SCIRun Visualization Basics

SCIRun 5.0 Documentation

Center for Integrative Biomedical Computing
Scientific Computing & Imaging Institute
University of Utah

SCIRun software download:

http://software.sci.utah.edu

Center for Integrative Biomedical Computing:

http://www.sci.utah.edu/cibc

This project was supported by grants from the National Center for Research Resources **(5P41RR012553-14)** and the National Institute of General Medical Sciences **(8 P41 GM103545-14)** from the National Institutes of Health.

Author(s):
Ayla Khan, David Weinstein

Contents

Chapter 1

SCIRun Overview

This tutorial demonstrates how to build a simple SCIRun dataflow network.

1.1 Software requirements

1.1.1 SCIRun 5.0

All available downloads for SCIRun version 5.0 and the SCIRunData archive are available from SCI software portal. Make sure to update to the most up-to-date release available, which will include the latest bug fixes.

Currently, the easiest way to get started with SCIRun version 5.0 is to download and install a binary version for Mac OS X. Sources are also available for Linux, however this option is recommended only for advanced Linux users.

Unpack the SCIRunData archive in a convenient location. Recall from the User Guide that the path to data can be set using the `SCIRUN_DATA` environment variable or by setting `SCIRUN_DATA` in the *.scirunrc* file.

Chapter 2

User Interface

Scope: Data Layers - Favorite Modules - Saved Subnetworks - Converting Networks from SCIRun 4

Chapter 3

Simple Dataflow Network

Scope: Read Data File - Slice - Visualize - Bounding Box - Isosurface

3.1 Slice Field

The purpose of this section is to read, manipulate, and visualize a structured mesh dataset originating from SCIRunData.

3.1.1 Read Data File

Create a **ReadField** module by using the **Module Selector** on the left hand side of the screen. Navigate to **DataIO** subsection using the scroll bar in the Module Selector and instantiate a ReadField (Figure 3.1). Recall from the **User Guide** that a module can also be selected by giving a text input into the filter in the Module Selector (Figure 3.2).

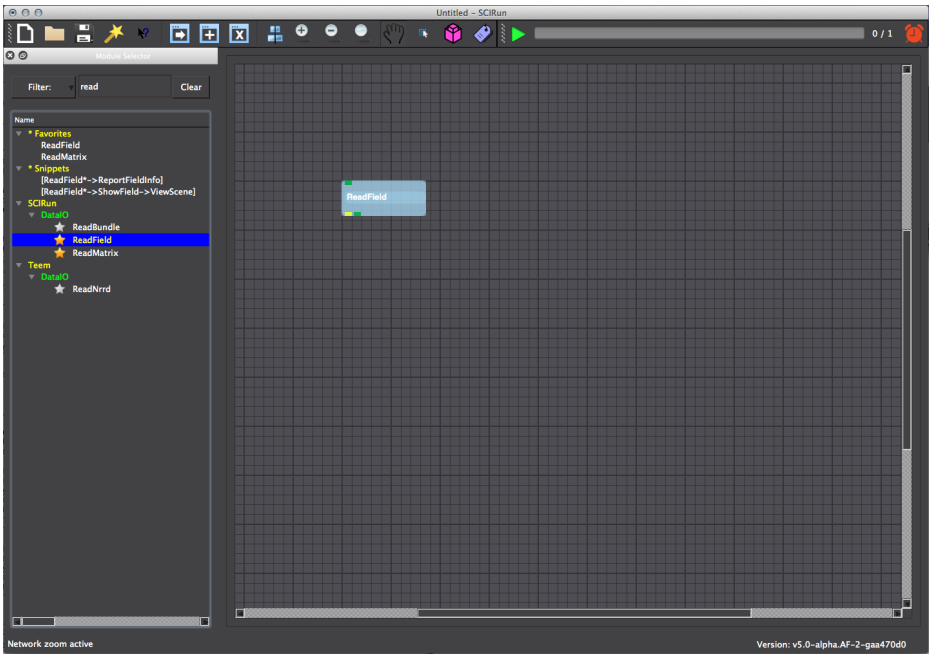

Figure 3.1. Locate ReadField module using scroll bar in the Module Selector.

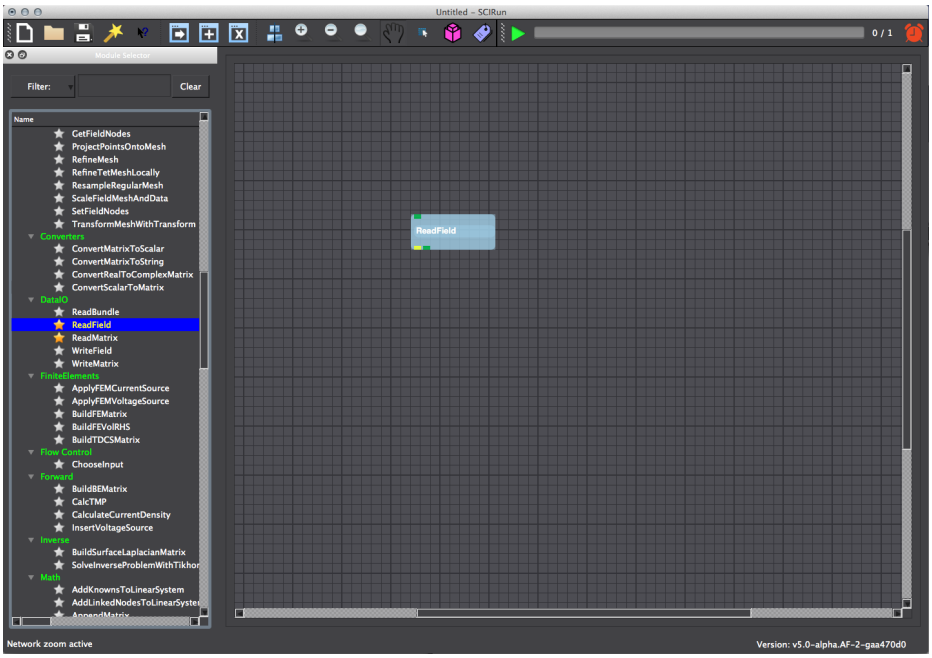

Figure 3.2. Locate ReadField module using text input into filter.

Within the ReadField **user interface (UI)**, click the open button to navigate to the SCIRunData directory and select the dataset *volume/engine.nhdr* (Figure 3.3). Notice that many different file formats can be imported by changing the file type within the Read-

Field selector window. When using Mac OSX El Capitan, press the options button in the ReadField selector window to change the file type. Change the file type to Nrrd file. The ReadField UI can be closed after selection to provide for a larger network viewing frame.

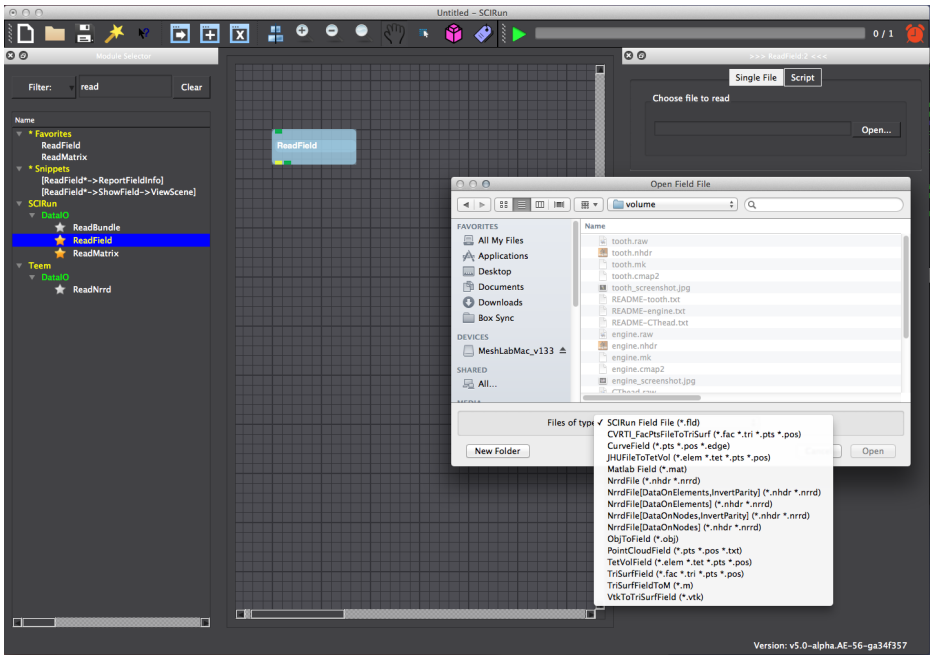

Figure 3.3. The ReadField selector window can be used to select and read many data files.

3.1.2 Slice Field

Slice the engine field by node index along a given axis by instantiating the module **GetSlicesFromStructuredFieldByIndices** in the **NewField** category and connecting it to ReadField (Figure 3.4). This can be done by using the Module Selector filter or scrolling through the list of modules in the Module Selector.

