parameters of only a specific type of model component, e.g. boundary conditions, material parameters, etc. This filter can be helpful in finding a specific parameter.

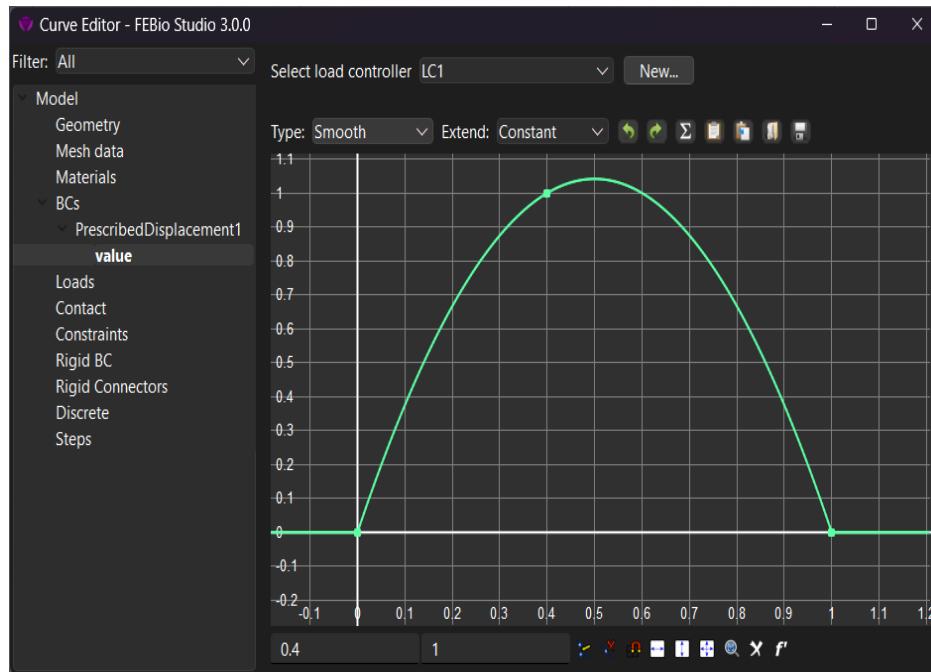


Figure 3.13: The curve editor shows all the load curves used in the model.

The right panel shows the current load controller associated with the parameter selected in the left panel. At the top, the load controller can be selected that will be associated with this parameter. Clicking the *New* button allows you to add a new load controller and associate it with the parameter. Depending on the type of load controller selected, either the controller's properties or a graphical representation of the load controller will be shown in the right panel.

One particularly useful type of load controller is the *load curve*, which consists of *time,value* pairs that will be interpolated. FEBio Studio often associates a default load curve to certain parameters of boundary conditions and loads. This default load curve ramps up the value from zero to one over the unit time interval. However, all curves can be modified using the tools available at the bottom of the Curve Editor. The curve's data points are represented as dots on the view. These data points can be selected by clicking on them with the left mouse button. They can also be moved by click+dragging them. The current (time, value) pair of the selected point is displayed on the toolbar at the bottom of the view. This toolbar offers the following features.

When this button is toggled, each click on the curve view will add a new node. Note that you don't click this button to add nodes. Nodes can also be added by shift+click with the left mouse button on the curve view.

This button will delete the node that is currently selected in the curve view.

The snap-to-grid option will allow you to move a node on the intersections of the grid lines.

Zooms the curve view out so that all nodes are visible within the bounds of the curve view.

The toolbar at the top of the Curve Editor provides additional tools for modifying the active curve.

This will load data from a file. The file must be a simple text file with one line of data for

each point. On each line, specify the time-load value pair delimited by a space. You can enter as many lines as you want.



Save the active load curve to a text file.



Copy the curve data to the clipboard. This allows the curve data to be pasted in another application that supports clipboard operations.



Store the load curve data so it can be pasted to another curve.



Paste the curve data that was copied to the active load curve.



Undo the last change to the active load curve.



Redo the last change that was undone.



Open the equation editor where load curve data can be generated via a mathematical equation (see below).

The current view can be zoomed in or out, either by using the zoom buttons at the bottom of the view, or using the mouse wheel. When you scroll the mouse wheel while hovering over one of the axes, the graph will only zoom in that axis' direction. Depressing the left mouse button while moving the mouse, pans the view.

At the top of the curve view, you can see two drop-down lists. The first list allows the user to set the curve type which defines the interpolation mode for the currently displayed curve. The choices are as follows:

- **linear**: use a linear interpolation between the curve points
- **step**: use a constant interpolation between the curve points. The value of the curve is defined by the value of the point closest and to the right of a particular ordinate.
- **smooth**: A cubic polynomial is fitted through the data points resulting in a smooth interpolation.
- **cubic spline**: A cubic spline is fitted through the data points.
- **control points**: The data points define the control points of a spline.
- **Approximate**: An approximation of the control points interpolation.
- **smooth step**: A cubic polynomial is fitted between each pairs of data points. The slope at the end points is zero.

The second list displays the *extend mode* options. The extend mode defines the value of the curve outside its defined range. The choices are:

- **Constant**: the value is clamped to the range of the curve as defined by the first and last point.
- **Extrapolate**: the value is extrapolated from the end-points of the curve.

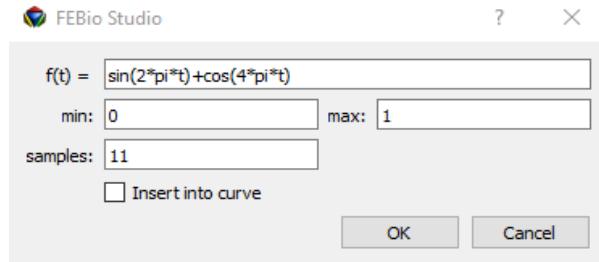


Figure 3.14: The equation editor allows user to easily generate load curve data via a mathematical expression

- **Repeat:** the curve is repeated on either end of the curve's domain
- **Repeat offset:** same as repeat, except that the curve is offset by the end-point values.

If the previous tools are not sufficient to describe the evolution of the load curve in detail, the *Equation Editor* can be used (Figure 3.14). This tool is accessed from the toolbar and allows the user to enter a mathematical function of time. Use the symbol *t* to reference time. This function will be evaluated and discretized to generate a set of points that interpolate the function approximately.

### 3.13 The Mesh Inspector

The Mesh Inspector window displays some statistics of the mesh of the currently active object (Figure 3.15). It shows, for instance, what elements the mesh is composed of and the user can inspect element metrics such as element volume, jacobian, etc.

The mesh info pane at the top lists the total number of nodes, faces, and elements in the current mesh. It also shows a list of all the different element types and their count.

The variable pane allows the user to select the quality metric that is displayed in the bar chart. To limit the plot only to a certain type of element, select the corresponding element in the element list of the mesh pane.

The statistics pane shows the min, max and average values of the selected variable. The selection pane can be used to select elements in a certain range of the currently plotted variable. This can be useful to identify and quickly select elements of interest.

### 3.14 The Measure Tool

The *Measure Tool* can be accessed from the main toolbar and offers various tools for measuring distances, angles, areas, and more.

The following is a list of the available on the Measure Tool. When selected a property list will be displayed where the user can enter additional information required for the selected tool.

- **Point Distance:** This tool can be used to evaluate the distance between two selected nodes. To use it, select two nodes on the mesh. The tool will display the coordinate differences, the distance between the points (length), and the stretch, which is the ratio of the current distance over the distance at the first time point.

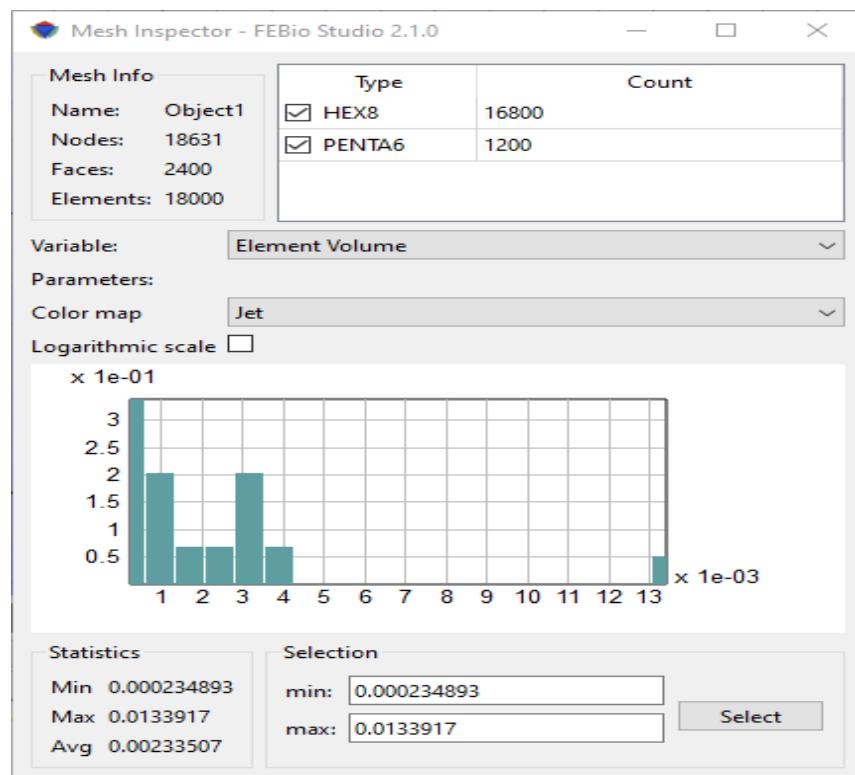


Figure 3.15: The Mesh Inspector tool allows the user to inspect certain quality measure of the mesh.

- **3 Point Angle:** This tool measures the angle between the two lines defined by three nodes. Select three nodes,  $a$ ,  $b$ , and  $c$ . The angle between the lines formed by nodes  $(a,b)$  and  $(b,c)$  is calculated.
- **4 Point angle:** Measures the angle between the lines defined by four nodes. Select four nodes,  $a$ ,  $b$ ,  $c$ , and  $d$ . The angle between the lines formed by nodes  $(a,b)$ , and  $(c,d)$  is calculated.
- **Measure Area:** This tool calculates the area from a selection of faces. To use it, first select some faces, then click the “Apply” button. The tool will show the number of selected faces and the total area.
- **Element Volume:** Calculates the volume of the selected elements.
- **Sphere Fit:** This tool can be used to fit a sphere to the model. If the “selection only” is checked, the sphere will be fit only to the selection, otherwise all the nodes on the surface of the model will be used. Press “Fit” to calculate the best fitting sphere. The location of the center, the radius, and the objective value (which measures the average deviation of the sphere), will be calculated.
- **Tet Overlap:** Calculates the number of tetrahedral elements that overlap.
- **Surface Volume:** Calculates the volume bound by an enclosed surface.
- **Plane normal:** Calculates the normal of the plane defined by three selected nodes.
- **Pt. Congruency:** Calculates the congruency of a node.

## 3.15 FEBio Studio Settings

The FEBio Studio Settings dialog can be opened from the **Tools** → **Settings** menu or by pressing the F12 shortcut button. Different categories can be selected by clicking the tab buttons. By selecting a category a list of options is shown on the right hand side of the dialog box. The categories are:

- *Background*: change the background settings such as color, gradient, etc.
- *Camera*: Set options that control the motion of the camera when switching between views.
- *Colormap*: Edits default color maps or create new ones.
- *Display*: Shows options that affect how objects and meshes are rendered in the Graphics View.
- *FEBio*: Sets default FEBio configuration file and the paths for generating FEBio plugins.
- *Lighting*: Set the lighting conditions for rendering the Graphics View
- *Palette*: Set the default color palette.
- *Physics*: Allows the user to select options that draw physics related items in the Graphics View.

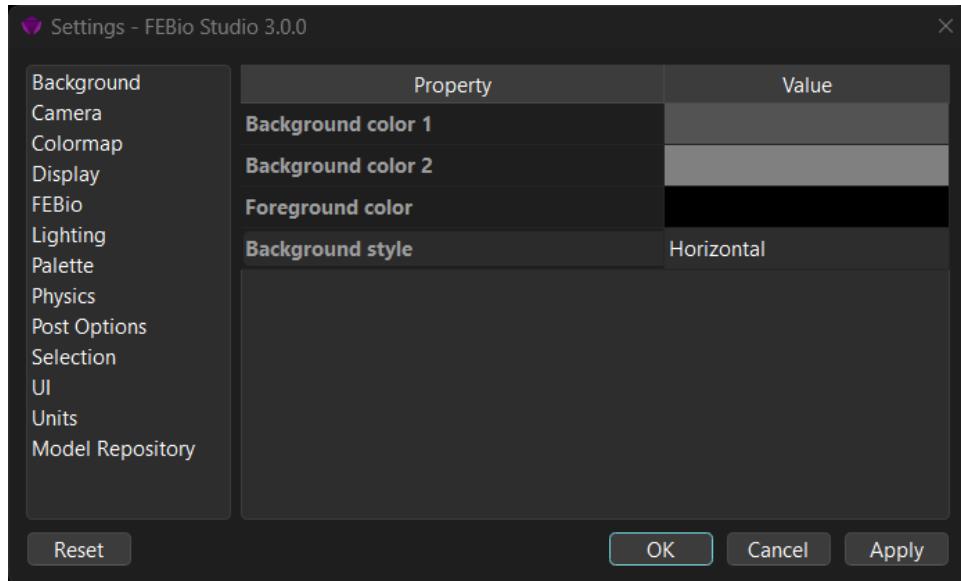


Figure 3.16: The FEBio Studio Settings dialog allows users to customize many aspects of the FEBio Studio environment.

- *Post Options*: Set special settings only used in the Post configuration.
- *Selection*: Options that affect the selection of nodes, elements, etc.
- *UI*: Shows options that affect how users interact with some UI components.
- *Units*: Set the unit system options.
- *Model repository*: set options that affect the interaction between FEBio Studio and the online FEBio model repository.
- *Auto update*: Options to update your installation to the latest release or development version.

The buttons on the bottom of the dialog allow users to accept or reject the changes. The **Reset** button will restore all default settings. (It will not clear the recent file lists.) Click **OK** to accept all new changes and close. Click **Cancel** to close the dialog without accepting changes. Click **Apply** to apply the changes without closing the dialog box.

### 3.15.1 Background Settings

The background settings set how the background of the Graphics View is rendered.

- **Background color 1**: Sets the Graphic's View background color 1.
- **Background color 2**: Sets the Graphic's View background color 2
- **Foreground color**: Sets the Graphic's View foreground color
- **Background style**: Sets the style of the Graphic's View background.

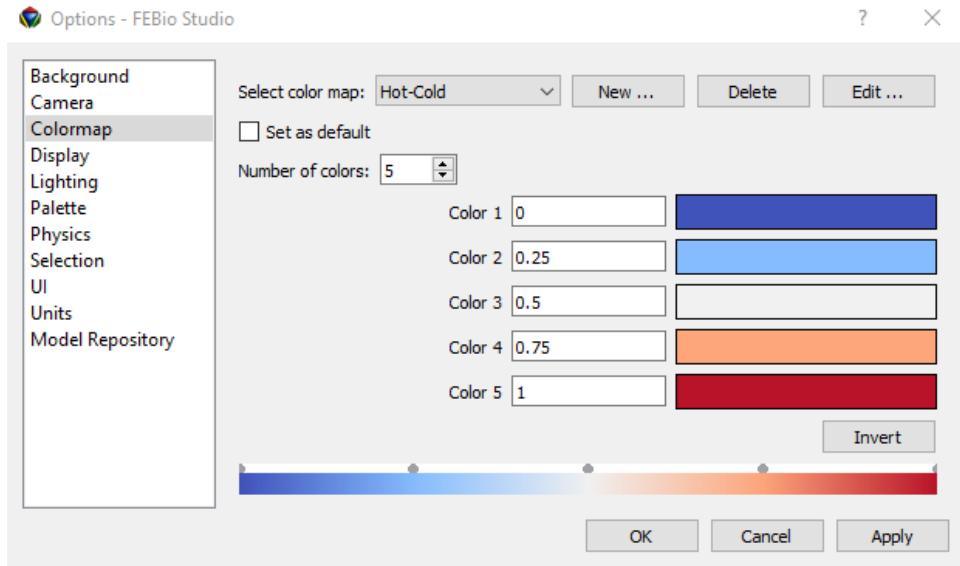


Figure 3.17: The Colormap settings allow users to edit the colormaps that are used for rendering plots in the Graphics View.

### 3.15.2 Camera Settings

These settings control how camera motion is animated when the user switches between views (e.g. when zooming in on an object)

- **Animate camera:** Chooses whether the camera should be animated when switching between views.
- **Animation speed:** Sets the speed of the camera when switching between views. The value must be between 0 and 1.
- **Animation bias:** Controls the initial acceleration and deceleration of the camera's motion. The value is between 0 and 1.

### 3.15.3 Colormap Settings

FEBio Studio uses a palette of colormaps for rendering plots (contour plot, isosurface, slice, streamlines, etc.). This palette can be edited in this tab of the dialog. Users can modify the colormaps or create new colormaps. New colormaps are automatically saved and reloaded between FEBio Studio sessions.

### 3.15.4 Display Settings

The display settings affect some of the aspects of how the model is displayed on the screen. The following options can be modified:

- **Size of nodes:** Sets the size (in pixels) of point rendering (e.g. nodes, tags).
- **Line width:** Sets the thickness (in pixels) for rendering lines.

- **Mesh color:** sets the mesh color for rendering the mesh.
- **Show normals:** Show a line on each surface facet that indicates the orientation of the local normal.
- **Normals scale factor:** Sets the scale factor that is used for rendering the normals.
- **Enable highlighting:** When enabled, geometry items are highlighted when hovering the mouse over them. The item that is highlighted depends on the current selection context.
- **Identify backfacing faces:** When enabled, the backs of surfaces are rendered in a different color than the front side. This can be useful to distinguish between the fronts and backs of surfaces, especially in models that use shells.
- **Multiview Projection:** sets the multiview projection option
- **Object transparency mode:** Selects under what conditions objects will be rendered as transparent.
- **Default Text Color Option:** Chooses how the default text color for Graphic View widgets is determined.
  - **Theme:** The UI theme determines the color. For light modes, it will be black; for dark modes, it will be white.
  - **Custom:** Choose a custom text color.
- **Custom Text Color:** This is the color that will be used when the user chooses the *Custom* option for the *default text color option* setting.
- **Tag Font Size:** Set the font size used for rendering the labels on tags in the Graphics View.

### 3.15.5 FEBio Settings

- **Load FEBio config file on startup:** Sets whether the FEBio configuration file is loaded when FEBio Studio starts. This may be necessary for running FEBio using the default configuration.
- **Config file:** Set the location for the default FEBio configuration file.
- **FEBio SDK Include path:** Set the location of the FEBio SDK Include file path.
- **FEBio SDK Library path:** Set the location of the FEBio SDK Library file path
- **FEBio create plugin path:** Set the default location for creating FEBio plugins using the FEBio Plugin Creation Wizard.

### 3.15.6 Lighting Settings

Lighting options affect how the model is lit in the Graphics View.

- **Enable Lighting:** toggle lighting on or off. When lighting is off, parts are rendered without any shading effects.

- **Diffuse Intensity:** The diffuse intensity of the light source.
- **Ambient Intensity:** The ambient intensity of the light source.
- **Light direction:** The direction vector that points to the light source.
- **Environment map:** Sets the location of an image file that will be used as the environment map. Images need to be either JPG or PNG, and for best results, should be high-quality HDR images, created specifically for use as an environment map. (Changing this requires a restart of FEBio Studio.)
- **Use Environment Map:** Determines whether the environment map is used when rendering objects in the Graphics View. Requires a valid environment map image to work properly.

### 3.15.7 Palette Settings

The palette defines the default colors that are assigned to different parts. Different palettes can be chosen, as well as loaded, or saved. A custom palette can be created from the current model.

- **Current palette:** Select the palette to use for new models. Changing this does not affect the currently loaded model. To apply the selected palette to the current model, press the *Apply palette to materials* button.
- **Load palette:** This button loads a palette from a file. The file is an xml-formatted file with *PostViewResource* as the root tag. You can define multiple palettes, each starting with the *Palette* tag. This tag requires a *name* attribute. Then, define a *color* child tag with the RGB values of the color.
- **Save palette:** Save the currently selected palette to an external file.
- **Create palette from materials:** Create a palette from the materials of the currently loaded model.
- **Apply palette to materials:** Apply the currently selected palette to the materials of the current model. If the model contains more materials than are defined in the palette, the palette colors are recycled.

### 3.15.8 Physics Settings

The Physics options control how certain model components are displayed in the Graphics View.

- **Show rigid joints:** Sets whether or not to show glyphs representing the rigid joints in the model.
- **Show rigid wall:** Sets whether or not to show rigid walls.
- **Show material fibers:** Sets whether or not to show material fibers on the selected object.
- **Fiber scale factor:** Sets the scale factor when rendering the material fibers.
- **Show material axes:** Sets whether or not to show the material axes.
- **Show fibers/axes on hidden parts:** Whether or not to show material fibers or axes on the hidden parts of the currently selected object.

### 3.15.9 Post Settings

These are settings that only apply to the Post environment.

- **Default colormap range:** Sets the default range option that will be used by the colormap item in the Post model tree.

### 3.15.10 Selection Settings

The selection options affect what mesh items (elements, faces, edges) can be selected in the Graphics View.

- **Select Connected:** Sets whether connected items will be selected simultaneously when a single item is selected.
- **Tag info:** Sets what information is displayed on a selection.
- **Ignore Backfacing Items:** Sets whether items that are facing away from the user can be selected.
- **Ignore Interior Items:** Sets whether items on the inside of the mesh can be selected.
- **Respect partitions:** Sets whether “select connected” option will not cross the mesh’ partitions.

### 3.15.11 UI Settings

The UI options set various options that affect the UI.

- **Emulate apply action:** Emulates apply action
- **Clear undo stack on save:** When checked, the undo stack is cleared when saving the model.
- **Recent file list:** This button clears the recent file lists that are shown on the Welcome page and in the various *Open Recent File* menu options.
- **Autosave interval:** The time interval (in seconds) between auto-saves of the model.

### 3.15.12 Units Settings

The units options control how units are displayed in FEBio Studio.

- **Change for:** Selects whether the changes affect the current model, all models, or new models.
- **Unit system:** Selects the unit system that will be used. If a unit is selected, a list of the base units, as well as some derived units will be shown.

### 3.15.13 Model Repository Settings

The model repository options control how FEBio Studio interacts with the online FEBio model repository.

- **Repository Folder:** Sets the location where a local copy of the model repository is saved. If empty, you will be prompted to set a location upon connecting to the model repository.

### 3.15.14 Auto Update Settings

FEBio Studio will automatically check for a software update when it is launched. This section of the options dialog will allow you to manually check for an update.

- **Update to Latest Release:** There are periodic, official releases of FEBio Studio and FEBio. This button will initiate a check to see if there are any updates available.
- **Update to Development Version:** FEBio Studio and FEBio are in almost daily development, and a new development version of the software is made available each night. This button will initiate a check to see if there are any updates available to the latest development version. Please note that while these versions contain the latest bug fixes and features, they are potentially unstable.

If a new update is available, click the “Close and Update” button on the update dialog to close FEBio Studio and launch the updater.

## Chapter 4

# Creating, Loading, and Saving Models

### 4.1 Starting a new model

A new model can be started from the **File → New Model...** menu or by clicking the corresponding toolbar button. This will open the New Model dialog box. This dialog presents the model templates that are available. A model template is a collection of modules that will be activated in a model. A module corresponds to certain physics modeling features or UI components. Model templates and modules greatly simplify the process of creating a FE model by presenting only relevant features to the user.

Choosing Cancel on this dialog will activate all modules and expose all the physics features and modeling capabilities. You can also change the active modules after starting a new model from the menu **Physics → Edit Physics Modules**.

### 4.2 Loading a model

Model files be loaded from the **File → Open Model File** menu. A standard file open dialog box appears where a file can be selected and opened.

Several formats are supported, including:

- **FEBio Studio Files (fsm, fs2):** This is the native FEBio Studio file format. (See note 1 below)
- **FEBio Input Files (feb):** This is the FEBio Input file format. (see note 2 below)

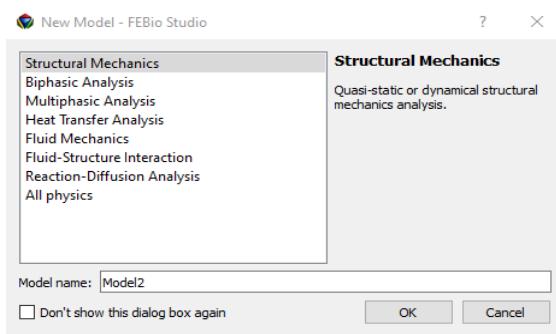


Figure 4.1: The New Model dialog allows users to choose a model template.

- **FEBio Plot Files (xplt):** This is the output format used by FEBio to store results. (see note 3 below)
- **Abaqus Files (inp):** This is the native format used by Abaqus.(Only a limited set of features are currently supported)
- **Nike3D Files (n):** This is the Nike3D input file format.

*Comments:*

1. The FSM file format is the native file format used by FEBio Studio 3. (Previous versions used the FS2 extension.) This format is an extension of the legacy software PreView's file format, which usually had the .prv extension. The prv files can still be read in. The fsprj format was used in the beta version of FEBio Studio before it was replaced with fsm. This extension is considered deprecated but for now is processed as an fsm file.
2. FEBio Studio supports reading in FEBio Input files formats 1.2, 2.0, 2.5, 3.0, and 4.0.
3. When reading in FEBio plot files (.xplt extension), FEBio Studio may add additional plot variables, depending on the contents of the plot file (see Appendix B for more information on these data fields):
  - (a) If a *displacement* field is defined, then FEBio Studio will add the following plot variables: *position*, *initial position*, and *Lagrange strain*.
  - (b) If a *stress* field is defined, then FEBio Studio will add the *pressure* field.
  - (c) The *stress* field is assumed to be the Cauchy stress.

### 4.3 Saving a model

A model can be saved from the **File → Save** or **File → Save as** menus. The former can be used to save the file to the same location as it was loaded. For new models that have not been saved yet, this will execute the **File → Save as** command on the first attempt to save the model. The **File → Save as** menu allows the user to select a file name for storing the model.

# Chapter 5

## Creating and Editing Geometry

FEBio Studio offers several geometry and mesh creation tools. Most of these tools can be found on the Build panel and its sub-panels. The Build panel offers the following sub-panels:

**Create** This panel has tools for creating primitive geometries.

**Edit** The Edit panel is used to change geometry parameters of shape.

**Mesh** The Mesh panel is used to mesh the selected object.

**Tools** Several specialized tools for creating and editing geometry and meshes can be found here.

The following sections describe the features of these panels in more detail.

### 5.1 Creating Geometry

Although FEBioStudio is not designed to be a mesh generator, it has a few features to create simple geometries and meshes. Most of these options are available from the *Create* panel, which is a sub-panel of the *Build* panel. At the top it shows the various creation options.

- *Primitives*: Create simple geometric shapes.
- *Discrete Objects*: Create springs
- *CAD Objects*: Create geometry via CAD-like generation methods.

The most commonly used creation option is *primitives*. When selected, a list of available primitives is shown. When selecting a primitive, its creation parameters are shown. After entering the desired values, click the *Create* button to create the primitive and add it to the model. The new object will be automatically selected in the Graphics View and shown in the Model Viewer.

After a primitive is created, its parameters can be changed in the *Edit* panel.

### 5.2 Importing Geometry

Instead of creating geometry directly in FEBioStudio it is more common to create your geometry in dedicated software and import the mesh into FEBioStudio. FEBioStudio can import several different mesh file formats. Geometry can be imported using the **File → Import Geometry** menu. Appendix A lists an overview of all the supported file formats including the supported features of the particular formats.

## 5.3 Editing Geometry

The *Edit* panel, one of the panels on the Build panel, gives access to the object's creation parameters and editing options.

The other feature to note at the top of both the Edit and Mesh panels is the Object rollout. This shows the name of the active object as well as its type. The type of the object affects what tools will be available for editing and meshing the geometry.

An object can be converted to a different type via the *Convert* button. Two options are available:

- *Editable surface*: An object of this type will have a surface mesh that can be modified. A separate volume mesh can be generated for this object.
- *Editable mesh*: An object of this type will only contain a volume mesh and this mesh can be modified if necessary.

Editable surfaces and editable meshes will be explained in detail in section [5.4](#).

### 5.3.1 Editing Primitives

For primitives the same parameters as when the object was created are shown on the *Edit* panel. This allows you to change the primitives dimensions at any time. If the object is not a primitive, no options will be shown here.

Note that changing the primitive parameters will discard the FE mesh if one was already created. The primitive will have to be meshed again.

### 5.3.2 Editing Editable Surfaces

An *editable surface* is defined via a surface mesh. This surface mesh can be modified via the tools on the *Edit* panel.

Note that changing the surface mesh of an editable surface will discard the FE mesh if one was already created. The object will have to be meshed again.

The sections below discuss each of the available tools in more detail.

#### 5.3.2.1 Auto Partition

The surface of an editable surface can be partitioned (i.e. divided) into several partitions. By default, FEBio Studio will automatically generate a partitioning based on finding the crease edges. If for some reason the resulting partitioning is not correct, the user can re-run the auto-partitioning tool.

The auto-partition tool partitions the geometry into surfaces by identifying the crease edges. A crease edge is an edge between two surface facets whose normals make an angle that is larger than the crease angle. To manually create surface partitions, use the Partition tool instead.

#### 5.3.2.2 Decimate

Decimation is the process of decreasing the number of triangles in the mesh. This tool decimates the triangular surface mesh using an approximate Centroidal Voronoi Diagram based approach.

- **Scale:** This option helps in creating a uniform triangular mesh. Scale value specifies the desired reduction in the total number of nodes in the final decimated mesh. For example a 0.4 value of scale will result in the final decimated mesh with 40% of current number of nodes.

### 5.3.2.3 Edge Collapse

This filter collapses all edges with a length that is below the given threshold value. This can be useful for eliminating small triangles or very narrow triangles, which may improve the quality of the mesh.

### 5.3.2.4 Fix Mesh

This tool provides several options to fix specific common problems in surface meshes.

- *Remove duplicate elements*: removes elements that are repeated in the mesh.
- *Remove non-manifold elements*: removes element that have an invalid connectivity.
- *Fix winding*: This option works only for triangular mesh. This option checks if the nodes of the triangle are in the same sequence clockwise/anti-clockwise direction for all the triangular facets.
- *Fill all holes*: Finds and fills the holes in the triangular mesh by adding new triangular faces.

### 5.3.2.5 Flip Edges

Flips the selected edge. This can be useful to eliminate slivers or otherwise locally improve the mesh quality.

### 5.3.2.6 MMG Remesh

The MMG Remesh tool uses the MMG library<sup>1</sup> to regenerate the surface mesh. The following options can be set.

- **Element size:** Desired element size.
- **Min element size:** The minimum element size allowed. MMG cannot always guarantee that this criterion is met.
- **Global Hausdorff value:** Sets the maximum allowed distance from the triangle mesh to the locally interpolated quadratic surface. Setting this to a smaller value creates a refined mesh in areas of large curvature. Note that this is a distance, so the units are length.
- **Gradation:** Sets the rate of transition between a refined and coarser area. A value close to one creates a very wide transition area. Larger values allow for narrower transition areas.
- **Only remesh selection:** If checked (and a current selection is active), only the selection will be remeshed.

---

<sup>1</sup><https://www.mmgtools.org/>

This tool can be used for globally remeshing the surface, or for doing a local mesh refinement. To perform a local mesh refinement, select the faces that should be refined, and make sure to check the “only remesh selection” option.

### 5.3.2.7 Partition

The Partition tool creates a partition (i.e. a selectable geometry surface) from the current node, edge, or face selection.

This tool is useful when the auto-partition tool did not create a desirable partitioning of the surface mesh.

### 5.3.2.8 Project Curve

This tool projects a curve on the mesh and modifies the mesh locally so that the curve becomes a crease edge of the surface mesh.

First, select the curves (i.e. edges of other geometry) and assign them to the selection box. Then, click Apply.

### 5.3.2.9 Refine

This tool uniformly refines the mesh by splitting all elements. Triangles are split into 3 triangles, and quads are split into 3 smaller quads.

### 5.3.2.10 Smooth

Performs a simple mesh smoothing, where each node is moved towards the average of its neighbors. The new node position is defined by,

$$\mathbf{r}_{new} = (1 - \lambda)\mathbf{r}_{old} + \lambda\mathbf{r}_c$$

Here,  $\mathbf{r}_{new}$  is the new position,  $\mathbf{r}_{old}$  is the old position,  $\mathbf{r}_c$  is the average of the node neighbors, and  $\lambda$  is a weight factor.

The following parameters can be set:

- **Iterations:** Enter the value to repeat the smoothing operation several times.
- **lambda:** The weight factor  $\lambda$  in the equation above.
- **preserve shape:** When checked, the new nodal positions are projected back onto the original mesh. This is useful to counteract the shrinking that typically occurs when applying this smoothing filter.
- **preserve edge:** When checked, nodes that were on an edge on the original geometry, are forced to remain on that edge.

### 5.3.2.11 Weld Nodes

The *Weld Nodes* tool merges the selected nodes that are within the specified tolerance. Any edges or faces that collapse as a result, will be removed from the mesh as well.

## 5.4 Creating and Editing a Mesh

The *Mesh panel* gives access to FEBio Studio's meshing capabilities. As noted before, FEBioStudio was not designed to be a powerful mesh generator, but regardless, it has some simple mesh generation and editing features.

### 5.4.1 Meshing Primitives

When the object is a primitive, that is, created with one of the geometry creation tools, the command window will list the mesh creation parameters for that particular object. Most primitives will be meshed with a so-called *butterfly mesh*. These primitives have a simple rectangular box as center. The rest of the mesh is a projection from this box to the respective geometry. For instance, for the sphere the surface of the inner box is projected onto a sphere. This projection is segmented to create several layers of elements. Note that all solid primitives are composed of 8-noded hexahedral elements and all shell primitives are composed of 4-noded quadrilateral elements, although some primitives may offer different element options. After you change the mesh parameters simply press the *Apply* button to create the new mesh. To show the mesh in the Graphics View select the **View → Toggle Mesh lines** menu or use the '*m*' shortcut.

