BrainStimulator - A SCIRun5-based Toolkit for Modeling of Transcranial Electromagnetic Stimulation

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 the National Institute of General Medical Sciences of the National Institutes of Health under grant number **(P41GM103545)**

Author(s):
Moritz Dannhauer, Spencer Frisby

Contents

Chapter 1

Overview

This tutorial demonstrates both non-invasive electrical brain stimulation, known as transcranial direct current stimulation (tDCS), and non-invasive magnetic brain stimulation, known as transcranial magnetic stimulation (TMS), can be modeled using SCIRun5. It explains briefly the basics of tDCS and TMS, such as the mathematical fundamentals, how to set up tDCS/TMS simulations, and how to visualize results based on a simple geometric model: a label-mask data set containing multiple spheres forming a mickey mouse head. Besides the Mickey Mouse toy model, BrainStimulator contains a realistic human head model that was derrived from the well-known Colin27 brain atlas in flavors of detail.

1.1 Transcranial Direct Current Stimulation (tDCS)

Transcranial direct current stimulation (tDCS) is a non-invasive technique that effects human brain functions. It is an emerging therapeutic approach to support the treatment of a wide range of neurological conditions, such as mood disorders, and is also known to enhance cognitive functions such as memory and motor skill learning. tDCS aims to modulate specific regions of the brain by injecting low amplitude direct current through surface electrodes attached to the scalp of the subject.

1.1.1 Mathematical Modeling of tDCS

For the purpose of computing a tDCS forward solution, the following equation system has to be solved: $M \cdot U = I$, where M is the tDCS forward Matrix (module: BuildTDCSMatrix) and U is the solution potential vector given the injected current vector, the right hand side I (module: SetupTDCS). M combines volume conduction properties (FEM stiffness matrix, module: BuildFEMatrix) and electrical boundary conditions (module: SetupTDCS), including the complete electrode model to simulate current injection that solves the Poisson equation:

$$
\begin{aligned}
\nabla \cdot (\sigma \nabla u) &= 0, (x \in \Omega), \\
u + z_l \sigma \frac{\partial u}{\partial n} &= U_l, (on\ \partial\Omega_{e_l}, x \in e_l), \\
\int_{e_l} \sigma \frac{\partial u}{\partial n} &= I_l (l = 1, 2, ..., L), \\
\sigma \frac{\partial u}{\partial n} &= 0, (x \in \partial\Omega \setminus \cup_{l=1}^{L} e_l).
\end{aligned}
$$

In more detail, M is composed of the regular FEM stiffness ($A \in N \times N$, with N representing the number of FEM nodes) and additional electrical boundary conditions specified as submatrices A_2, B, C ($B \in N \times L$, $C \in L \times L$, with L representing the number of electrodes). M is composed such as:

$$
\begin{aligned}
M &= \begin{pmatrix} A & -B \\ -B^T & C \end{pmatrix} \\
\begin{pmatrix} A & -B \\ -B^T & C \end{pmatrix} \begin{pmatrix} U_n \\ U_e \end{pmatrix} &= I = \begin{pmatrix} I_n \\ I_e \end{pmatrix}, \\
A(i,j) &= A_1(i,j) + A_2(i,j) = \int_\Omega \sigma \nabla \phi_i \cdot \nabla \phi_j \ d\Omega + \\
& \sum_{l=1}^{L} \int_{e_l} \frac{1}{z_l} \phi_i \phi_j \ d\Omega_{el}, \\
B(i,l) &= \frac{1}{z_l} \int_{e_l} \phi_i \ d\Omega_{el}, \\
C(i,l) &= \frac{1}{z_l} \int_{e_l} \ d\Omega_{el},
\end{aligned}
\tag{1.1}
$$

Here, the donations are: $M \in \mathbb{R}^{N+L \times N+L}$, $A_1, A_2 \in \mathbb{R}^{N \times N}$, $B \in \mathbb{R}^{N \times L}$, $C \in \mathbb{R}^{L \times L}$, U_n potentials at nodes, U_e represents potentials at electrodes, I_e is currents at electrodes and ϕ denotes linear basis functions.

The posed mixed electrical boundary problem can be efficiently solved with SCIRun5 modules (see more below). The implemented complete electrode model enables us to model a non-uniform current distribution across the surface of the electrodes. It also allows adjustments to be made for the electrode-scalp contact relationship using a resistive electrode impedance z_l, which is often available in experimental settings.

1.2 Transcranial Magnetic Stimulation (TMS)

Transcranial Magnetic stimulation (TMS) is a non-invasive technique that influences human brain function. TMS is an FDA-approved tool used in basic and clinical neuroscience. A

TMS coil can be freely positioned close to the scalp by the experimenter (often guided by a navigation system) to target a specific brain region of interest (ROI).