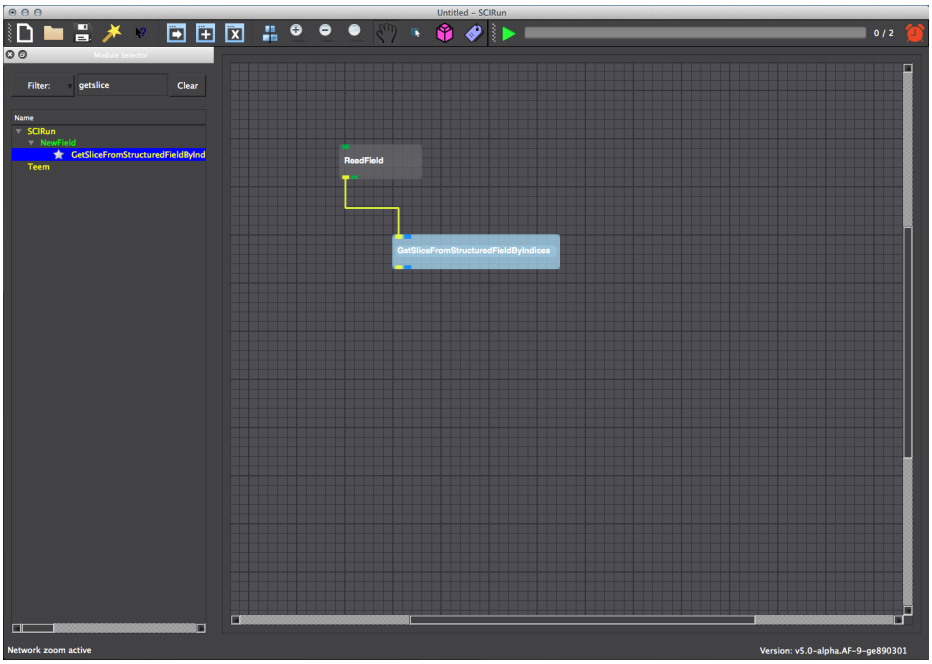

Figure 3.4. Using the ReadField port's pop-up module menu to instantiate GetSliceFromStructuredFieldByIndices.

3.1.3 Visualize Field

To visualize the field geometry, instantiate module **ShowField** in the **Visualization** category and module **ViewScene** in the **Render** category (Figure 3.5). ShowField takes a field as input, and outputs scene-graph geometry. ViewScene displays the geometry and allows a user to interact with the scene.

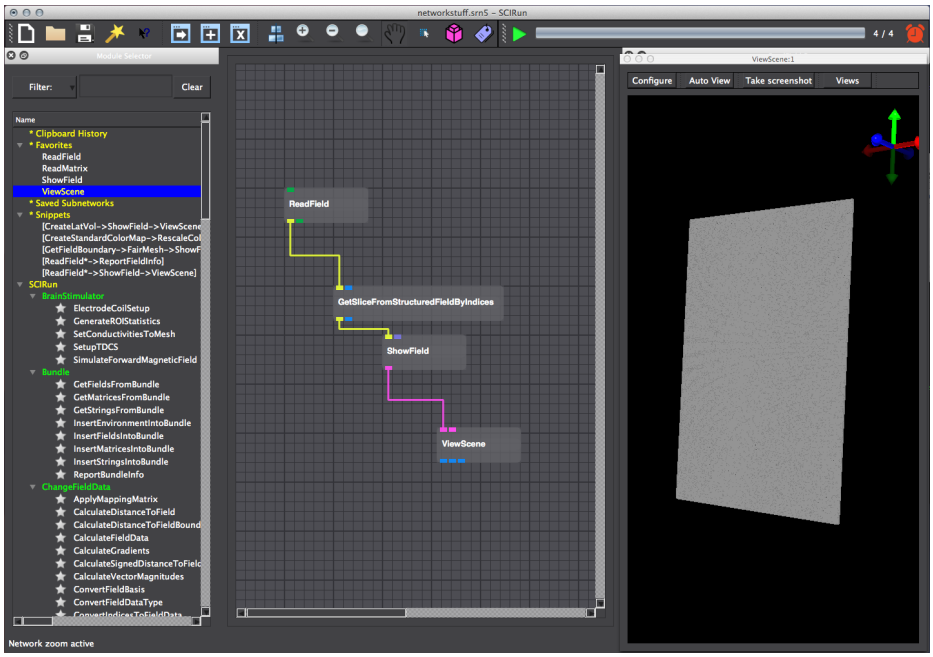

Figure 3.5. SCIRun can be used to visualize the structured mesh.

Apply a colored scale to the data values on the geometry using **CreateStandardColorMaps** and **RescaleColorMaps** modules in **Visualization** (Figure 3.6). Colors can be manipulated using the CreateStandardColorMap UI and RescaleColorMap UI (Figure 3.7). Change the coloring scheme to Blackbody using the drop-down menu in the CreatSrandard-ColorMap UI.

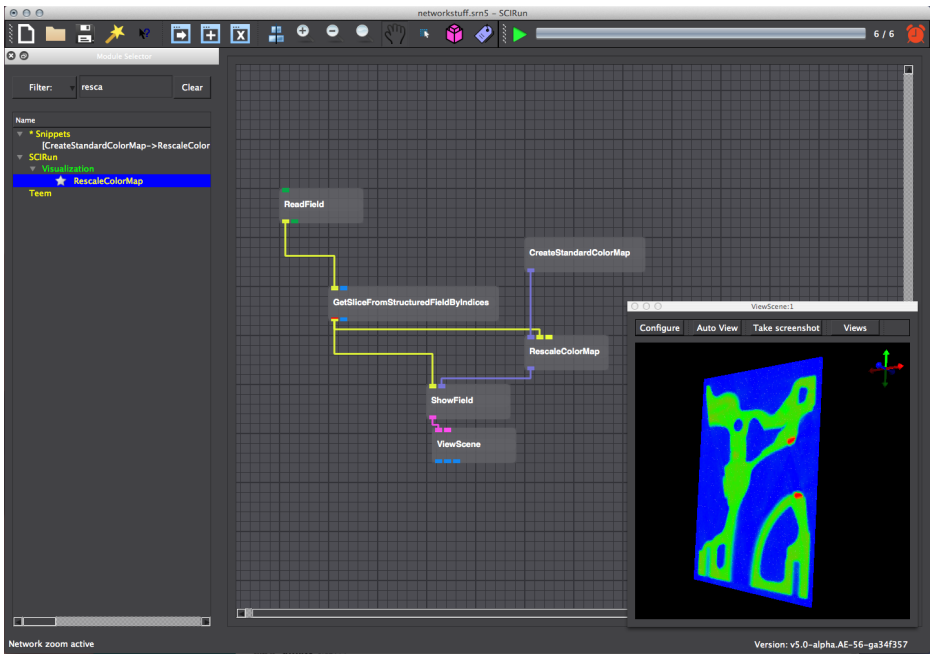

Figure 3.6. Apply and rescale a colormap to data values on the geometry.

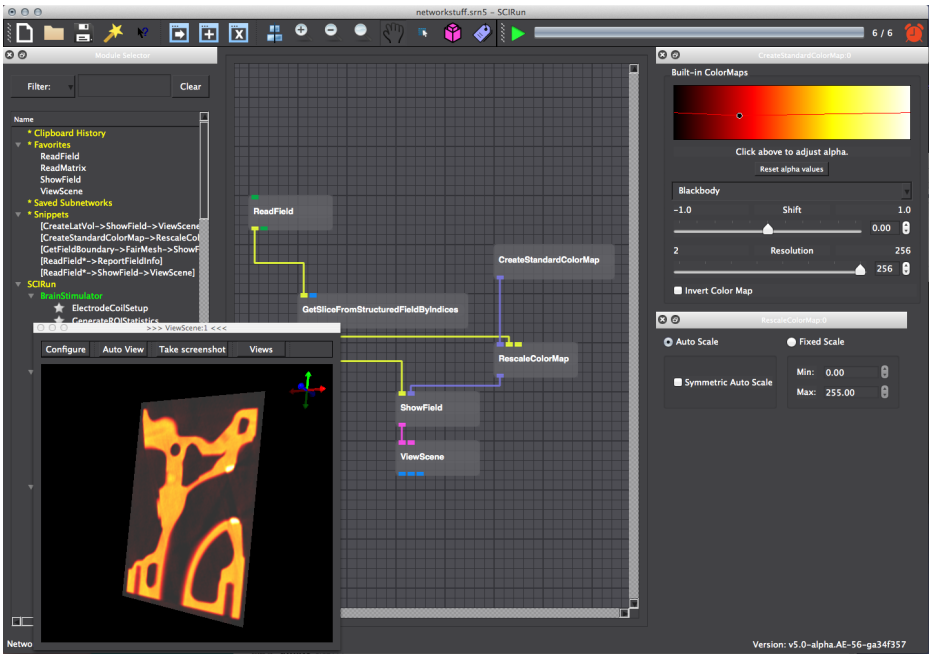

Figure 3.7. Manipulate the color scaling using both the CreateStandardColorMaps and RescaleColorMaps modules.

Return to the default color scale. Use the sliders in the GetSlicesFromStructuredFieldByIndices UI to change slice position within the geometry. Compare with figure 3.6.

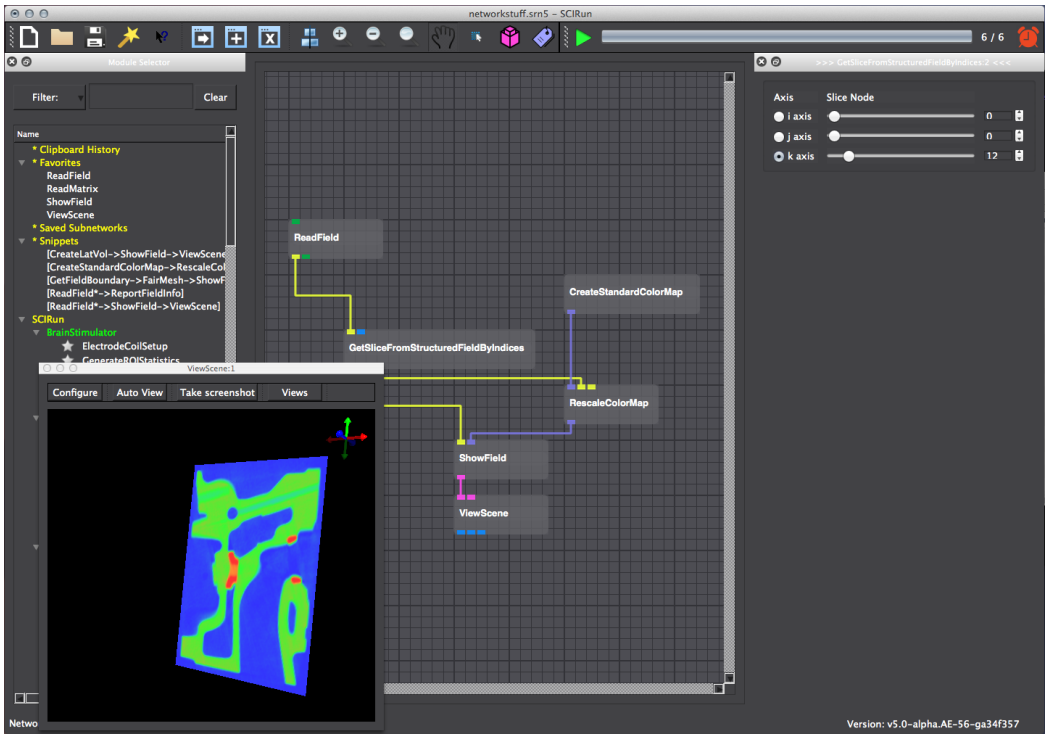

Figure 3.8. Different cross sections can be visualized within the geometry using GetSlices-FromStructuredFieldbyIndices.