### 5.4.2 Meshing CAD Geometry

CAD objects are meshed with the NetGen library. The NetGen mesher is a multi-pass mesh generator. It generates a base mesh and then tries to improve the mesh over several iterations. The following parameters control the mesh generation process.

- **Mesh Granularity:** set level of detail. This sets several internal parameters that affect the final element density.
- **Use local mesh modifiers:** Check to use local mesh modifiers.
- **Grading:** Set grading level (range is  $0 < \text{grading} \leq 1$ ). This global parameter describes how fast the mesh-size decreases. (Warning: Picking a very small value, e.g.,  $< 0.01$ , could cause Netgen to crash.)
- **Max element size:** Set the largest allowed element size.
- **Min element size:** Set the smallest allowed element size.
- **Nr. 2D optimization steps:** Set the number of passes to improve surface mesh.
- **Nr. 3D optimization steps:** Set the number of passes to improve volume mesh.
- **Second order tets:** When checked, NetGen will generate tet10 elements.
- **Elements per edge:** Number of elements to generate per edge of the geometry.
- **Elements per curve:** Elements to generate per curvature radius.
- **Quad dominant shell mesh:** For surface geometries, produce a quad-dominant mesh.

Click *Apply* to generate the mesh.

### 5.4.3 Meshing Editable Surfaces

An *editable surface* is an object that is defined via a surface mesh. The surface mesh can be edited via the tools on the Edit panel. If the mesh is closed and composed of triangles, the object can be meshed using **Tetgen**, available from the Mesh panel. This tool will generate a tetrahedral mesh. The following options can be set.

- **Element size:** the desired size of the elements.
- **Quality:** the desired quality of the elements. The quality of the element is defined as the ratio of the radius of the circumscribed sphere over the shortest edge length. The theoretical minimum is  $\sqrt{6}/4 \approx 0.621$ . Note that this is a suggestion to TetGen, and in general TetGen cannot guarantee that all elements will satisfy this criterion. In fact, setting it too low may cause TetGen to fail.
- **Element type:** Set the desired element type. Note that TetGen only generates 4-node tetrahedral elements (TET4), however, FEBioStudio can modify the element type after TetGen completes.
- **Split Faces:** Allow TetGen to split surface facets if it can improve the mesh quality. For curved surfaces this option should not be used, since TetGen does not try to maintain the curvature while splitting facets.
- **Hole:** To hollow out a part of the mesh, check this box and set the coordinates of a point inside the hole.

Click Apply to generate the mesh.

Editable surfaces can also be converted to shells. This is the only option available for surface meshes that are not closed.

### 5.4.4 Editable Meshes

If the object is a so-called *editable mesh* the Mesh panel will list some mesh editing tools. Editable meshes don't have a geometry object associated with it so the mesh, or at least its surface, defines the geometry implicitly. This has some important consequences related to applying boundary conditions to an editable mesh. Any change to the mesh may also change the corresponding geometry and as a consequence any data that was associated with the previous geometry may become invalid.

**Important Note.** *It is best to first make the necessary modifications to an editable mesh before you apply any boundary conditions or loads. Any modifications to an editable mesh may invalidate the selections assigned to boundary conditions and loads.*

Editable meshes can be edited on several levels, namely the *object level*, the *element level*, the *face level*, the *edge level*, and the *node level*. The element, face, edge, and node level are also referred to as the sub-object levels. When an object is an *editable mesh* (or editable surface), the Graphics control bar, will show additional buttons that allow you to select mesh items.



Enter the *element level*.



Enter the *face level*.



Enter the *edge level*.

- Enter the *node level*.

When none of the sub-object levels are active, the *object level* is automatically active. If you are in one of the sub-object levels, you can return to the object level by deselecting the selected button on the selection tab or pressing the Esc button.

The Graphics control bar also provides several options that affect the way mesh items can be selected.

-  *Select connected*: select all items that are connected to the selection. An angle criterion is used in addition to a connectivity criteria. The angle for this criterion can be set in the edit field next to this button.
-  *Select via closest path*: This will select all items via a closest-path criterion between two selected points.
-  *Select backfacing*: ignore items that are on the back of the mesh (and therefore not visible from the current viewing position).

To edit the geometry select the *Mesh* panel. Several tools will be displayed that allow you to modify the mesh. Most of these tools require a specific mesh selection mode to be active.

#### 5.4.4.1 Add Node

Adds a node to the mesh. Enter the position of the node and press *Apply*. Note that the node will not be connected to the rest of the mesh.

This tool can be useful to create an anchor for a spring that is connected to a node on the mesh.

#### 5.4.4.2 Align

This tool flattens and aligns the selection to be on the same plane.

#### 5.4.4.3 Auto Partition

Partitions the surface, edges, and nodes of the mesh based on an angle criterion. If the *Repartition elements* is checked, the elements will be partitioned based on their connectivity.

#### 5.4.4.4 Boundary Layer

For certain types of analyses (e.g. biphasic or computational fluid dynamics), it may be important to refine the mesh near the boundary to capture rapid boundary layer effects. The Boundary Layer tool can be used to generate a thin layer of refined elements. The tool requires a facet selection. It then finds the layer of solid elements adjacent to the selected faces, and then subdivides these elements, using the given control parameters:

- **bias**: Sets the refinement gradation level. If equal to one, all elements will have the same size. If larger than one, elements will shrink as they get closer to the surface.
- **segments**: Sets the number of divisions each boundary element will be divided in.

Note that this tool may introduce different types of elements. For instance, when applied to a tetrahedral mesh, the boundary layer may contain tetrahedral, wedge, and hexahedral elements. The quality of the elements in the boundary layer may also be degraded.

If a tetrahedral mesh was generated with TetGen (Section 5.4.3), the boundary layer tool may fail. In that case, remesh it with MMG (Section 5.4.4.14) and try the boundary layer tool again.

#### 5.4.4.5 Convert Mesh

Convert the element type to a different type.

When converting between linear and quadratic elements, nodes are inserted in (or removed from) edges and faces. The number of elements is not affected.

When converting between linear element types, elements may be split and the total number of elements will increase.

#### 5.4.4.6 Create shells from faces

This tool creates a layer of shells from a face selection. The shells will lie on top of the adjacent solid element.

#### 5.4.4.7 Detach Elements

This tool detaches the element selection from the rest of the mesh. The detached elements will be assigned to their own part.

#### 5.4.4.8 Discard Mesh

This discards the volume mesh and only retains the surface as a shell mesh.

#### 5.4.4.9 Extrude Faces

The selected faces are extruded. New solid elements are inserted. Triangular faces will extrude into wedge elements and quad faces will extrude into hex elements.

#### 5.4.4.10 Fix Mesh

This tool offers several options for repairing common problems in volume meshes.

#### 5.4.4.11 Inflate

Create a biased mesh in an existing tetrahedral mesh, by extruding the selected surface inward, producing pentahedral elements, then remeshing the inner domain with TetGen. This can be useful for analyzing boundary layer responses in the vicinity of the selected surface.

To use this tool, select the faces adjacent to the elements that will be inflated.

- **distance:** Sets the inflation distance. The element size normal to the facet selection will grow to this value. This value also determines the mesh size for the remeshing of the inner domain.
- **segments:** The top layer elements will be divided into this many segments.

- **mesh bias:** Sets the gradation level of the inserted elements.
- **symmetric mesh bias:** Applies a symmetric mesh bias to the inserted elements.
- **weld tolerance:** The inner domain is welded to the inflated mesh domain using this tolerance.
- **crease angle:** The two domains are rebuilt into a single domain (as described in the section [5.4.4.16](#) below), using the selected crease angle.

#### 5.4.4.12 Invert

This tool can be used to invert the selected elements. Elements that are inverted (i.e. turned inside-out) have negative volumes and cannot be used in an FE simulation. This can happen when the mesh was generated with a tool that assumes a different element nodal connectivity. This tool can be used to fix this problem.

#### 5.4.4.13 Mirror

This tool mirrors the mesh with respect to the selected mirror plane and center.

#### 5.4.4.14 MMG Remesh

The MMG remesh tool can be used to re-create the mesh or apply a local mesh refinement or coarsening on a tetrahedral mesh. The following parameters can be set.

- **Element size:** Desired element size.
- **Min element size:** The minimum element size allowed. MMG cannot always guarantee that this criterion is met.
- **Global Hausdorff value:** Sets the maximum allowed distance from the surface mesh to the locally interpolated quadratic surface. Setting this to a smaller value creates a refined mesh in areas of large curvature. Note that this is a distance, so the units are length.
- **Gradation:** Sets the rate of transition between a refined and coarser area. A value close to one creates a very wide transition area. Larger values allow for narrower transition areas.
- **Only remesh selection:** If checked (and a current selection is active), only the selection will be remeshed.

This tool can be used for global remeshing, or for doing a local mesh refinement or coarsening. To perform a local mesh refinement/coarsening, select the elements that should be refined, and make sure to check the “only remesh selection” option.

#### 5.4.4.15 Partition

The Partition tool creates a partition from the current node, edge, face, or element selection.

This tool is useful when the auto-partition tool did not create a desirable partitioning of the mesh.

#### 5.4.4.16 Rebuild Mesh

This tool can be used to fix any issues related to mesh connectivity. It rebuilds the internal mesh data structures. Note that this tool may affect the mesh partitioning.

This tool should not be used often. It should only be used if any other mesh operation leaves the mesh with an invalid internal data structure. Before applying this tool, you can use the Mesh Diagnostic tool to check if there are problems with the mesh.

#### 5.4.4.17 Refine Mesh

Refines the triangular shell mesh uniformly by dividing each triangular face into 4 smaller triangles. In this method, a new node is added at the center of each edge of the triangle and new triangles are created using these new nodes.

#### 5.4.4.18 Revolve Faces

Similar to the Face Extrude tool, but the selected faces are extruded by revolving them around an axis.

**axis** Sets the axis orientation

**center** Sets the center of rotation

**angle** Angle of rotation

**pitch** displacement along axis

**segments** divisions during rotation

#### 5.4.4.19 Rezone

This tool can be used to locally rezone a mesh. This is useful for refining or coarsening a mesh near a boundary, without the need to remesh the area. This tool simply moves nodes around in order to obtain the desired zoning.

#### 5.4.4.20 Set Axis

This tools can be used to generate local material axes on each element. Different algorithms can be chosen using the *generator* option.

**vector** The first option for generating material axes is *vector*. This option allows you to specify the same set of material axes for all elements of the mesh. In general, this option is useful if the desired material axes are not aligned with the global XYZ coordinate system. For this option you are prompted to enter the *a* and *d* vectors that uniquely define the XYZ axes triad. The vector *a* represents the local X-direction, and the vector *d* is used to produce the local Z-direction from the cross product  $c = a \times d$ . Then, the Y-direction is obtained form the right-hand rule from the cross-product  $b = c \times a$ . FEBioStudio automatically normalizes the vectors.

**angles** This option is similar to the vector option described in the previous section. It produces a uniform set of material axes using the azimuthal angle theta ( $\theta$ ) and declination angle phi ( $\phi$ ) of a spherical coordinate system as shown in the FEBio User's Manual.

**node-numbering** This option is generally useful only when the mesh consists of hexahedral elements, such as hex8, hex20 and hex27 elements, created in a regular pattern. Users need to specify three nodes, n0, n1 and n2, which must have values between 1 and 8, representing the 8 corner nodes of a hexahedral element. FEBioStudio treats the line segment between nodes n0 and n1 as the vector  $a$ , and the line segment from n0 to n2 as the vector  $d$ , from which it produces local XYZ axes as explained in vector above.

**cylindrical** The last option under the generator pull-down menu of Set Axes is the cylindrical option. This option may be used to produce local material axes that are aligned with a cylindrical coordinate system whose long axis (the local Z-direction) is along the user-specified vector  $a$  and whose radial direction (the local X-direction) is along the user-specified vector  $d$ .

#### 5.4.4.21 Set Axis from curvature

There are many geometries for which the options described in the Set Axes section for generating material axes may not be suitable. For example, consider a torus geometry that includes two families of fibers, each oriented along one of the two principal radii of the Torus. Thus we may wish to generate local material axes that are aligned along the principal directions of curvature of the toroidal surface. The following options can be set.

To use this tool, first make a face selection.

**generator** This option determines who to approximate the local surface in order to estimate the principal axes of curvature. For general surface, choose *spline*, but if the surface is a quadric surface (e.g. cone, cylinder, ellipsoid), the *quadric* option may perform better.

**whole\_part** The box labeled Whole part ensures that local material axes are assigned to all elements inside the geometry, not just the ones right underneath the selected faces.

This tool may be particularly useful with meshes that represent anatomical features, as typically acquired from medical images.

#### 5.4.4.22 Set Fibers

This tools can be used to generate local material fibers on each element. Different algorithms can be chosen using the *generator* option.

**vector** The first option for generating fibers is *vector*. This option allows you to specify the same fiber vector for all elements of the mesh. For this option you are prompted to enter the fiber vector directly. FEBioStudio automatically normalizes the vector.

**node-numbering** This option is generally useful only when the mesh consists of hexahedral elements, such as hex8, hex20 and hex27 elements, created in a regular pattern. Users need to specify two nodes, n0 and n1, which must have values between 1 and 8, representing the 8 corner nodes of a hexahedral element. FEBioStudio treats the line segment between nodes n0 and n1 as the fiber vector .

#### 5.4.4.23 Shell Thickness

Set the shell thickness for the selected shell elements.

#### 5.4.4.24 Smooth

This option works only for triangular shell meshes. It smooths the mesh by iteratively moving points towards their neighbors and often results in better-shaped triangles and more evenly distributed nodes.

- *Iterations*: Number of iterations to apply the smoothing
- *lambda*: weight factor for scaling
- *Preserve shape*: tries to preserve the overall shape of the object.
- *Project*: project the smoothed nodes back to the original mesh

#### 5.4.4.25 TetGen

This tool applies the TetGen mesher to a tetrahedral mesh. It can be used to recreate or remesh a tet4 mesh locally.

- **minimum radius-edge ratio**: sets the desired quality measure for the tet mesh.
- **element size**: sets the desired element size.
- **split facets**: allows TetGen to split surface facets if necessary.
- **feather**: sets the number of element layers of the transition zone between the locally remeshed part and the rest of the mesh.

#### 5.4.4.26 Weld Nodes

Weld the selected nodes together that are within a distance specified by the *threshold* edit field. Welding is useful to connect touching parts together. However, be aware that you might create unexpected errors in your geometry this way since the effects of welding are not always visible.

# Chapter 6

# Generating and Editing Mesh Data

FEBio allows the use of mesh data maps in a model, i.e. values assigned to individual nodes or elements of a mesh that can be used in various contexts. For instance, it can be used to assign a different material parameter value to each element, or prescribe a different nodal value for each node of a prescribed displacement boundary condition. FEBio offers two mechanisms for defining mesh data: The data can be defined explicitly via a map, i.e. a single value defined for each element or node, or it can be defined implicitly through *mesh data generators*. The latter are procedural approaches that generate the actual mesh data map when FEBio starts. FEBio Studio supports both types of mesh data and in addition offers several tools for generating, editing, and visualizing mesh data<sup>1</sup>. This chapter describes the ways that FEBio Studio offers users to generate and edit mesh data.

## 6.1 Adding Mesh Data

To add a mesh data map to the model first select an object in the Graphics View, right-click on the **Mesh Data** item in the model tree, and select **Add mesh data map** from the popup menu. A dialog box appears that allows users to select the type of map, the data type (i.e. scalar, vector, tensor), the data format, and how to initialize the data map.

The following properties can be set.

**name** Specify the name of the mesh data map<sup>2</sup>.

**type** Select whether the map represents nodal, element, or surface data. This choice affects how the data can be used. For instance, nodal data will typically be used in boundary conditions, and element data will often be used to define heterogeneous material parameters.

**data type** Select the data type of the map, i.e. scalar, vector, or tensor.

**data format** (Only relevant for element and surface data.) The 'item' option will assign a single value to the element. The 'node' option will assign a unique value to each node of the domain. For instance, for element data, the data is assigned to the (shared) nodes of the

---

<sup>1</sup>Note that FEBio Studio does not currently visualize the data generated by mesh data generators as this data is only generated by FEBio when the model is run.

<sup>2</sup>In general, it is best to avoid giving the mesh data map the same name as the model's variable to which it will be assigned. For instance, if the map is to be used for a material parameter named 'E', don't name the map 'E' as well. Instead, name it for instance 'E\_map'.

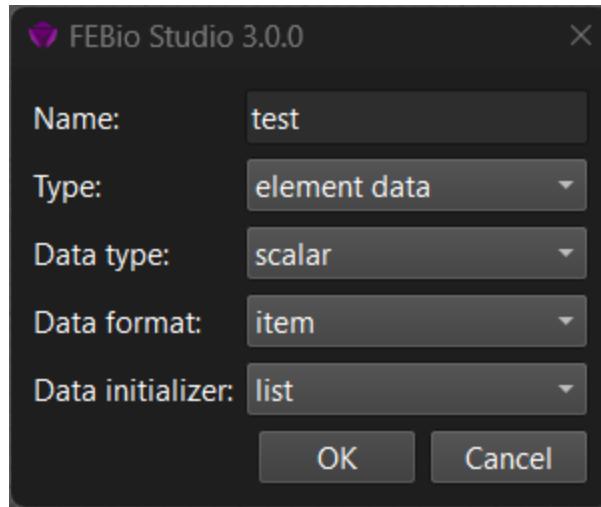


Figure 6.1: This figure shows the dialog box that can be used to add a mesh data map to a model.

elements. The mesh data therefore represents a continuous datafield over the domain. The 'mult' option assigns a unique value for each node of each element. Neighboring elements may have different values for each node they share. The resulting datafield may be discontinuous across element boundaries.

**data initializer:** Specifies how FEBio Studio will generate the data for the mesh data map. The 'list' option will generate a value for each item of the map and is typically what you want. The 'constant' option will not generate a value for each item, but instead will assign a single constant value to the entire map. This is only useful for very specific applications.

Once all properties are set, click OK to add the mesh data map to the model. At this point, although the map is visible in the model tree, it has not yet been assigned to any part of the mesh. The next step is to select nodes, elements, or surface facets, depending on the type of the map, and assign it to the map via the selection pane. After a selection is assigned to the map, the actual data is generated and can be edited. By default, the data values will all be set to zero. To change these values, right-click on the map in the model tree and select **Edit** from the popup menu. This will bring up a new dialog box that lists the current values for each item in the map.

Users can now edit the individual values of the map, or paste them from the clipboard. The data can also be copied onto the clipboard.

Although the Edit Data Map dialog allows fine grain control over the map's values, usually it is not the most convenient way to generate map data. The following sections describe other tools for generating mesh data.

## 6.2 FEBio Mesh Data Generators

As described above, FEBio offers an alternative way for generating mesh data maps, namely using *mesh data generators*. These are procedural (i.e. algorithmic) approaches for generating mesh data. A mesh data generator is added by right-clicking the *Mesh Data* item in the model tree, and selecting **Add mesh data generator** from the popup menu.

The dialog presents a list of the available mesh data generators. Note that the category column shows the type of mesh items the data generator can be assigned to. This is important as the type

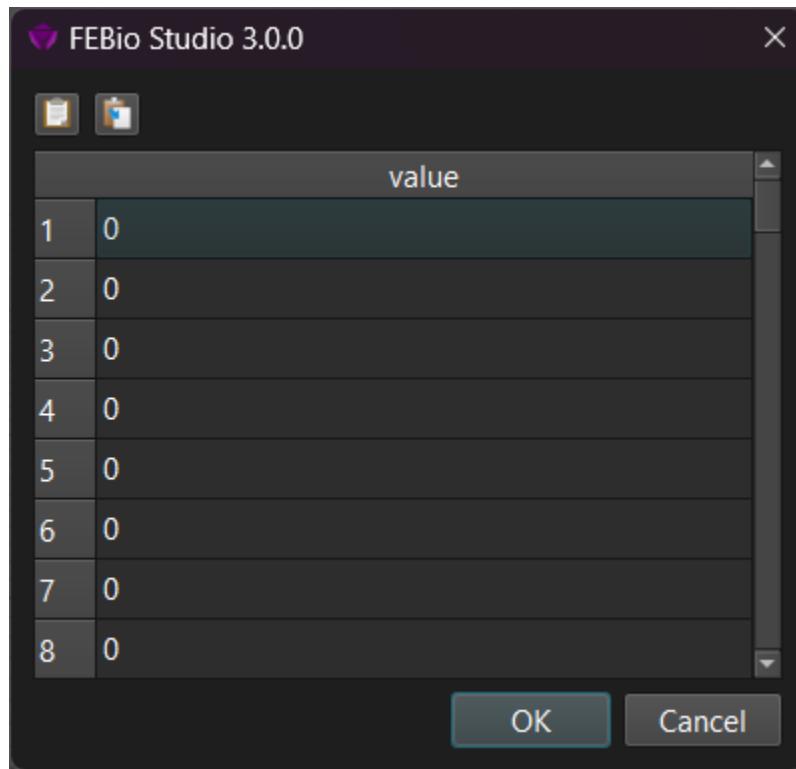


Figure 6.2: The Edit Data Map dialog can be used to modify the data values directly.

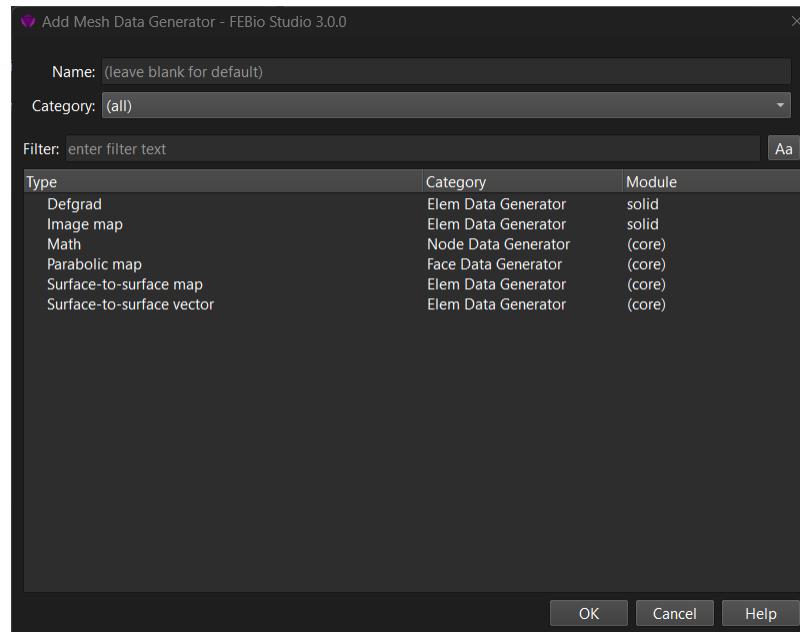


Figure 6.3: The Add Mesh Data Generator dialog box allows users to add an FEBio mesh data generator item to the model.

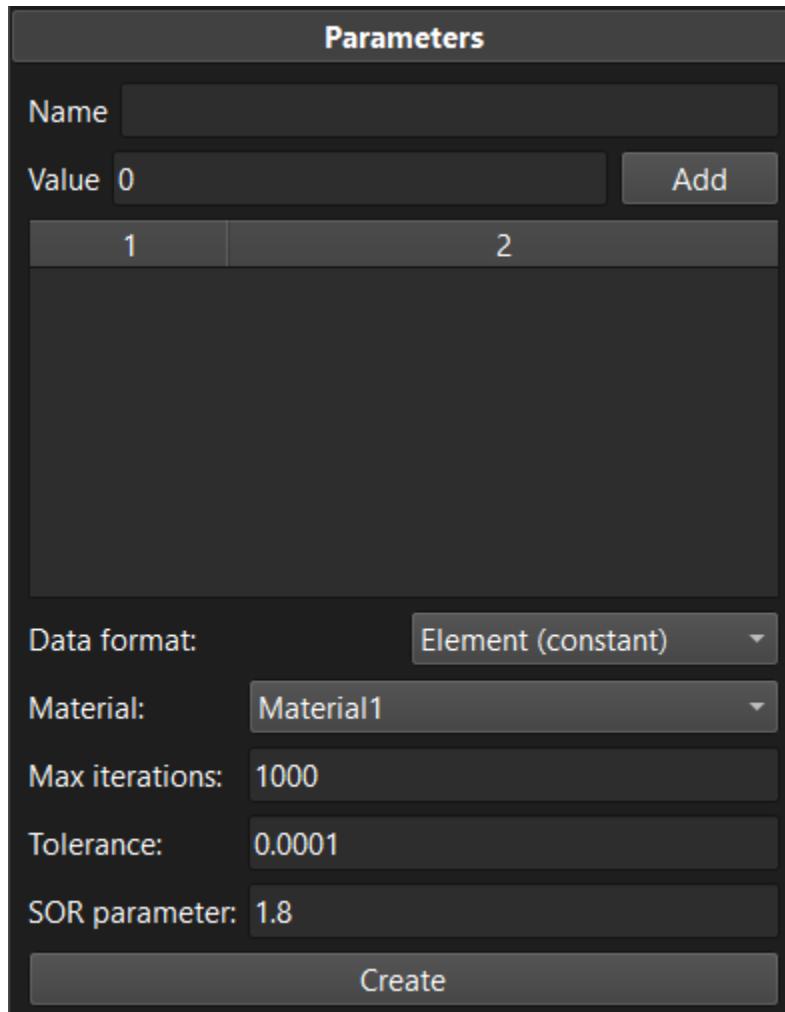


Figure 6.4: The scalar field tool can be used to generate a mesh data map.

defines how the map can be used in the model. Consult the FEBio manual for more details on the specific options and how to use them in a model.

### 6.3 The Scalar Field Tool

The **scalar field tool** is used to generate scalar data maps on a mesh. These maps can then be used for defining heterogeneous material parameters or other spatially varying model parameters.

The tool can be found on the *Tools* tab of the *Build* panel.

To use the tool, first enter a name for the resulting mesh data map in the *Name* field. Next, you can assign values to mesh selections. For example, if certain values are desired on specific surfaces, edges, or nodes, then the following steps will accomplish that goal: make a selection on the mesh, enter the desired value in the *value* field, click the **Add** button. A new entry will be added in the table. The values assigned to the selections can still be edited by double-clicking the value in the table and entering a new value. This procedure can be repeated as many times as needed to assign different values to selections. Note that it is not necessary to assign values to every item

(i.e. node or element) of the mesh. Instead, only values on some boundary surfaces or edges are needed. The values for the interior of the mesh are calculated by the tool. It accomplishes this by solving a Poisson problem on the mesh, using the provided values as boundary conditions on the corresponding mesh selections.

The following parameters further specify the data map that will be generated.

**Data format** Set how the tool will output the values, i.e. as nodal data or element data. This choice may affect how the data map can be used in the model.

**Material** Currently, data is only generated for the parts to which a particular material is assigned. This option allows you to select the material, and the corresponding parts on which data is generated.

The tool uses an iterative method for solving the Poisson problem. The remainder of the parameters control the solution process.

**Max iterations** Set the maximum number of iterations that can be used by the iterative solver.

Setting this value too low may result in an unconverged solution to the Poisson problem. The default value of 1000 will be sufficient for most problems, but models with large meshes may need to increase this value.

**Tolerance** The convergence tolerance for the iterative method.

**SOR parameter** Sets the SOR (Successive Over Relaxation) parameter, which may significantly decrease the number of iterations necessary for convergence. The default value will likely work well for most applications.

To run the tool, click the **Create** Button. This will solve for the nodal or element values, given the boundary conditions to the Poisson problem, create the mesh data map, assign the values, and add it to the model.

## 6.4 The Fiber Generator Tool

Despite the specific reference to “fiber”, this tool can be used to generate a vector field on a mesh. The vector field can be stored in a data map, which can then be used to define e.g. material fiber orientations.

The tool works very similarly to the scalar field tool. First, users assign (scalar) values to specific mesh selections. It then solves a Poisson problem on the mesh using the provided values as prescribed boundary conditions on the selections. This results in a scalar field. The tool then proceeds by numerically evaluating the gradient of this scalar field. This defines the initial vector field. Additional parameters allow further modifications of this vector field.

- **Generate mat axes:** Generate material axes instead of vectors and assign these to the element local coordinate system.
- **Generate cross product:** The final vector field will be the cross product of the initial vector field and the user-defined “normal vector”.
- **Normal vector:** The vector that will be used to generate the cross product vector field.
- **Out-plane rotation (deg):** Specifies the additional rotation angle (in degrees) to apply to the vector field that rotates it out of the plane that is perpendicular to the normal vector.

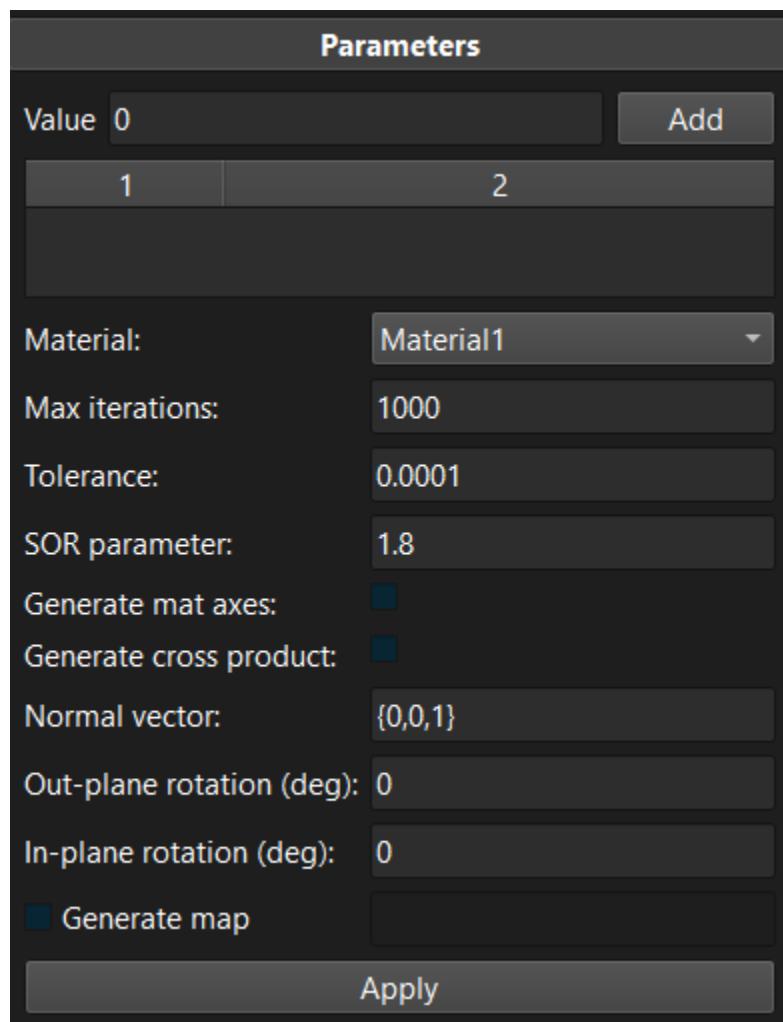


Figure 6.5: The scalar field tool can be used to generate a mesh data map.

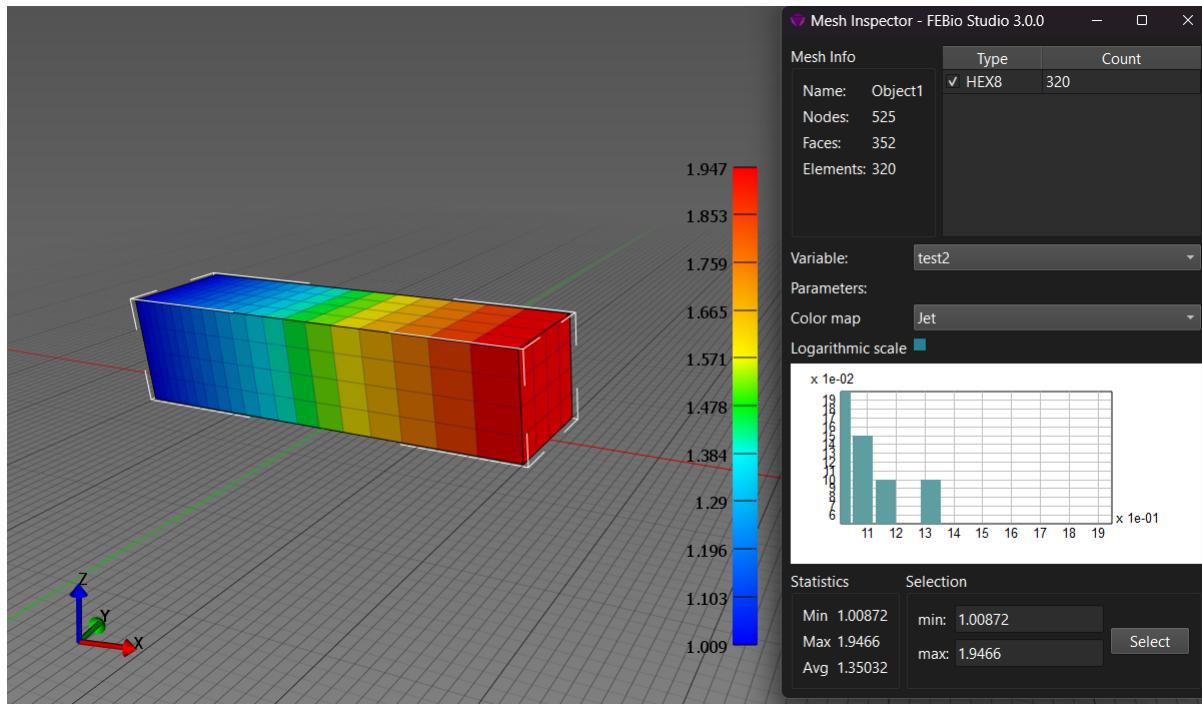


Figure 6.6: Mesh data can be visualized using the Mesh Inspector.

- **In-plane rotation (deg):** Specifies an additional rotation angle (in degrees) to apply to the vector field that rotates the vector in the plane that is perpendicular to the normal vector.
- **Generate map:** Generate a map and define the map's name. (If not checked, the vectors are assigned to the element's fiber property, but this will be deprecated in the future.)