1.2.1 Mathematical Modeling of TMS

Modeling TMS involves the computation of magnetic fields generated by the TMS coil for a particular ROIs (e.g. biological brain tissues). The magnetic field originating at the TMS coil can sufficiently be approximated by a number of magnetic dipoles. Those magnetic dipoles describe the magnetic field that the coil is emitting and depend on shape and other technical specifications of the coil. Typical shapes of TMS coils can consist of a single or a double-ring coil. Each subcoil of the most common double-ring coil (8-shaped coil) is wound in opposite direction, which focuses the stimulation at the mid-point of the entire TMS coil. To compute the electrical current density J that is induced by the TMS coil the following equation needs to be solved:

$$J \quad = \quad -\sigma(\nabla\phi + \frac{dA}{dt}),$$

with $\nabla\phi$ being the gradient of the electrical potential (ϕ) and $\frac{dA}{dt}$ being the time derivative of the magnetic vector potential generated by the coil both being multiplied by $-\ \sigma$, the electrical conductivity tensor. Firstly, the magnetic vector potential (module: SimulateForwardMagneticField) is computed at each node location of the ROI mesh. For each of these node locations, a mean conductivity tensor ($\bar{\sigma}$) of the surrounding mesh elements is computed and multiplied by the magnetic vector potential. The result can be seen as a *primary current* ($J = \sigma \cdot A$, here: assumed to be constant for the considered time interval) induced at the ROI node location. This primary current depends solely on the position, orientation and magnetic field profile of the TMS coil for a given conductivity distribution in the ROI. Secondly, the induced primary currents (J) modeled as electrical boundary (volumetric source,module: BuildVolRHS) conditions, *secondary currents* can also be computed with SCIRun.The computation of secondary currents by solving the FEM problem is rather time consuming and experiments showed that its impact is small (compared to primary currents) on the current density J. In this tutorial, both, primary and secondary current contributions are computed.

Chapter 2

Software requirements

SCIRun Compability

The modules demonstrated in this tutorial are available in SCIRun version 5.0 and are not compatible with any older version of SCIRun. Be sure to update your SCIRun version to the latest build available from the SCIRun website, which will include the latest bug fixes and will make sure that the capabilities demonstrated in this tutorial are up to date.

Required Datasets

This tutorial relies on several datasets that are all part of the SCIRunData bundle. To obtain these datasets, go to the SCIRun website and click the **Download** button. Instead of the SCIRun source or binary files, download the SCIRunData archive files. The latter is available as a zip file or as a gzip file.

Chapter 3

Finite Element Modeling

We used a standard approach to solve finite element problems for tDCS (for details review the Defibrillation tutorial, chapter 2). For TMS, an established forward modeling approach is employed (Thielscher et. al, 2004,Clinical Neurophysiology and later publications, www.simnibs.de).

3.1 Preparation of Finite Element Model

The finite element mesh (file name: mesh.mat) was generated using our meshing package (Cleaver) based on the label mask file (file name: mickey4tDCS.nrrd). The file mickey4tDCS.nrrd consists of the labels background, mickey's body, right ear, left ear, two tDCS target regions that have distinct label mask values 1, 2, 3, 4, 5, 6. It is used as a geometrical basis for tDCS and TMS. Any meshes used for simulations are generated within SCIRun5.

3.1.1 Tissue segmentation: Mickey Mouse toy model

The simulation setup of the tDCS mickey mouse example 6 sites to place electrodes can be identified, which are parallel to the axis. There are two target regions usable for tDCS and TMS, one located superficially and one deeply inside Mickey's head 3.1.

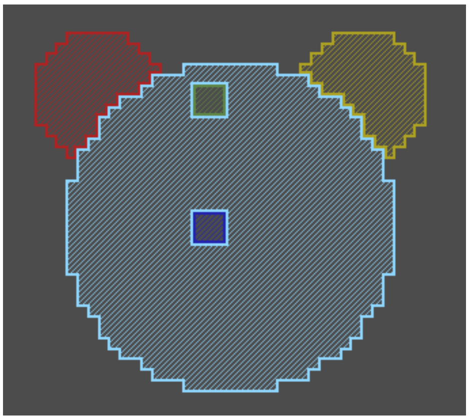


Figure 3.1. Toy example - Mickey mouse segmentation (depicted tissue labels: red/light brown=left/right ear, light blue=body, dark blue/green=deep/superficial ROI)

3.1.2 Generating finite element meshed (FEM) within SCIRun5 from segmentations

The SCIRun5 network shown in 3.2 implements a way to generate a tehtrahedral mesh based on a segmentation (*.nrrd file) that need to be loaded so that the data elements are defined on the elements. The loaded field will be separated based on the label values (1..6) and for each a triangle surface is generated. To create a valid input for the meshing tool cleaver 1.0 (module: InterfaceToCleaver) all triangle surfaces need to be represented implicitly as indicator functions. Those indicator functions can be defined as signed distances to the surfaces of the materials. The creation of signed distance fields requires a regular grid where the signed distances are stored. That grid (module "CreateLatVol") is created in the top part of this network and aligned with the segmentation (module "AlignMeshBoundingBoxes") as well.