3.2 Show Bounding Box

Add the **EditMeshBoundingBox** module under **ChangeMesh** (Figure 3.9). Connect it to the ReadField module and direct the output to the ViewScene module. Execute the network to visualize the bounding box of engine.nhrd. Adjust the size of the bounding box by pressing the + or - buttons under Widget Scale in the EditMeshBoundingBox UI (Figure 3.10).

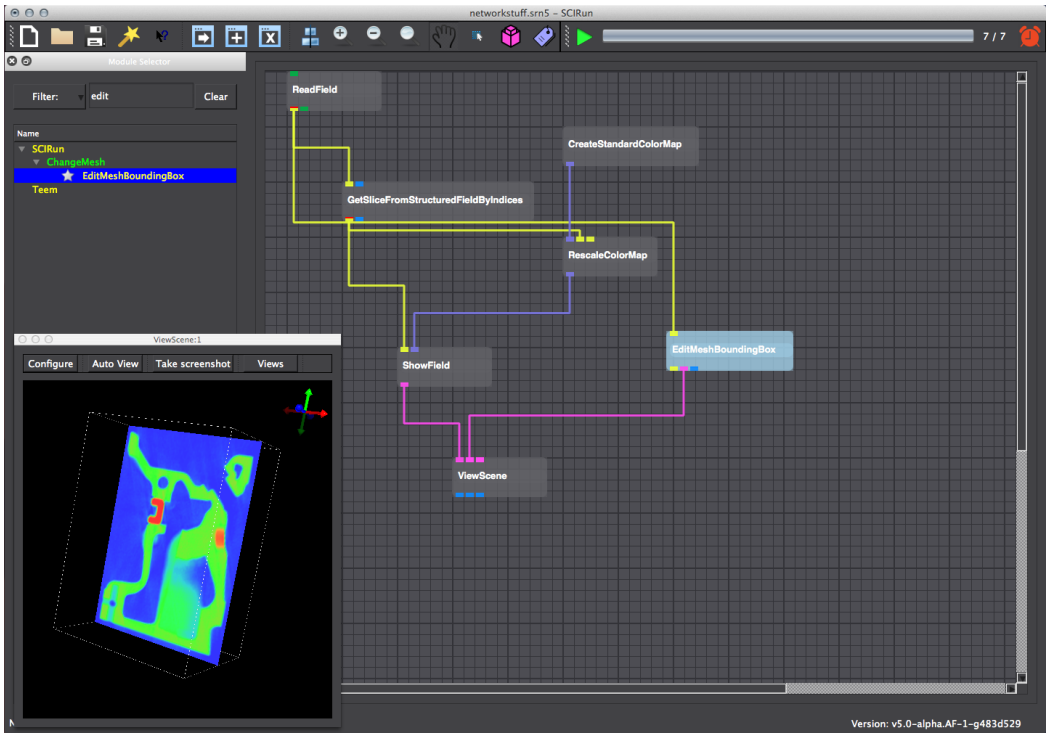

Figure 3.9. Visualize the mesh's bounding box.

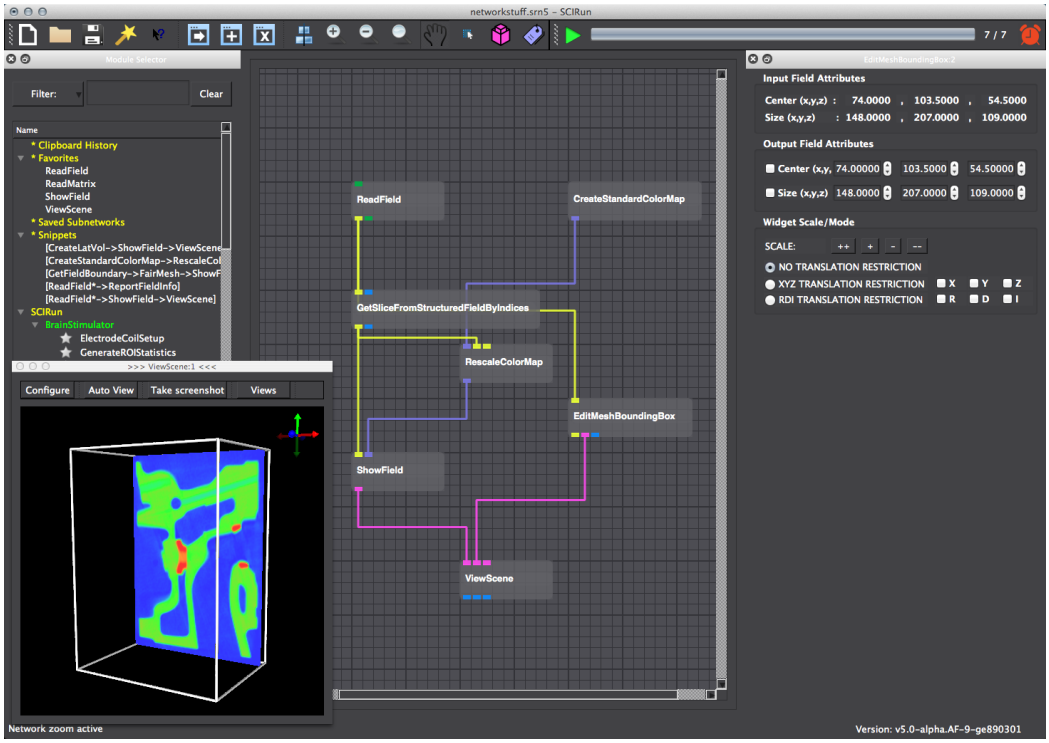

Figure 3.10. Change the scale of the mesh's bounding box using the Scale Widget in the EditMeshBoundingBox UI.

3.3 Isosurface

Construct an isosurface from the field by instantiating and connecting a **ExtractSimpleIsosurface** module to the ReadField module. The isovalue must be changed within the ExtractSimpleIsosurface UI. Open the field information by clicking on the connection between the ReadField and ExtractSimpleIsosurface and press I to bring up information. Enter a value from within the data range like 120. Visualize the isosurface by connecting it to a new ShowField module ported into the ViewScene module (Figure 3.11). Execute the network. Color isosurface output geometry by connecting the RescaleColorMap module to the ShowField module (Figure 3.12). To better view the geometry, turn off the edges within the ShowField UI (Figure 3.13).

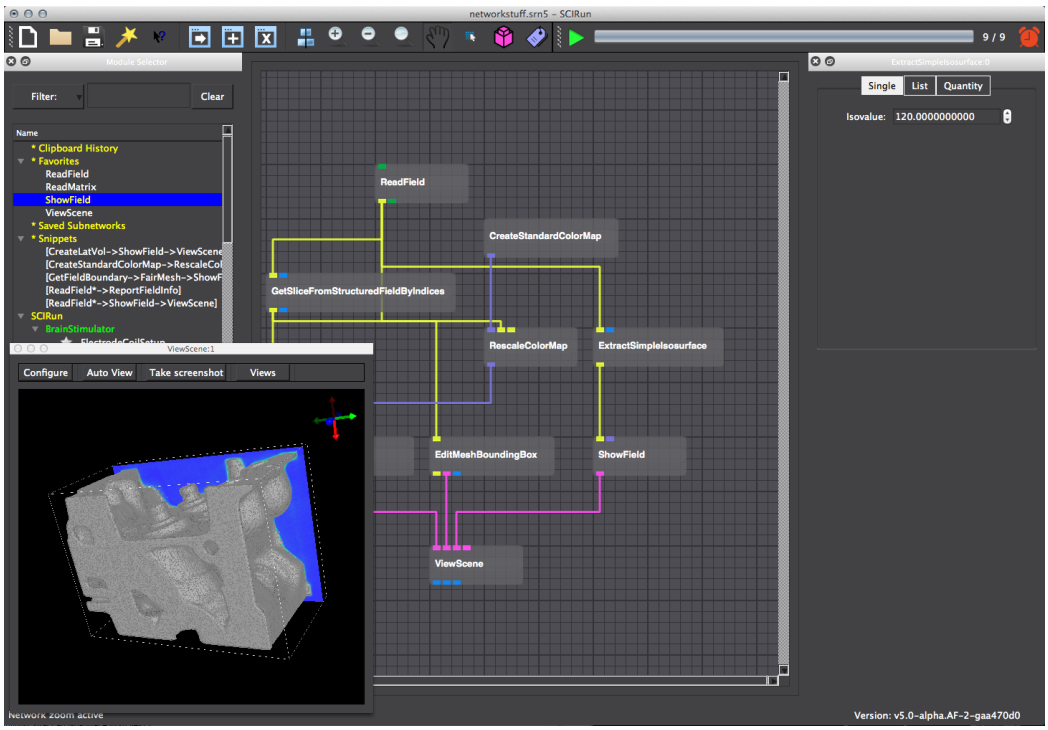

Figure 3.11. Extract an isosurface from field.