Note that unlike the scalar field tool, the values assigned to mesh selections don't matter that much, but are simply chosen to guide the generation of the vector field.

## 6.5 Visualizing Mesh Data

Mesh data maps can be visualized in FEBio Studio<sup>3</sup> using the Mesh Inspector. This tool can be accessed from the menu **Tools** → **Mesh Inspector** or from the corresponding icon on the main toolbar. The *variable* option will list all the datafields that can be visualized and includes, near the bottom, the scalar mesh data maps. Simply select the name of the mesh data map to visualize the data in the Graphics View.

<sup>3</sup>The data maps generated by FEBio mesh data generators can currently not be visualized in FEBio Studio since this data is not generated until FEBio runs.

# Chapter 7

## Materials

Materials are used to define the constitutive properties and behavior of the various parts in the model. This chapter explains how to add, edit, and assign materials to the different parts of your model.

### 7.1 Adding materials

Materials are added using the *Material Browser*. The Material Browser is accessed from the **Physics** → **Add Material** menu, using the *Ctrl+M* shortcut or by right-clicking on the *Materials* item in the Model Editor and selecting *Add Material* from the popup menu. The Material browser is displayed in Figure 7.1. The Material Browser allows you to select a material and optionally provide a name for the material. If no name is given, FEBio Studio will generate a default one. The name of a material can also be changed later. After you selected the material, click **OK** to create the material and add it to the model. The material will become selected in the Model Viewer.

### 7.2 Setting material parameters

After you create a material, you can set the material parameters in the Model Viewer. More specifically, when you select a material in the Model Viewer, the material parameters are displayed in the Properties rollout. Note that materials can have child components, often referred to as material properties. Initially, these properties may not be set, in which case a drop-down box next to their name will appear. Selecting the <select...> option, will bring up another dialog box that allows you to select which particular material property to assign.

Note that you can assign a color to each material. In the Object rollout on the Model Viewer, you will see a color selector next to the material's name. If you click on this you can change the color associated with that material. When assigning a material to a part (see below), the part will be shown using the material's color.

### 7.3 Assigning materials

After you created a material and defined the material parameters you need to assign the material to your model. You can assign a material either to an object or to a part. To assign to an object, first make sure the object selection option on the toolbar is enabled . Next, select the object to

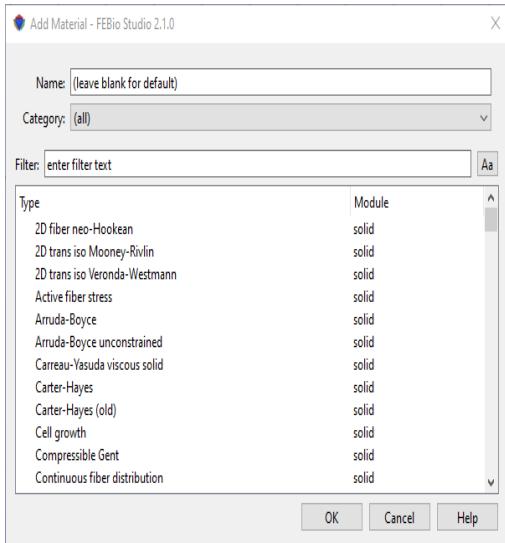


Figure 7.1: The Material Browser is used to add materials to the model.

which you wish to assign a material. Find the material in the Model Editor. Notice the selection panel in the Model Viewer. This panel contains a list that displays the parts to which this material is assigned. Initially, this list will be empty indicating that the material is not being used yet. When you now click on the + button, the selected object will be added to the list. More precisely, all the parts of the selected object will be added. To assign a material to only a part of an object, follow a similar procedure except now enable the part selection option on the toolbar . Select the part in the Graphics View and then add the part by clicking on the + button on the material's selection list.

## 7.4 Creating a Solute Table

Solutes may be included in several types of materials in FEBio, including biphasic-solute, triphasic, and multiphasic materials. A global table of solutes is created by accessing the **Physics → Solute Table** menu. When adding materials that include solutes (Section 5.1), these may be selected from the solute table.

## 7.5 Creating a Solid-Bound Molecule Table

Solid-bound molecules (SBM) may be included in multiphasic analysis. A global table of SBMs is created by accessing the **Physics → Solid-Bound Molecule Table** menu. When adding materials that include SBMs, these may be selected from the SBM table.

## 7.6 Adding Chemical Reactions

Chemical reactions may be included in some types of analysis (e.g. multiphasic analysis). Chemical reactions can only be added to previously created multiphasic materials as described in Section

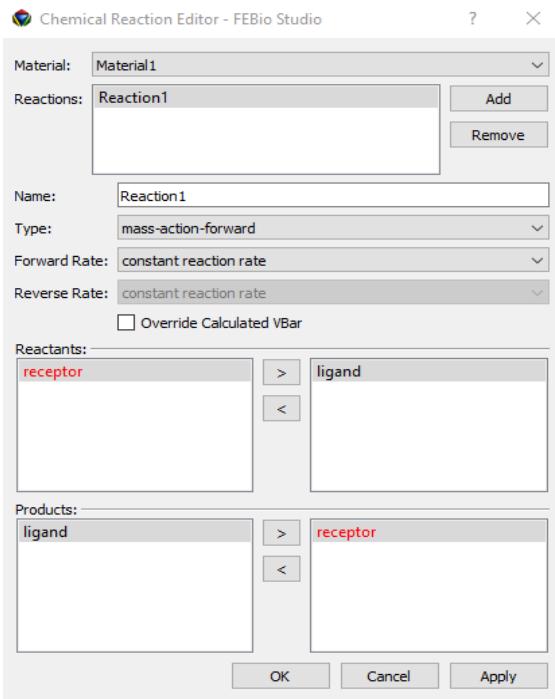


Figure 7.2: The Chemical Reaction Editor allows users to create and edit chemical reactions used in multiphasic analysis.

7. Chemical reactions are defined between solutes (Section 7.4) or solid-bound molecules (Section 7.5). Only solutes and SBMs that have been included in the parent multiphasic material may be involved in a chemical reaction.

To create a chemical reaction, select the **Physics → Chemical Reaction Table** menu. The Chemical Reaction Editor will appear (Figure 7.2). First select the multiphasic material to which a chemical reaction will be added. Then, press the Add button, next to the reactions table. A new reaction will be added. The name of the reaction can be altered in the Name field. Select the type of reaction and choose the forward and reverse rates (the latter only if applicable.)

Under *Reactants* you will see two panels. The left panel shows all the solutes and sbms available in the model. The materials in red are not defined in the multiphasic material and should not be used to create a chemical reaction for this material. To include a species in the reaction move it to the right panel by pressing the > button between the two panels. You can remove a species from the reaction by pressing the < button. Repeat for all reactants of the chemical reaction. The *Products* panels work identically.

To create the chemical reaction and add it to the multiphasic material, simply press the OK button. Once a chemical reaction has been created, its material parameters may be set in the Model Viewer as described in Section 7.2. Multiple chemical reactions may be defined within the same multiphasic material. The Chemical Reactions Editor can also be used to modify or remove reactions from a material.

# Chapter 8

# Boundary Conditions and Loads

This chapter describes the various boundary conditions and loads that can be applied with FEBioStudio. These include fixed constraints, prescribed constraints, prescribed surface loads and tractions, body forces, etc.

## 8.1 Boundary Conditions

Boundary conditions are used to constrain the solution on parts of the boundary. For example, in structural mechanics problems a boundary condition can be used to fix nodes from moving.

To apply a boundary condition, select the **Physics** → **Add Nodal BC** menu or the **Physics** → **Add Surface BC**. Most of the boundary conditions will be under the Nodal BC option. Nodal boundary conditions can be applied to nodes, surfaces, edges, or parts. Some special boundary conditions that can only be assigned to surfaces are found under the Surface BC menu option. In either case, a dialog box appears that shows a list of available boundary conditions.

At the top of the dialog box the name of the boundary condition can be entered. Alternatively, this field can be left blank to accept a default name. The name can also be changed later. Next, a drop-down list displays all the steps for which you can define a boundary condition. If you choose the *initial* step, the step will be applied in the initialization phase of the analysis and will remain active for all subsequent analysis steps. If you choose any other step, the boundary condition will only remain active during that step.

After you selected the step and the type of boundary condition, simply press OK to add the boundary condition to the model and edit the parameters in the Model Viewer.

In general, there are two types of boundary conditions. There are the *fixed constraints* and the *prescribed constraints*. For a fixed constraint (e.g. zero displacement), the corresponding degree of freedom is kept zero throughout the entire analysis. For a prescribed constraint (e.g. prescribed displacement), the value of the corresponding degree of freedom is defined by the user. You may wonder why the fixed constraints are available, since you can achieve the same result by defining a zero value for a prescribed constraint. The reason is that the degrees of freedom for fixed constraints are removed from the linear system of equations, reducing the computational time to solve the linear system. On the other hand, since the equations are removed, no reaction loads are calculated for fixed constraints. If you need to know for instance the reaction force on a boundary, you need to use a prescribed displacement even if the displacement is zero.

It is important to know that with most prescribed constraints a default load curve will be associated. The actual value for the constraint at any given time is the product of the scale factor which you will enter in the properties dialog for the boundary condition and the value of the load curve at

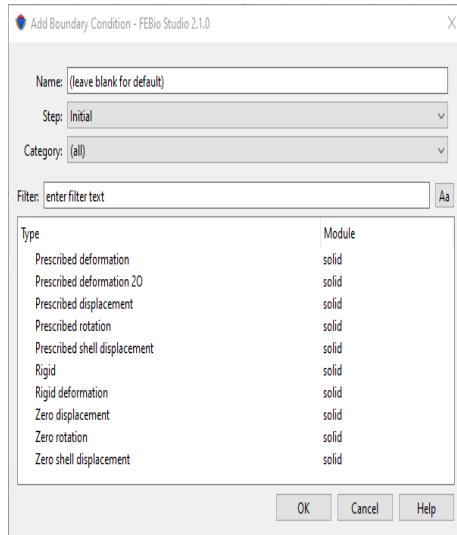


Figure 8.1: The Add Boundary Condition dialog box.

that time. Since by default the load curve will ramp from zero to one, the constraint value will ramp from zero to the specified value in a linear way. If you wish to modify the default curve you can edit it in the Curve Editor. See section [3.12](#) on details of dealing with load curves and the Curve Editor.

For details on the specific types of boundary conditions, please consult the FEBio User's Manual. The easiest way to access it, is to select a boundary condition and click the Help button in the dialog box. That should open the relevant page in a browser window.

## 8.2 Surface Loads

Surface loads are applied in a similar way as boundary conditions. First select the items to which you wish to apply the load. Then, select the **Physics** → **Add Surface Load** from the menu bar. This opens up a dialog box from which you can select the available loads (Figure [8.2](#)).

First, enter a name for the new surface load or leave the field empty to accept the default value. Then, select the step to which you wish to apply the load. Remember that applying a load to the initial step will cause the load to propagate through all the other steps. If you select an analysis step, on the other hand, the load will only remain active during that step.

Next, select the type of load you wish to add from the list and click the OK button to confirm your choice. The load parameters can be entered in the properties rollout on the Model Viewer. For details on the specific types of surface loads, please consult the FEBio User's Manual. The easiest way to access it, is to select a load and click the Help button in the dialog box. That should open the relevant page in a browser window.

## 8.3 Initial Conditions

For dynamic problems, the user can define initial conditions in a similar way as boundary conditions. First, select the part, surface, edge or node to which you want to apply the initial condition. Then, from the Physics menu, select “Initial Conditions” to open the *Initial Conditions* dialog box.

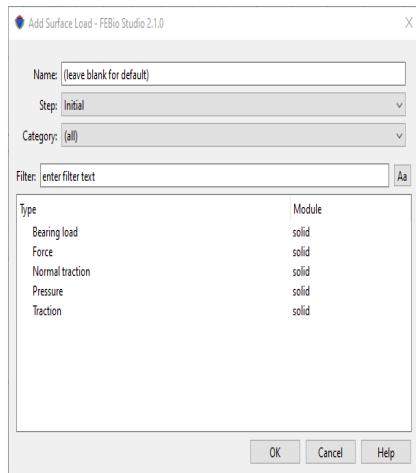


Figure 8.2: The Add Surface Load dialog box.

This dialog box presents a list of available initial conditions that can be applied to the current selection.

First, enter a name for the initial condition or accept the default. Then, select the step to which this initial condition needs to be applied. Next, select the initial condition from the list of available choices. For details on the specific types of initial conditions, please consult the FEBio User's Manual. The easiest way to access it, is to select an initial condition and click the Help button in the dialog box. That should open the relevant page in a browser window.

## 8.4 Assigning Boundary Conditions

To assign a boundary condition (or initial condition or load) to a selection, first select the boundary condition (or load, or initial condition) in the Model Viewer. Notice that a selection box appears below the tree view, which lists the model components to which this boundary condition is applied. Initially, this list may be empty. This list can be edited following the procedure detailed in section 3.8.3.

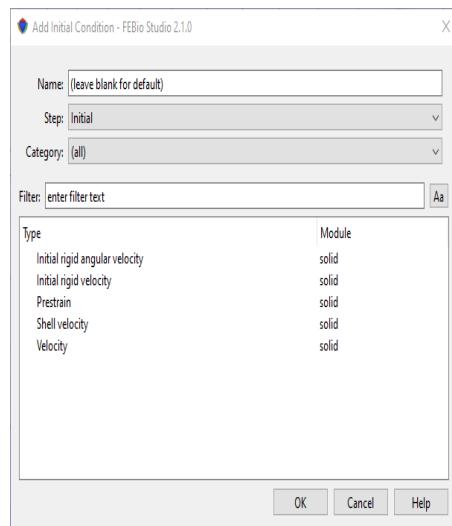


Figure 8.3: The Add Initial Condition dialog box allows users to create initial conditions for transient and dynamic analyses.

# Chapter 9

# Contact and Constraints

## 9.1 Contact

In many cases, different parts of a model can come into contact with each other. Once parts make contact, there are many different ways in which their motion can be constrained. Parts can slide across each other, stick to one another, etc. All these different contact behaviors can be modeled via contact definitions.

Most contact definitions can be divided into one of two categories.

**tied** Two, nonconforming, surfaces can be “glued” together with a tied interface. This is most commonly used when two parts are connected to each other at an interface, but their meshes are not conforming.

**sliding** A sliding interface allows two opposing surfaces to come into contact, and then slide across each other.

To add a contact definition, select the **Physics → Add Contact** menu. A dialog box shows up that allows you to select the step for which the contact definition is to be active (or select *Initial* if the definition is to be active during all steps). To add a particular contact condition, select an option from the list and click on the *Add* button.

In FEBio, contact constraints are enforced using an *Augmented Lagrangian* approach. This implies that the Lagrange multipliers are only approximated to a user-specified tolerance. The following parameters will appear in nearly all contact interfaces.

- **augmented Lagrangian:** Turn the augmented Lagrangian method on or off. When off, a penalty method is used for constraint enforcement.
- **augmentation tolerance:** Set the convergence tolerance for the Lagrange multipliers.
- **Penalty factor:** The penalty factor controls the rate of convergence. A high penalty factor will try to reach the tolerance quickly, but if chosen too high might introduce instability into the system. If it is too low, convergence to within the specified tolerance might not be reached.

For sliding interfaces, there will usually be an option for *two-pass* option. In a usual contact implementation, the required integrations are only performed over one of the surfaces, usually referred to as the *primary* surface. If the contacting surfaces are perfectly smooth, it does not matter which surface acts as the primary or secondary surface. However, in an FE simulation the surfaces are

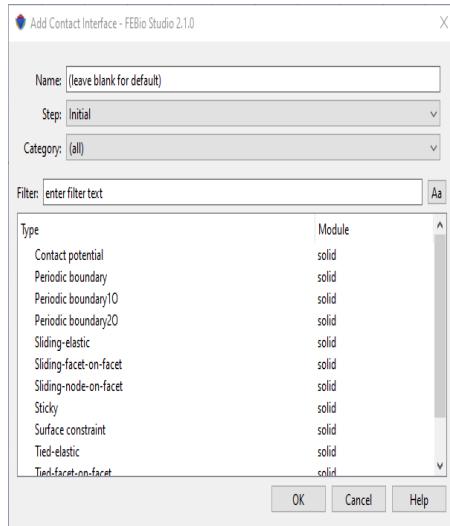


Figure 9.1: The Add Contact dialog box allows users to create initial conditions for transient and dynamic analyses.

discretized and are most likely non-conforming. Thus, the choice of primary and secondary surfaces is important and will introduce bias in the solution. It is usually advisable to select the more finely meshed surface as the primary. In the *two pass algorithm*, an attempt is made to reduce the bias by performing the contact calculations twice, with the roles of primary and secondary surface switched for the second pass. Although it may appear that the two pass algorithm is always the best choice, this is not always so. Certain contact applications perform better using a single pass. See the *FEBio Theory Manual* for a more detailed description of the contact model.

After the contact parameters are entered the user needs to define the *primary* and *secondary* contacting surfaces. To do this, first close the contact interface dialog and select the contact interface in the Model Editor. In the Model Editor you will now notice two selection boxes, one for the primary surface and one for the secondary surface. The boxes work similarly as for boundary conditions. For instance, to add a surface of your model to the primary surface, select the surface in the Graphics View and press the '+' button in the primary's surface selection box.

For detailed descriptions of the different contact definitions follows, please consult the FEBio User's Manual.

## 9.2 Nonlinear constraints

Aside from contact, FEBio offers several types of nonlinear constraints that in some way constrain the solution of the finite element problem. In FEBio Studio, these constraints are differentiated based on whether they apply to surfaces or to volumetric bodies. Constraints that don't quite fit in either category are referred to as general constraints.

Similar to contact, these nonlinear constraints are enforced using an Augmented Lagrangian method.

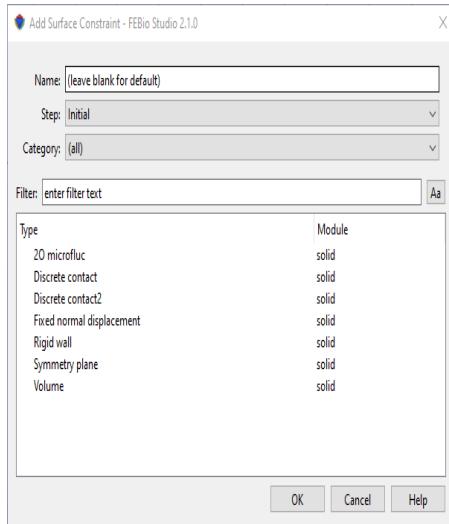


Figure 9.2: The Add Surface Constraint dialog box allows users to add nonlinear constraints on surfaces of the model.

### 9.2.1 Surface Constraints

A *surface constraint* constrains the solution of the finite element problem (e.g. displacement for a structural mechanics problem) at a surface. To add a surface constraint, go to the menu **Physics** → **Add Surface Constraint**, or right-click on the *Constraints* item in the model tree and select *Add Surface Constraint*.

For detailed descriptions of the surface constraints, please consult the FEBio User's Manual. You can press the Help button to take you directly to the relevant page in the FEBio User Manual in an separate web browser window.

### 9.2.2 Body Constraints

A *body constraint* constrains the solution of the finite element problem (e.g. displacement for a structural mechanics problem) on all the nodes of a volumetric body part. To add a body constraint, go to the menu **Physics** → **Add Body Constraint**, or right-click on the *Constraints* item in the model tree and select *Add Body Constraint*.

For detailed descriptions of the body constraints, please consult the FEBio User's Manual. You can press the Help button to take you directly to the relevant page in the FEBio User Manual in an separate web browser window.

### 9.2.3 General Constraints

A *general constraint* constrains the solution of the finite element problem in some special way. To add a general constraint, go to the menu **Physics** → **Add Body Constraint**, or right-click on the *Constraints* item in the model tree and select *Add General Constraint*.

For detailed descriptions of the general constraints, please consult the FEBio User's Manual. You can press the Help button to take you directly to the relevant page in the FEBio User Manual in an separate web browser window.

# Chapter 10

## Rigid Bodies

Rigid bodies are bodies that are assumed to be undeformable. They can only move and rotate and therefore are described by 6 degrees of freedom, 3 for translation and 3 for rotation. Rigid bodies are useful for modeling very stiff objects In FEBio (and FEBio Studio). Rigid bodies have their own set of constraints, initial conditions, and loads. They can also be connected via special model components called rigid connectors.

Rigid bodies are implicitly created when a *rigid body* material is added to the model. This implies that in principle rigid bodies do not need any geometry. They are defined completely via the corresponding rigid material definition. The only exception is that if the rigid body interacts with other, deformable, parts of the model via contact, then the rigid body needs to have geometry. Since the rigid body is defined via its material, you can easily make a part of the model rigid by assigning it a rigid material.

### 10.1 Rigid Body Constraints

The degrees of freedom of rigid bodies can be constrained in the same way as any other degree of freedom in the model. Before you can apply a constraint to a rigid body, you need to define a rigid body. In FEBioStudio, rigid bodies are implicitly defined when you create a rigid body material. A rigid body constraint can now be applied using the **Physics → Add Rigid Constraint** menu. It is important to note that when a rigid body is created none of its degrees of freedom are initially constrained.

In the *Add Rigid Constraint* dialog box first select the step in which this constraint will be active (or the *Initial* step if the constraint is to be active in all steps). Next, select a constraint from the list of available rigid constraints. Then click OK. The constraint will be added to the *Rigid* item in the model tree.

Please consult the FEBio User's Manual for more information on each rigid constraint. The easiest way to access it is to click the Help button on the dialog box. This will open the FEBio User Manual in a separate browser window.

### 10.2 Rigid Body Initial Conditions

Rigid bodies can also be used in dynamic simulations. Therefore it might be useful to set the initial conditions of the rigid body's kinematic variables. This can be accomplished by adding a rigid initial condition.

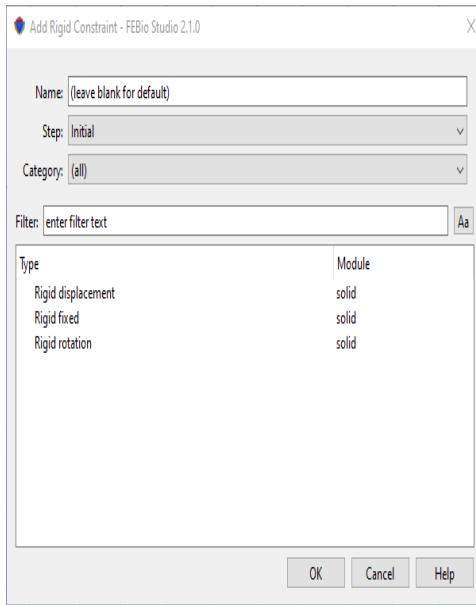


Figure 10.1: The Add Rigid Constraint dialog box allows users to constrain the degrees of freedom of a rigid body.

A rigid body initial condition can be added using the **Physics → Add Rigid Initial Condition** menu.

In the dialog box that appears first select the step in which this constraint will be active (or the *Initial* step if the initial condition is to be active in all steps). Next, select an initial condition from the list of available options. Then click OK. The initial condition will be added to the *Rigid* item in the model tree.

Please consult the FEBio User's Manual for more information on each rigid initial condition. The easiest way to access it is to click the Help button on the dialog box. This will open the FEBio User Manual in a separate browser window.

### 10.3 Rigid Body Loads

A rigid body load can be added using the **Physics → Add Rigid Load** menu.

In the dialog box that appears first select the step in which this rigid load will be active (or the *Initial* step if the initial condition is to be active in all steps). Next, select a rigid load from the list of available options. Then click OK. The rigid load will be added to the *Rigid* item in the model tree.

Please consult the FEBio User's Manual for more information on each rigid load. The easiest way to access it is to click the Help button on the dialog box. This will open the FEBio User Manual in a separate browser window.

### 10.4 Rigid Connectors

Rigid bodies can be connected to each other via *rigid connectors*. These connectors either constrain the motion (e.g. joints), or apply forces to the participating rigid bodies (e.g. springs).

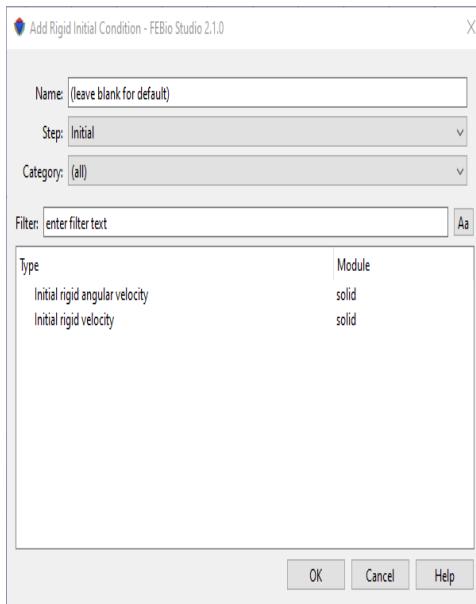


Figure 10.2: The Add Rigid Initial Condition dialog box allows users to set initial values for rigid body kinematic variables.

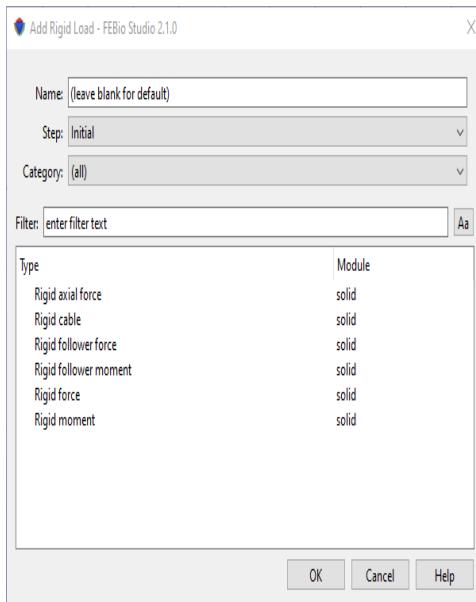


Figure 10.3: The Add Rigid Load dialog box allows users to apply various types of loads to a rigid body.

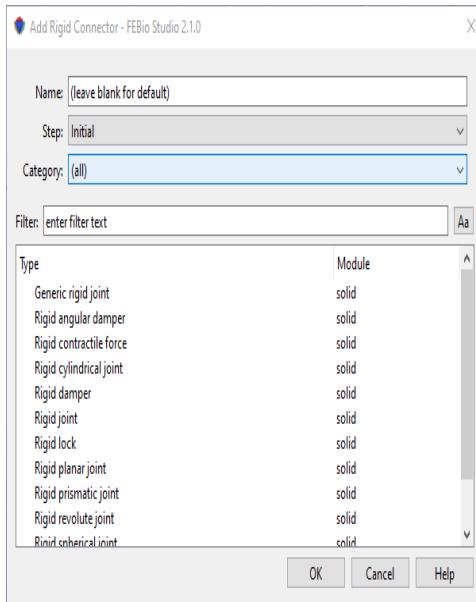


Figure 10.4: The Add Rigid Connector dialog box allows users to connect rigid bodies in various ways.

Rigid connectors can be added to the model using the **Physics → Add Rigid Connector** menu.

In the dialog box that appears first select the step in which this rigid connector will be active (or the *Initial* step if the initial condition is to be active in all steps). Next, select a rigid connector from the list of available options. Then click OK. The rigid connector will be added to the *Rigid* item in the model tree.

Please consult the FEBio User's Manual for more information on each rigid connector. The easiest way to access it is to click the Help button on the dialog box. This will open the FEBio User Manual in a separate browser window.

## 10.5 The Rigid Boundary Condition

As noted above, rigid bodies don't require geometry (except when used in contact) and are implicitly defined via rigid body materials. In order to tie a deformable body to a rigid body, a special boundary condition, the *rigid* boundary condition, is used. The rigid boundary condition is added as any other boundary condition (e.g. via the menu **Physics → Add Nodal BC**). Any selection of a deformable object (e.g. surface) that is assigned to this boundary condition will be tied to the rigid body. The rigid body that the selection ties to, is set as a parameter of the rigid boundary condition.

The rigid boundary condition should not be used to tie two rigid parts together. The easiest way to tie rigid parts together is to assign them the same rigid body material.

# Chapter 11

## Defining Analysis Steps

The actual analysis that is to be performed is defined through *steps*. A model may define as many steps as needed, and each step defines what is to be calculated for that particular step.

### 11.1 The Initial Step

When you start FEBioStudio or create a new model using the **File → New** menu, you'll notice that FEBioStudio by default creates one step, namely the *Initial* step. This initial step does not perform an actual analysis; instead it is a placeholder for the initialization phase of the analysis. Currently, the most important aspect of this step is that all boundary conditions, loads, constraints, etc. that are defined in this step will remain active during all subsequent analysis steps. This is different from an analysis step in that e.g. a boundary condition defined in an analysis step will only remain active during that step.

### 11.2 Adding an Analysis Step

An analysis step is added using the **Physics → Add Step** menu. Alternatively, you can also add a step by right-clicking on the *Steps* item in the Model Editor and select *Add* from the popup menu. A new step will be added to the *Steps* item in the model tree.

After the step was added, many settings can be modified to control how FEBio will solve the analysis step. Below, some of the most important settings are discussed. For more details, please consult the FEBio User Manual.

#### Analysis

The analysis settings allow you to choose the specific analysis type. For instance, for structural mechanics problems, users can choose between steady-state or dynamic.

#### Time Stepping

These settings control the basic time stepping and allows users to set the number of time steps and the size of each time step size. If no time step controller is specified, FEBio will use a fixed time stepping approach to solve the model.

#### Auto-time stepper

The auto-time stepper provides a way to adaptively adjust the time step size depending on how the solution progresses. The “default” auto-time stepper provides the following parameters.

- *Max retries*: the maximum number of times that a particular timestep will be retried.
- *Optimal iterations*: The expected average number of iterations required to converge a single time step.
- *Min stepsize*: the minimum stepsize that may be taken by the auto timestep controller.
- *Max stepsize*: the maximum stepsize that may be taken by the auto time step controller. Note that a load curve may be attached to this parameter. This load curve is often referred to as a *must point curve*. In addition to setting the max time step size, the must point curve also defines points through which the solution must pass. Also note that when a must curve point is defined, FEBio will ignore the value of max step (dtmax) variable.
- *Aggressiveness*: When set to 1, the time step size will be scaled by the *cutback* parameter.
- *Cutback*: scale factor for reducing time step size. Only used when *aggressiveness* is set to 1.

### Solver Settings

This property allows users to select from a list of solvers that can be used to solve the analysis. For example, for structural mechanics models, users can choose between an implicit and explicit solver. Each solver has additional parameters and properties. Please consult the FEBio User Manual for details of the particular solvers.

### Output settings

The output generated by FEBio can be controlled by the user. In general, FEBio produces a plot file, which contains the results of the FE analysis, and a log file. The log file stores the screen output from FEBio, but can be customized to store additional data. Alternatively, this additional can be written to separate files. See chapter 11 for more details on how to customize the FEBio output files.

The output settings of the step allow users to control the intervals at which FEBio will generate output during the step.

# Chapter 12

# Configuring Output

FEBio generates several output files. The two most important ones are:

- **logfile:** This file contains the screen output of FEBio, but can also output data requested by the user.
- **plotfile:** This file contains the results of the FEBio analysis.

To some extent, the contents of these output files can be customized by the user, as described in the following sections.

## 12.1 FEBio Logfile

The primary purpose of the FEBio logfile is to provide a copy of the screen output generated by FEBio. However, users can also request to write additional output data to this file (or a separate output file).

To add data to the logfile, expand the output section in the model tree, right-click on the *logfile* item, and then select “Edit Output” from the context menu. This brings up the Edit Output Dialog box (with the logfile tab selected).

To add a data output variable, select the Type and List, and enter a semi-colon (;) delimited list of variables.

- The *Type* selects the category of output variables and users can select *Node*, *Element*, *Rigid Body*, or *Rigid Connector*.
- The *List* option selects the named selection for which the data will be exported.
- The variables must be a list of semi-colon separated values. For a list of allowed variable names, please consult the FEBio user manual.

After the type, list, and variables are selected, click the Add button to add the data variable to the table.

Items in the table can be removed by first selecting them and then clicking the *Remove* button.

If the data should be written to a separate file instead of the logfile, a filename can be provided in the File column. This field can be edited by double-clicking on the corresponding table item.

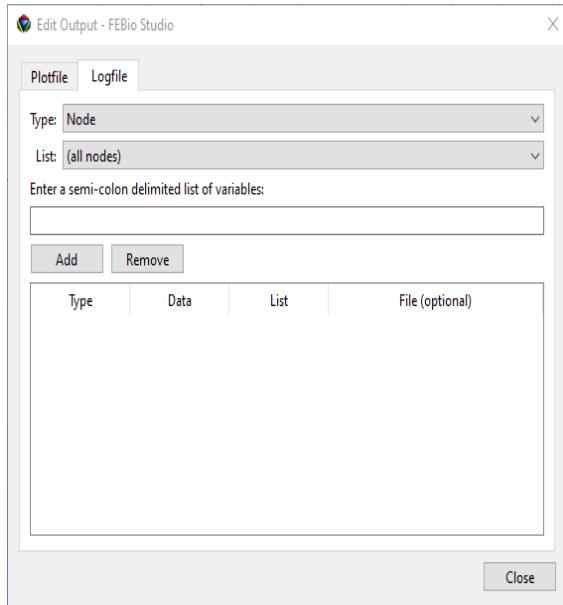


Figure 12.1: The Edit Output dialog, with the Logfile tab selected. This tab allows users to add data variables to the log file (or a separate data output file).

## 12.2 FEBio Plotfile

The FEBio plotfile contains the results of the analysis. It is a binary format and it is highly customizable. The contents of this file is entirely defined by the user.

**Although FEBio Studio tries to select some default, commonly used variables based on the analysis type, it is highly recommended to always check the list of selected plot variables to ensure that the desired data will be written. Failure to do so, may result in loss of data and the need to re-run the analysis!**

To edit the contents of the plot file, expand the Output section of the model tree, right-click the plotfile item, and select “Edit Output” from the context menu. The Edit Output dialog box will be shown, with the plotfile tab selected.

Output variables that will be written to the plot file can be selected by checking the box next to the corresponding variable name.

To find a particular variable, you can enter the first letter(s) in the Filter field. This will limit the list of displayed variable names to those that match the filter.