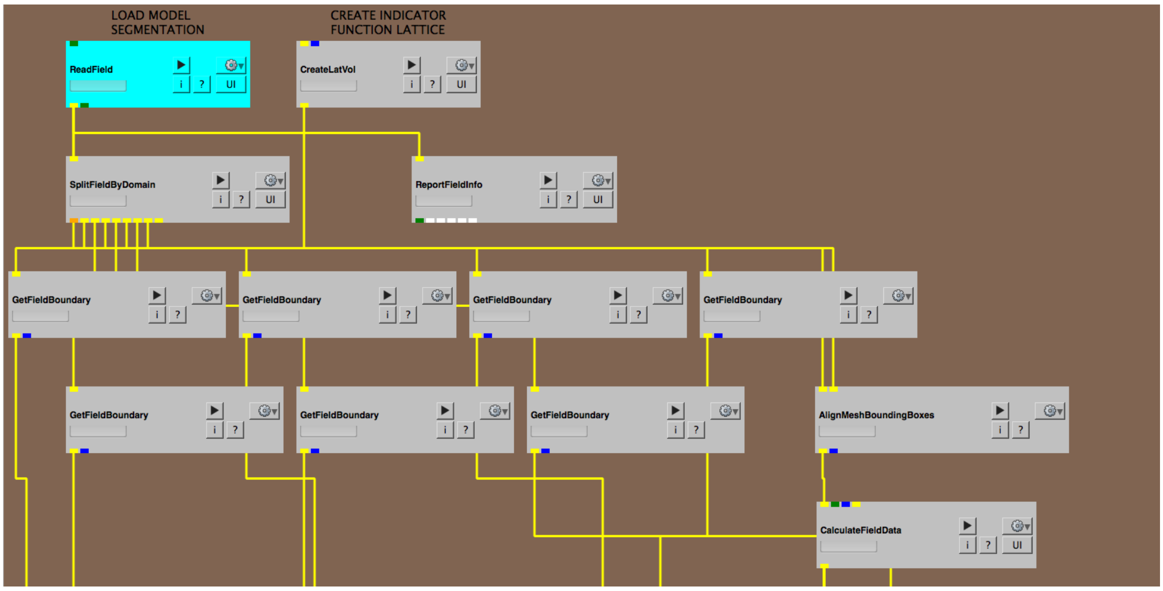


Figure 3.2. Mickey mouse segmentation: Tetrahedral mesh generation, SCIRun5 network (1/2)

The second part of the network (3.3) takes all tissue surfaces and the regular grid (latvol) to compute signed distances that will serve as indicator functions. Each output (signed distance field) piped into a CalculateFieldData module to be converted to float. All signed distance fields are piped into Cleaver (Cleaver expects to have positive signed distance values inside and negative values outside of it, for each material (also background) a positive distance value needs to be defined in one input).

3.2 Simulating Transcranial Direct Current Stimulation (tDCS)

The SCIRun5 network discussed in the following can be found at `src/ExampleNets/BrainStimulator/TDCS`

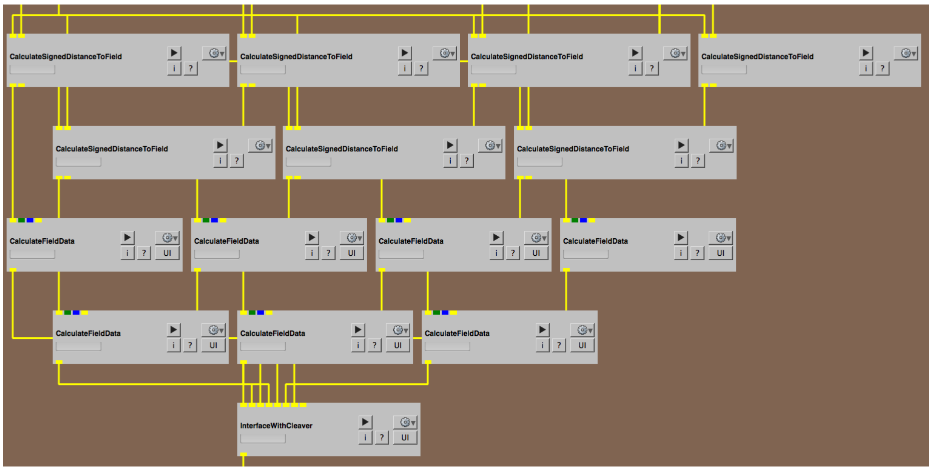

CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
InterfaceWithCleaver

Figure 3.3. Mickey mouse segmentation: Tetrahedral mesh generation, SCIRun5 network (2/2)