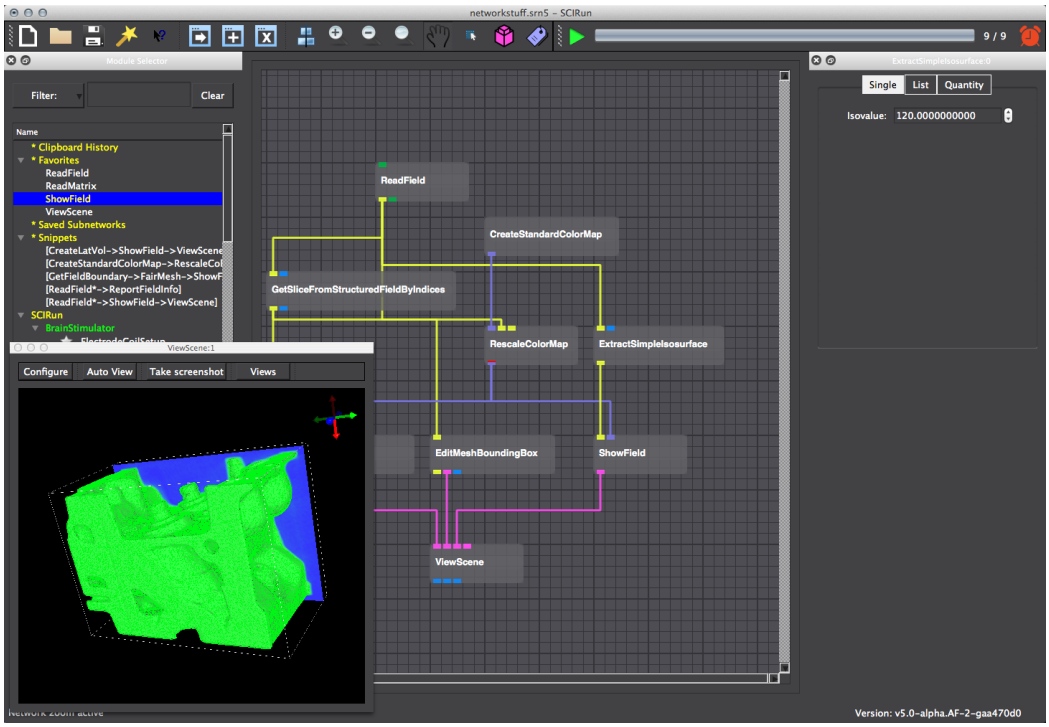

Figure 3.12. Change the isovalue within ExtractSimpleIsosurface UI.

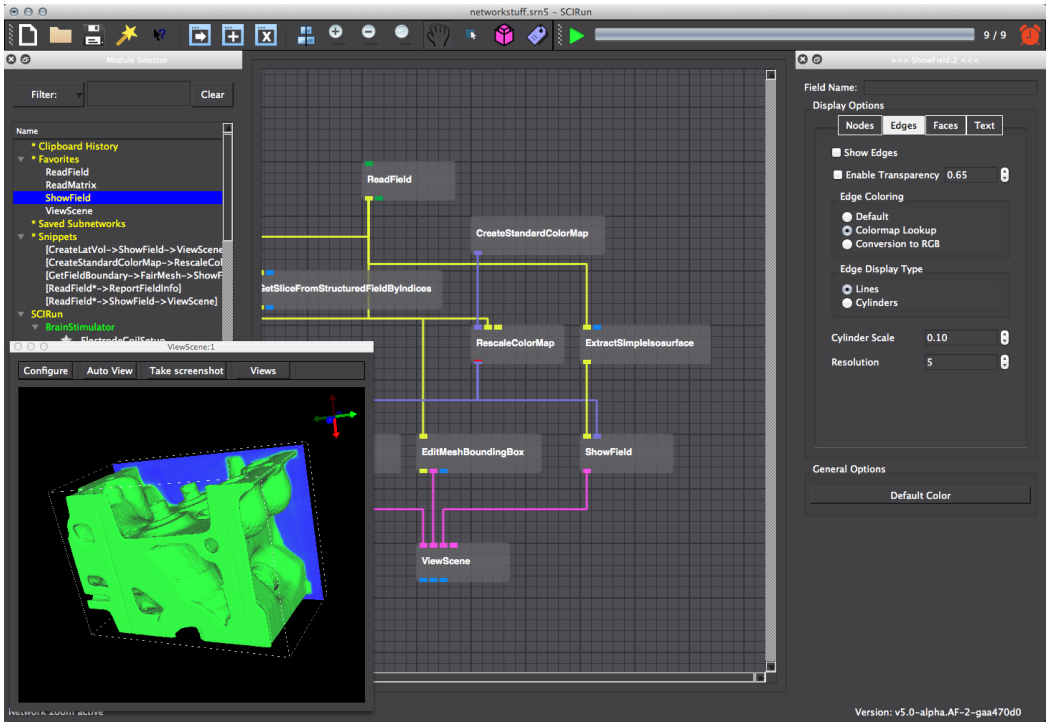

Figure 3.13. Adjusting parameters within the ShowField UI helps to better visualize the isosurface.

Chapter 4

Create, Manipulate and Visualize Field

Scope: Generate Lattice Volume - Isosurface - Visualize Geometry

4.1 Create Field

Create and manipulate a structured mesh type in this exercise. Start by creating a lattice volume using **CreateLatVol** module. Assign data at nodes using **CalculateFieldData** module. Connect CalculateFieldData to CreateLatVol. Input the expression $RESULT = sqrt(X * X + Y * Y + Z * Z)$ to compute data for each node within the CreateFieldData UI.

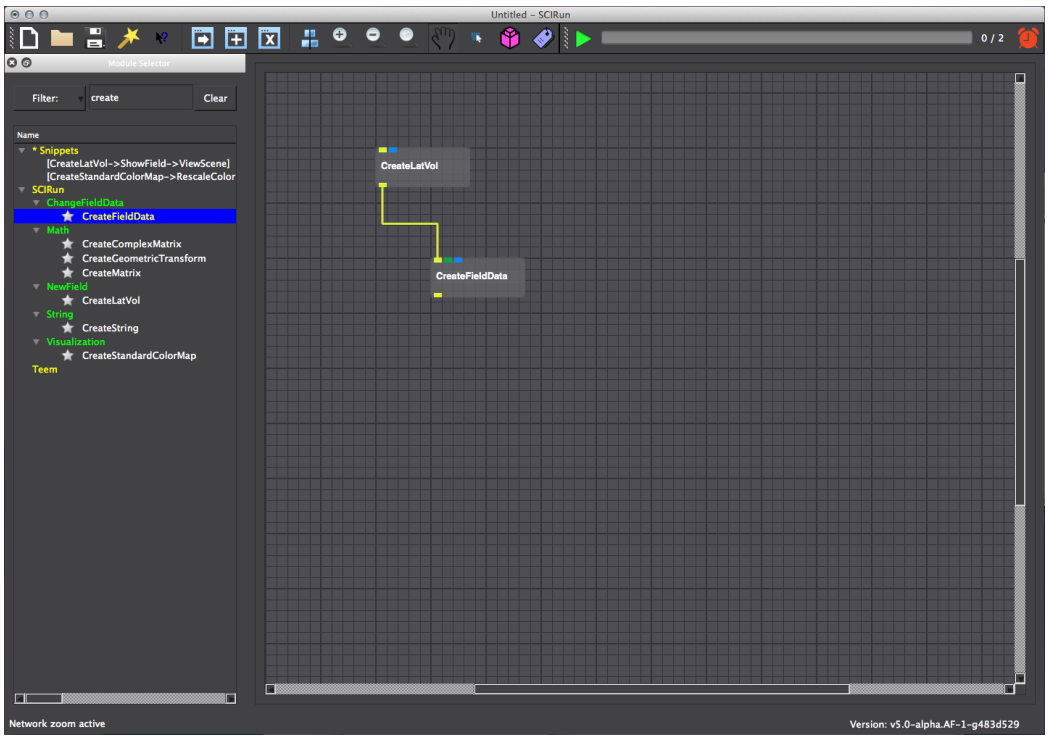

Figure 4.1. Create lattice volume field using CreateLatVol module.

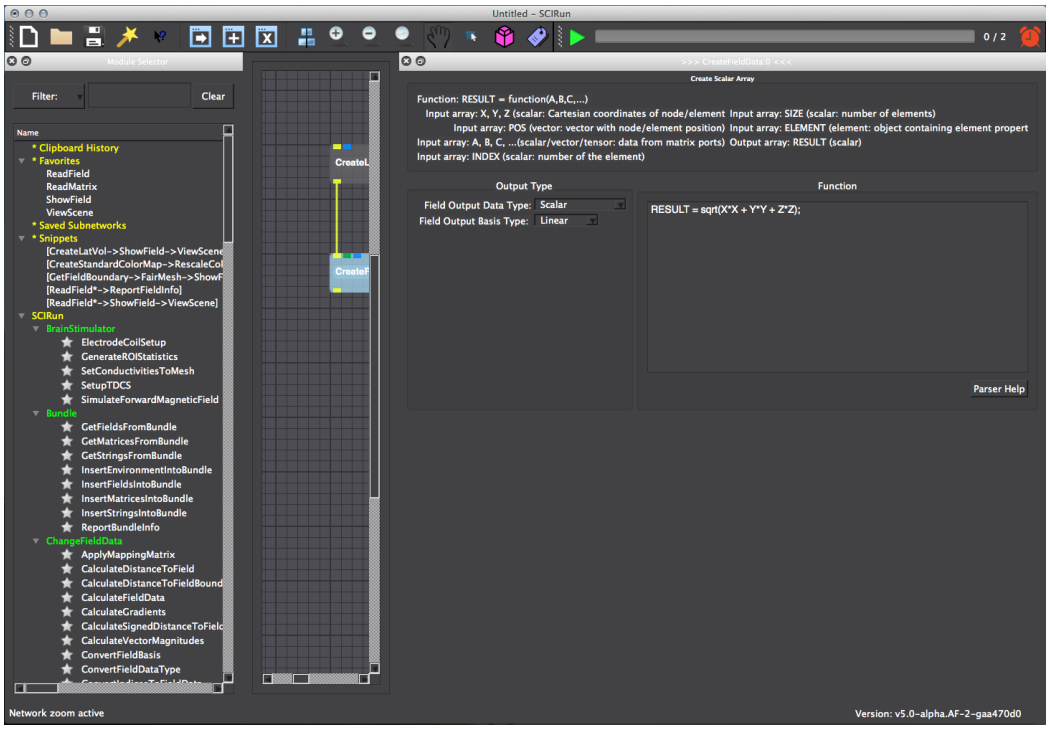

Figure 4.2. Create a new field by inputting an expression into the CreateFieldData UI.