If the plot variable requires a selection (e.g. surface, or part) to extract the data from, click the “Add Selection” button, and then choose from the list.

Custom plot variables can be added as well, by clicking the “Add...” button. A dialog box shows up that allows users to enter the name of the plot variable. This can be useful for adding plot variables that are not directly supported by FEBio Studio. In addition, this feature is currently the only way to define plot variables using the extended syntax supported by FEBio that allows adding a filter and an alias to the plot variable. For example, if the user wants to output the fiber vector of a particular component of a solid mixture, the user can add a custom plot variable that uses the full syntax:

```
fiber vector['solid[0]']=v1
```

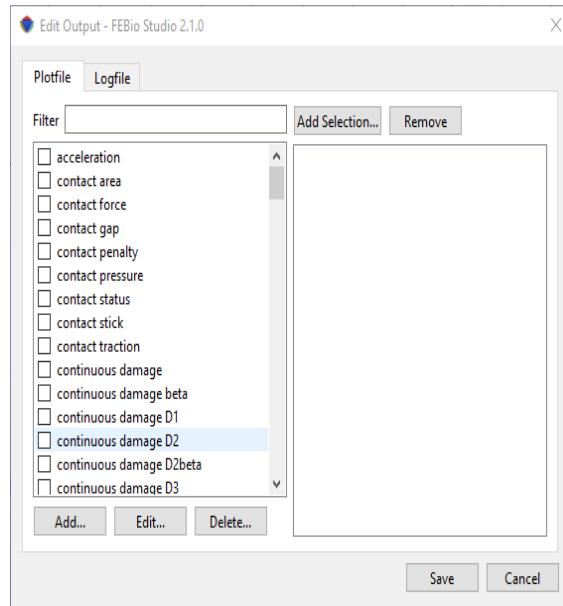


Figure 12.2: The Edit Output dialog, with the Plotfile tab selected. This tab allows users to add data variables to the plot file.

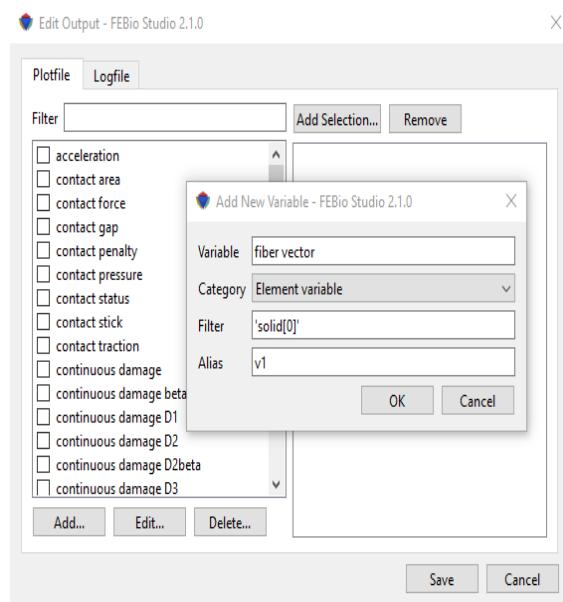


Figure 12.3: Users can add custom plot variables to the FEBio plotfile.

# Chapter 13

## Running FEBio from FEBioStudio

### 13.1 Running FEBio

FEBio can be called from within FEBioStudio by selecting the **FEBio → Run FEBio** menu, or from the main toolbar by clicking the corresponding tool button . The Run FEBio dialog box shows up (Figure 13.1) where users can set the following fields.

- **Job name:** The name of the job. This is used to generate the febio input and output file names. The job name will also appear in the model tree.
- **Working directory:** Specify the location where FEBioStudio will store the FEBio input and output files.
- **Launch Configuration:** The launch configuration defines how FEBio Studio will launch FEBio. As of FEBio Studio 2.0, the default launch configuration will use the version of FEBio that FEBio Studio was build with. Other launch configurations can be created that start FEBio as a separate process on either the local machine or even a remote server.
- **Save document before running FEBio:** Check this to save the model file automatically before running FEBio.
- **FEBio file format:** Select the format for the FEBio input file.
- **Write Notes:** Check this option for writing the notes associated with the model components.

By pressing OK, FEBioStudio will first export the current model to the file and location as specified, then call FEBio. The FEBio output will be shown in the log panel. If this panel is not visible, you can activate it from the **View → Log** menu.

Clicking the *Advanced* button will show additional options for controlling FEBio.

- **Debug Mode:** Check this option to run FEBio in debug mode. In debug mode, FEBio will write additional information to the log and plot files that can help in debugging a model.
- **Config File:** The configuration file contains information to configure FEBio before running the model, including linear solver parameters and plugins.
- **Task name:** The name of the FEBio task that needs to be executed.
- **Task control file:** An optional file that provides information to the FEBio task.

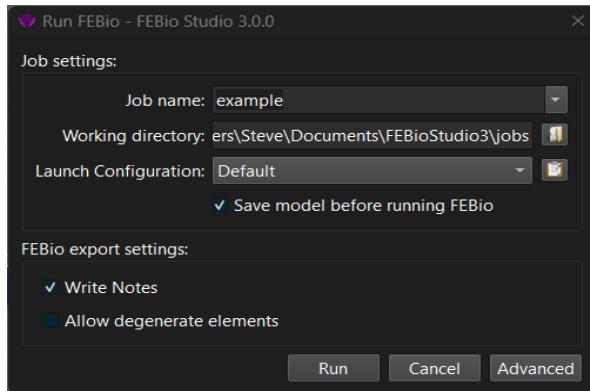


Figure 13.1: The Run FEBio dialog box lets you start FEBio from within FEBioStudio.

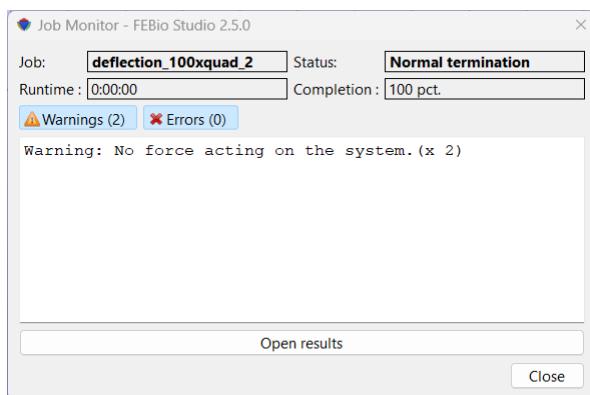


Figure 13.2: The Job Monitor is displayed after the FEBio job completes.

- **override command:** Check this option to override the FEBio command line that FEBio Studio will use, and specify the command line options directly.

To run the model in FEBio press the Run button. The model will first be written to an FEBio input file and then FEBio will be called on the input file. After FEBio returns a dialog box appears with some information on how the job ran.

The information shown in the Job Monitor dialog box includes:

**Job** The name of the job that just completed

**Status** The final status. This will be *Normal Termination* if the FEBio was able to complete the job successfully, or *Error Termination* when the job failed.

**Runtime** The total time it took for the job to run

**Completion** A percentage indicating how far the analysis was completed.

The text field below the status information shows a list of all the warning and error messages that were generated by FEBio. (This currently only works with the Default launch configuration.) The two buttons labeled *Warnings* and *Errors* are used to toggle the corresponding messages from the text field.

The **Open Results** button can be used to open the results of the completed run in FEBio Studio. If the run was not successful, the user might still be able to load the partial results.

## 13.2 FEBio Launch Configurations

Launch configurations contain the information needed to run FEBio on your local machine or on a remote server. For local configurations, you just need define a path to the FEBio executable. For remote configurations, you'll have to provide additional information on the server that you wish to run on.

To edit launch configurations, click the *Edit* button next to the *Launch Configuration* field in the *Run FEBio* dialog. This will launch a new dialog which will allow you to edit launch configurations (Figure 13.3).

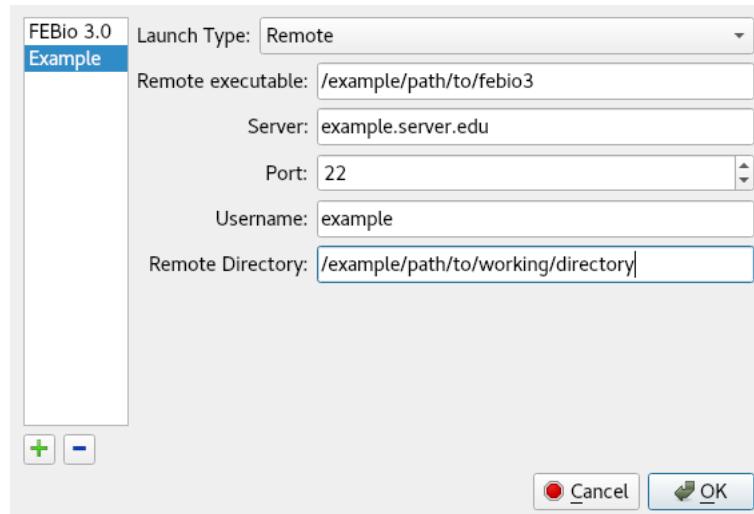


Figure 13.3: The launch configuration editor, shown editing a remote configuration.

The left side of the dialog shows your existing configurations. To add a new configuration click the plus icon below the list of configurations. To remove a configuration, select it in the list, and then click the minus icon. To rename a configuration, double-click on the configuration name. To reorder the configurations, drag and drop them in the list.

There are several types of launch configurations, each with a different set of fields to edit. With the exception of the *Local* launch configuration type, all types run your jobs on a remote server.

- The *Local* launch configuration type will launch the specified FEBio executable to run your job. This will be run on your current machine. The only field required is:
  - *FEBio Executable*: The path to the local FEBio executable that you wish to run with this configuration
- The *Remote* launch configuration type will copy your job files to a remote server via SFTP into the specified directory, and then run the job via SSH with the specified remote FEBio executable. To use a remote configuration, fill out the following fields:
  - *Remote Executable*: The remote path to the FEBio executable that you wish to run with this configuration.
  - *Server*: The domain name or IP address of the server that you wish to connect to via SSH/SFTP.

- *Port*: The port on your server on which the SSH service is being hosted. By default, this is set to 22, which is the standard port for the SSH protocol.
  - *Username*: Your username on the remote machine. Your user account must have permissions to connect to the server via SSH.
  - *Remote Directory*: The remote directory to which your FEBio input file will be copied, and which will act as the working directory for your FEBio run. Unless otherwise specified in your input file, your log and plot files will be created here. This directory need not exist, and will be created automatically if it does not.
- The *PBS Queue* and *SLURM Queue* launch configuration types will copy your job files to a remote server via SFTP into the specified directory, and then add a new job to a PBS or SLURM queuing system. These launch configuration types each have all of the same fields as the *Remote* type, along with a text box in which a PBS or SLURM script may be edited. At launch, the contents of this text box are put into a file, transferred via SFTP to the remote server, and run with either the *qsub* (for PBS) or the *sbatch* (for SLURM) command. By default, the text box contains a valid PBS or SLURM script with some common options set. For convenience, several macros have been defined, which you may use in your script. Before being written to a file and being copied to the server, these macros are replaced with a string unique to either the launch configuration, or the current job. These macros are:
  - `${JOB_NAME}`: This macro will be replaced with the name of the current job as defined in the *Run FEBio* dialog.
  - `${REMOTE_DIR}`: This macro will be replaced with the contents of the *Remote Directory* field in the launch configuration.
  - `${FEBIO_PATH}`: This macro will be replaced with the contents of the *Remote Executable* field in the launch configuration.
- The *Custom Remote* launch configuration type will copy your job files to a remote server via SFTP into the specified directory, and then run a set of specified commands in series on the remote server via SSH. This launch configuration type has all of the same fields as the *Remote* type, except for the *Remote Executable* field. It also has a text box in which a custom series of commands may be written, one on each line. There are a few things to keep in mind when creating a custom script:
  - The same macros available in the *PBS Queue* and *SLURM Queue* launch configuration types are available for use in this script, with the exception of the  `${FEBIO_PATH}` macro.
  - Lines starting with the '#' symbol are comments and will be ignored.
  - Empty lines are ignored.
  - The full string of each line will be run as a separate command on the remote system.
  - The script written in this field is not a bash script, but rather a set of discrete commands which will be executed in series. It is of course possible to run a bash script with this launch configuration type, by placing the desired script on the remote server, and then running the script by calling it in one of the lines of the text box.
  - This launch configuration type allows for great flexibility. Using it, you can do pre and post processing of your FEBio jobs on a remote server with the click of a single button. Use this launch configuration type with caution, as it does not have any safety nets;

anything you tell the remote server to do, it will do, as long as your user account has permissions to do it.

When an FEBio job is started using any of the remote launch configuration types, an SSH connection is made to the remote server. This requires authentication. If you have set up private key authentication for the specified server and username, the authentication will happen automatically. Otherwise, you will be prompted for your user account password during the first attempt to connect to the server. This password will be encrypted and stored in RAM for the duration of your FEBio Studio session, and will automatically used for subsequent server interactions until you close FEBio Studio.

In order to retrieve the log and plot files from any of the remote launch configuration types, click the *Get Remote Files* button under the *Job* properties in the *Model Tree*. The status of the PBS or SLURM queue may be checked by clicking the *Get Queue Status* button under the *Job* properties in the *Model Tree*. The queue status will be printed to the *Output* panel.

### 13.3 The FEBio Monitor

The FEBio Monitor offers an alternative way to run FEBio models from FEBio Studio. Using the FEBio Monitor, users can have a live graphical view of the model's solution progress, as well as monitor the convergence norms using graphs. The solution progress can be paused at which point users can inspect the detailed model's state, including the current stiffness matrix.

The FEBio Monitor can be a helpful tool for debugging a model as it offers a direct visual way to inspect the solution progress, something that is otherwise difficult to obtain. Problems can be identified as soon as they occur, without the need to take any further post-processing steps.

The FEBio Monitor can be invoked from the menu **FEBio → Run FEBio Monitor**. A dialog box appears that is similar to the usual run dialog, but offers additional controls.

The *Job Settings* pane allows users to enter the job's name and its working directory, similar to a standard FEBio job. The working directory is the place where the output files will be stored. It is important to note that the FEBio Monitor will run a standard FEBio job that will produce the usual output files.

The *Monitor Settings* pane allows users to configure the control parameters of the session. Most of these parameters can also be modified once the job is started, as discussed below.

**Pause event** This is the event that will be used when a pause is requested. After the job is started, the user can pause the run. The job will then be paused at the next occurrence of the selected event.

**Start job in paused state** This option will pause the job at the first occurrence of the selected event.

**Pause after time** This will pause the job once it passes the entered time value. Note that this time refers to the simulation time, not the actual runtime of the job.

**Debug level** Run the job in debug mode or not.

**Record states** By default, the FEBio Monitor will only display the current state of the model. When this option is checked, all the states will be recorded and can be inspected when the job is paused.

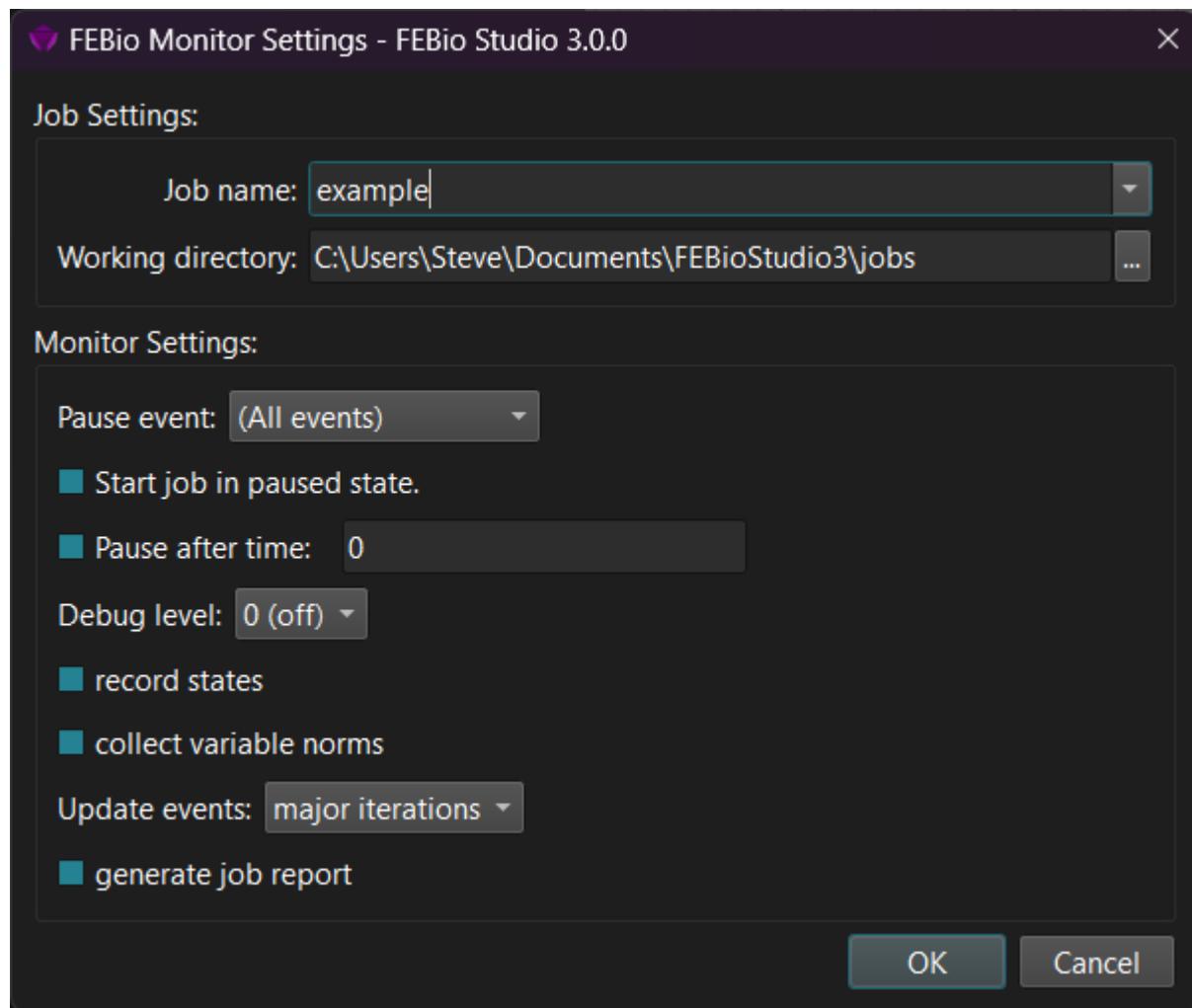


Figure 13.4: The FEBio Monitor dialog allows users to configure the FEBio Monitor session and run the FEBio model.

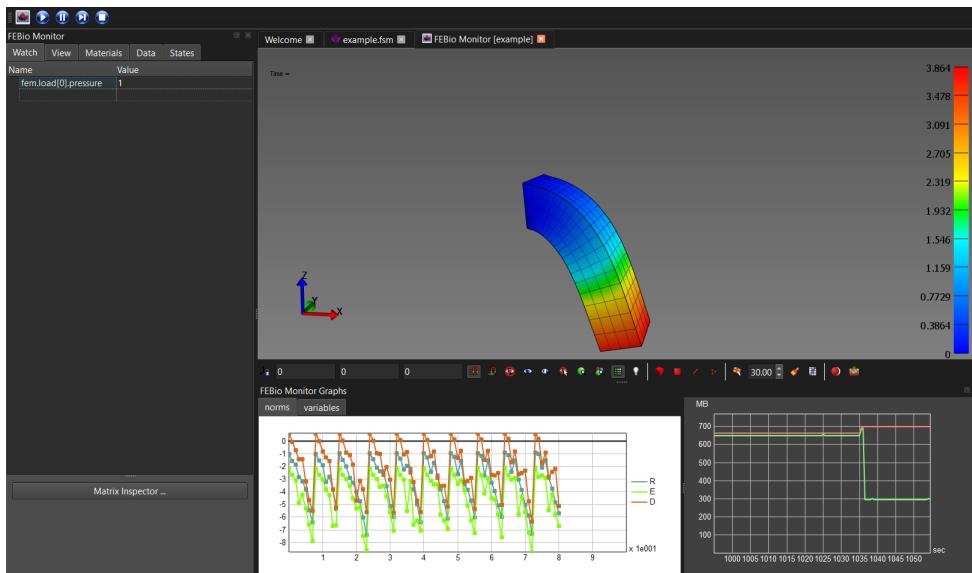


Figure 13.5: The FEBio Monitor shows a graphical view of the current state of the FEBio model’s solution.

**Collect variable norms** Check this option if you want to display the norms of the model’s individual solution variables. (By default, only the residual, energy, and total solution norms are shown.)

**Update events** Select when the model’s state will be updated in the monitor. This only affects how often and when the monitor will query the solution’s state.

**Generate job report** Collect additional data while running the FEBio job and create a report once the job is completed.

Once the settings have been chosen, click the OK button to start the monitor session. The FEBio Monitor will load and start the FEBio job. Note that the job is run using the default launch configuration. The FEBio Monitor currently does not support any other launch configurations.

In many ways, an FEBio Monitor session resembles a post session and offers many of the same features. However, there are a few noteworthy differences (see 13.5). The FEBio Monitor panel on the left has a *Watch* tab that by default displays all the model parameters that are load-controlled and shows the current value. (Note that this panel is only updated when the model is paused.) Users can also add additional entries in the table by double-clicking the empty slot below the last entry and entering the model parameter’s name. In addition, on the bottom of the *Watch* panel, the *Matrix Inspector* can be accessed. The Matrix Inspector gives users a visualization of the current stiffness matrix as well as some tools to inspect and diagnose the matrix.

As part of an FEBio Monitor session, a new toolbar is visible in the toolbar area near the top of the window.



From the left to right, the buttons allow the user to take the following actions.

- Open the FEBio Monitor Settings dialog. This dialog is similar to the FEBio Monitor dialog mentioned above, but only shows the parameters that can be edited in an active session.

- The 'play' button will continue a paused model. The model will not be automatically paused on the next pause event.
- The 'pause' button will pause the model. Note that FEBio will only pause on the pause event selected in the FEBio Monitor dialog.
- The 'next' button will continue a paused model. The model will be paused automatically on the next pause event.
- The 'stop' button will stop a running model and terminate the monitor session.

In addition to the Watch panel, additional graphs are shown below the Graphics View that show various metrics of the active model. The graphs display the convergence norms of the model, typically the residual (R), energy (E), and overall solution norm (D), as well as additionally selected norms. The *memory monitor* shows the current and maximum allocated memory. (Note that this is the memory consumption of FEBio Studio, not of FEBio.)

## 13.4 Running Batch Jobs

FEBio Studio has the option to manage and run several FEBio jobs at the same time. These are referred to as batch jobs. In addition to running multiple jobs simultaneously, the batch feature also offers the ability to compare the results between different batch runs.

### 13.4.1 Create a new batch job

To create a new batch job, use the menu **File → Batch Run → New Batch ...**. A dialog box appears where users can add files from different sources.

- *Add files from the current project*: this will add all the feb files from the current project file.
- *Add files from folder*: this will prompt the user to select a folder on their system. All feb files from the folder will be added.
- *Add files*: this will give users the option to select individual files from the file system.

After adding files, the files are displayed in the dialog box. Additional files can be added or files can be selected and removed using the corresponding buttons at the top of the dialog box. Finally, click OK to create the batch job. The batch job will be displayed in the batch job view, as described below.

### 13.4.2 Open or Save a batch file

An existing batch file can be loaded directly from the file system using the menu **File → Batch Run → Open Batch File**.

A standard file open dialog will be presented where the user can select a batch job file (extension **.fsbatch**). Click OK to load the batch file. The batch job will be displayed in the batch job view, as described below.

To save a batch job, simply use the menu **File → Save** or **File → Save as**.

	file	status	timesteps	iters	nrhs	reforms	norm	runtime (sec.)
1	C:\Users\Steve\source/repos\FEBioChem\Example\ex01.feb	Finished	100	100	200	100	2.675	0.51 (0.36)
2	C:\Users\Steve\source/repos\FEBioChem\Example\ex02.feb	Finished	100	100	200	100	11.146	14.25 (15.52)
3	C:\Users\Steve\source/repos\FEBioChem\Example\ex03.feb	Finished	100	100	200	100	9.3654	1.96
4	C:\Users\Steve\source/repos\FEBioChem\Example\ex04.feb	Finished	100	100	200	100	7.5235	1.86 (1.76)
5	C:\Users\Steve\source/repos\FEBioChem\Example\ex05.feb	Finished	100	100	200	100	115.04	0.96 (0.86)
6	C:\Users\Steve\source/repos\FEBioChem\Example\ex06.feb	Finished	100	100	200	100	12.324	2.53 (2.56)
7	C:\Users\Steve\source/repos\FEBioChem\Example\ex07.feb	Finished	100	100	200	100	4.6151	2.47 (2.61)
8	C:\Users\Steve\source/repos\FEBioChem\Example\ex08.feb	Finished	10	10	20	10	3.1941	0.33 (0.26)
9	C:\Users\Steve\source/repos\FEBioChem\Example\ex09.feb	Finished	10	10	20	10	3.1689 (3.1688)	0.33 (0.28)
10	C:\Users\Steve\source/repos\FEBioChem\Example\ex10.feb	Finished	10	10	20	10	3.2053	0.44 (0.34)
11	C:\Users\Steve\source/repos\FEBioChem\Example\ex11.feb	Finished	10	10	20	10	3.1768	0.37 (0.36)
12	C:\Users\Steve\source/repos\FEBioChem\Example\ex12.feb	Finished	100	100	200	100	13.716	7.85 (7.99)
13	C:\Users\Steve\source/repos\FEBioChem\Example\ex13.feb	Finished	100	100	200	100	15.332	7.91 (8.31)
14	C:\Users\Steve\source/repos\FEBioChem\Example\ex14.feb	Finished	100	164	264	100	7.751	6.51 (6.33)
15	C:\Users\Steve\source/repos\FEBioChem\Example\ex15.feb	Finished	100	164	264	100	7.7328	6.20 (6.34)
16	C:\Users\Steve\source/repos\FEBioChem\Example\ex16.feb	Finished	100	100	200	100	20.676	6.58 (7.09)
17	C:\Users\Steve\source/repos\FEBioChem\Example\ex17.feb	Finished	100	100	200	100	20.676 (20.675)	6.85 (7.06)
18	C:\Users\Steve\source/repos\FEBioChem\Example\ex18.feb	Finished	88	88	176	88	0	0.95 (0.92)
19	C:\Users\Steve\source/repos\FEBioChem\Example\ex19.feb	Finished	88	88	176	88	0	0.94 (0.97)
20	C:\Users\Steve\source/repos\FEBioChem\Example\ex20.feb	Finished	10	10	20	10	0.0011135	0.18 (0.20)

Figure 13.6: The Batch Job View shows the list of files that are part of the batch, as well various metrics that can be used to compare runs.

### 13.4.3 The Batch Job View

After a new batch job is created or a batch file was loaded, the contents of the batch job is displayed in a special window called the **batch job view**.

The view shows a table that lists all the FEBio files in the batch job, as well as several columns with metrics that were collected the last time the batch job was run. (For a new batch job, the columns will be empty.)

At the top of the view, a toolbar shows several buttons that allow you to manage various aspects of the batch job. From left to right:

**Options** The options button opens a dialog box where you can set several settings for running the batch job, such as the number of concurrent processes, the number of threads per process, and the path to the FEBio executable.

**Start** The start button is used to start the batch job. You can either choose to run all the files or just the selected ones.

**Cancel** Once the batch runs, this button can be used to cancel either all pending jobs or all running and pending jobs.

**filter** The filter option allows users to limit the files that are shown based on certain criteria. The criteria can be either a string or regex expression to filter the file list, or a condition to filter based on the metrics. To enter a regex start the filter text with a slash ('/'). To define a condition, start with an equal sign ('='). The condition is formed as follows: *metric operator value*. The *metric* can be any of the names listed on the column headers. The operator is any of <, <=, >, >=, =, ==, != ('=' and '==' are both for *equal than* and '!= means *not equal*). The *value* is any valid numeric value for the corresponding metric. For example, to list only jobs that ran for more than a minute, set the filter to '=runtime>60' (don't include quote symbols).

**add** This button can be used to add additional files to the batch job.

**remove** This button can be used to remove the currently selected files.

**copy** This button can be used to copy the batch jobs and metric to the clipboard as a tab-separated list.

Right-clicking on any batch file prompts a popup menu from which additional actions can be selected. These actions allow you to open the feb file or the corresponding log or plot file in FEBio Studio.

#### 13.4.4 Running the batch job

To run the batch job, click the Start button in the toolbar at the top of the batch view window. Two options are listed:

- **Run all:** select this option to run all the currently visible files in the batch view. If no filter criteria are active, this will run all the files in the batch.
- **Run selected:** only run the files that are currently selected

Before a batch job is run, several options can be set that will affect the batch job. To change these settings, click the **Options** button on the batch view's toolbar. A dialog opens where you can edit the following settings:

- **Nr. of processes:** This is the max number of concurrent processes that will be started when you run the batch job.
- **Nr. threads/process:** This is the number of threads that will be used for each process.
- **FEBio executable:** sets the path of the FEBio executable that you want to use.

Before starting a batch job, it's important to choose these settings wisely. This means that you must know the system resources that you have available. There is often nothing to gain by starting up more processes or choosing more threads than you have processors on your computer. For instance, if you only have 4 cpus, then the product of the number of processes and threads per process should not exceed 4. The default value is 1 for the number of processes, and (default) for the number of threads per process. This usually means that the number of threads is determined by the OS and the applicable environment variables (e.g. OMP\_NUM\_THREADS).

When the batch job is started, all files that will be run change their status to PENDING (shown in the first column next to the file name). When a job is picked up by a process, it changes its status to RUNNING. Files that complete normally will have their status changed to FINISHED. If the file failed to start or error terminated, the status is set to FAILED. Running and pending jobs can be canceled from the toolbar. Pending jobs that were canceled will set their status to CANCELED. Jobs that were already running when canceled will be set to FAILED.

As the individual jobs are run, the columns containing various metrics will be populated. If the batch job was loaded from an existing file, and if the new metrics are different, both the new and the old values are shown, using the format "new value (old value)". The background color will also change. It will be red if the new value is higher than the old value, and it will be yellow if the new value is less than the old value. This makes it easy to compare different runs. Keep in mind that the runtimes will often be different, and some metrics may also show small differences when the jobs were run with multiple threads per process. These are often expected changes and not an indication of a problem. However, if there are more significant changes or the changes occurred when running on a single thread, then that could be a sign of a deeper problem.

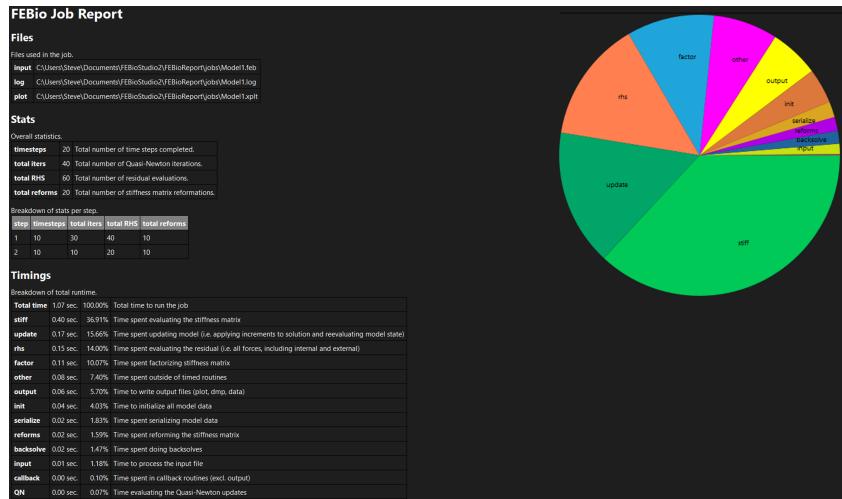


Figure 13.7: The FEBio Job Report provides additional information on a completed FEBio job.

## 13.5 FEBio Job Reports

FEBio Job Reports provide additional details about an FEBio job that was completed. This report can provide insights into how well (or how poorly as the case may be) FEBio ran and help with tracking down possible problems.

By default, FEBio Studio does not generate job reports. To generate a job report after FEBio finishes, make sure to check the corresponding option in the Run FEBio dialog box before running the job (under the Advanced settings). Job reports can also be generated from FEBio log files. Simply drag-and-drop the log file onto the FEBio Studio window and a job report will be created. Job reports generated from log files may not be as detailed as the ones generated directly from an FEBio job.

The job report shows the following information:

- The files that were used and created during the job.
- Some stats about the job as well as a breakdown of the stats per analysis step.
- A color-coded bar chart showing the number of iterations/rhs/reffms per time step. (red means the time step failed, green means the time step completed.)
- Detailed timings of specific FEBio operations.
- A pie chart of these timings.
- A list of all the warnings and errors that were generated during the FEBio job.

# Chapter 14

## The Post Environment

This chapter presents a more detailed overview of the GUI when a plot file is loaded. This configuration of the FEBio Studio UI is called the *Post Environment*.

### 14.1 The Post Environment UI

When a plot file is loaded, the UI of FEBio Studio is modified to hide irrelevant tools and show the post processing tools. In this section we take a closer look at the UI when in Post mode.