3.2.1 SCIRun5 tDCS network: Part 1/5

The only difference of figure 3.4 to figure 3.2 are the 2 modules in the top right corner. One creates a matrix of electrode locations (x, y, z) and the other one reads the shape of a tDCS electrode prototype.

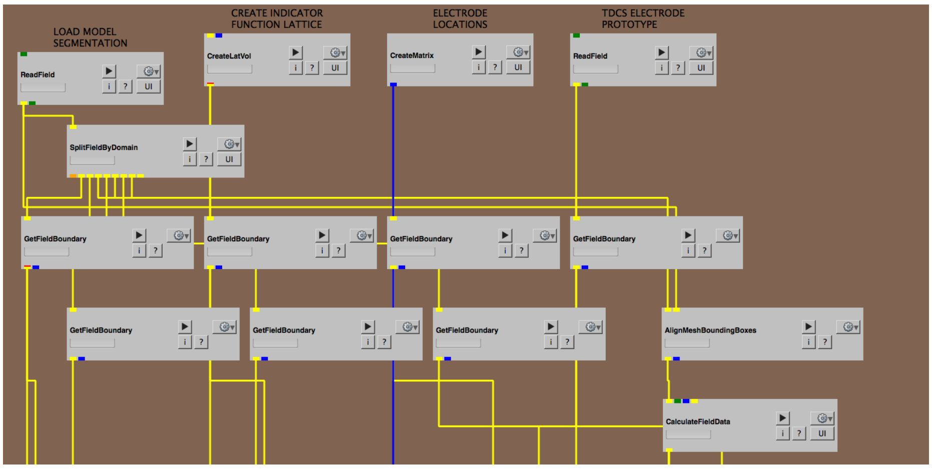


Figure 3.4. tDCS SCIRun5 network (1/5)

In figure 3.7, the module ElectrodeCoilSetup is highlighted which positions each electrode on the scalp and generates a Field of definied electrode shapes that fit the scalp surface and can be used later to create a signed distance field for the electrodes (as an input for Cleaver).

Figure ?? demonstrates this process of the model mesh generation is performed by the InterfaceWithCleaver module that takes all tissues as well as the electrode inputs (signed distance fields) and outputs a tetrahedral mesh. This tetrahedral mesh is split into labels whereas the background, which is not needed for computations, is stripped. The scalp and electrode tissue surfaces are created to set up FEM computations.

In figure ?? presents the computation core of the network. The created tetraherdral

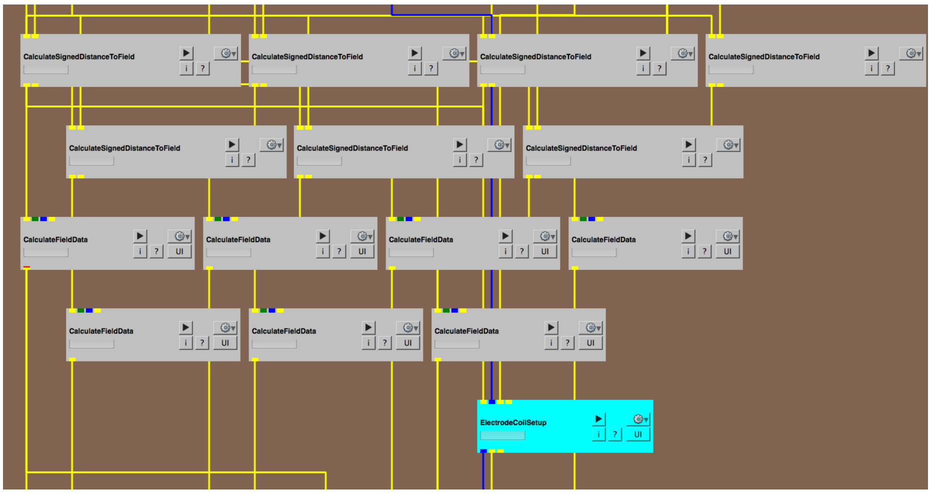

CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
ElectrodeCoilSetup

Figure 3.5. tDCS SCIRun5 network (2/5)