4.2 Isosurface

Generate the isosurface by instantiating and connecting an ExtractSimpleIsosurface module to CalculateFieldData (Figure 4.3). Adjust the isovalue within the ExtractSimpleIsosurface UI so that the isosurface can be visualized (Figure 4.4). Add a color map and visualize the isosurface as in section 3.3 (Figure 4.5). Show the mesh bounding box as in section 3.2 (Figure 4.6).

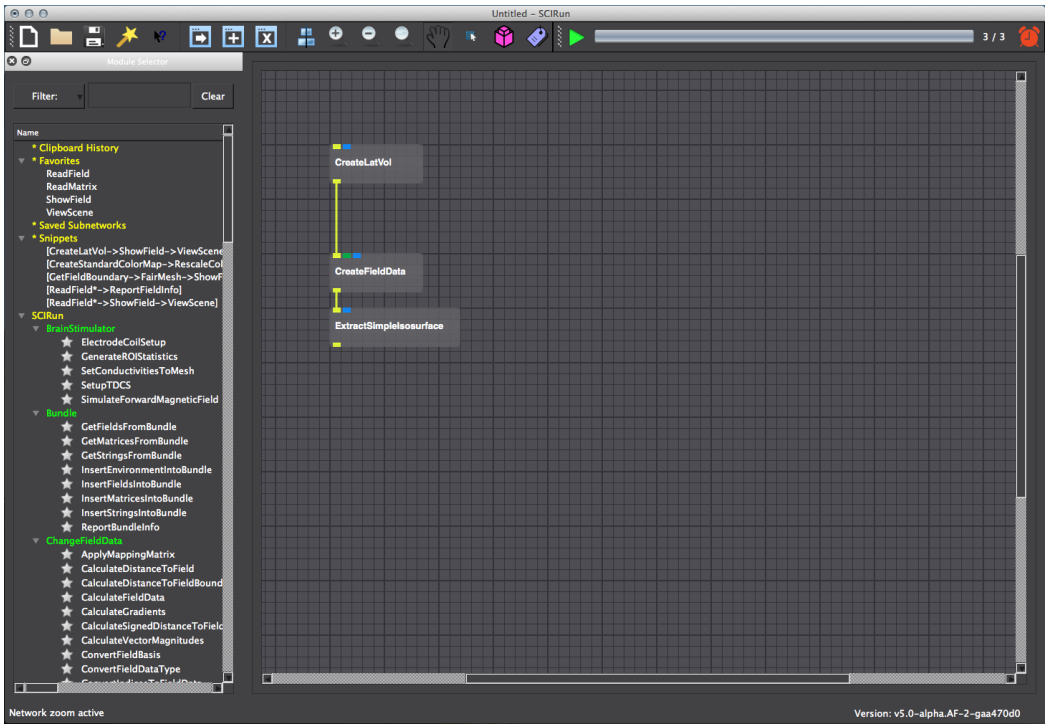

Figure 4.3. Extract an isosurface from the field data.

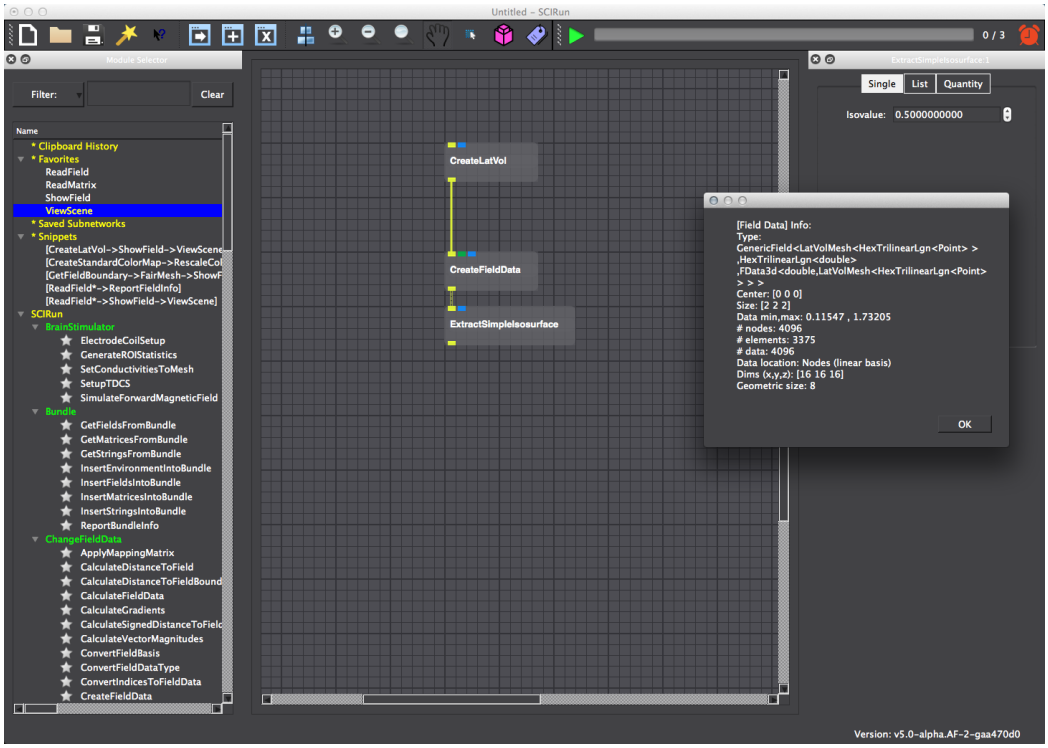

Figure 4.4. Change the isovalue so that an isosurface can be visualized.

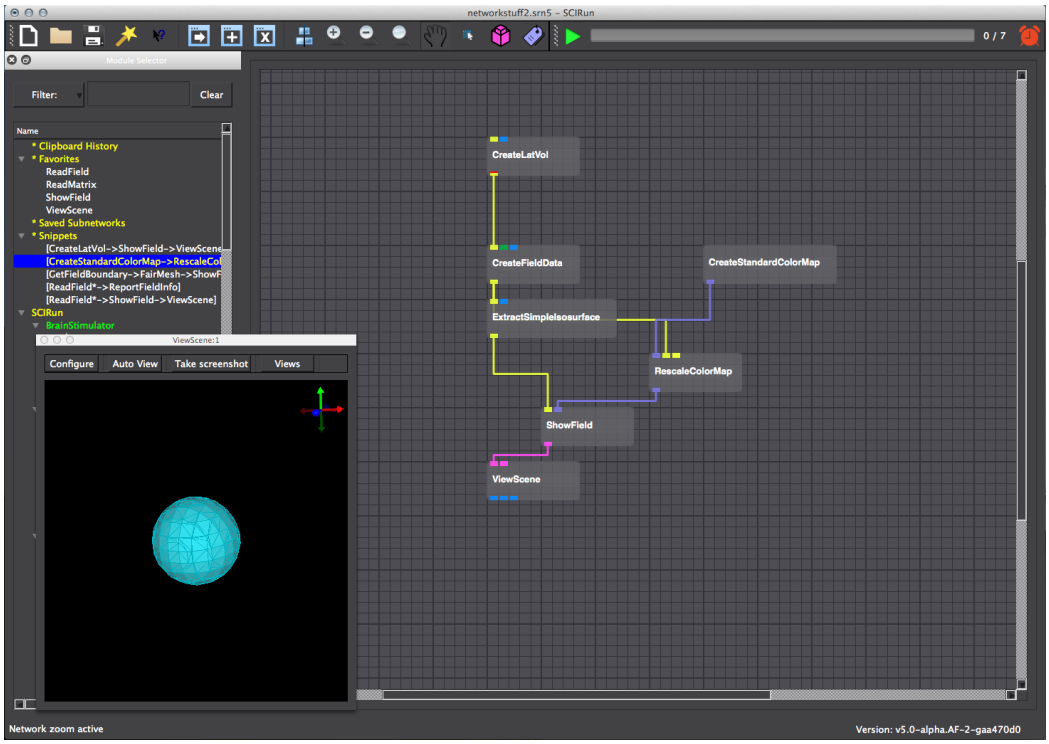

Figure 4.5. Visualize the isosurface.