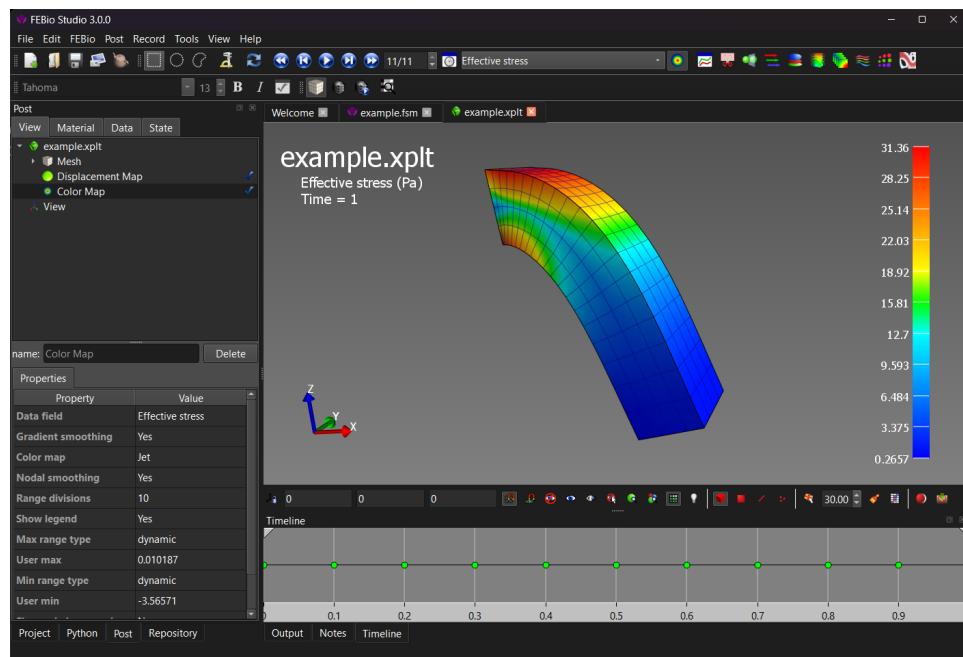


Figure 14.1: The GUI after a file has been loaded.

In the *Post* environment, the Model and Build panels, as well as the Build toolbar, are hidden. The Post panel, Post Tool bar, and the Post menu are shown.

The Post panel has the following tabs.

- *View:* shows a hierarchical section of the content of the model.

- *Material* : displays an overview of the different materials or parts of the model.
- *Data* : allows users to remove, add, or filter data fields.
- *State* : lists all the states in the model and can be used to remove states.

Each of the UI components described above will be described in more detail in subsequent chapters. The menu and toolbars are described in the following sections.

## 14.2 The Post Menu

The post menu allows the user to set many of the options that effect what is being displayed.

- *Plane Cut* - add a clipping plane to the model.
- *Mirror plane* - adds a mirror plane to the model that mirrors the geometry.
- *Vector plot* - adds a vector plot object to the model.
- *Tensor plot* - adds a tensor plot object to the model.
- *Isosurface plot* - adds an isosurface plot object to the model.
- *Slice plot* - adds a slice plot object to the model.
- *Displacement map* - add a displacement map to the model. (Note that for standard FEBio plotfiles, a displacement map is automatically added to the model when loading the file.)
- *Streamlines plot* - adds a stream line plot, which can be useful for visualizing flow fields.
- *Particle flow plot* - adds a particle flow plot, which shows particles flowing in a vector field.
- *Volume flow plot* - adds a volume flow plot, which renders a volumetric “fog” whose density is proportional to the selected field value.
- *Point probe* - adds a point probe, which allows users to track a location on a model and sample the current value.
- *Curve probe* - adds a curve probe, which tracks and samples the value along a user-defined curve.
- *Ruler* - can be used to track relative position and distance between two nodes of the mesh.
- *Muscle Path* - Defines a geodesic curve between two points, i.e. a shortest-path that wraps around geometry. Can be used as a simple representation of a muscle and allows calculation of departure points and moment arms.
- *Vector glyph* - Can be used to show vector data at particular points in the model. (E.g. to show velocity, acceleration, or force on a rigid body.)
- *Helical axis* - Calculates the location and orientation of the helical axis of a rigid body and displays it in the Graphics View.

- *Plot group* - Create a new plot group in the model tree. If a plot group is selected in the model tree when adding a new plot object, the plot object will be added to the plot group. Post model items can also be moved between groups by right-clicking on the item and select “Move to group” from the popup menu.
- *Image slicer* - adds an image slicer, which shows a slice from a 3D image stack.
- *Volume renderer* - adds a volume renderer to the Graphics View, which displays a 3D rendering of a image stack.
- *Image isosurface* - renders an isosurface of a 3D image stack.
- *New Graph* - opens a new graph window.
- *Scatter plot* - Opens a new Graph window that shows a scatter plot of the selected data.
- *Summary* - provides 2D plots of minimum, maximum and average of the data fields.
- *Statistics* - displays a bar chart of the currently displayed data.
- *Integrate* - shows the integration tool.
- *Integrate surface* - Shows the *Integrate Surface* tool, which performs an integration of the currently selected datafield multiplied by the local normal. The components of the resulting vector field are shown as three individual line plots.
- *Import points* - This tool imports point data that can be superimposed on the model. Enter a name for the point data, select the source file, and click *Apply*.
- *Import lines* - This tool imports line data that can be superimposed on the model. Enter a name for the line data, select the source file, and click *Apply*.

### 14.3 The Post Toolbar

The *Post* toolbar offers various shortcuts to the Post menu items.

- |                                                                                     |                                    |
|-------------------------------------------------------------------------------------|------------------------------------|
|  | Return to the first time step      |
|  | Step back by one step              |
|  | Start (and stop) the animation     |
|  | Step forward by one step           |
|  | Go to the last time step           |
|  | Open the time dialog box           |
|  | Activate the contour plot          |
|  | Adds a plane cut plot to the model |
|  | Adds a mirror plane to the model   |
|  | Adds a vector plot to the model    |

-  Adds a tensor plot to the model
-  Adds a isosurface plot to the model
-  Adds a slice plot to the model
-  Adds a stream line plot to the model
-  Adds a particle flow plot to the model
-  Adds an image slicer to the model
-  Adds a 3D volume renderer to the model
-  Adds an isosurface rendering of a 3D image stack.
-  Opens a new graph window
-  Opens the Integrate window

## 14.4 The Graphics View

The Graphics View (GV) is the area of the screen where the model is displayed.

### 14.4.1 Elements of the GV

Aside from a 3D rendering of the model, the GV has several other components to it. These components are referred to as widgets. The image below displays the GV and all of its default widgets.

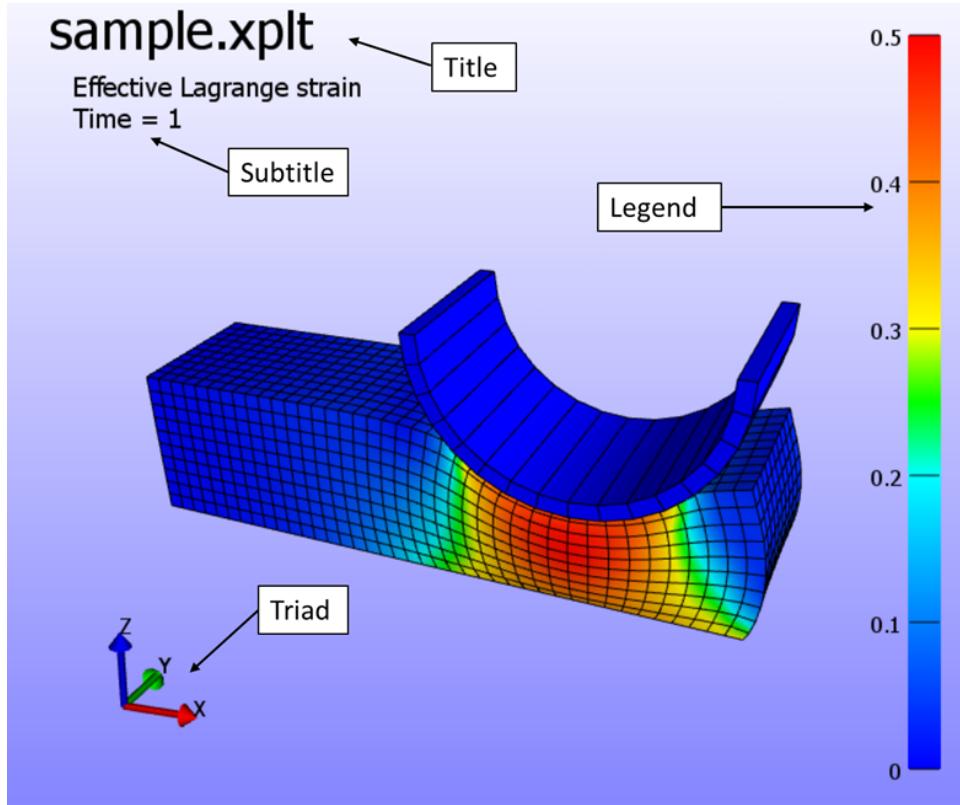


Figure 14.2: The Graphics View components.

The *Title* widget displays the title of the model. The *Subtitle* displays additional information, such as selected field and current time value. The *Triad* indicates the current orientation of the scene. Finally, the *Legend* displays a colored bar and the range of the selected field.

### 14.4.2 Customizing the GV

The user can customize the GV by selecting and moving the different widgets around, and by adding new widgets. This section describes how the user can customize the GV. The following GV widgets are currently available:

- *Text box* - displays a user-defined text.
- *Triad* - displays the orientation of the current view.
- *Legend* - displays a colored legend bar.

#### 14.4.2.1 Selecting and moving widgets

You can select a GV widget by clicking on it with any mouse button. A selection box appears over the widget. The widget can be moved by dragging the box, while holding the mouse button down. If you click and drag the small triangular shaped area in the lower right corner of the selection box, you can resize the widget. Double-clicking on a widget brings up a properties dialog box, where you can modify the widget's properties.

#### 14.4.2.2 Setting the GV widget's properties

After selecting a widget, you can alter its properties by clicking the Properties button from the Widget toolbar. A dialog box appears. You can also bring up this dialog box by double-clicking the widget. Below is an example of the properties dialog box for a text widget.

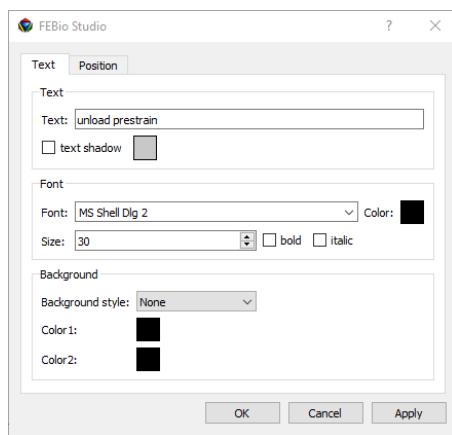


Figure 14.3: Text properties dialog box.

The font type, font size, font color and attributes can be set in this dialog box. You can also set the text to be displayed in the box. The text can consist of literal text and keywords. The keywords, which start with a percentage sign (%), will be replaced by a predefined text. The available keywords are listed in table 1.

keyword	description
%title	Title of the problem. The title is set in the <b>File → Model Info</b> dialog box.
%field	The field variable currently being displayed.
%time	The time value of the current state.
%state	The current state number.
%filename	The name of the file.
%filenitle	The title of the file, i.e. the filename without the extension.
%filepath	The path of the file.

Table 14.1: Text box keywords.

You can also use escape sequences, which start with a backslash (\).

- \n - start a new line.
- \t - start the following text at the next tab position.

#### 14.4.2.3 Adding GV Widgets

FEBio Studio starts with a predefined set of GV widgets. You can add new widgets by clicking one of the GV buttons in the toolbar. Currently, FEBio Studio only supports the addition of a new text widget by clicking on the “Add text box” button, located on the main toolbar.

#### 14.4.2.4 Deleting GV Widgets

GV widgets can be removed by first selecting them and then selecting **Edit** → **Delete** from the menu bar. Note that you cannot delete the predefined widgets, namely the title, subtitle, legend and triad. However, you can hide these objects by selecting the corresponding button on the toolbar.

# Chapter 15

## Saving Graphics

### 15.1 The Capture Frame

The capture frame is the area of the screen that will be captured when taking a screenshot or recording an animation. The capture frame is not always visible, but can be displayed by selecting the **View → Toggle Capture Frame** from the menu. The same menu will hide it again. The capture frame can be shown by selecting the keypad-'0' shortcut. When visible, the capture frame can be moved and resized by selecting one of the borders and dragging the mouse button. To resize it, you must grab the little triangular area in the lower right corner of the capture frame. If you wish to specify the location and size more accurately, you can open the properties dialog box, by selecting the *Properties* button on the widget options toolbar or by double-clicking on one of its borders.

### 15.2 Taking a snapshot

To take a snapshot of the current Graphics View push the *snapshot* button on the toolbar. A standard Save dialog box appears and the desired filename can be entered or selected. Images can be saved as BMP, TIFF or JPEG images. To only capture a part of the Graphics View, make sure the capture frame is active. To activate the capture frame, use the menu **View → Show Capture Frame**, or use the corresponding '0' (number zero) shortcut. The capture frame can be moved by dragging its border or resized by dragging the lower-right corner.

### 15.3 Using the Ray Tracer

Instead of taking a screenshot, you can also create a ray-traced rendering of the current scene. A ray-traced rendering can create higher quality, arbitrary resolution images, which may be more suited for journal articles or other high quality publications. Ray-tracing is often more suitable for rendering shadows and transparent materials.

To create a ray-traced rendering, click the ray-trace button on the main toolbar. A dialog will appear where several settings can be chosen that will affect the final resolution and image quality of the rendering.

**Width, Height** set the resolution of the final image. By default, the dimensions of the active Graphics View are entered, however users can enter any dimensions here. Note that higher

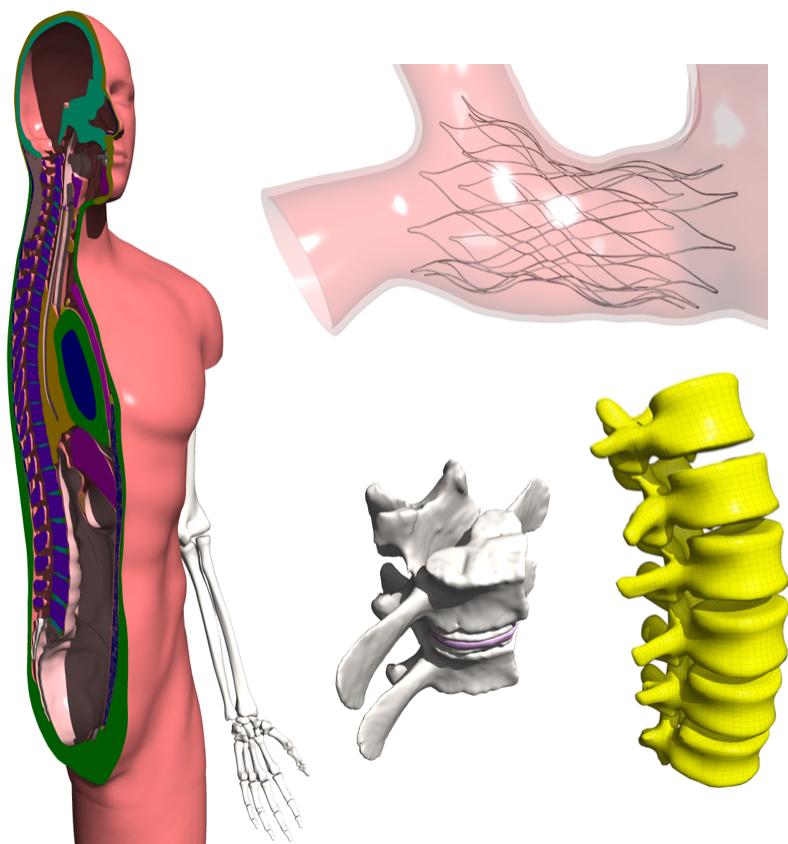


Figure 15.1: A collection of ray-traced images, generated by FEBio Studio.

resolutions will take longer to render.

**Shadows** Check this option if you want shadows to be rendered.

**Shadow Strength** Set the strength of the shadows (range 0..1 from very light to very dark)

**Multisample** Select the multi-sample level. Higher levels of multi-sampling will create better quality images but will take longer to render.

**Background** Select the background option:

**Default**, same as Graphics View

**transparent** make the background transparent

**background color** use the *background color* setting.

**Background Color** Sets the background color. Only used when the **Background** setting is set to *background color*.

## 15.4 Recording an animation

FEBio Studio has the capability to record an animation of the current GV. To record an animation, first position and resize the capture frame so that it covers the desired area of the GV that will be captured. Next, select the **Record** → **New** menu. This opens a standard file dialog box where you can select a file format and enter the target filename. If the selected file format is an image format (.bmp, .tiff, etc.), the target filename will be the file template from which the actual filenames will be generated. Each frame will be stored in a separate file, where the frame number is appended to the file template.

After you have selected a target file, you are ready for recording. Note that the capture frame will now be locked, so you can no longer move or resize it. If it is visible, it will turn red. The recording will begin in a paused state, allowing you to make some changes to the GV before recording begins.

To start recording, select **Record** → **Start** from the menu or press the corresponding shortcut. Now, all the action in the GV will be recorded to the target file. For example, if you press the play button, the GV will loop over all timesteps and each step will be recorded to the file. You can also rotate the GV and this will also be recorded to the file.

To pause the recording, select **Record** → **Pause** from the menu. To finally stop the recording, select **Record** → **Stop** from the menu. This will close the target file and unlock the capture frame.

## 15.5 Camera Control

In FEBio Studio, the camera determines the position and orientation from which the model is viewed. The user can position and orient the camera in several ways. Camera positions can also be saved to make it easier to recover certain preferred positions.

Action	Mouse/Keyboard
To rotate the view	Left MB
To pan the view	Middle MB or ALT+Right MB
To zoom	Right MB
To rotate in the plane of view	ALT+Left MB
To rotate view left/right by small increment	Left/Right arrow key
To rotate view up/down by small increment	Up/Down arrow key

Table 15.1: Mouse and keyboard shortcuts to control view.

### 15.5.1 Basic Camera control

The easiest way to position the camera (and thus change the view) is by using the mouse. Moving the mouse while holding down one of its buttons will modify the view. The action depends on which mouse button is held down. The following lists the possible actions.

The camera position and orientation is displayed under the View item in the Model Viewer. By selecting this item in the Model Viewer, the current camera position and orientation is shown in the properties panel under the Model Viewer. All these fields can be edited manually for precision control over the camera.

### 15.5.2 Element tracking

The element tracking feature in FEBio Studio allows you to track the position and orientation of an element. The camera will then move with this element as it deforms through time.

In order to use element tracking, first select the element that you wish to track. Then, select the **View → Track Selection** menu item. To stop element tracking, clear the selection and then select the same menu item (or use the Ctrl+T shortcut).

### 15.5.3 Camera key-framing

The current position of a camera can be stored in what is called a *viewpoint*. Simply select the **View → Save viewpoint** menu item and a new key item will be created in the Model Viewer. As with the View item, selecting a Key item allows you to edit the position and orientation of that stored camera position. In addition, by selecting the **View → Prev/Next viewpoint**, you can smoothly interpolate between camera key positions. This feature comes in handy when recording an animation and different camera positions need to be visited in a repeatable manner. Camera keys are stored in the FEBio Studio session file so they can be retrieved later.

# Chapter 16

## The Post Panel

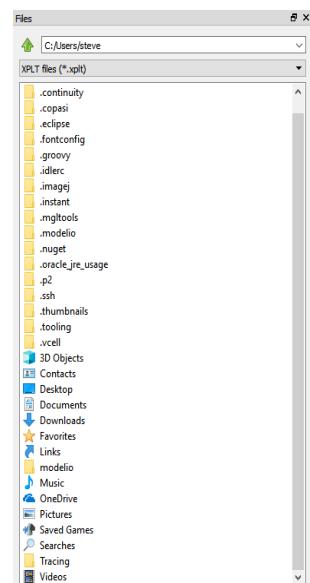
The Command panels are a set of dockable windows that display important information about the model and can also be used to modify the model. The different panels are described next.

### 16.1 The View Tab

The *Model Viewer* gives an overview of the contents of the model. It also gives access to the property window which is used to set many of FEBio Studio's display settings. The Model Viewer is accessed by clicking the "Model" tab button on the Tab Window.

The top part of the Model Viewer shows a tree-view of the contents of the model. Each item can be selected, and when selected, the property window at the bottom lists the available properties for the selected item. Properties can be edited by selecting the corresponding item in the value column.

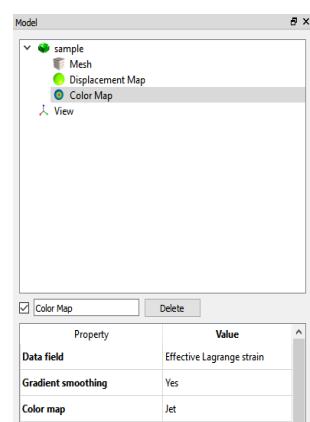
The name edit field in the middle of the Model Viewer can be used to change the name that refers to the selected item. The *delete* button next to the name field can be used to delete a selected item. Note that this only works for items that the user has added to the model. See Chapter 6 for more details on adding items to a model. The *enable* button, the checkbox located next to the name field, can be used to activate the selected item in the Graphics View. What this precisely means will depend on the particular item, but in general this means that if enabled, the item will be displayed in the Graphics View, and if disabled it will be hidden. Note that if an item is disabled, its name will be drawn in italics in the model viewer.



### 16.2 The Material Tab

The *Material Viewer* gives an overview of the different materials (or parts) in the model. It can be accessed by selecting the "Materials" tab.

All the materials in the model are listed in the large window on the left. A material can be selected by clicking on the entry in the list. Multiple materials can be selected by holding down the shift button or the ctrl button.



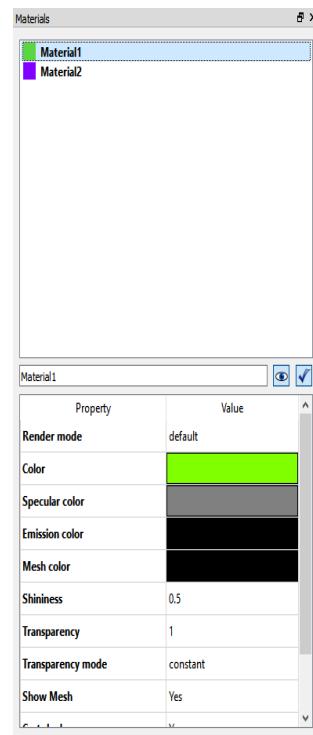
The name of the material can be modified by first selecting it, and then editing the name field, which is located in the center of the material viewer. The enable button, next to the name field, can be used to toggle the state of the material. When a material is disabled, the corresponding part will not be evaluated when visualizing the model data. In that case, it will display with the material colors as defined in the Material Viewer. When enabled, the part will be evaluated, and will be colored according to the local node or element values. See Chapter 6 for more details on this process. The *visible* button, next to the *enable* button, can be used to toggle the visibility of the corresponding part in the Graphics View.

When a material is selected, the property window on the bottom of the material viewer lists the available properties that the user can change.

To edit the appearance of the material, click the colored sphere to open the Appearance Editor. This allows users to set various settings that affect the visual appearance of the material.

Additionally, users can set the following properties:

- *Render Mode*: Allows users to override the default render mode and choose between *solid* and *wireframe*.
- *Opacity*: sets the opaqueness value for the material. Slide the bar to change the value.
- *Opacity Mode*: Chooses between different transparency options
- *Show Mesh*: if checked, the mesh lines will be drawn on top of the model (when mesh lines are visible).
- *Mesh color*: sets the color of the mesh lines
- *Clip*: allows the material to be clipped when a clip plane is defined for the model.



## 16.3 The Data Tab

The *Data Manager* lists all the available data fields that are defined in the model in a table with four columns.

The *Data Field* column gives the name of the data field. When selected, the name can be edited in the field below the data list.

The *Type* of the field identifies the type of the data. The following types are currently defined:

- *float*: single precision floating point
- *vec3f*: a 3D vector with three float components
- *mat3fs*: a 3D symmetric matrix with six float components

The *Class* column identifies the region type for which the data field is defined:

- NODE: data is defined at the nodes of the model

- FACE: data is defined for each facet of the model
- ELEM: data is defined for each element in the model

The *Format* column displays the storage format for the data field. The following values are supported:

- ITEM: One value is stored for each item of the data field's region.
- NODE: One value is stored for each node of the data field's region.
- MIXED: One value is stored for each node of each item of the data field's region.
- REGION: Only one value is stored for the entire region.

In addition, new data fields can be added by clicking the Add button. A menu shows up from which the user can select from several options:

- *Standard*: select from a list of pre-defined data fields. See Appendix A for an overview of currently supported data fields.
- *From file*: load a data field from a data file.
- *Equation*: Enter a mathematical expression that will be evaluated over the mesh.

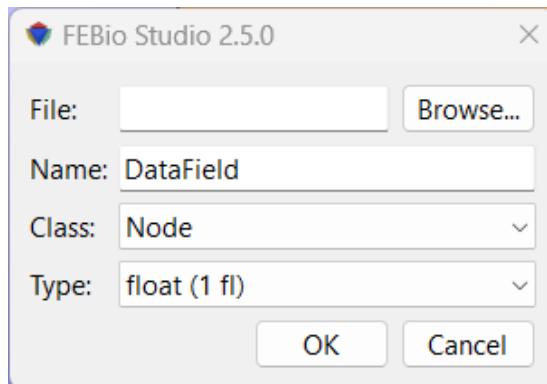


Figure 16.1: Add Data from file dialog box.

### 16.3.1 Adding data from a text file

When selecting the **Add → From file** menu, a dialog box will be shown where you can enter the file name and additional information for parsing the file. The text file containing the data can be selected using the Browse button. The data must be formatted using a comma delimited list, each line corresponds to one item (node, element) and each value on each line corresponds to a single state. The *Name* edit box is used to give a name to the user field, the *class* list identifies whether the data corresponds to a node, element or face, and the *type* list is used to define whether the data is a scalar (float), 3D vector (vec3f) or 3D matrix (mat3fs).

For example, assuming *class* is node, and *type* is float.

```
1, 0.1, 0.2, 0.3
2, 3.2, 3.3, 3.4
3, 1.2, 2.3, 2.4
```

This file defines for nodes 1, 2 and 3 three values, one for each state (assuming the model has three states). Another example, assuming *class* is element, *type* is vec3f.

```
1234, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6
1235, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2
1236, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8
```

This file defines for three elements (elements 1234, 1235 and 1236) a vec3f value for two states: element 1234 has the value (0.1, 0.2, 0.3) for state 1 and (0.4, 0.5, 0.6) for state 2.

### 16.3.2 Adding data via an equation

When selecting the **Add → Equation** menu from the data panel, a dialog box appears where the user can enter the name of the new data field and a mathematical expression that will be evaluated over the entire mesh.

Currently, this only allows the creation of scalar nodal data (Type = float, Class = NODE, Format = ITEM).

You can use the symbols *x*, *y*, and *z* to reference the (time dependent) nodal coordinates, and *t* to reference time.

### 16.3.3 Filtering data

New data fields can be defined by filtering existing data fields. Note that this currently can only be done with the data fields that were loaded from file (not with the standard data fields that FEBio Studio defines).

To create a filtered data field, select a data field in the Data panel and then click the **Filter...** button. A dialog box appears where the following information has to be entered.

- *Name*: The name of the new data field.
- *Filter*: The type of filter to apply to the original data field. (See below)

For each filter additional data needs to be entered. The following filters, and required data, are supported.

- *Scale*: The data will be scaled by multiplication by a scalar.
  - *scale*: the scale factor
- *Smooth*: The data will be smoothed via a Laplace operator.
  - *theta*: weight of the smoothing operator.
  - *iterations*: number of times to apply the operator.
- *Arithmetic*: Apply a simple arithmetic operation.
  - *operation*: select the operation to perform
  - *operand*: The data set that will be used as the right operand (the filtered data field is the left operand).
- *Gradient*: Evaluate the gradient vector of a scalar data field.

### 16.3.4 Exporting Data

Data can be exported from the Data panel. To export data, first select the field variable from the data panel that will be exported. (It doesn't matter which data field is being displayed in the Graphics View.) Then, click the Export button at the top of the Data panel. This will bring up a file-save dialog. Select the location and enter the filename of the file where the data will be stored. Click Save to continue. The next dialog shows some options that allows the content of the output file to be customized. The options are as follows:

- **Selection only:** When checked, only the data associated with the current selection will be exported. Make sure that the selection matches the data "Class". E.g. for element data, elements should be selected; for node data, nodes should be selected. The data class of a data variable is shown in the Class column of the data panel. If this option is not checked, the data of all corresponding mesh items will be output.
- **Write all states:** Write the data for all the time steps of the simulation.
- **Write current state only:** Only export data for the currently active time step.
- **Write states from list:** Enter a comma-separated list of states (i.e. time steps) that will be exported. (e.g., 1,3,5,7. You can also use the short-hand equivalent notation 1:7:2),

After clicking OK, the data will be exported to the output file. The data is exported to a text file as a comma-separated list of values. Each row corresponds to a mesh item. E.g., for nodal data, each row will contain the data of a node of the mesh. For element data, each row contains the data for an element. The values on each row are as follows:

1. Item ID (e.g. node number, or element number)
2. The value(s) of the selected data field for that item at the first time step. The number of values depends on the value type of the field.
  - (a) For scalar fields, there will be one value.
  - (b) For vector fields, there will be 3 values (comma separated).
  - (c) For mat3f fields (3x3 matrix), there are 9 values in row-first ordering (m11, m12, m13, m21, m22, m23, m31, m32, m33)
  - (d) For mat3fs (3x3 symmetric matrix), there are six values (m11, m22, m33, m12, m23, m13)
3. The values are exported for each selected time step

For example, the output for exporting the nodal displacement vector (a vec3f variable) of node 123, for 4 time steps (vectors {0,0,0}, {0.1,0,0}, {0.2,0,0}, {0.3,0,0}) would be as follows.

```
123,0,0,0,0.1,0,0,0.2,0,0,0.3,0,0
```

## 16.4 The State Tab

The *State Manager* allows users to see all states defined in the model. It shows a table and lists in each row the state and its corresponding time value. At the top of the panel several buttons are shown that allow users to modify the state content of the model.

To delete a state, select the state in the table and press the Delete button. This will remove the state from the list.

A state's time value can be edited by pressing the *Edit* button. A dialog box opens where the user can edit the time value.

A new state can also be added. Select the *Add* button. A dialog box appears where a time value can be entered for the new state. The new state will be inserted on the defined time point.

# Chapter 17

## Post Processing

In this chapter the different post-processing tools are discussed that FEBio Studio offers to display the model's data. The user can define a displacement map that FEBio Studio will use to deform the model. The user can also define a colormap that defines the color of the different parts of the model according to the corresponding element or nodal value. The user can also add different types of plots to display data. Plane cuts can be made to conveniently hide parts of the model and to inspect the interior of the model.

### 17.1 Properties of the Model

When selecting the root item in the View tab of the Post panel, the user can edit the properties of the model. The following properties are defined.

- *Element subdivisions*: defines the number of subdivision levels. FEBio Studio subdivides the elements in the model to improve the quality of the renderings. However, increasing the number of subdivisions will also increase the time it takes to render the model so use this property judiciously.
- *Render mode*: Sets the render mode to solid or wireframe.
- *Render undeformed outline*: when selected *yes*, FEBio Studio will draw an outline of the model in its undeformed state.
- *Outline color*: set the color for the model outlines
- *Node color*: set the color of the nodes
- *Selection color*: set the color of selected items
- *Render smooth* - when selected *yes*, FEBio Studio will render the model smoothly. This means that it will vary the surface normal of the model to create a smooth surface. When selected *no*, flat shading is used to render the model. Note that smooth shading can be significantly slower than flat shading due to the additional calculations that FEBio Studio needs to do.
- *Shells as hexes* - renders all shells as hexahedral elements using the shell thickness (if available) to extrude the shell surface in the normal direction.

- *Shell reference surface* - defines how FEBio Studio needs to interpret the shell surface in relation to the shell volume.
- *Smoothing angle* - identifies the hard edges in the model. Edges with adjoining faces that have a surface normal, whose angle is more than the smoothing angle, are considered hard.

## 17.2 Displacement Map

FEBio Studio uses the displacement map to define the deformation of the model at each state. The displacement map can be accessed in the Model Viewer by selecting the item entitled *displacement map* (this is a sub-item of the *model* item). Click on it to display the properties in the Properties Window, located at the bottom of the Model Viewer. The Displacement map has two parameters that the user can change.

- *Data field*: is the data field that FEBio Studio will use to calculate the model's deformation.
- *Scale Factor*: sets the displacement scale factor, which can be used to scale the displacements when displaying the model in the Graphics View. (This scale factor does affect strain measures.)

Note that you must enable the displacement map by clicking on the *enable* button in the Model Viewer (this is the button with the checkmark). Similarly, by clicking the button again the displacement map can be deactivated in which case the model will not be deformed when displaying the different time steps.

## 17.3 Color Map

The *Color Map* defines how FEBio Studio will calculate the color that is used to display the model in the Graphics View. The Color Map's properties can be accessed by selecting the item in the Model Viewer entitled *Color Map*. The properties will then be listed in the Properties Window below. The following properties are defined for the color map.

- *Data field* - allows the user to select the data field that FEBio Studio will use to define the color of the model. The color of the model is defined by the data field and the color gradient (see below).
- *Gradient smoothing* - Colors the material by drawing a fringe plot of the selected data field. When gradient smoothing is on, the fringe colors are smoothed to produce a continuous transition between fringes. When this option is off, a discrete set of colors is used instead.
- *Color map*: defines the color map that will be used to color the model.
- *Nodal values* - FEBio Studio defines nodal data and element data. When displaying element data, the element data is projected to the nodes to produce a smooth rendering of the data. However, when this option is turned off, element data will be displayed by using a single color for each element. This will produce a discretized drawing of the data, but will be truer to the actual data since the element data is not interpolated before displaying.

- *Range type* - FEBio Studio keeps track of the range of the selected data field (that is the minimum and maximum values). The user can use this option to select a *dynamic* range (range is updated for each state independently), a *static* range (range is calculate over all the states) or *user range* (user defines the minimum and maximum values).
- *Range divisions* - changes the number of fringes that will be drawn on the model.
- *Show legend* - allows you to toggle the displaying of the legend bar in the Graphics View.
- *Legend Orientation*: sets the orientation of the legend bar (horizontal or vertical)
- *User max/ User min* - defines the minimum and maximum range value when using the *user* range option for *range type*.

When FEBio Studio starts, the colormap is turned off by default. To activate it, select the *enable* button in the Model Viewer. Similarly, to disable the color mapping, simply press this button again. When the colormap is turned off, FEBio Studio will use the material colors to display the model in the Graphics View.

The color map displays scalar values. For tensor variables, the user can select the sub-components of the tensor as well as some typical scalar quantities that can be derived from the tensor. The following tables explain these components for the different tensor types.