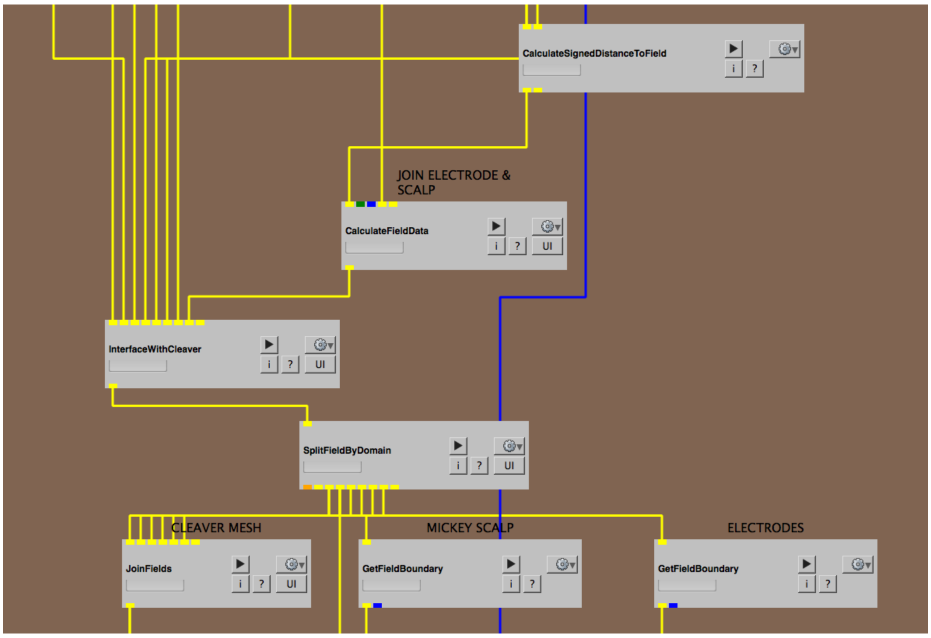


Figure 3.6. tDCS SCIRun5 network (3/5)

mesh is populated with isotropic conductivity values from the SetMeshConductivities module (user input possible to change conductivities) and piped directly into the BuildFEMatrix module to generate the FEM stiffness matrix A_1. The module SetupTDCS takes all the information on the tDCS electrode definition and creates outputs to be able to set appropriate electrical boundary conditions (user input possible to define electrode current intensities and reference node). Those outputs are piped mainly into the module BuildTDCSMatrix to create the TDCS forward matrix A. A and I (SetupTDCS) are piped into SolveLinearSystem to solve $A \cdot U = I$ iteratively for the potential vector U.

The computed solution U is split into the nodal potential vector U_n that is only needed for further computations (figure 3.8). Then, the nodal potentials are assined to the previ-

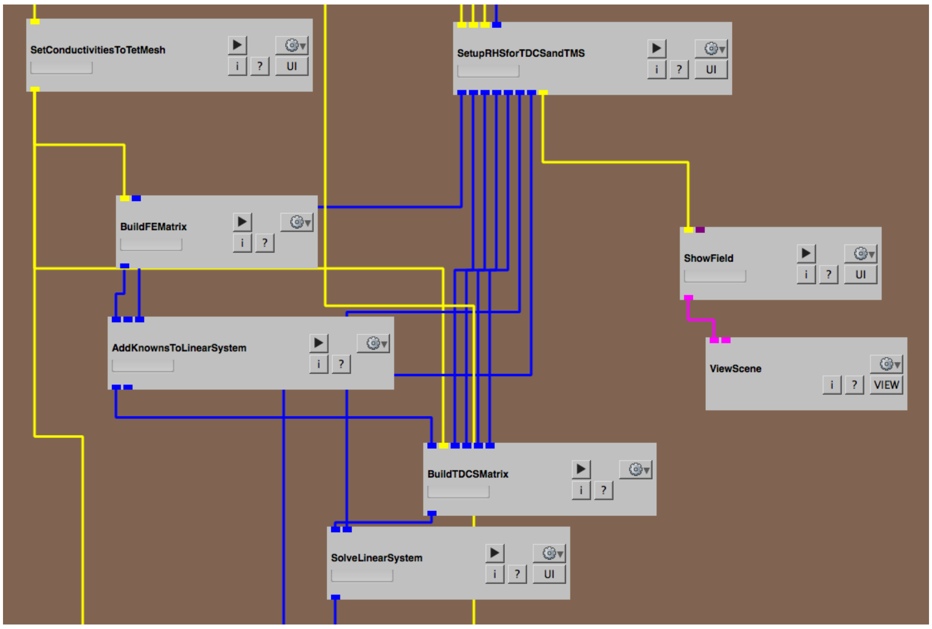


Figure 3.7. tDCS SCIRun5 network (4/5)

ously generated tetrahedral mesh and mapped onto a triangle surface (here Mickey's body) for inspection. The mapped potentials defined on Mickey's body surface are mapped to its nodes in order to be visualized in ViewScene.