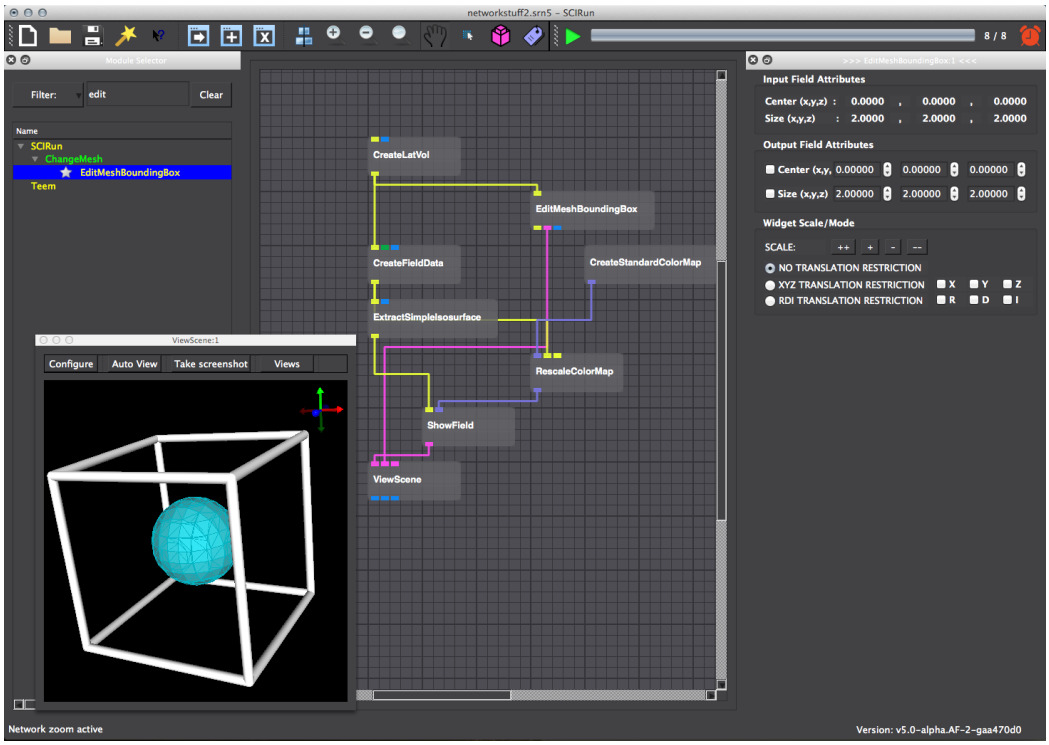

Figure 4.6. Visualize the mesh's bounding box.

4.3 Slice Field

Extend the functionality of this network by slicing the field using GetSliceFromStructured-FieldByIndices as in section 3.1.2.

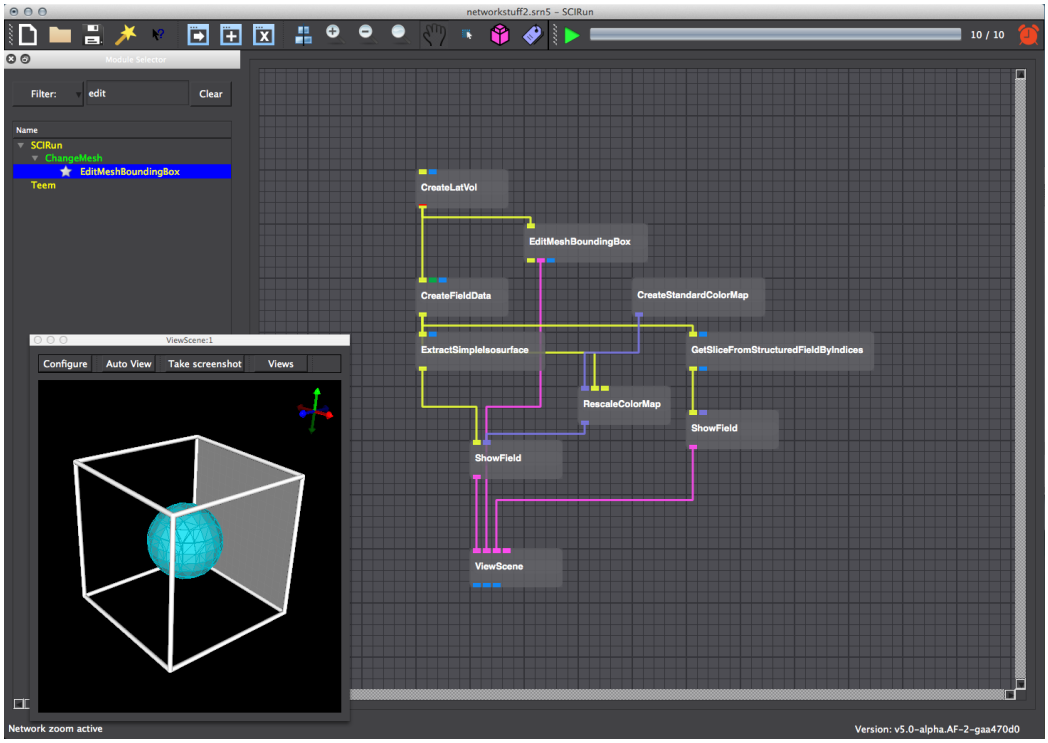

Figure 4.7. Insert GetSliceFromStructuredFieldByIndices into the network.

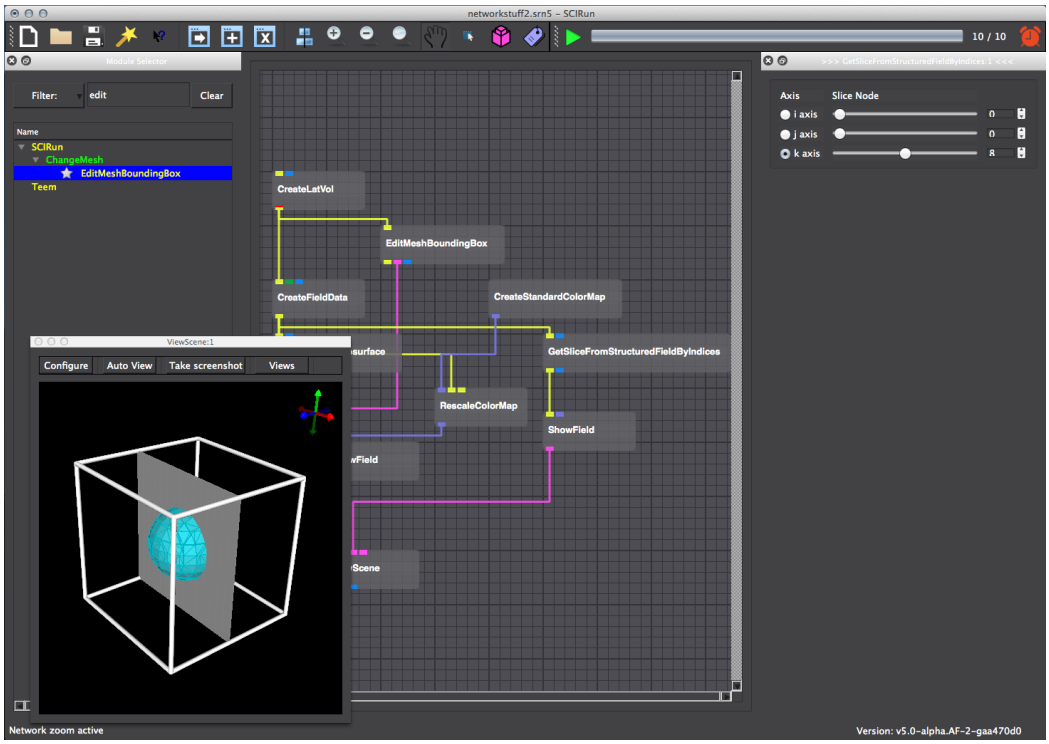

Figure 4.8. Change the slice index using the GetSliceFromStructuredFieldByIndices UI.

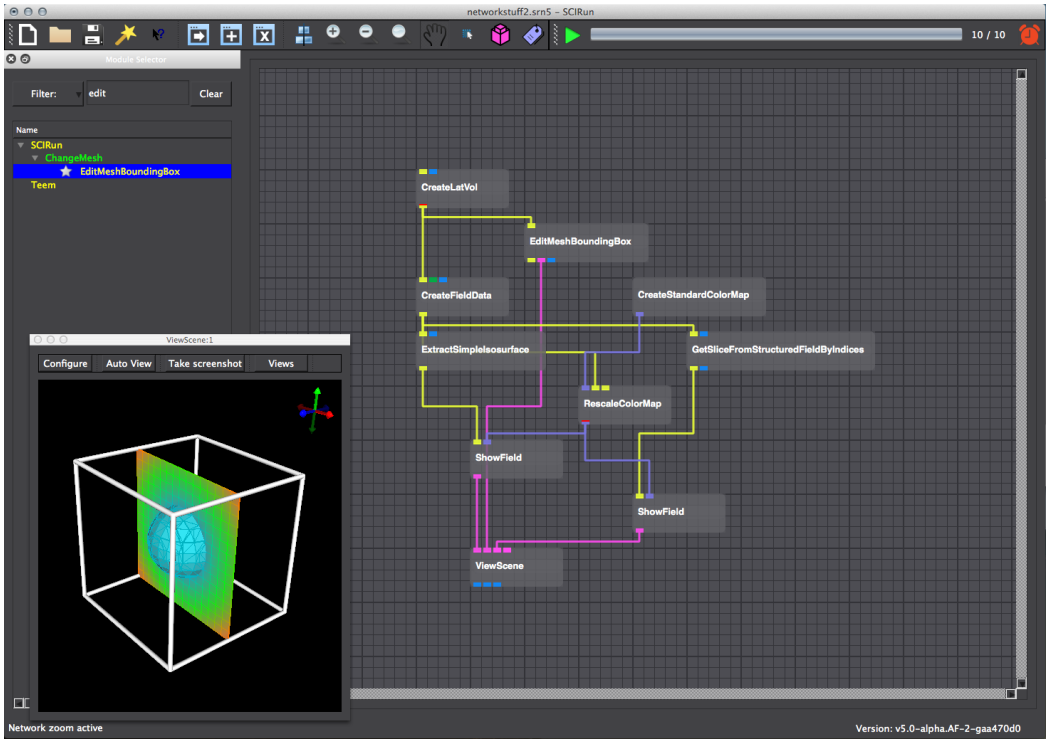

Figure 4.9. Attach the RescaleColorMap module to the ShowField module.