Vector field components (vec3f)	Description
X	The X-component of the vector
Y	The Y-component of the vector
Z	The Z-component of the vector
XY	$\sqrt{X^2 + Y^2}$
YZ	$\sqrt{Y^2 + Z^2}$
XZ	$\sqrt{X^2 + Z^2}$
Total	$\sqrt{X^2 + Y^2 + Z^2}$ (i.e. the vector norm)

Table 17.1: Vector field components

## 17.4 Plane Cuts

FEBio Studio allows the user to add a plane cut to the model. The plane cut defines a clipping plane which will hide all geometry on the positive side of the plane. To add a plane cut to the model, select the **Post → Plane cut** from the menu. A new item will show up the Model Viewer that the user can select and edit the properties off. The following properties are defined for the plane cut.

- *Show plane* - shows or hides the plane in the Graphics View. Note that selecting *no* for this option does not disable the plane cut. It simply toggles the displaying of the plane. To disable the plane cut, you need to click the *enable* button (see below).
- *Cut hidden*: If yes, hidden parts will be cut. If no, hidden parts are not cut.
- *Show Mesh*: show the mesh on the plane cut.
- *Transparency*: Set the transparency value of the plane.

Symmetric second-order tensor components (mat3fs)	Description
X	The X-component of the tensor
Y	The Y-component of the tensor
Z	The Z-component of the tensor
XY	The XY-component of the tensor
YZ	The YZ-component of the tensor
XZ	The XZ-component of the tensor
Effective	The effective or “von-Mises” value: $\sqrt{\text{dev}\mathbf{A} : \text{dev}\mathbf{A}}$
1-Principal	The first (i.e. largest) eigenvalue $\lambda_1$
2-Principal	The second (i.e. middle) eigenvalue $\lambda_2$
3-Principal	The third (i.e. lowest) eigenvalue $\lambda_3$
1-Dev Principal	The first eigenvalue of the deviatoric tensor.
2-Dev Principal	The second eigenvalue of the deviatoric tensor.
3-Dev Principal	The third eigenvalue of the deviatoric tensor.
Max Shear	The “max shear” component: $\max 0.5(\lambda_i - \lambda_j) $

Table 17.2: Symmetric second-order tensor field components

- *X-normal* - sets the  $x$  component of the plane normal. Aside from entering the value directly, this edit field can also be used as a slider. You can change the value by click+dragging with the left mouse button.
- *Y-normal* - sets the  $y$  component of the plane normal. Aside from entering the value directly, this edit field can also be used as a slider. You can change the value by click+dragging with the left mouse button.
- *Z-normal* - sets the  $z$  component of the plane normal. Aside from entering the value directly, this edit field can also be used as a slider. You can change the value by click+dragging with the left mouse button.
- *offset* - sets the relative position of the plane with respect to the center of the model. Aside from entering the value directly, this edit field can also be used as a slider. You can change the value by click+dragging with the left mouse button.

Multiple plane cuts can be added to the model (up to six) and each plane can be positioned independently of the others. Note that the color of the plane in the Graphics View hints to its orientation using a RGB color coding: a red plane is a plane whose normal is directed in the  $x$ -direction, a green plane has its normal in the  $y$ -direction and a blue plane has its normal in the  $z$ -direction. A general orientation uses a combination of red, green and blue to indicate the orientation. You can also hide the displaying of the plane by setting the *Show plane* property to the appropriate value.

To enable the plane cut, press the *enable* button in the Model Viewer. To disable it, press the button again. You can delete the plane cut altogether by clicking the *delete* button on the Model Viewer panel.

## 17.5 Mirror Plane

The mirror plane allows users to mirror the geometry along one of the principal axis (X, Y, Z). To add a mirror plane, select the menu **Post → Mirror Plane**, or select the corresponding button on the post toolbar. The mirror plane defines the following options.

- *Mirror plane* - Define the reflection direction.
- *Show plane* - Show the mirror plane itself or not.
- *Transparency* - Set the transparency of the mirror plane.
- *Offset* - Defines a translation of the mirror plane.

To enable the mirror plane, check the check button next to the name field.

## 17.6 Vector Plot

A *vector plot* of the model can be added by selecting the **Post → Vector plot** menu. This will add a vector plot item to the Model Viewer. The user can select the item in the Model Viewer to edit the properties of the vector plot. The following properties are defined.

- *Data field* - selects the vector field that will be rendered.
- *Allow clipping*: If yes, the vector field will be clipped by any active plane cuts.
- *Density* - sets the density of the vector field. A density of one will draw a vector at each node of the model. Lower values will draw a vector only at randomly selected nodes. The lower the value, the less vectors are drawn.
- *Glyph* - a small graphic that will be drawn at each node to represent the vector. This option allows the user to select the glyph.
- *Glyph color* - allows the user to set how the color of each glyph is determined. Currently, the options are:
  - *Solid* - draws each glyph in the same color (also see *Solid color*).
  - *Length* - draws the glyph with a color that relates to its length (also see *Gradient*).
  - *Orientation* - draws the glyph using RGB color coding to indicate the orientation of the vector.
- *Solid color* - is the color that is used when *Solid* is selected as the glyph color.
- *Normalize* - when selected *yes*, all vectors will be normalized. This implies that all vectors will be drawn with the same length. Otherwise the size of the glyph will be representative to the size of the corresponding vector.
- *Auto-scale*: scale the vectors automatically.
- *Scale*: scales the size of the glyphs.

The *enable* button on the Model Viewer panel can be used to toggle the vector plot on and off. The *delete* button can be used to remove the vector plot from the model.

## 17.7 Isosurface plot

An isosurface plot draws a surface through all the points of the model that have the same value. An isosurface plot can be added by selecting the **Post → Isosurface plot** from the menu. A new item will appear in the Model Viewer that the user can select to edit the surface plot's properties. The following properties are defined.

- *Data field* - selects the data field that will be used to calculate the isosurfaces.
- *Allow clipping*: If yes, isosurfaces will be clipped by any active plane cuts.
- *Gradient* - sets the color gradient that will be used to color the isosurfaces.
- *Slices* - defines the number of isosurfaces to draw.
- *Show legend* - shows a legend bar for the isosurface plot.
- *Smooth* - when selected *yes*, the surfaces will be drawn using smooth shading. When selected *no*, the isosurfaces are drawn with flat shading. Note that when using smooth shading, additional calculations need to be performed which may slow down the rendering of the plot.

The isosurface plot is enabled by selecting the *enable* button on the Model Viewer panel. Similarly, disabling the plot can be done by pressing the same button again. You can delete the isosurface plot by pressing the *delete* button on the Model Viewer panel.

Since the isosurface plot is drawn inside the model, it is advisable to hide it. You can hide the model by hiding each individual material, or by clicking the *enable* button when the *model* is selected in the Model Viewer.

## 17.8 Slice plot

A slice plot draws the intersection of the model with a series of planes. The planes can be oriented by the user. To add a slice plot, select the **Post → Slice plot** from the menu. A new item will show up in the Model Viewer which the user can select to edit the slice plot's properties.

- *Data field* - selects the data field that will be used to color the planes.
- *Gradient* - sets the gradient that will be used to color the planes.
- *X-normal* - sets the x-component of the plane normal.
- *Y-normal* - sets the y-component of the plane normal
- *Z-normal* - sets the z-component of the plane normal
- *Show legend* - shows or hides the legend bar for this plot.
- *Slices* - selects the number of slicing planes.

The slice plot can be enabled or disabled by pressing the *enable* button in the Model Viewer. The plot can be deleted by selecting the *delete* button. Since the slice plot is drawn inside the model, it is advisable to hide the model. You can hide the model by hiding each individual material, or by clicking the *enable* button when the *model* is selected in the Model Viewer.

## 17.9 Tensor plot

The tensor plot renders a glyph based on a second order tensor data field. Users can plot eigenvectors (for symmetric tensor fields), or the columns or rows of the matrix.

- **Data field:** selects the data field that will be used for the tensor plot.
- **Calculate:** sets the option to generate vector data that is used to generate the glyphs.
- **Color map:** sets the color map used for coloring the glyphs.
- **Allow clipping:** Allows the plot to be clipped by cutting planes or not.
- **Show hidden:** Show the plot on hidden materials or not.
- **Scale:** sets the scale factor of the glyphs.
- **Density:** sets the glyph density. When this value is less than one, glyphs are drawn on random elements, selected based on this density value.
- **Glyph:** sets the glyph that is used for rendering.
- **Glyph Color:** sets the option of how the glyphs are colored.
- **Solid Color:** sets the color that is used when the *glyph color* setting is set to *Solid*.
- **Auto-scale:** scales the glyphs automatically based on the overall mesh size.
- **Normalize:** Normalize the vectors when drawing.

## 17.10 Streamline Plot

Streamlines are useful for visualizing fluid flows. The streamlines are calculated by integrating a vector field at particular seed points. The seed points are determined automatically based on the vector field. Everywhere the flow enters the mesh, a seed point is placed at the center of the corresponding facet.

- **Data field:** selects the data field that will be used for the streamline plot.
- **Color map:** sets the color map used for coloring the glyphs.
- **Allow Clipping:** Allows the plot to be clipped by cutting planes or not.
- **Step size:** Sets the integration step size. A smaller value will produce more accurate flows, at the cost of a large computational expense.
- **Density:** Sets the density of streamlines. When this value is less than one, streamlines are seeded on random facets, selected based on this density value.
- **Velocity threshold:** The minimum value for the velocity for seeding.
- **Range type:** Sets how the range of the corresponding colormap is determined.
- **Range divisions:** Sets the number of intervals for the colormap (and associated legend)
- **User Range min:** The minimum value for the range when the *range type* is set to *user*.
- **User Range max:** The maximum value for the range when the *range type* is set to *user*.

## 17.11 Particle Flow Plot

Particle flows are useful for visualizing fluid flows. The particles are seeded at the influx boundary of the flow, which is determined automatically. Everywhere the flow enters the mesh, a seed point is placed at the center of the corresponding facet. The particles then move with the flow, their motion determined by integrating the fluid flow.

- **Data field:** selects the data field that will be used for the particle flow plot.
- **Color map:** sets the color map used for coloring the glyphs.
- **Allow Clipping:** Allows the plot to be clipped by cutting planes or not.
- **Seed step:** The time step at which the particles are seeded.
- **Velocity threshold:** The minimum value for the velocity for seeding.
- **Seeding density:** Sets the density of particles at the seed step. When this value is less than one, particles are seeded on random facets, selected based on this density value.
- **Step size:** Sets the integration step size. A smaller value will produce more accurate flows, at the cost of a large computational expense.
- **Show path lines:** Show the path of the particles or not.

## 17.12 Additional Windows

### 17.12.1 Summary Window

The Summary Window displays a graph of the minimum, maximum, and average values as a function of time for the selected expression. This summary of values can be calculated using only selected elements or nodes, or for the entire model when no elements or nodes are selected. It can be opened by selecting the **Post → Summary**. Figure 9 shows the *Summary Window*.

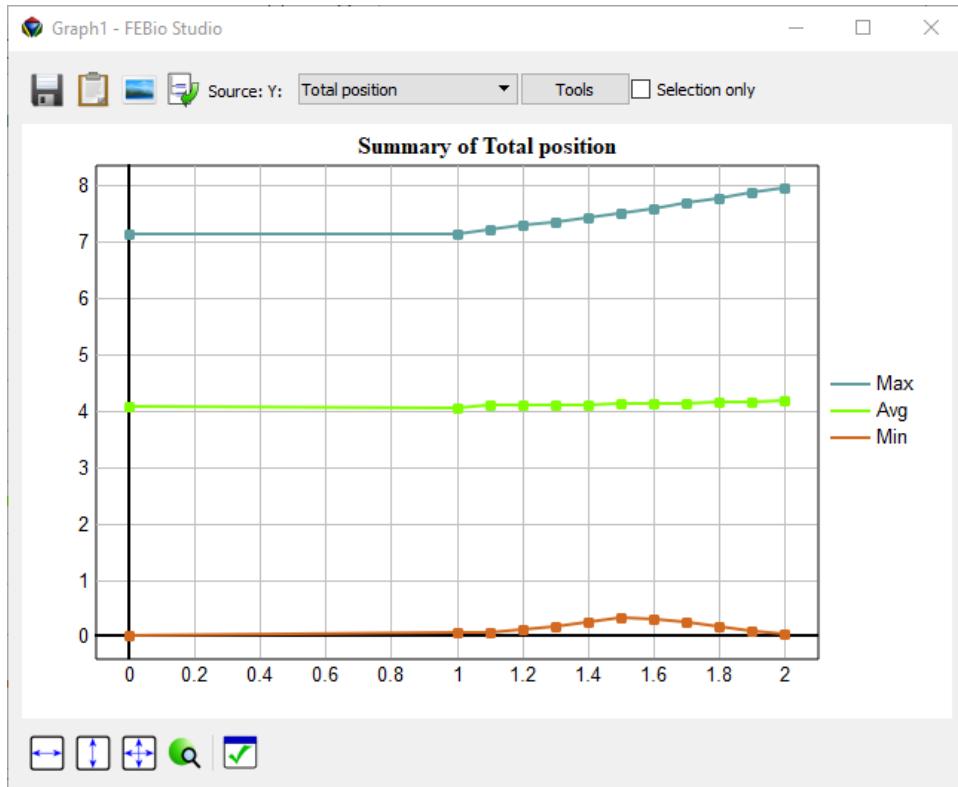


Figure 17.1: The Summary Window can be opened from the **Post → Summary** menu.

To select the expression to display, click on the drop-down box in the upper left corner and select the desired expression. Each data point can be selected by clicking on it. The exact values will appear next to a selected data point.

To save the summary data to file, click the *Save* button located at the bottom of the *Summary View*. This will open the File Save dialog box. After the user enters a filename, the data is saved to file as a simple ASCII file. The data can also be copied to the clipboard with the “Copy to Clipboard” option.

The *Options* button shows a dialog box where the user can change some options.

The graph area can be scaled or moved by click+dragging the right and left mouse button respectively. The buttons in the lower left corner of the Summary window can be used to restore the x-range, y-range or both.

### 17.12.2 Graph Window

The Graph window can be used to show time history plots and scatter plots for selected items. To open a *Graph window*, select the **Post → new Graph** menu. Multiple graph windows can be displayed simultaneously.

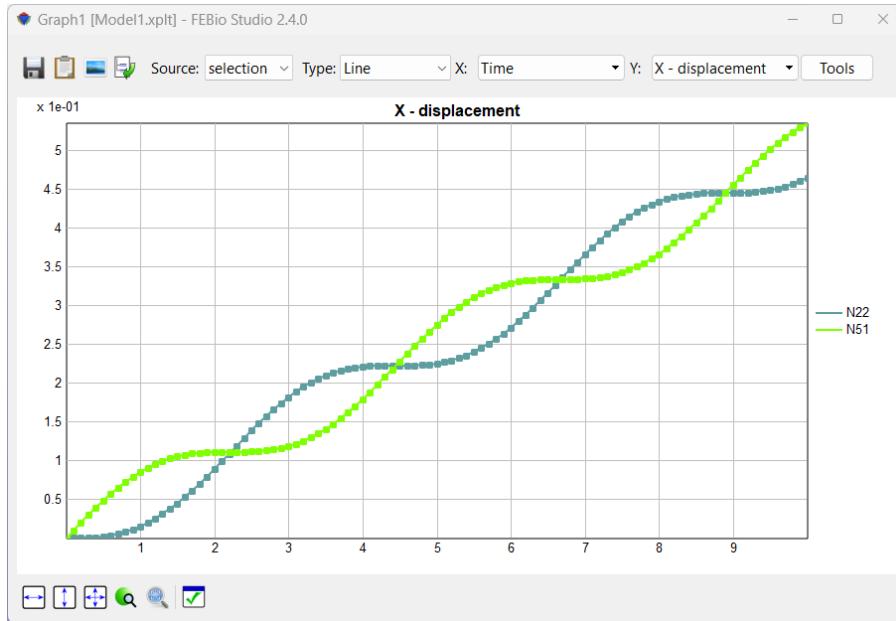


Figure 17.2: A Graph window displaying the time history of selected mesh items.

The *Graph window* displays the selected expression for the selected mesh items (see below on how to select mesh items). On the right, the legend shows the item numbers (preceded by an 'E' for 'element', 'N' for 'node', 'F' for 'face', 'C' for edge) and they are shown in the same color as the corresponding curve.

Each Graph window has a toolbar that offers the following functionality.

- The *Save* button will save the displayed data to a text file.
- The *Clipboard* button can be used to store the data values on the clipboard. This data can then be pasted in other software that allows clipboard operations.
- The *Type* selection box allows the user to select from different types of plots. The following types are currently supported:
  - *Line* - displays a curve that represents the evolution of the selected data field as a function of (pseudo-) time. For this type, the user can only select the value for the y-axis; the x-axis will show the time.
  - *Scatter* - displays an x-y plot. In this case, the user can select different data fields for both the x- and y-axis.
  - *Time-scatter*: like scatter plot, but points are connected by time value
- *Tools*: Shows the graph tools panel, which is discussed in more detail below.

The graph area can be scaled or moved by click+dragging the right and left mouse button respectively. The buttons in the lower left corner of the Graph window can be used to restore the x-range, y-range or both.

### 17.12.3 Graph Tools

Each Graph window offers a tools panel that allows access to additional features that affect what is displayed in the graph area.

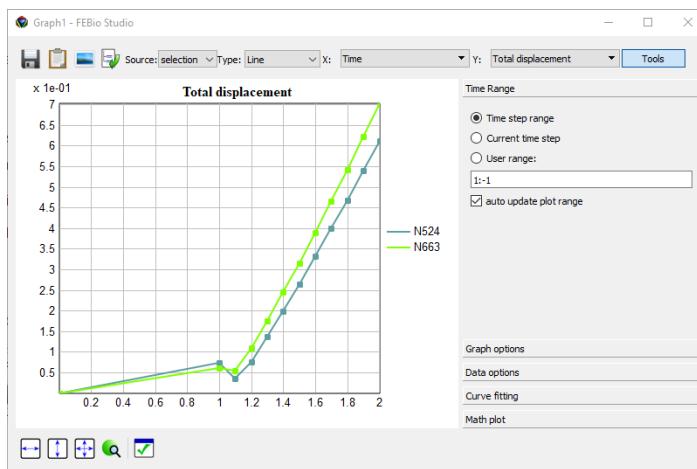


Figure 17.3: A Graph window with the Tools panel expanded.

The following tools are currently supported.

- *Options*: Set various settings that affect what is shown and how the data is displayed.
- *Linear Regression*: Do a linear regression on the first shown curve.
- *Math plot*: Enter a mathematical expression, using  $x$  as the ordinate, which will be displayed on the top of the graph.

### 17.12.4 Selecting mesh items

You can select nodes, edges, faces, and elements. The item that will be selected is controlled by the selection buttons in the *Toolbar*:

-  Switches to node-selection mode.
-  Switches to edge-selection mode.
-  Switches to face-selection mode.
-  Switches to element-selection-mode.

To add an item to the current selection, just shift+click the item. When tags are enabled, a dot followed by the item's number will appear next to the item. To enable the tags, select the corresponding button on the toolbar. To remove an item from the current selection, ctrl+click the node or element. You can (de-) select multiple items at the same time by dragging the mouse cursor while holding down the shift or ctrl key and the left mouse button. A colored rectangle will

appear indicating what elements or nodes will be selected. Note that only *visible* elements or nodes that fall inside this rectangle will be selected. This means that only elements or nodes on the surface of the mesh can be selected.

When no other windows are open (such as e.g. Graph windows, etc.), pressing the ESC-key will clear the selection. You can also select the **Edit** → **Clear Selection** menu to clear the entire selection.

Also note that the *Edit* menu lists several options to manipulate the current selection, such as hiding, un-hiding, inverting, etc.

### 17.12.5 Integration Tool

The *Integration tool* allows you to calculate the integral over a selected region. To use it, first select nodes, edges, faces, or elements. Then, activate the Integration tool from the **Post** → **Integrate** menu. A window appears with a graph that represents the integral of the selected region as a function of time. Depending on the selection, the graph represents different things. For nodes, it is the sum of the values all the selected nodes, for edges it is the line integral, for faces it is the surface integral of the selected surface, and for elements it is the volume integral over the volume of the selected elements. The *Save* button saves the results to a text file and the *Clipboard* button copies the data to the clipboard.

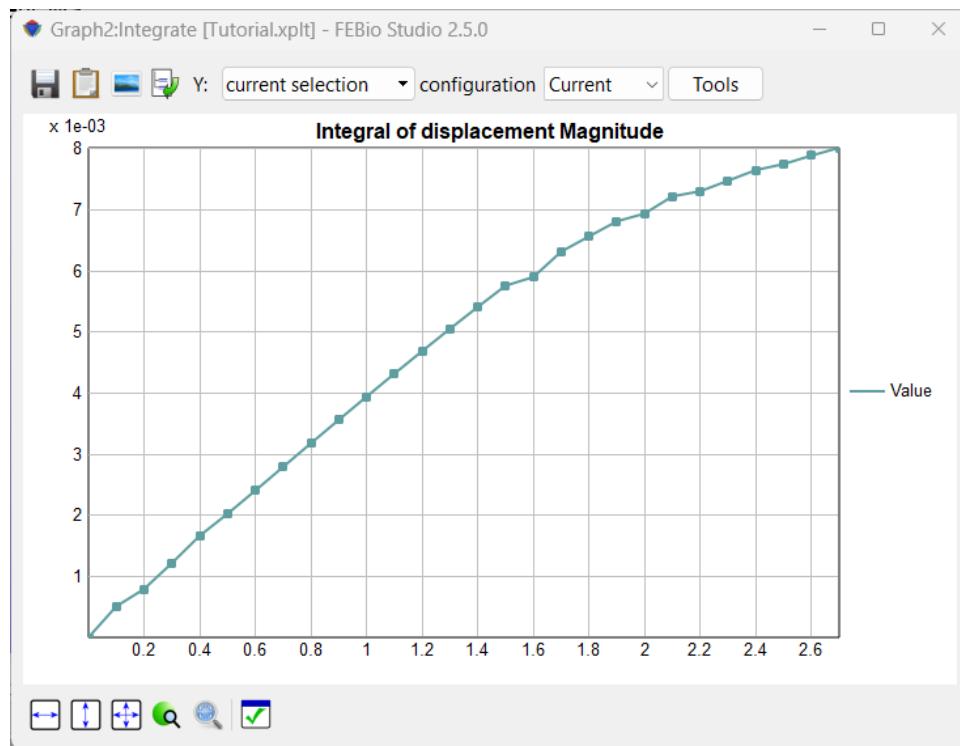


Figure 17.4: The Integration tool can be used to calculate the definite integral over the selected region.

## 17.13 Post Session Files

The post model, i.e. the results and all the plots and other items added, can be saved to a Post Session File (PSF). The PSF is an xml-formatted file that contains the contents of the post model, and all the information necessary for FEBio Studio to reconstruct the model. The default extension for a PSF is *fsps*.

To create a PSF, first load a model into the Post side of FEBio Studio (e.g. by loading an *xplt* file). Then, go to **File** → **Save as...**, select the “FEBio Studio Post Session” file filter, choose a name, and enter Save. Similarly, to open a PSF, go to the menu **File** → **Open Model**, select the “FEBio Studio Post Session” filter, choose the file, and click Open.

The rest of this section describes the content of the PSF file in some detail.

### 17.13.1 PSF File Structure

The Post Session File (PSF) is an xml-formatted file. The root tag is *febiostudio\_post\_session* and it takes a required *version* tag, which has to be set to “1.0”.

```
<febiostudio_post_session version="1.0">
...
</febiostudio_post_session>
```

The remainder of the file will contain tags that change default attributes or add other model components (e.g. plots). The following table shows a list of the supported tags. The subsequent sections will discuss each tag in more detail.

tag	Description
model	Open a model file
material	Set material properties
datafield	Set datafield parameters
plot	Add a plot

Table 17.3: Supported PSF File tags

#### 17.13.1.1 model

The first XML element in the file will usually be the *model* tag, which is used to read a file, e.g. an FEBio *xplt* file. The *file* attribute specifies the filename.

```
<model file="/path/to/some/file"/>
```

At this point, FEBio Studio will try to read the file. If reading the file fails, FEBio Studio will not process the rest of the session file.

The *model* tag can also contain a *type* attribute instead of a *file* attribute. This is used for creating models using a specialized tool. This can be used for instance for creating models via the *Kinemat* tool. See the examples below for how this works.

### 17.13.1.2 material

The *material* tag allows users to set default material properties for the materials that were automatically created when reading the model file. The following table shows a list of available material parameters.

property	Description
diffuse	Set the diffuse color for the material
ambient	Set the ambient color for the material
specular	Set the specular color for the material
emission	Set the emission color for the material
mesh_color	Set the color for rendering mesh lines
shininess	Set the shininess parameters (between 0 and 1)
transparency	Set the material transparency (between 0 and 1)

Table 17.4: Material properties

### 17.13.1.3 datafield

The *datafield* element is used to set parameters for data fields that are loaded from the model data file. This element requires a *name* attribute that identifies the specific data field.

*Example:*

```
<datafield name="Lagrange strain">
  <ref_state>2</ref_state>
</datafield>
```

### 17.13.1.4 plot

The *plot* tag will add a specific type of plot to the model. The plot type is specified by the *type* attribute. The following plots can be added.

*Comments:*

1. The line plot requires a *source* property that defines the file from which the line data will be read.

## 17.13.2 PSF Examples

A minimum PSF file that reads in an xplt file will look like this.

```
<febiostudio_post_session version="2.0">
  <model file="/path/to/some/file.xplt"/>
</febiostudio_post_session>
```

Assume an xplt file contains two materials. The following example reads the xplt file and sets the default color properties for the materials.

<b>type</b>	<b>Description</b>
iso-surface	Add an iso-surface plot
lines	Add a line plot (1)
points	Add a point cloud plot
mirror	Add a mirror plot
planecut	Add a planecut plot
probe	Add a probe plot
ruler	Add a ruler plot
slices	Add a slices plot
streamlines	Add a streamline plot
tensor	Add a tensor plot
vector	Add a vector plot
volume-flow	Add a volume flow plot

Table 17.5: Plot types

```
<febiostudio_post_session version="2.0">
    <model file="/path/to/some/file.xplt"/>
<material id="1">
<diffuse>240,164,96</diffuse>
<ambient>240,164,96</ambient>
</material>
<material id="2">
<diffuse>128,128,0</diffuse>
<ambient>128,128,0</ambient>
</material>
</febiostudio_post_session>
```

The next example shows how to load an xplt file and add a line plot that reads data from a separate file.

```
<febiostudio_post_session version="2.0">
<model file="/path/to/some/file.xplt"/>
<plot type="lines">
<source file="/path/to/lines/file.ang"/>
</plot>
</febiostudio_post_session>
```

This example illustrates the use of the *kinemat* tag to read and construct a post model.

```
<febiostudio_post_session version="2.0">
<model type="kinemat">
<model_file>geometry.k</model_file>
<kine_file>kine_data.txt</kine_file>
<range>1,999,1</range>
</model>
</febiostudio_post_session>
```

# Chapter 18

# Visualizing 3D Image Data

FEBio Studio has several capabilities for rendering and overlaying 3D image data. In order to visualize a 3D image, you must first load a 3D image stack into the active model. Then, you need to attach one of the 3D image rendering tools to the image stack.

## 18.1 Loading 3D image data

FEBio Studio uses SimpleITK to read and manipulate image data, and so it supports reading a large number of image formats. At the moment, FEBio Studio does only support single channel, 8-bit images. When a higher bit image is loaded in, it is scaled down to 8 bits. Multichannel images (including color images) cannot be read in at all. To load image data, select the **File → Import Image** from the main menu, and then select the type of image that you would like to import, which will show the standard File Open dialog box. Locate the image file on the file system and click OK.

By default, the image data will be loaded into the active model. Thus, if you wish to view the image on top of a FE model, you first need to load that model, make sure it is active one, and then load the image data. When the image data is loaded in, a 3D volume renderer will be added to your model, displaying the image.

### 18.1.1 Supported Image Types

**Raw** Raw images contain no header information, and so in order to import them, you must provide information about the physical location of the image, and the physical and pixel dimensions of the image. When you import a raw image, a dialog box will appear where you can enter this information.

**DICOM** DICOM images are read using SimpleITK. In order to load in a sequence of DICOM images, simply select one of the DICOM files from the File Dialog. FEBio Studio uses built-in SimpleITK functionality to parse the header data of the DICOM images to locate and read all of the files that belong to the same dataset as the file you specified.

**TIFF** TIFF images are read using SimpleITK. To load in a TIFF image, simply select it from the File Dialog.

**OME TIFF** OME TIFF images are read using SimpleITK. To load in an OME TIFF image, you should select the XML file containing the OME header data from the File Dialog.

**Image Sequence** If you have a sequence of 2D image files that together make up a single 3D image, you can load them in using this option. Simply select all of the images files that correspond to your image stack in the File Dialog. To ensure that they are loaded in the proper order, please make sure that they are organized alphabetically by file name. Do not use this option to load in a sequence of DICOM files. Use the DICOM option instead.

## 18.2 Image GUI Environment

FEBio Studio has various GUI elements dedicated to viewing and manipulating image data. Some of these GUI elements only appear when an image is selected from the model tree in an open model.

### 18.2.1 Properties Box

The image properties box will appear in the Model panel when you select an image in the model tree. The box has three tabs: *Properties*, *Filters*, and *Histogram*. The Properties tab shows the physical bounds of the image in real space, and allows you to edit them. The Filters tab allows you to add, edit, and apply various filters to the images (please see the section on Filters for more information). The Histogram tab displays a histogram of the data from the selected image.

### 18.2.2 3D Image Settings Panel

The 3D Image Settings Panel will only be visible after an image has been selected. It allows you to customize various display settings for the current image. These display settings will be remembered for each image, and will be used across all display modes for the image. The settings include:

- **Alpha Scale:** Sets the overall alpha scale, which scales the transparency of the image.
- **Gamma correction:** Sets the gamma correction factor, which applies a nonlinear scaling of the image intensity.
- **Hue/Saturation/Luminance:** Sets the hue (or color), saturation, luminance level of the primary color.
- **Intensity:** Set the minimum and maximum cutoff image intensity. Voxels with an intensity below the minimum (maximum) value will be rendered with a transparency set to the value of the minimum (maximum) value of the *alpha range* setting.
- **Alpha Range:** Sets the minimum and maximum alpha (transparency) values.
- **Clip X/Y/Z:** Sets the min and max image clipping values for the X, Y, and Z planes. This allows users to only show a region of interest instead of the whole 3D image.

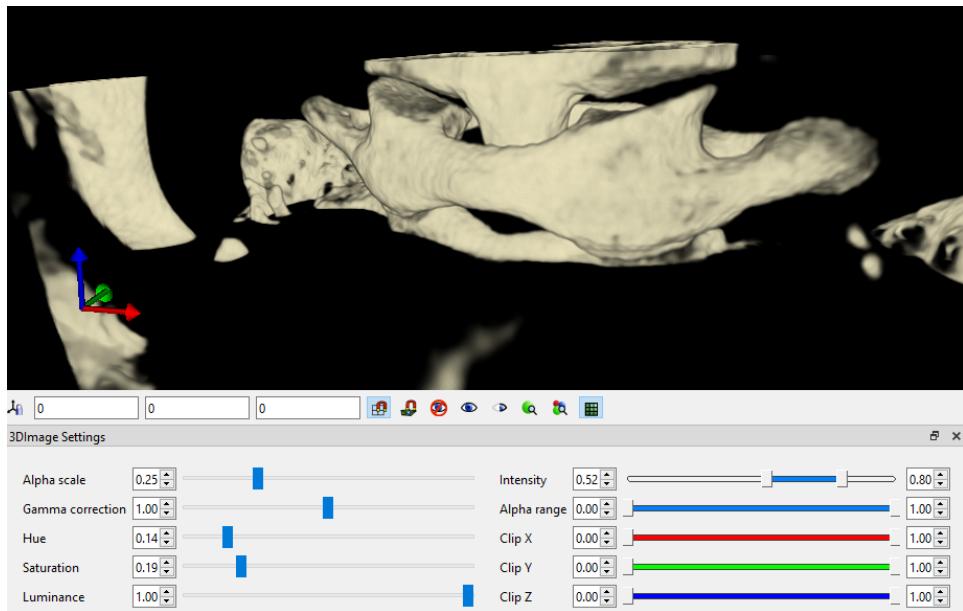


Figure 18.1: The 3D Image Settings panel can be used to modify the appearance of 3D volumetric images.

### 18.2.3 Slice View

To access the Slice View, click the Slice View button on the Image Toolbar. This view shows slices of the image in each of the cardinal directions, along with a 3D render of the image which shows the corresponding location of each of the slices.

Below each of the slices, the direction of the slice is displayed along with the index of the slice in that direction and a slider. You can change which slice is displayed by either editing the slice index number, dragging the slider, or hovering over the slice and scrolling with your mouse wheel. As a given slice is moved, the corresponding slice on the 3D render will move with it.

### 18.2.4 Slice Sequence View

To access the Slice Sequence View, click the Slice Sequence View button on the Image Toolbar. This view allows you to automatically step through the slices of your image at a fixed interval. This is particularly useful if your image data consists of 2D images taken at different times.

Below the slice, there are various controls for this view. The Play/Pause button will start or stop the animation. The Interval value is the amount of time in milliseconds that the view will pause between each slice. Its default value is 40ms, which corresponds to 25 frames per second. The Slice Direction dropdown allows you to change from which direction the slice be shown. The number and slider next to the Slice Direction dropdown show the current slice index and relative slice position. The slice index can be changed manually by editing the slice index number, dragging the slider, or hovering over the slice and scrolling with your mouse wheel. If, however, the view is currently animating, the value will be automatically advanced as soon as it is edited.

## 18.3 Filters

FEBio Studio allows you to apply various filters to your images. These features are still under development, and as such there are not yet many filters available. In order to add, edit, or apply filters to your image, select the image from the Model Tree, then select the Filters tab in the 3D Image box in the Model panel.