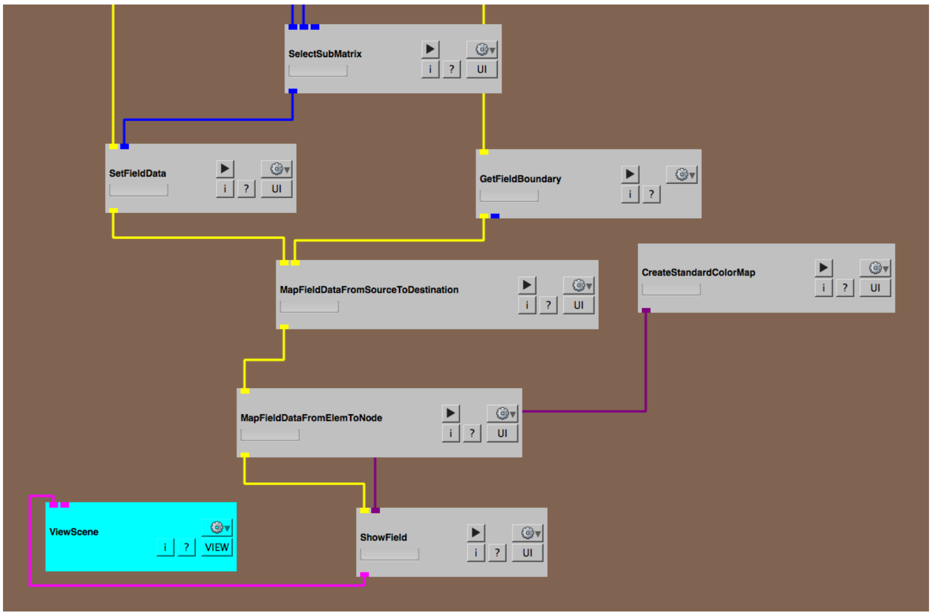


Figure 3.8. tDCS SCIRun5 network (5/5)

3.3 Simulating Transcranial Magnetic Stimulation (TMS)

The SCIRun5 network discussed in the following can be found at `src/ExampleNets/BrainStimulator/TMS_` As previously described, the tetrahedral mesh is created in a similar fashion from the segmentation. In figure 3.9, the TMS coil location is defined and the tms coil prototype is loaded. The magnetically induced primary current is computed and a finite element mesh is set up. The primary current is fed into the BuildFEVolRHS to set up the volumetric primary currents (right hand side) to solve for the secondary currents.

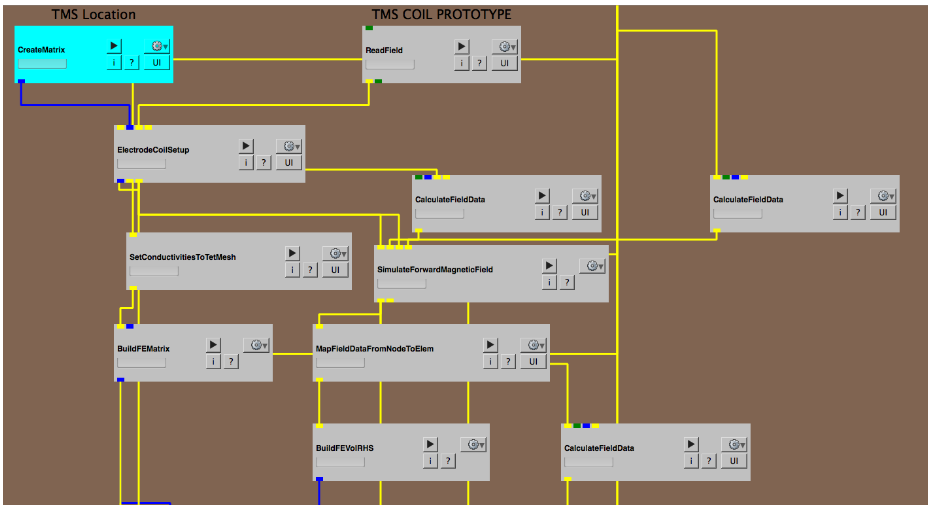


Figure 3.9. TMS SCIRun5 network (1/4)

In figure 3.10, the secondary currents are solved and electrical gradients are calculated.