4.4 Clip Field

Clip out a subset of the original field by converting the lattice volume to an unstructured mesh using **ConvertMeshToUnstructuredMesh** (Figure 4.10) and adding **ClipField-ByFunction** (Figure 4.11) to the network. Set the clipping location setting in ClipField-ByFunction to *all nodes*. Use the expression $DATA1 > 1 \&\& X < 0$ to clip the field (Figure 4.12).

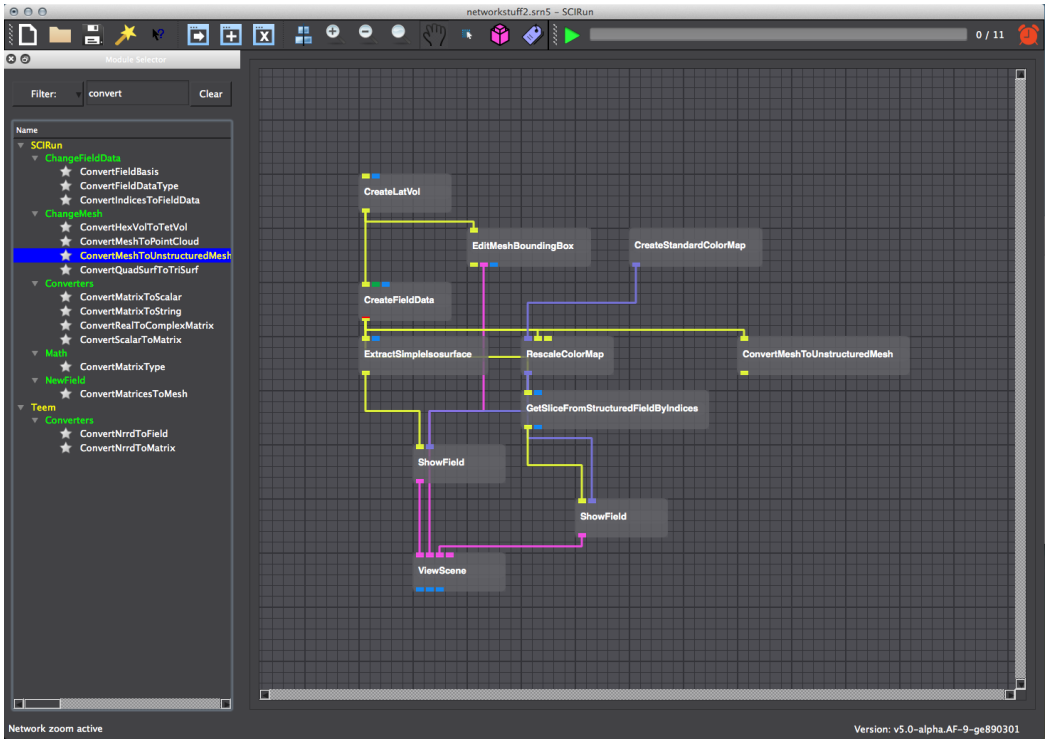

Figure 4.10. Convert the original field to an unstructured mesh.

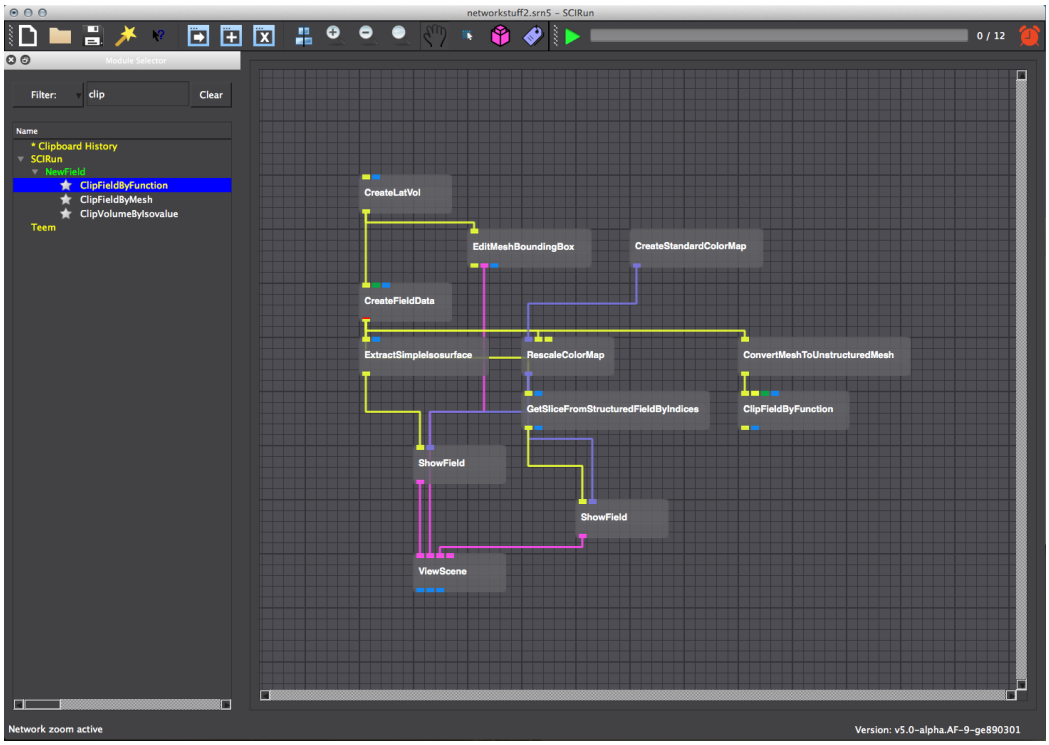

Figure 4.11. Insert a ClipFieldbyFunction module.

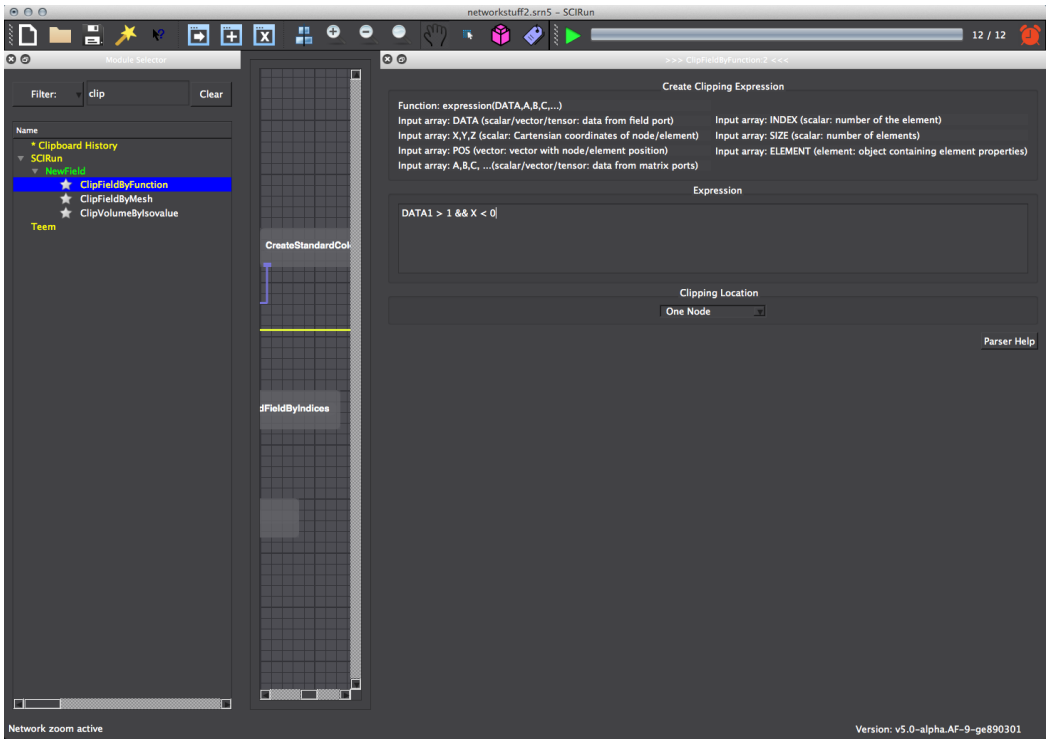

Figure 4.12. Clip the field by entering an expression in the ClipField UI.

4.4.1 Extract Boundary

At this point, it will be necessary to map the fields by interpolating the the boundary surface field to the clipping field. First, use **BuildMappingMatrix** to build a matrix that maps a linear combination of data values in the clipping field to a value in the boundary field. Then use **ApplyMappingMatrix** to multiply the data vector of the clipping field with the mapping matrix to obtain the data vector for the boundary surface field (Figure 4.13). Use GetFieldBoundary to extract the boundary surface from the lattice volume and use it as input into the ApplyMappingMatrixModule and BuildMapping Matrix (Figure 4.14). Port the output from the BuildMappingMatrix module to ApplyMappingMatrix and visualize the resultant field using a ShowFieldModule (Figure 4.15). Add a colormap to and enable transparency in ShowField UI for further functionality (Figure 4.16)