### 18.3.1 Adding, Removing, and Editing Filters

The Filter GUI consists of two boxes, the Filters box and the Filter Properties box, and an Apply button. Due to the time-intensive nature of some filters, filters are only applied to the image when you click the Apply button. The effects of any addition, removal, editing, or reordering of filters will not be applied to the image until the Apply button is clicked. The following actions can be done in the Filter GUI:

- To add a new filter, click the green plus icon to the right of the Filters box. This will open a dialog box which will allow you to select the type of filter you'd like to add.
- To remove a filter, select it from the list of filters, and click the red X icon.
- To change the order in which filters are applied, simply drag and drop the filters in the Filters box to reorder them. The filters will be applied in order from top to bottom.
- To change a filter's properties, select the filter from the Filters box. The filter's properties will appear, and can be edited in the Filter Properties box.

### 18.3.2 Available Filters

The image filtering functionality in FEBio Studio is new, and under active development. As such, there are only a small number of filters currently available. Many more filters will be added soon.

There are three filters currently available: a threshold filter, a mean image filter, and a Gaussian blur filter.

The threshold filter allows you to set a minimum and maximum intensity value for your image. Pixel values below your minimum will be set to 0, those above your maximum will be set to 255, and all other pixel values will be scaled between 0 and 255 linearly.

The mean image filter will assign each pixel the value of the average of all surrounding pixels within a given radius. The X, Y, and Z radii can each be set individually.

The Gaussian blur filter applies a Gaussian blur to the image. The sigma value of the Gaussian distribution can be specified in the filter properties.

Any other filters that may appear should be considered experimental for now.

## 18.4 Other 3D Image Visualization Options

After a 3D image stack is loaded you can attach several image renderers to visualize the image data in the Graphics View. Before you add an image renderer, make sure the correct image stack is selected in the Model Viewer. In the following section, the different image renderers are described.

### 18.4.1 Image Slicer

The image slicer shows a single slice of the 3D image stack in the Graphics View. You can select the image orientation as well as relative position in the image stack.

- *Image orientation* - Choose the orientation of the image slice
- *Image offset* - Choose the relative position of the image offset
- *Color map* - Set the color map that will be used to render the slice.

### 18.4.2 Volume Renderer

The Volume Renderer shows a volume rendering of the image stack in the Graphics View. The following options can be set.

- *alpha scale* - Sets the overall alpha scale, which scales the transparency of the image.
- *min intensity* - Minimum cutoff image intensity. Voxels with an intensity below this value will be rendered with a transparency set to the value of the *min alpha* parameter.
- *max intensity* - Maximum cutoff image intensity. Voxels with an intensity above this value will be rendered with a transparency set to the value of the *max alpha* parameter.
- *min alpha* - the alpha (transparency) value used by voxels with an intensity below the *min intensity* parameter.
- *max alpha* - the alpha (transparency) value used by voxels with an intensity above the *max intensity* parameter.
- *Amin, Amax* - The alpha (transparency) range for voxels with an intensity between *min intensity* and *max intensity*.
- *Color map* - Color map for mapping the grayscale voxels to a color value.
- *Lighting effect* - If Yes, the voxel color is attenuated based on local normal estimation, which simulates a lighting effect.
- *Lighting strength* - The strength of the lighting effect, as a value between 0 and 1.
- *Ambient color* - The ambient color added for the lighting effect.
- *Specular color* - The specular color added for the lighting effect.
- *Light direction* - The direction of the light used in the lighting effect.

### 18.4.3 Image Isosurface

The image isosurface features renders an isosurface of the 3D image data. This feature has the following parameters.

- *Isosurface value* - The image intensity value that determines the isosurface.
- *Smooth surface* - Smooth the facet normals to create a more smooth rendering of the iso-surface.

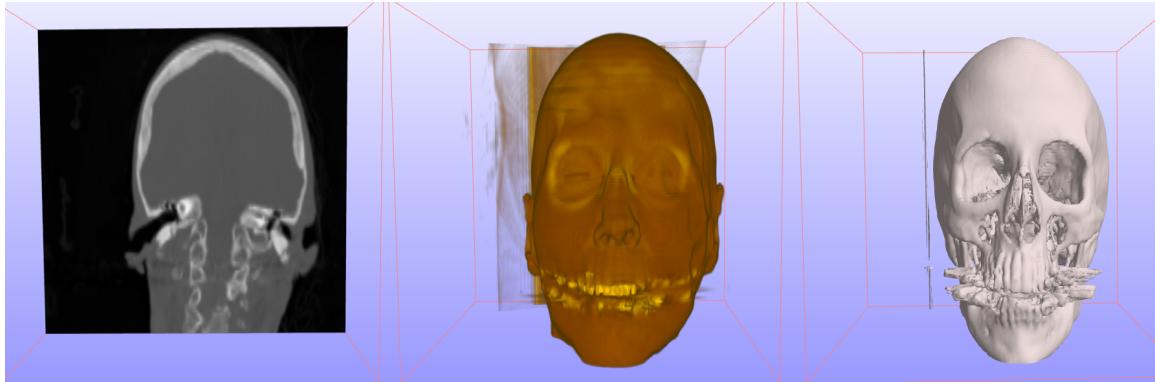


Figure 18.2: The various 3D image renderers. From left to right, Image Slicer, Volume Renderer, and Image Isosurface. (source: cthead from Stanford Volume Dataset).

- *Surface color* - set the color of the isosurface.
- *Close surface* - Close the surface when it intersects with the boundaries of the image domain.
- *Invert space* - Invert the image space.
- *Allow clipping* - Allow this surface to be clipped by any active cutting planes.

## 18.5 Other 3D Image Tools

### 18.5.1 Image Map Tool

The Image Map Tool (found on the Tools tab of the Build Panel) is used to map intensity values from a 3D Image onto an FE mesh that exists inside the image domain. To use this tool you must first import a 3D Image, select the FE object onto which you want the intensity values to be mapped, and then run the tool. After running the tool, a new Mesh Data object will be created and added to the model tree.

The tool has the following parameters.

- *Name* - The name of the resulting data map.
- *Image Model* - The 3D image that will be used in the image map.
- *Method* - The method used by the algorithm to sample the image (see below for more details).
- *Project Surface Nodes Inward* - A parameter used by the *Sample Image at Nodes* method to help nodes on the surface of a mesh to be assigned the correct value (see below for more details).
  - *Surface Threshold* - The intensity value cutoff at which to stop projecting a surface node inward.
  - *Max Search Depth* - The maximum number of voxels to search when projecting nodes inward. The default value of 5 is usually sufficient.

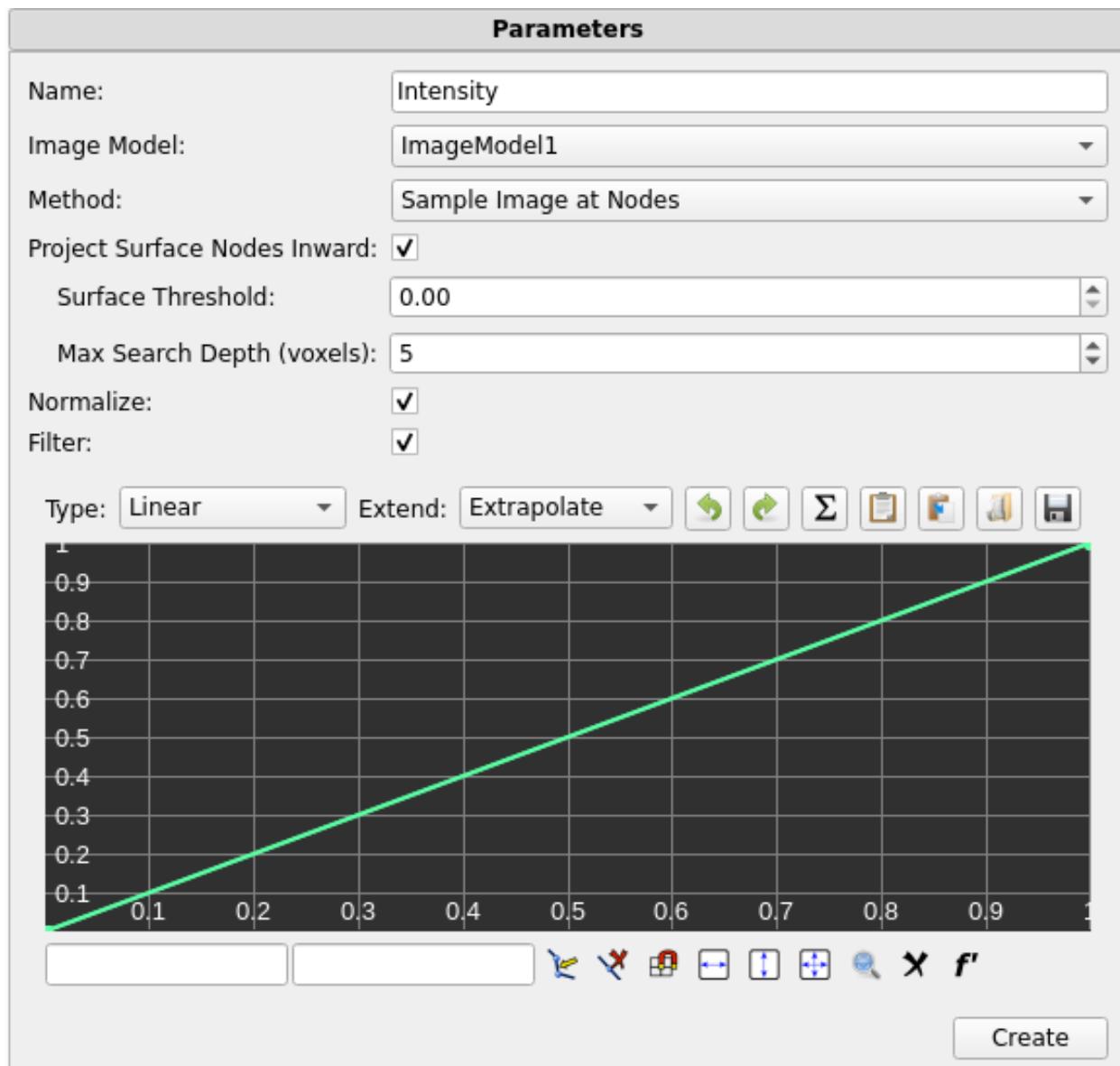


Figure 18.3: Parameters for the Image Map tool

- *Normalize* - When checked, the values of the resulting data map will always range between 0 and 1. When left unchecked, the values of the data map will be read directly from the image.
- *Filter* - When checked, an editable graph similar to the Curve Editor (Section 3.12) will be shown. This graph can be used to create a curve through which the data sampled from the image will be passed before being written to the data map. If the *Normalize* option is checked, the normalization will occur before the data is passed to the filter.

There are three different methods that can be used for sampling the image. Each method may be useful with different combinations of mesh and voxel density. It is important to choose the correct method to ensure that the tool accurately samples your image. The *Sample Image at Nodes* method is best in most situations, but due to its complexity, it is discussed last.

The *Sample at Element Centroids* method samples the image at the location of each element's centroid and assigns the value to that element, resulting in a single value for each element. The value of the sample is linearly interpolated between voxels in the image. This method works well when the FE elements in the mesh are similar in size to the image's voxels, or smaller than the voxels. If the elements in the mesh are considerably larger than the voxels, high frequency data in the image will be lost.

The *Average Intensity Over Elements* method loops over every element in the mesh, and finds all of the voxels in the image whose centroids lie inside the current element. It then takes the average of those voxel values, and assigns that average to the element. This method works well when the FE elements are considerably larger than the image's voxels, and tends to act as a sort of low-pass filter for the image data. This method is not recommended for use when the FE elements are of a similar size to the image's voxels, or smaller.

The *Sample Image at Nodes* method samples the image at the location of each node in the FE mesh, and assigns the value to that node, resulting in a single value for each node in the mesh. The value of the sample is linearly interpolated between voxels in the image. This method works well when the FE elements in the mesh are similar in size to the image's voxels, or smaller than the voxels. This method also provides the *Project Surface Nodes Inward* parameter which can be used to help nodes on the surface of a mesh to be assigned the correct value, as explained below.

It is often the case that an FE mesh does not simply cover the entire domain of the image, but rather outlines some prominent feature in the image, such as a bone (see Figure 18.4).

It is also often the case that the surface of the FE mesh does not perfectly follow the outline of the feature in question. As shown in the Figure 18.5, there is a gap between the surface of the FE mesh, and the surface of the pelvis in the 3D image.

The nodes on the surface of this FE mesh lie slightly outside the bounds of the bone in the image, and so when the Image Map Tool samples the values for these nodes, it will sample a value from the background of the image, rather than the bone. The *Project Surface Nodes Inward* option helps to mitigate this. When this option is enabled, the Image Map tool projects each node on the surface of the FE mesh inward along its surface normal until it encounters a voxel whose value meets or exceeds a user-specified threshold, and assigns that voxel's value to the node.

In more detail, the *Project Surface Nodes Inward* option works as follows. For each node on the surface of the FE mesh it will first check the value of the voxel at the node's location. If the value is greater than or equal to the value specified in the *Surface Threshold* parameter, then it will assign that voxel's value to the node, and move on to the next node. However, if the value of that voxel is less than the *Surface Threshold*, then it will travel in a direction opposite of the node's surface normal, until it encounters a new voxel. It will check the value of that voxel, and if it meets or exceeds the *Surface Threshold* then it will assign that voxel's value to the node, otherwise it

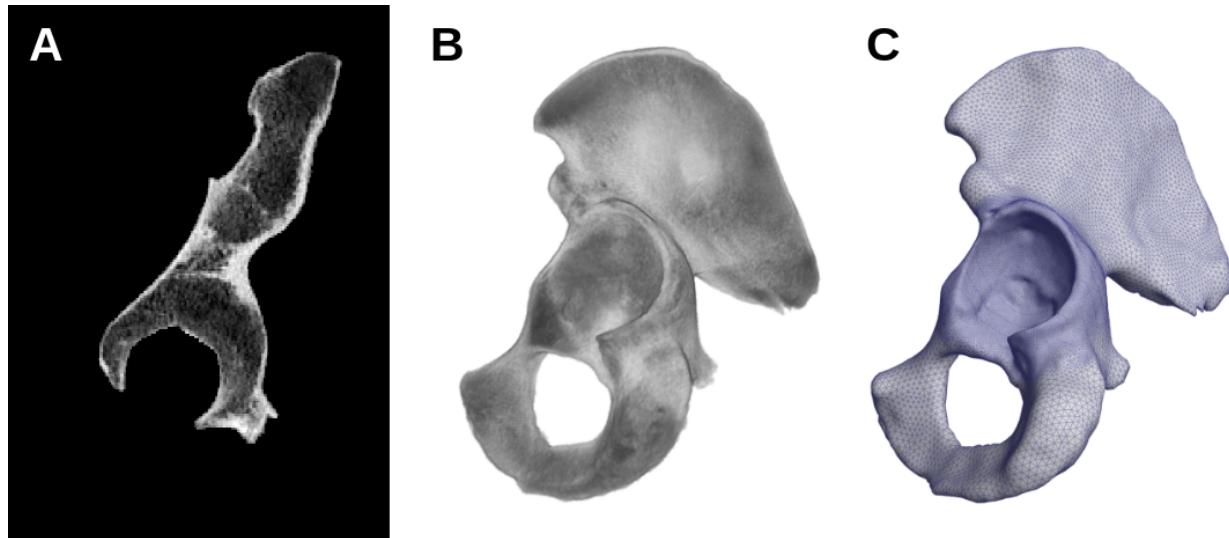


Figure 18.4: Part of a human pelvis shown in three views. A: a 2D slice of the image along the X axis. B: a 3D render of the image. C: an FE mesh covering the surface of the bone.

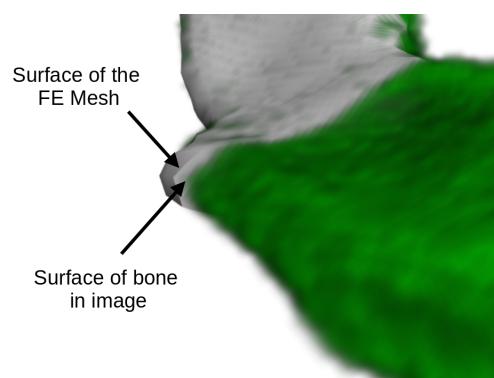


Figure 18.5: The FE Mesh and 3D Image (shown in green for contrast).

will move on to the next voxel in line. This search continues until either a suitable voxel is found, or until it has searched a number of voxels equal to the value stored in the *Max Search Depth* parameter. If the tool is unable to find a suitable voxel before that point, it assigns the node the value of the voxel closest to the node.

After running the Image Map tool, the results will be stored in a Mesh Data object in the Model Tree. To visualize the values on the mesh, select the object and open the Mesh Inspector (Section 3.13). Then in the *Variable* drop-down menu, select the name that was specified in the Image Map tool's *Name* parameter.

## 18.5.2 Fiber ODF Analysis

The Fiber ODF (Orientation Distribution Function) analysis feature in FEBio Studio is a tool designed to bridge the gap between experimental imaging data and finite element modeling. This analysis calculates ODFs directly from 3D image data, providing an accurate and detailed representation of the anisotropic material properties.

The analysis removes the need to rely on idealized or parametric approximations of fiber orientation distributions by enabling the direct use of experimentally measured, high-resolution, non-parametric ODFs. This approach eliminates the loss of detail and accuracy that often accompanies parametric fitting, providing a more faithful representation of real-world materials. It allows users to subdivide image domains into spatial subregions, customize processing parameters, and visualize the resulting ODFs alongside the original image data. The resulting ODF information can then be copied into a constitutive model designed to utilize this orientation information in FEBio.

A full description of this algorithm is beyond the scope of this document, and users should consult the manuscript published on this topic for a more in depth explanation of the process [2, 1]. However, a brief explanation follows. After some minimal preprocessing with a Butterworth filter, a 2D Fourier transform is applied to the volumetric image. The power spectrum is computed from the Fourier domain representation of the image. The power spectrum is then summed along the frequencies to determine the contribution to each direction in space. This results in a condensed power spectrum defined on the unit sphere. The power spectrum is then discretized on the unit sphere by dividing the surface into equal-area triangles using a tessellated icosahedron. The ODF is then obtained by applying the Q-ball algorithm to the condensed power spectra. The resulting ODF is then represented using spherical harmonics based on Legendre polynomials.

### 18.5.2.1 Running a Fiber ODF Analysis

To start a Fiber ODF analysis, load the desired 3D image into FEBio Studio, then right-click the image in the *Model Tree* and select *Fiber ODF Analysis*. A new *Fiber ODF Analysis* item will appear in the *Model Tree*. Selecting this item opens the *Properties Panel* and *Fiber ODF Analysis Panel*.

In the *Parameters* tab of the *Fiber ODF Analysis* Panel, the number of image subdivisions and their percent overlap may be specified. Clicking the *Generate subvolumes* button creates these subvolumes, and allows the *Next* button to appear. If the image has been subdivided into more than one region, a drop-down menu will also appear, allowing the user to specify a given subvolume as the “current” subvolume. Clicking the *Next* button shows parameters controlling the band-pass Butterworth filter, the harmonic order of the spherical harmonic representation, and a check box controlling whether or not a fitting analysis will be performed.

A Butterworth filter is applied to the image before the Fourier analysis. It removes unwanted noise by attenuating high and low frequency noise in the image. The high and low frequency

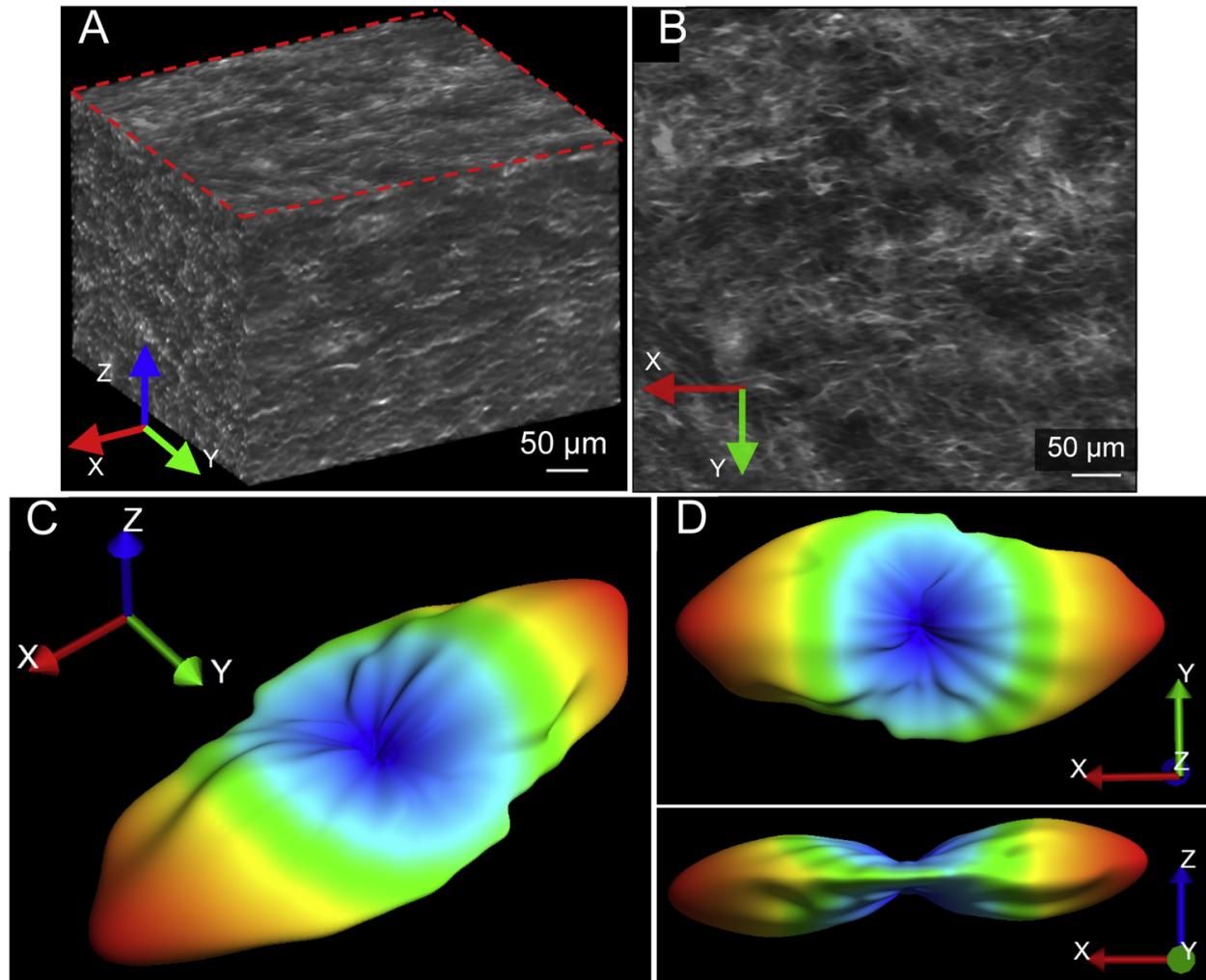


Figure 18.6: Type I collagen hydrogel image volume and its fibril orientation distribution function (ODF) characterized using our image analysis technique. (A) Isometric view of the image dataset, exhibiting collagen fibrils that are largely oriented along the X-direction. The dataset was acquired with a multiphoton microscope and pre-processed to correct for systematic distortions and improve contrast. (B) Top view of the first slice in the image stack, highlighted in a red dashed box in panel A. The image demonstrates the dataset is composed of fibrils that have considerable dispersion and some spatial heterogeneity. (C) Isometric view of the fiber ODF characterized over the entire volume. The ODF is visualized with the radial representation, where the probability of each direction is represented by its distance from the origin and a colormap. (D) Top and front views of the ODF. These views show the predominant orientation of the fibrils is along the X-axis and the dispersion is primarily within the XY plane.

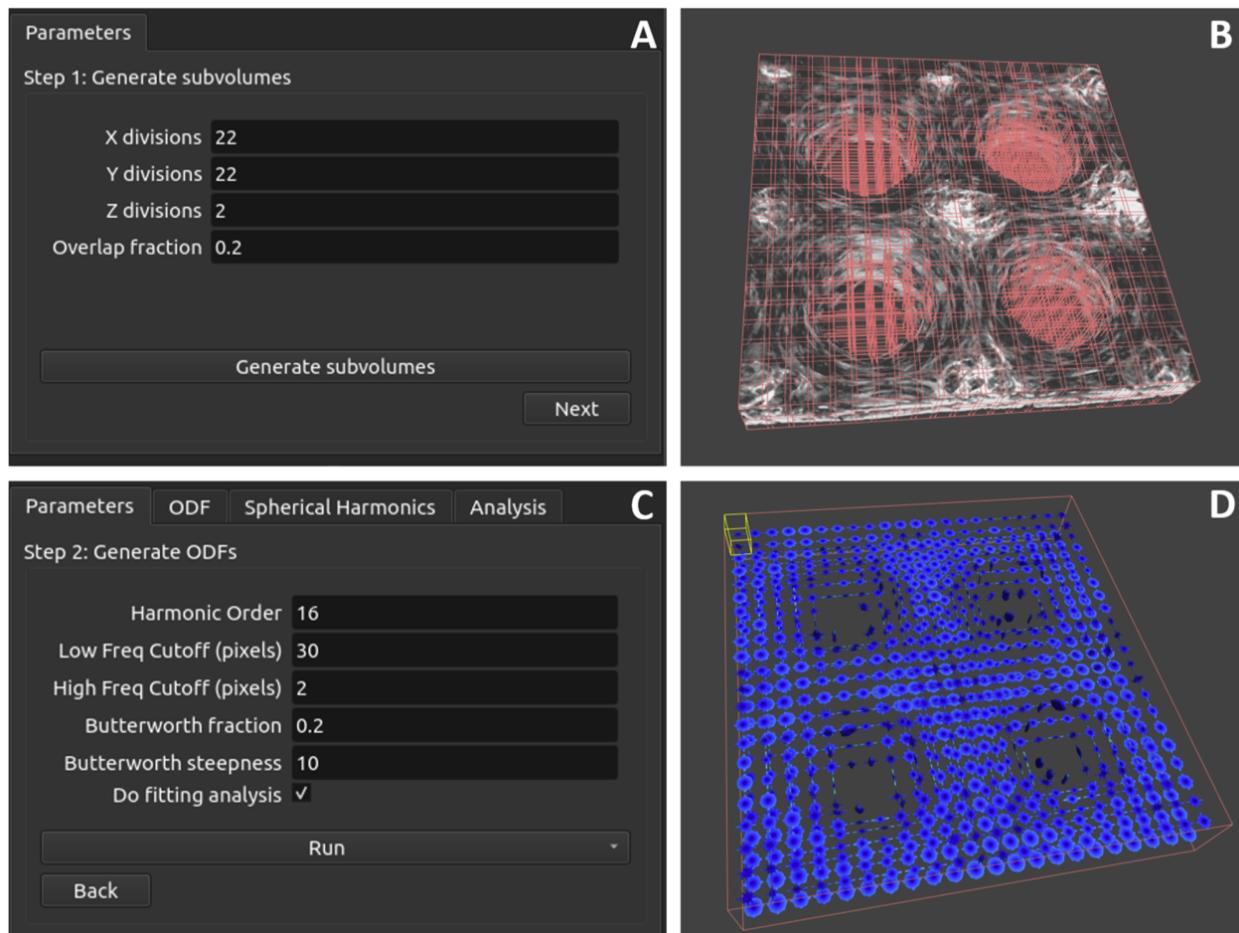


Figure 18.7: Parts of the user interface that control the inputs for ODF analysis in FEBio Studio. (A) The first page of the Parameters tab in the Fiber ODF Analysis provides control how the image is divided into subdomains. (B) The graphics view in FEBio Studio, showing the image and a preview of the user-specified subdomains. (C) The second page of the Parameters tab in the Fiber ODF Analysis panel allows the user to specify parameters that control the Fiber ODF analysis. The Run button allows the user to run the analysis with the specified parameters either for every subvolume, or for an individual subdomain. (D) Visualization of the results of the ODF analysis showing radial representations of the ODFs for all subdomains based on the parameters in panel C.

cutoff values can be specified along with the Butterworth fraction and steepness. The Butterworth fraction defines the fraction of the frequency spectrum retained. A low fraction (e.g. 0.2) focuses on broad trends and removes small, sharp variations (high-frequency noise). A high fraction (e.g. 0.8) retains more detail but risks including noise. The Butterworth steepness controls how sharply the filter transitions between retaining and attenuating frequencies. A high steepness value creates a sharp cutoff, effectively eliminating unwanted frequencies but may risk distorting the signal if the cutoff is too aggressive. A low steepness value creates a smoother transition, which may allow some noise to pass but avoids abrupt changes that could distort the ODF.

In order to reduce the storage size of the resulting ODFs, they are represented using spherical harmonics based on Legendre polynomials. The user is able to choose a harmonic order for the spherical harmonic representation of the ODFs, and it is important to insure that a reasonable value is chosen for a given dataset. A higher harmonic order (e.g. 20) allows for finer angular resolution, capturing sharp directional features in the ODF, but increases computational cost and memory usage. A lower harmonic order (e.g. 10) produces a smoother, lower-resolution ODF representation, reducing computational demand, but may miss fine details. It is generally best to choose the lowest harmonic order that captures the most important features of a given ODF.

In some cases, the ODF calculated from the image data may be adequately described by either an ellipsoidal fiber distribution (EFD), or a Von-Mises distribution (VM3), rather than the full, non-parametric ODF. Using either the EFD or VM3 representations of the ODF can reduce the necessary computation time during an FE analysis. The Fiber ODF Analysis tool can also perform a fitting analysis on the calculated ODFs. If this option is enabled then, for each ODF, an optimization routine is run to calculate the parameters for both EFD and VM3 representations that most closely match the original ODF. If an EFD or VM3 representation is used in place of the full ODF, care should be taken to ensure that this approximation does not significantly alter the FE results.

Once these parameters are set, the analysis may be started by clicking the *Run* button, and then choosing to either run the analysis for the currently selected subvolume, or for all subvolumes. When the analysis finishes, 3D graphical representations of the ODF(s) will appear in the graphics view area, and several new tabs will appear in the *Fiber ODF Analysis* panel.

The *ODF* tab shows a 3D rendering of the currently selected subvolume. This rendering is controlled by the visualization options described in the next section. The *Spherical Harmonics* tab displays the spherical harmonic coefficients for the currently selected ODF. The *Analysis* tab shows other information about the currently selected ODF such as its position, mean direction, fractional and general fractional anisotropy, and the EFD and VM3 fitting parameters if a fitting analysis was performed.

After an analysis is performed, the ODF information needs to be copied to a special *Fiber ODF* constitutive model so that it can be used in an FE Analysis. It is important to understand how the *Fiber ODF* constitutive model works before using it. Please see section 18.5.2.3 for more information on this material. The *Copy to Material* button allows the ODF information to be copied to an existing *Fiber ODF* material in the open model. It also allows the optimized EFD parameters to be copied to an existing EFD material in the open model, if the fitting analysis was performed.

The *Save to XML* button allows results from the analysis to be exported to an XML file so that it can be easily parsed by external tools for further analysis. The *ODFs* option exports the full-resolution ODFs with the nodal coordinates of the ODF, the position of each ODF, and the values of each ODF at each nodal coordinate. The *Spherical Harmonics* option exports the position and spherical harmonic coefficients of each ODF. The *Statistics* option exports the information displayed in the *Analysis* tab for each ODF.

### 18.5.2.2 The Fiber ODF Analysis UI

The following is a breakdown of the Fiber ODF Analysis GUI elements.

#### Properties Panel

This panel controls the visualization and rendering of ODFs.

- Rendering Options:
  - *Render Scale*: Adjusts the visual size of ODF plots in the graphics view.
  - *Render Mesh Lines*: Toggles the visibility of the ODF mesh lines.
  - *Render ODF As*: Changes how the ODF is rendered in the graphics view. Choose between:
    - \* Full-Resolution ODF
    - \* Remeshed ODF (see section 18.5.2.3 for more details)
    - \* Fitted EFD/VM Distributions
    - \* Glyph Representations for simplified visualization.
  - *Radial Mesh*: Scales radii of vectors proportional to ODF values.
- Bounding Box Options:
  - *Show Bounding Boxes*: Displays the boundaries of subdomains.
  - *Show Selection Box*: Highlights the active subdomain.
- Legend Options:
  - *Coloring mode*: Colors the ODFs based on either the ODF values, or their fractional anisotropy.
  - *Legend Divisions*: Controls the number of divisions in the legend.
  - *Legend Range*: Choose between automatic and manual legend ranges.
  - *Legend Min*: The minimum value on the legend.
  - *Legend Max*: The maximum value on the legend.

#### Fiber ODF Analysis Panel - Parameters Tab

The first tab allows users to configure the subdivision of the image domain and set parameters for ODF analysis.

- Subdivision Parameters:
  - *X, Y, Z Divisions*: Specifies the number of subdomains along each axis. This is important for analyzing spatially inhomogeneous samples.
  - *Overlap Percentage*: Defines the overlap between adjacent subdomains to ensure smooth transitions and consistent analysis.
  - *Generate Subvolumes*: Generates and displays the subdomains in the graphics view for inspection and adjustment.
- Other Analysis Settings:
  - *Harmonic Order*: Sets the maximum order of spherical harmonics for ODF representation, impacting resolution and computational cost.

- Frequency Cutoffs:
  - \* *Low-Frequency Cutoff*: Filters out large-scale noise.
  - \* *High-Frequency Cutoff*: Removes high-frequency noise that may obscure fine details.
- *Butterworth fraction*: Defines the fraction of the frequency spectrum retained. A smaller value keeps only the most dominant frequencies, which correspond to larger-scale structures in the image.
- *Butterworth steepness*: Controls how sharply the filter transitions between retaining and attenuating frequencies.
- *Do Fitting Analysis*: Toggles whether the software fits parametric distributions (e.g., EFD, VM) to the computed ODFs. This enables comparison between measured and idealized ODF models.
- *Run*: Click *Run* to start the analysis and choose whether to analyze all subdomains or a selected one.

### **Fiber ODF Analysis Panel - Other Tabs**

The other tabs on the *Fiber ODF Analysis Panel* appear only after the analysis has run.

- *ODF Tab*: Shows a 3D render of the currently selected ODF.
- *Spherical Harmonics Tab*: Displays the spherical harmonic coefficients for the currently selected ODF.
- *Analysis Tab*: Shows other information about the currently selected ODF such as its position, mean direction, fractional and general fractional anisotropy, and the EFD and VM3 fitting parameters if a fitting analysis was performed.