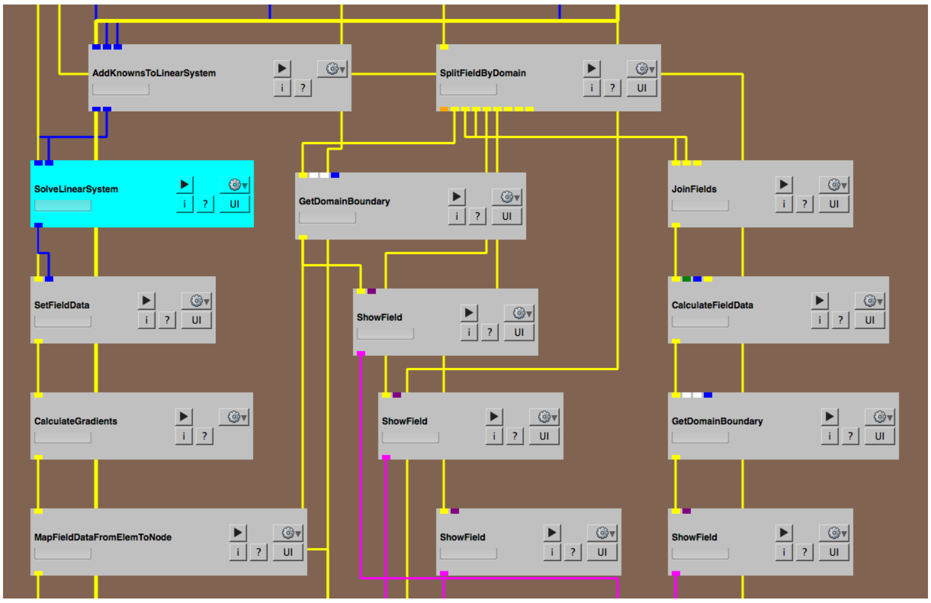


Figure 3.10. RMS SCIRun5 network (2/4)

In figure 3.11, primary and secondary currents are joined and mapped to the mesh.

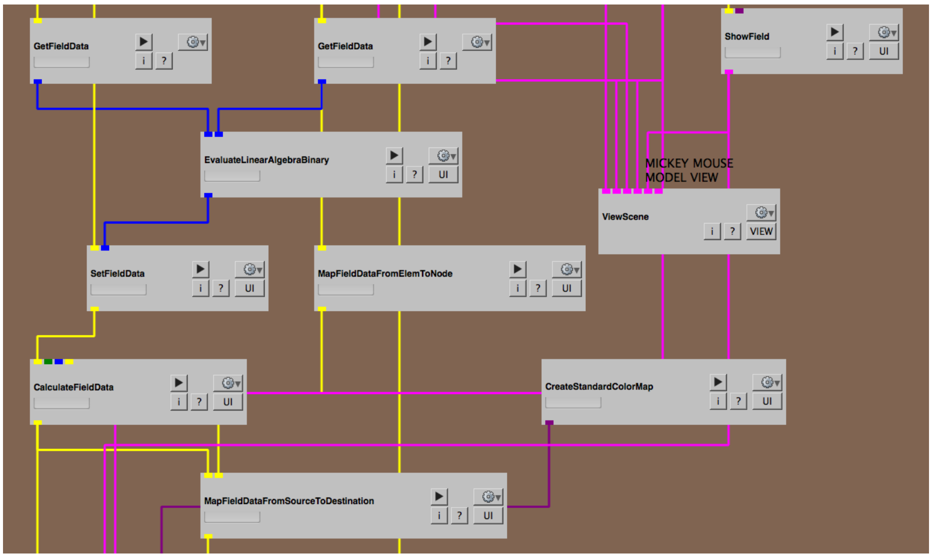


Figure 3.11. RMS SCIRun5 network (3/4)

In figure 3.12, the currents are mapped to Mickey's body and ROI based analysis can be performed.