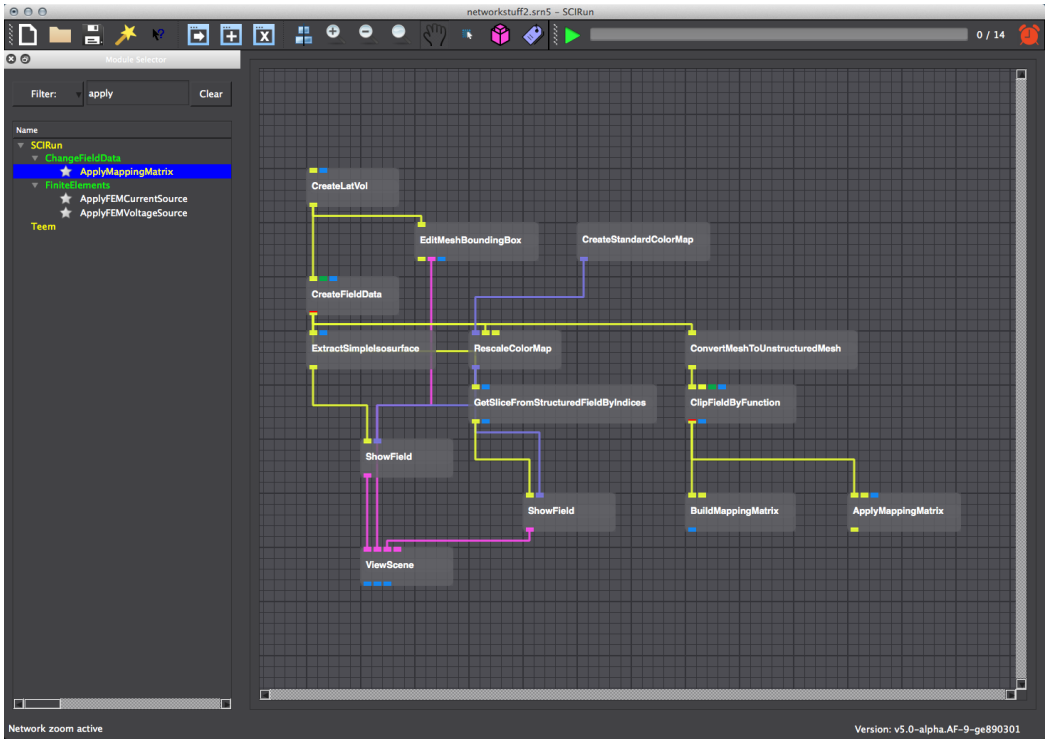

Figure 4.13. Build and apply the mapping network connections.

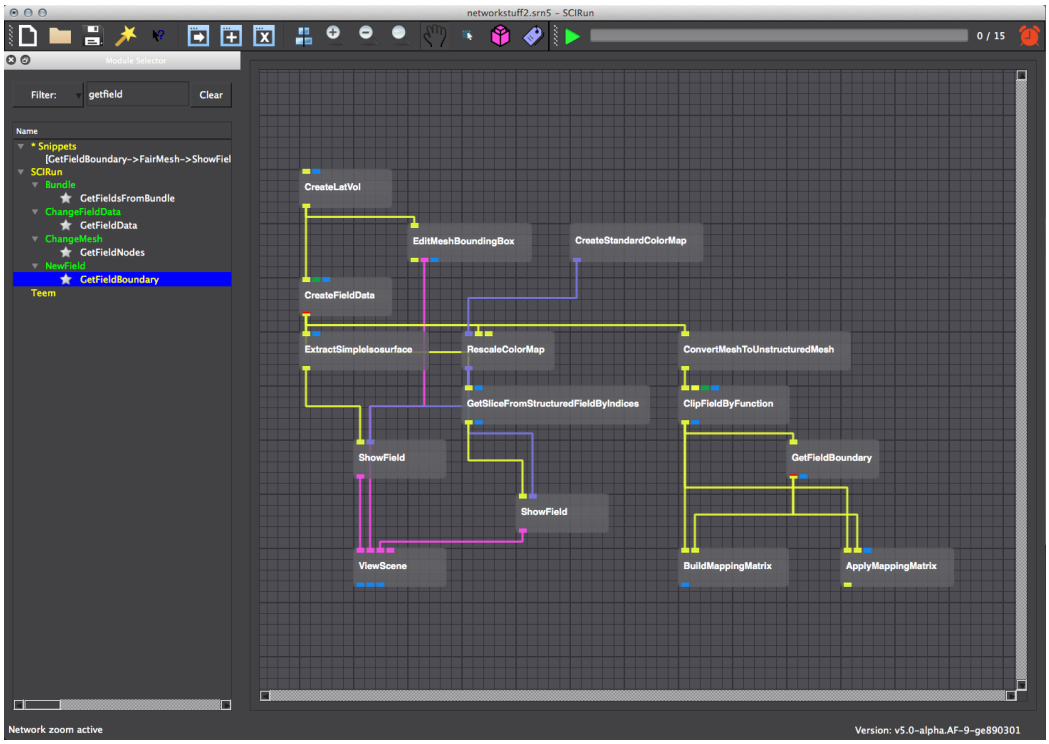

Figure 4.14. Add GetFieldBoundary to the network.

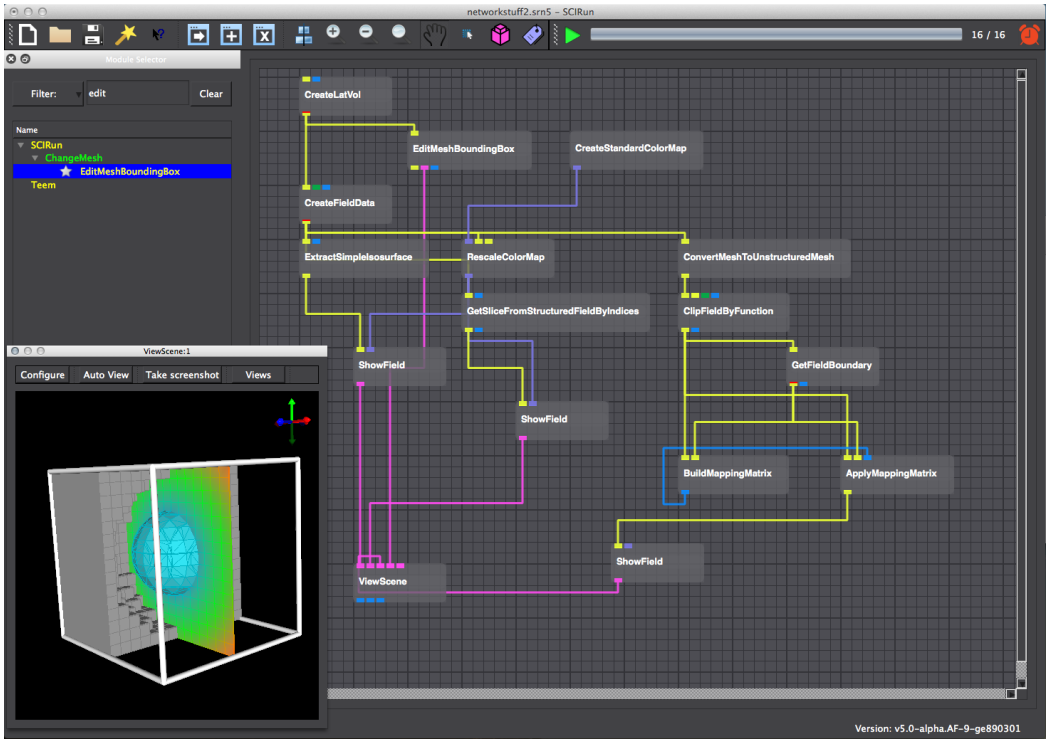

Figure 4.15. Connect all the modules for mapping and visualize the output.

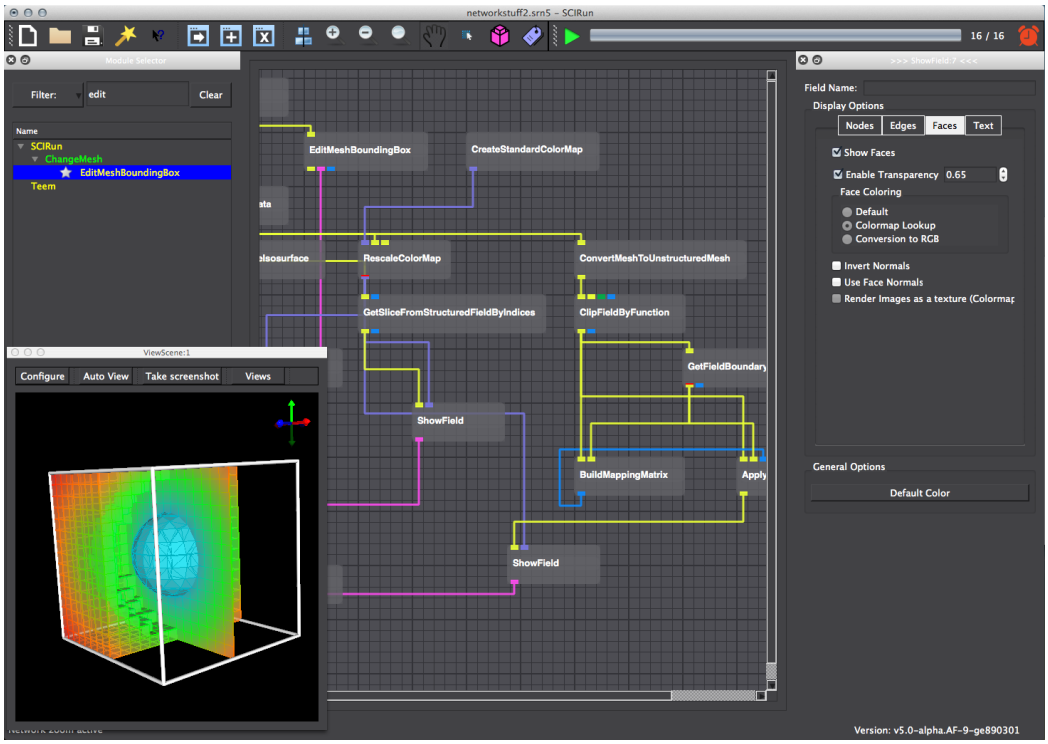

Figure 4.16. Add a colormap and enable transparency.

Finally, it is not strictly necessary to explicitly convert the original mesh to an unstructured mesh using ConvertMeshToUnstructuredMesh because ClipFieldByFunction can implicitly convert structured mesh types to unstructured mesh types before clipping the field. As a final exercise, delete ConvertMeshToUnstructuredMesh from the network and try to obtain the same result.