#### **18.5.2.3 The Fiber ODF Constitutive Model**

The *Fiber ODF* constitutive model is a special version of the *Continuous Fiber Distribution* constitutive model, with its own distribution type, and a unique integration scheme. (Please see the FEBio User Manual on continuous fiber distributions for more information). The Fiber ODF material takes any number of ODFs as parameters, each with a position in space, and a list of spherical harmonic coefficients. When an FE simulation is run using this material, some preprocessing is done on the ODFs in order to reduce the computational load during the simulation. The following is a description of these preprocessing steps. For a more detailed description, please see the associated manuscript [1].

In order to prevent sharp changes in the fiber direction between elements, a unique ODF is interpolated for each element in the mesh based on the element's physical distance from the nearby ODFs in the discrete ODF field. This interpolation is done using a weighted-mean approach utilizing the objective distance metric afforded by the Fisher-Rao inner product space [1]. This results in a unique ODF for each finite element in the domain of the constitutive model which smoothly transitions between ODFs defined in each of the original image's subdivisions. If only a single ODF is defined for the domain (in other words, if the image was not subdivided before the analysis) then this step is skipped and the original ODF is used for all elements in the domain.

Constitutive models using continuous fiber distributions require an efficient integration scheme over the unit sphere to compute stress contributions from the fiber family. Because there is no analytical representation of these non-parametric ODFs, this integration is performed by sampling

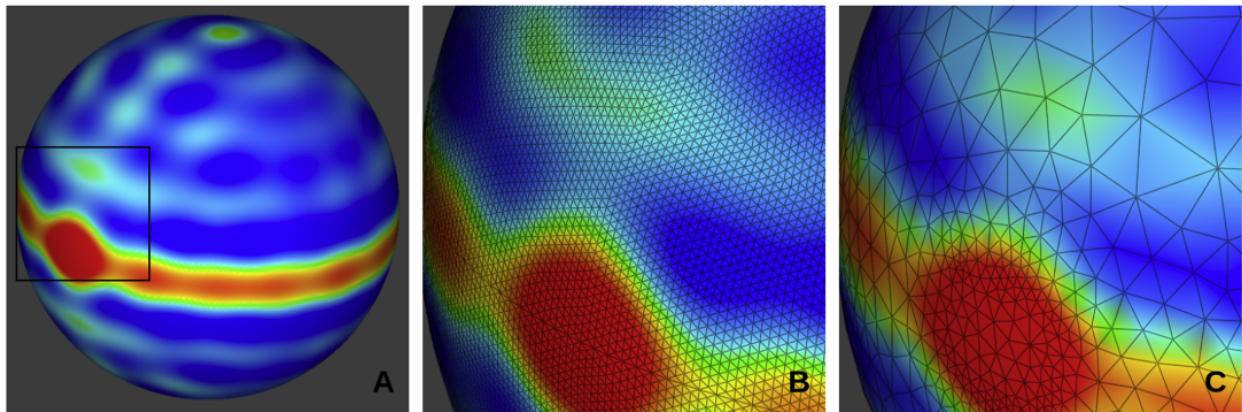


Figure 18.8: Spherical representations of an ODF showing the results of the remeshing algorithm. (A) A full-resolution, spherical representation of an ODF. (B) An enlarged portion of the same ODF showing the area inside the black square in panel A. In this panel, the ODF's mesh has been made visible to show the density of sampling points at the full-resolution of 40,962 points. (C) The remeshed surface of the same ODF showing the area inside the black square in panel A. The density of the ODF's sampling points has been greatly reduced in the regions of the ODF where there is little variation in the ODF's value, while maintaining high density in regions where there are sharp changes in value. This preserves the general shape of the ODF while significantly reducing the number of sampling points.

the values of the ODFs at its defined points. To ensure faithful representation of the ODF, all ODF calculations up to this point in the process use a high resolution, pre-defined set evenly-spaced points on the unit sphere. Integrating the stress contributions across the ODFs at their full resolution proved to be prohibitively computationally expensive, as this integration must occur at every integration point in the FE mesh for each stress calculation in the analysis.

To address this issue, we developed a mechanism to reduce the number of sampling points on the ODF using an approach based on finite element remeshing technology. The gradient of each ODF is calculated, and a triangulated mesh of the ODF probability surface and its gradient is passed to the mmg remeshing library. The mmg algorithm remeshes a surface of triangular elements, adjusting the relative nodal density of the ODF mesh based on the magnitude of the gradient, resulting in a higher sampling density in areas of high curvature while reducing the overall number of sample points, thereby preserving sharp changes in orientation (Figure 18.8). The visualization parameter *Render ODF As -> Remeshed ODF* in the properties panel in FEBio Studio allows the user to see this remeshed version of the ODF as a 3D render. The number of integration points in the remeshed ODFs varies depending on the ODF topology, but the number of points is generally reduced by about two orders of magnitude. Since the resampling of the ODFs only takes place once, at the beginning of the FE analysis, the time required for the remeshing step is small relative to the overall time involved in a FE analysis. The reduction in points dramatically increases the speed of stress computations without significantly altering the results.

After the interpolation and remeshing steps, the material initialization is complete, and the FE analysis continues normally.

# Chapter 19

## Using Python in FEBio Studio

FEBio Studio has a built-in Python interpreter that can be used to run simple Python scripts that interact with FEBio Studio in various ways. In addition, FEBio Studio offers an external Python module that can be used to use FEBio Studio functionality without opening FEBio Studio. Both of these options are discussed in this chapter.

### 19.1 The Embedded Python Interpreter

To open the Python editor, select the menu **Tools** → **Python Editor**.

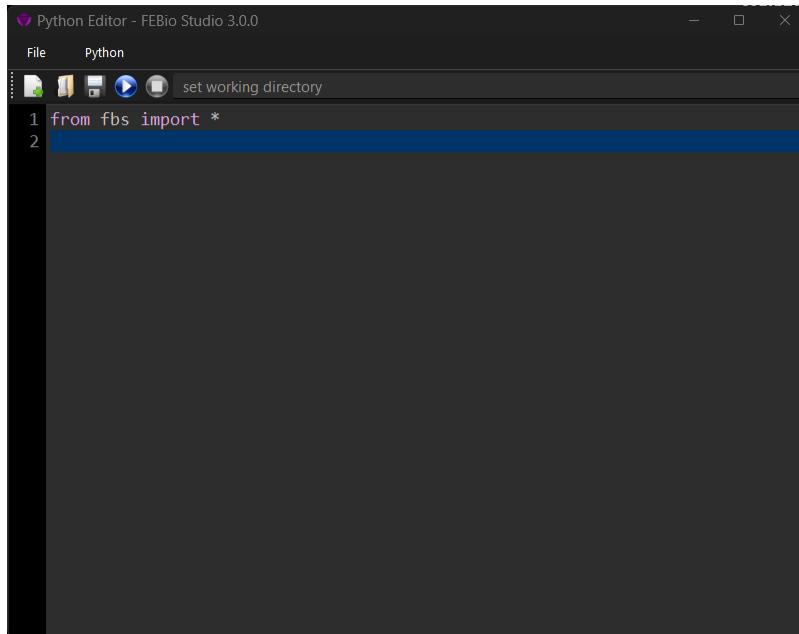


Figure 19.1: The Plugin Repository can be used to download and install FEBio plugins.

The Python Editor window shows a text editor for creating simple Python scripts. Python scripts can be saved and loaded from the **File** menu. To run the Python script, click the run button on the toolbar or select the menu **Python** → **Run Script**. Before you run a script, it is advised to set the working directory on the corresponding field on the toolbar. The output of the Python script can

be found on the Output panel. Make sure to select Python from the “Show output from” dropdown menu near the top of the Output panel.

Example Python scripts can be found on the Github repo <https://github.com/febiosoftware/python-tests>.

Some preliminary documentation of the Python module is available from [https://help.febio.org/docs/fbs\\_module](https://help.febio.org/docs/fbs_module)

## 19.2 The External Python Module

FEBio Studio also offers an external Python module that allows users to use FEBio Studio functionality directly from a Python script and without opening the GUI.

To use the module, you must first configure Python so that it can find the module. (Alternatively you can simply copy the module to the folder of the python script where you want to use it.)

Take note that the external Python module is built against specific versions of Python and thus require to be run with the same version.

# Chapter 20

# Using and Creating FEBio Plugins

FEBio supports plugins that allow users to add new features to FEBio. Plugins can be used to add new constitutive models, boundary conditions, loads, physics solvers, and much more. FEBio Studio supports these plugins and offers ways to load and integrate them seamlessly in the FEBio Studio UI. This chapter first discusses the plugin repository, which is main mechanism for downloading and installing FEBio plugins. Then, the tools are discussed that allow users to create, edit, and build plugins from within FEBio Studio.

## 20.1 The Plugin Repository

The Plugin Repository is the standard mechanism for downloading, installing, and sharing FEBio plugins. To open the Plugin Repository select the menu **FEBio → Plugin Repository**.

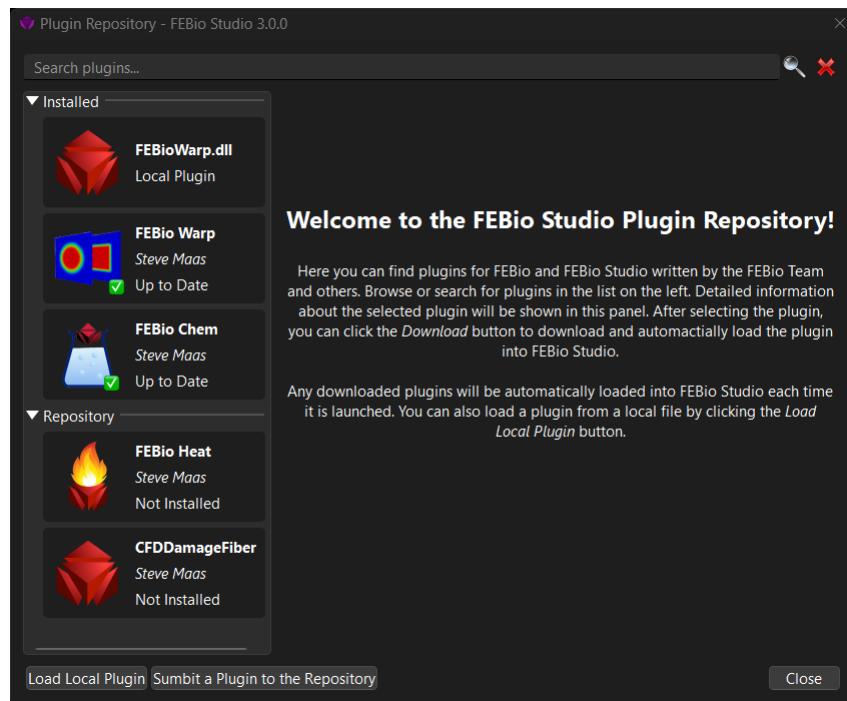


Figure 20.1: The Plugin Repository can be used to download and install FEBio plugins.

On the left hand side you see a panel that is divided into two sections. The top section shows all the plugins that are currently installed. The bottom section shows a list of plugins that are available from the online repository. Selecting a plugin will display additional information about the plugin on the right side of the dialog box. If the plugin is not installed, you can download it and install it. If the plugin is already installed and loaded, you can choose to unload and uninstall it. In addition, for installed plugins, the *Details* section provides more information on the specific features that the plugin implements.

If you built a custom plugin on your system, then you can also load it into FEBio Studio using the Plugin Repository. On the bottom of the dialog box is a button labeled **Load Local Plugin**. Clicking the button brings up a standard file open dialog box where you can locate and select the plugin file.

**Important!** It is important to note that only plugins can be loaded that were build for the specific version that FEBio Studio uses. Trying to load plugins that are not compatible may cause instability or a crash.

**Important!** When loading a model file (either fsm or feb) that needs a plugin, it is important to *first* load the plugin into FEBio Studio. Otherwise FEBio Studio will not be able to load the file.

## 20.2 Creating FEBio Plugins

FEBio Studio offers the ability to create and use custom FEBio plugins. In this section we'll detail the steps needed to do this. Before you can create plugins, there are a few prerequisite steps that need to be completed.

### 20.2.1 Prerequisites

First and foremost, you must install a development environment on your system. At the very least, you will need a C++ compiler, but it is best to use the same compiler that was used to build the FEBio Studio software itself. For Windows, we recommend using the Community Edition of Visual Studio. For Mac, we recommend using XCode, and on Linux we recommend using g++.

In addition to a c++ compiler, FEBio Studio also uses CMake (<https://cmake.org/download/>) to build the plugin code and thus CMake must be installed on your system as well. Make sure to configure CMake during installation so that it can be called from the command line.

Finally, you must have the FEBio SDK (Software Development Kit) installed. This is an optional component when you install FEBio Studio. If it is not installed, you may need to reinstall FEBio Studio making sure to enable that component.

After you installed a development environment, CMake, and the FEBio SDK, you must set several settings in FEBio Studio. Open the FEBio Studio Settings dialog box from the menu **Tools** → **Settings...** and select the FEBio option on the left side.

Here, you need to specify three paths:

- **FEBio SDK Include path:** This is path to the include folder of the FEBio SDK (e.g.: C:\Program Files\FEBioStudio3\ sdk\include)
- **FEBio SDK Library path:** This is the path to the folder that contains the library files (e.g.: C:\Program Files\FEBioStudio3\ sdk\lib\Release)
- **FEBio create plugin path:** This is the path to a folder that will be used as the default location to place plugin files.

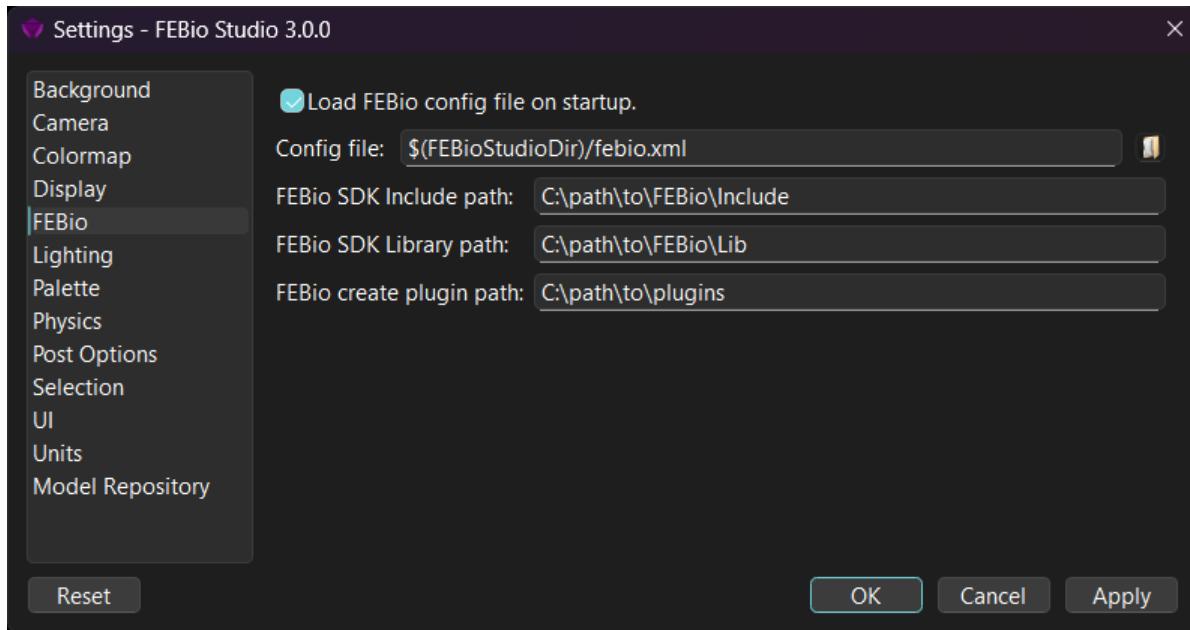


Figure 20.2: The FEBio settings page is used to specify the location of the FEBio SDK.

With all the necessary software installed and FEBio Studio configured, you can now move on to the next step, creating the plugin.

### 20.2.2 The FEBio Plugin Wizard

Creating an FEBio plugin project is as simple as following the steps in the FEBio Plugin wizard. To start the wizard, select the menu **FEBio → Create FEBio Plugin....**

The first page of the wizard allows you to choose a name for the plugin. Since this name will be used to generate the plugin files, make sure to use only characters that are valid in file names. In addition, you can also choose the location where FEBioStudio will generate the plugin files. Lastly, choose the FEBio module that you will expand. For instance, if you intend to add a new constitutive model for an elastic material, select the *solid* module.

After filling out the information, click the **Next** button to proceed to the next page.

On the second page, you can select the plugin type (e.g. Elastic Material, if you intend to implement a new constitutive model) and enter the name of the class. Make sure to use a valid C++ class name. Optionally, you can enter the type string of the plugin class. This is the string value that will be used when registering the plugin and the string that users will use as the value for the type attribute in the FEBio input file. By default, the name of the plugin class will be used.

Once this page is completed, click **Next** to continue.

The third page will list properties you can set to further configure the particular plugin you want to create. The properties that are listed will depend on the type of plugin you selected. For example, if you selected the "Elastic Material" plugin, the properties will allow you to choose the base class for the material class.

Once the properties are set, click **Finish** to exit the wizard.

At this point, the wizard will generate all the necessary files for building the plugin. A dialog box will inform you whether FEBio Studio was able to generate the plugin files or not. If the wizard

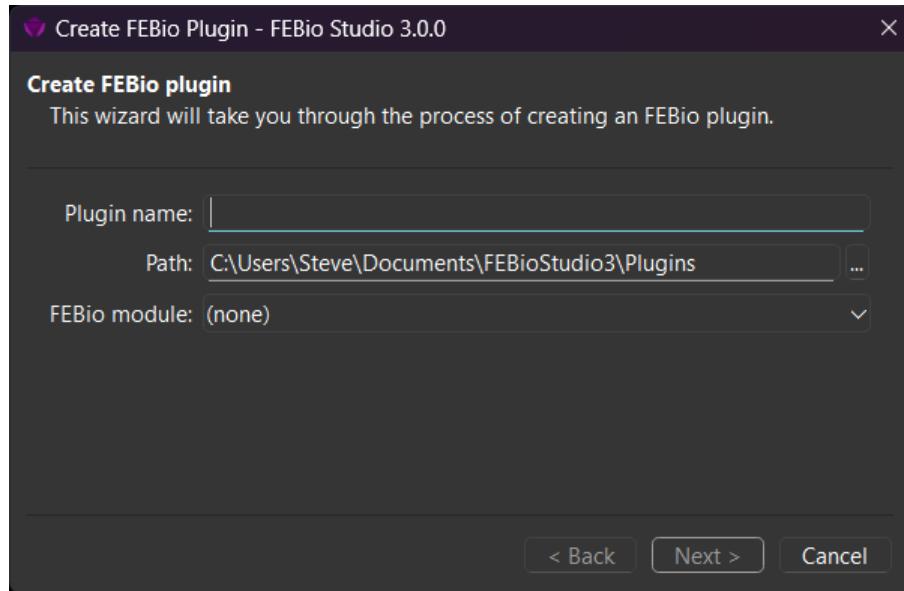


Figure 20.3: The Plugin Wizard can be used to create the template code of an FEBio plugin.

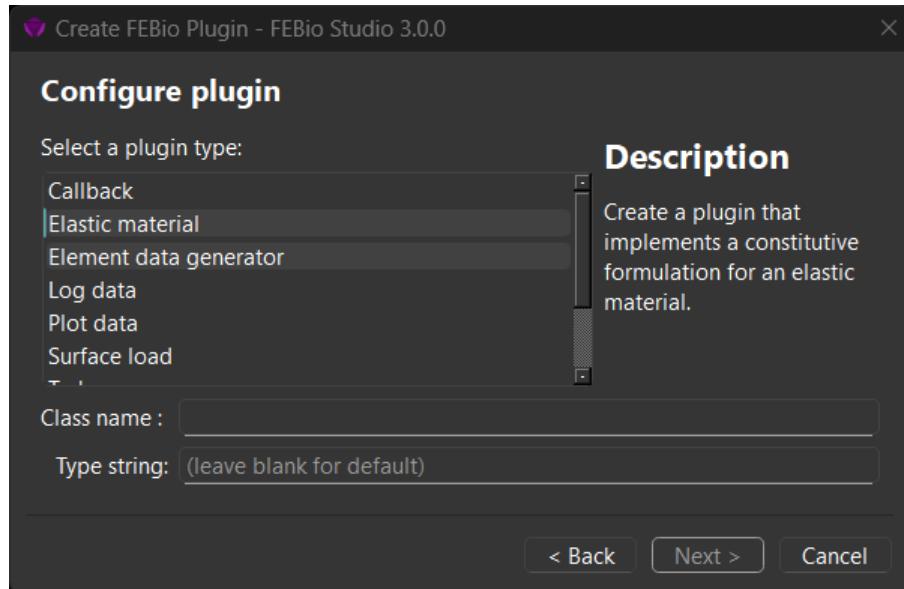


Figure 20.4: On the second page of the wizard you select the type of plugin you want to create.

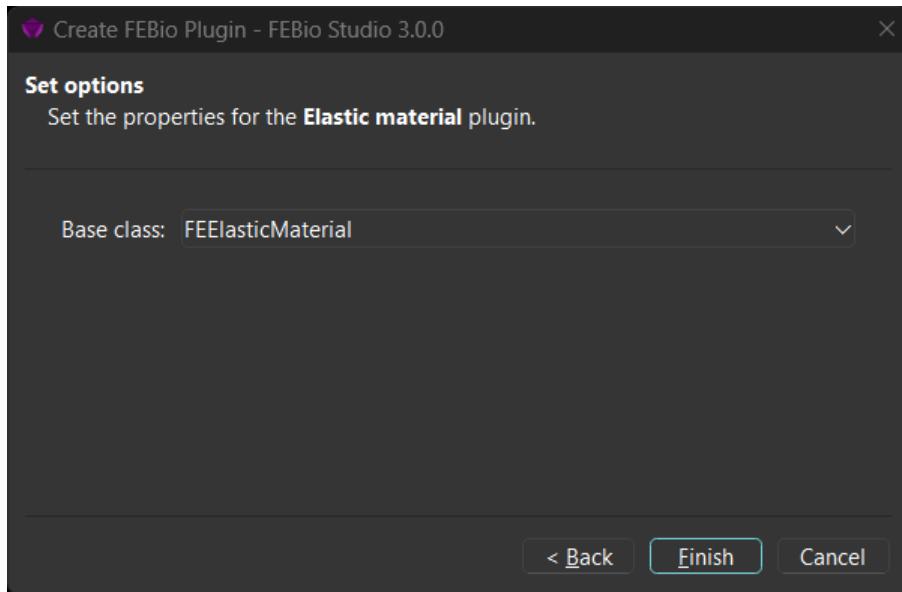


Figure 20.5: On the third page of the wizard you select additional options for the specify plugin type you want to create.

was successful, the following files should have been generated in the location that you specified on the first page of the wizard.

- **CMakeLists.txt** This is the CMake file that will generate the project files for your build environment.
- **main.cpp** This is the file that contains the required functions that FEBio needs to integrate your plugin.
- **MyPluginName.h** The header file containing the class declaration. (The actual name will use the class name you entered in the wizard.)
- **MyPluginName.cpp** The code file containing the class definition. (The actual name will use the class name you entered in the wizard.)

The next step is to build the plugin.

### 20.2.3 Building the plugin

The wizard will generate a template code that is ready to build. Of course, the plugin may not actually do anything yet, but at this point, you can verify that the plugin builds correctly with your build environment and can be loaded in FEBio Studio.

The first step will be to build the project files for your build environment. This essentially means that you have to run CMake on the CMakeLists.txt that the wizard generated. If you are familiar with CMake, you can do this outside FEBioStudio, however, you can also do this from FEBio Studio. In fact, you can build the plugin directly from FEBio Studio.

On the Project panel, you'll notice that your plugin files are listed there under a new item that is named after your plugin. Double-clicking a file will open it up in an editor in FEBio Studio.

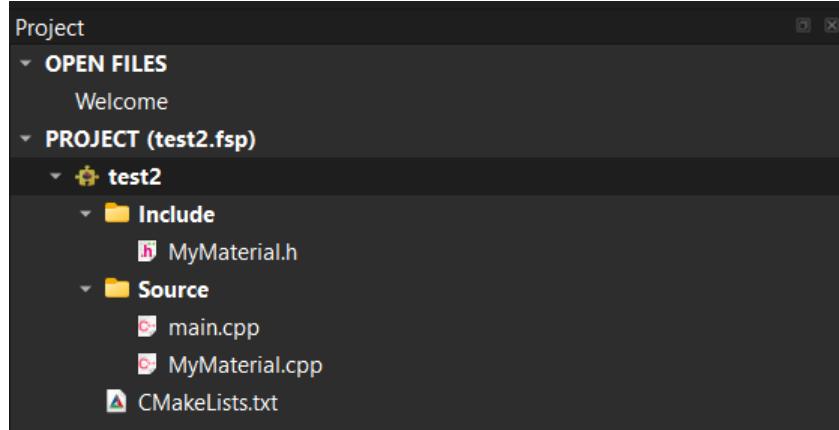


Figure 20.6: The project panel will show the files that have been created with the Plugin Wizard.

To build the plugin, right-click the top-level item with the plugin's name and select the Build option. When doing this, FEBio Studio will now first call CMake to generate the project files, and then call CMake again to build the plugin with your native build environment. You can monitor the progress on the Output panel. (Make sure "Build" is selected in the show output from the dropdown list.)

If the build was successful, you can now load the plugin into FEBio Studio. To load the plugin, right-click the item with the plugin's name in the Project panel and select the Load menu option. A dialog box will inform you whether the plugin was loaded successfully or not.

#### 20.2.4 Plugin Caveats

Although FEBioStudio has the ability to generate plugin files, edit the files, and build the plugin, it is still highly recommended to familiarize yourself with your build environment since some tasks can only be done there.

One important thing to keep in mind is that FEBio Studio will only build the *Release* configuration of the plugin, since only the *Release* build can be loaded into FEBio Studio. For the purpose of testing your plugin code, it is more convenient to build the *Debug* configuration. However, this can only be done from your build environment. To build your *Debug* configuration, you can still use the wizard to generate the plugin files and build the *Release* configuration, as this will generate all the project files for your build environment. However, after that, you'll have to open the IDE (or manually change the project files) to build and test the *Debug* configuration of your plugin.

Further documentation on how to create FEBio plugins can be found online in the [FEBio Developer's Manual](#).

# Appendix A

## Mesh Import Formats

This appendix details the different file formats that are supported by FEBioStudio. If you don't find your preferred format (or keywords) in this list, you may request them by posting a feature request on the FEBioStudio forum. Please keep in mind that we can only support publicly available text based formats.

### A.1 FEBio

FEBio is a nonlinear finite element solver designed specifically for computational biomechanics problems. FEBioStudio was designed to create FEBio input files and supports most of the FEBio features. See the FEBio User's Manual and Theory Manual for a detailed description of FEBio's features.

### A.2 NIKE3D

NIKE3D is a nonlinear finite element solver designed by Lawrence Livermore National Laboratories (LLNL). The following sections are supported:

- Control section
- Material deck: materials 1, 15, 18, 20 and 63 are supported.
- Node point deck
- Hexahedron deck
- Shell element deck
- Rigid node and facet deck
- Sliding surface deck: contact types 2 and 3 are supported.
- Stonewall and symmetry plane deck
- Load curve deck
- Concentrated nodal loads deck

- Pressure boundary condition deck
- Displacement boundary condition deck
- Base acceleration Body Force Loads deck

### A.3 HyperMesh ASCII

This is the hypermesh ASCII format. The following keywords are supported.

- node
- component
- tria3
- tetra4
- hexa8

### A.4 ABAQUS

The following ABAQUS®keywords are supported in FEBioStudio.

- ELEMENT
- ELSET
- END PART
- END INSTANCE
- HEADING
- INCLUDE
- INSTANCE
- MATERIAL
- NFILL
- NGEN
- NODE
- NSET
- PART
- SURFACE
- SURFACE INTERACTION

The following element types are recognized.

- C3D8
- C3D6
- C3D4
- C3D10
- R3D4
- S4
- S3
- S3R
- R3DR

## A.5 LSDYNA keyword

The following LSDYNA®keywords are supported.

- ELEMENTSHELL
- ELEMENTSHELLTHICKNESS
- ELEMENTSOLID
- END
- MATELASTIC
- NODE
- PART
- SETSEGMENTTITLE

## A.6 ANSYS

The following ANSYS® keywords are supported.

- EBLOCK
- NBLOCK

## A.7 DXF

Currently, FEBioStudio only reads in *polylines*, described by vertices and triangulated faces.

## A.8 Hypersurface ASCII

The following keywords are supported.

- Vertices
- Triangles

## A.9 GMsh

The following GMsh keywords are supported.

- MeshFormat
- PhysicalNames
- Nodes
- Elements

## A.10 BYU

The BYU format, developed at Brigham Young University, is used to describe a surface mesh. All features of this format are supported.

## A.11 VTK

The legacy VTK format is provided by the Visualisation Toolkit. FEBio Studio only supports the ASCII format. The following keywords are supported:

- DATASET- POLYDATA, UNSTRUCTUREDGRID
- POINTS
- POLYGONS / CELLS
- POINTDATA
  - SCALARS – ShellThickness
  - SCALARS – ScalarData
- CELLTYPES

Currently FEBio Studio only supports the triangular and quadrilateral shell mesh and hexahedral volume mesh.

## A.12 STEP

Step files are used to import CAD geometry. The STEP file reader is supported mostly through a third-party library, called OpenCascade<sup>1</sup> and should be able to read a wide variety of STEP files.

The STEP file reader will import each solid part defined in the STEP file as a separate object.

## A.13 BREP

BREP files are used to import CAD geometry. The BREP file reader is supported mostly through a third-party library, called OpenCascade and should be able to read a wide variety of BREP files.

The BREP file reader will import the entire geometry as a single object.

## A.14 IDEAS

This is the file format of the I-DEAS software.

The IDEAS file reader will import the nodal coordinates and element connectivity. The following element types are supported:

- 11 : rod element
- 21 : linear beam
- 22 : tapered beam
- 23 : curved beam
- 24 : parabolic beam
- 44 : plane stress linear quadrilateral
- 91 : linear triangular element
- 94 : linear quadrilateral element
- 111: linear tetrahedral element
- 112 : linear wedge element
- 113 : 15-node wedge element
- 115 : linear brick element
- 116 : 27-node brick element
- 118 : 10-node tetrahedron element

---

<sup>1</sup><https://www.opencascade.com/>

## A.15 NASTRAN

This is the file format from the Nastran finite element software. The following keywords are supported.

- GRID
- CTETRA
- PSOLID
- MAT1 : materials are mapped to “isotropic elastic” materials.
- CHexa

## A.16 MESH

Mesh file format. The following keywords are supported.

- Vertices
- Hexahedra
- Tetrahedra
- Triangles

## A.17 TETGEN

File format used by the TetGen software. TetGen typically splits the mesh into two separate files, a .node file, which contains the nodal coordinates and .ele file, which contains the element connectivity. The TetGen file reader expects the .ele file and will automatically look for a .node file that has the same file name. So, for example, if the .ele file is called “mymesh.ele”, then the file reader will look for a .node file called “mymesh.node” in the same file folder.

## A.18 RAW

The RAW format is used to define a 3D stack of image data. The RAW file reader will import the image data and generate a hexahedral mesh that has the same number of elements in x, y, and z as the image dimensions. The elements are partitioned based on the grayscale levels in the image. For instance, for a black-and-white image, two parts will be created, one for all the elements that correspond to the black part of the image, and one for all the elements that correspond to the white part of the image.

## A.19 COMSOL

This file format is used by the COMSOL software package. Currently, only the text version (mphtxt) can be imported.

Only the finite element mesh is extracted from the file.

## A.20 PLY

These are files stored in the Polygon File Format, and supports triangular and quadrilateral surface meshes.

Meshes from PLY files are imported as editable surfaces.

## Appendix B

# Standard Data Fields

FEBio Studio defines the following list of standard data fields that can be added to a post model.

Name	Description
Initial Position	The initial nodal position of the model
Aspect ratio	The element's aspect ratio
1-Princ curvature	The first principal curvature of the surface
2-Princ curvature	The second principal curvature of the surface
Gaussian curvature	The Gaussian curvature of the surface
Mean curvature	The mean curvature
RMS curvature	The Root-Mean-Square curvature
Princ Curvature difference	The difference of the principal curvatures
Congruency	The congruency
1-Princ curvature vector	The first principal curvature vector
2-Princ curvature vector	The second principal curvature vector

The following data fields require a “displacement” field.

Name	Description
Position	The current nodal position of the deformed model
Deformation gradient	The deformation gradient of the deformation map
Infinitesimal strain	The infinitesimal (engineering) strain tensor
Lagrange strain	The Green-Lagrange strain tensor
Right Cauchy-Green	The right Cauchy-Green deformation tensor
Right stretch	The right stretch tensor
Biot strain	The Biot strain tensor
Right Hencky	The Right Hencky tensor
Left Cauchy-Green	The left Cauchy-Green strain tensor
Left stretch	The left stretch tensor
Left Hencky	The left Hencky tensor
Almansi strain	The Almansi strain tensor
Volume	The element's (approximate) volume
Volume ratio	The ratio of current over initial element volume
Volume strain	The volumetric strain

The following data fields require a “stress” field.

Name	Description
pressure	Calculated from the “stress” field: $p = -tr\sigma$

# Bibliography

- [1] Adam Rauff, Michael R. Herron, Steve A. Maas, and Jeffrey A. Weiss. An algorithmic and software framework to incorporate orientation distribution functions in finite element simulations for biomechanics and biophysics. *Acta Biomaterialia*, November 2024.
- [2] Adam Rauff, Lucas H. Timmins, Ross T. Whitaker, and Jeffrey A. Weiss. A Nonparametric Approach for Estimating Three-Dimensional Fiber Orientation Distribution Functions (ODFs) in Fibrous Materials. *IEEE Transactions on Medical Imaging*, 41(2):446–455, February 2022. Conference Name: IEEE Transactions on Medical Imaging.