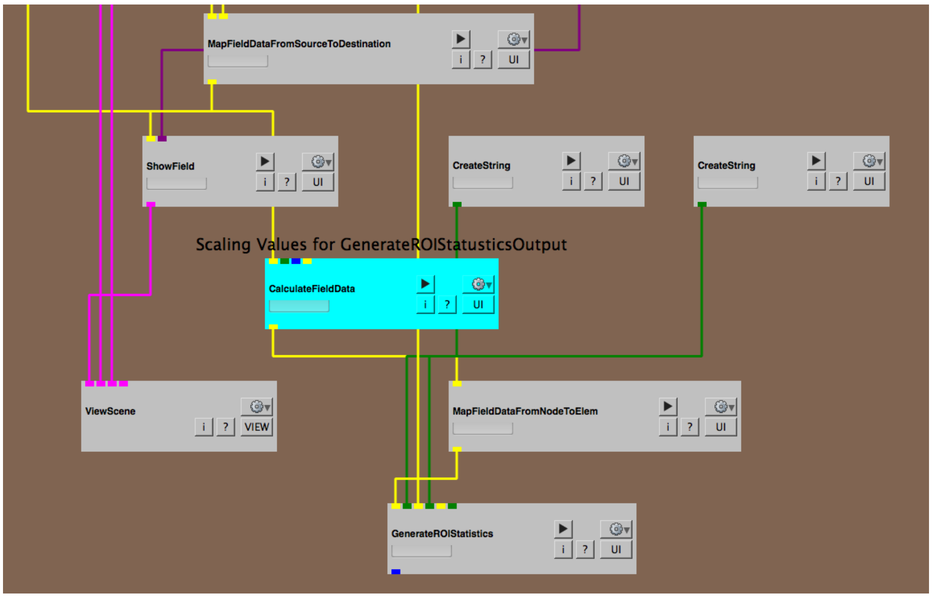

MapFieldDataFromSourceToDestination
ShowField
CreateString
CreateString
Scaling Values for GenerateROIStatusticsOutput
CalculateFieldData
ViewScene
VIEW
MapFieldDataFromNodeToElem
GenerateROIStatistics
i
?
UI

Figure 3.12. RMS SCIRun5 network (4/4)

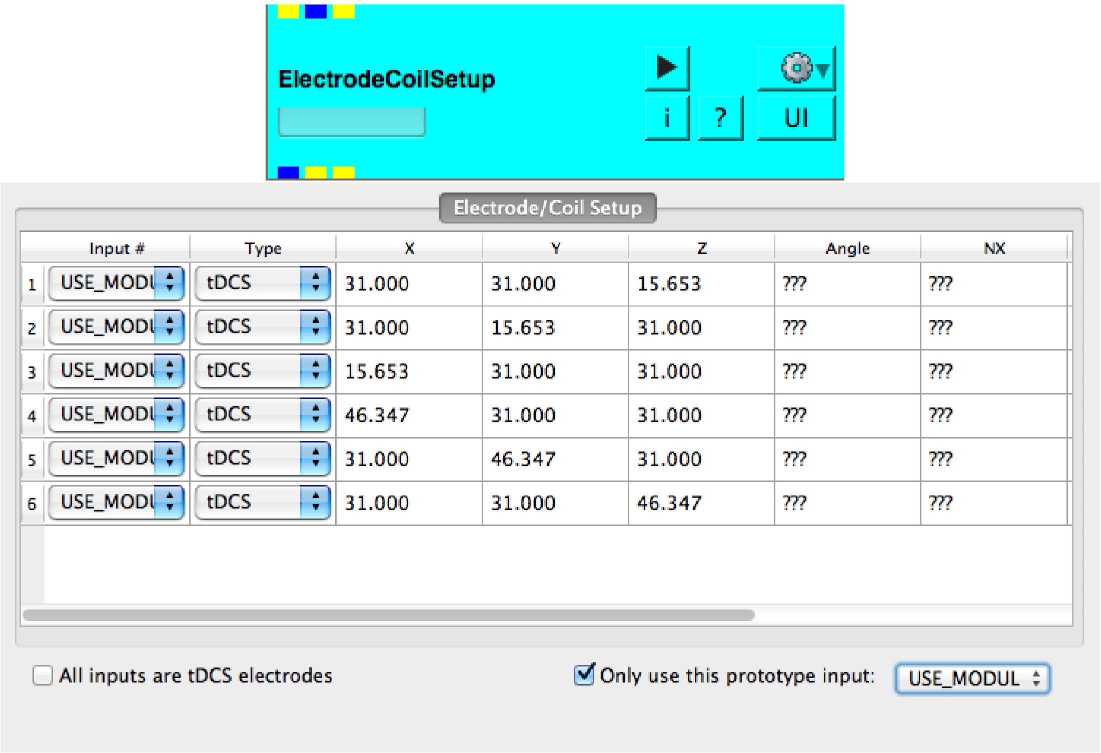
Chapter 4

Appendix

4.1 Modules

Modules are the workhorses of SCIRun. Below is given a brief description of the modules used in this tutorial. To obtain a complete understanding for each module visit SCIRun documentation.

4.1.1 ElectrodeCoilSetup



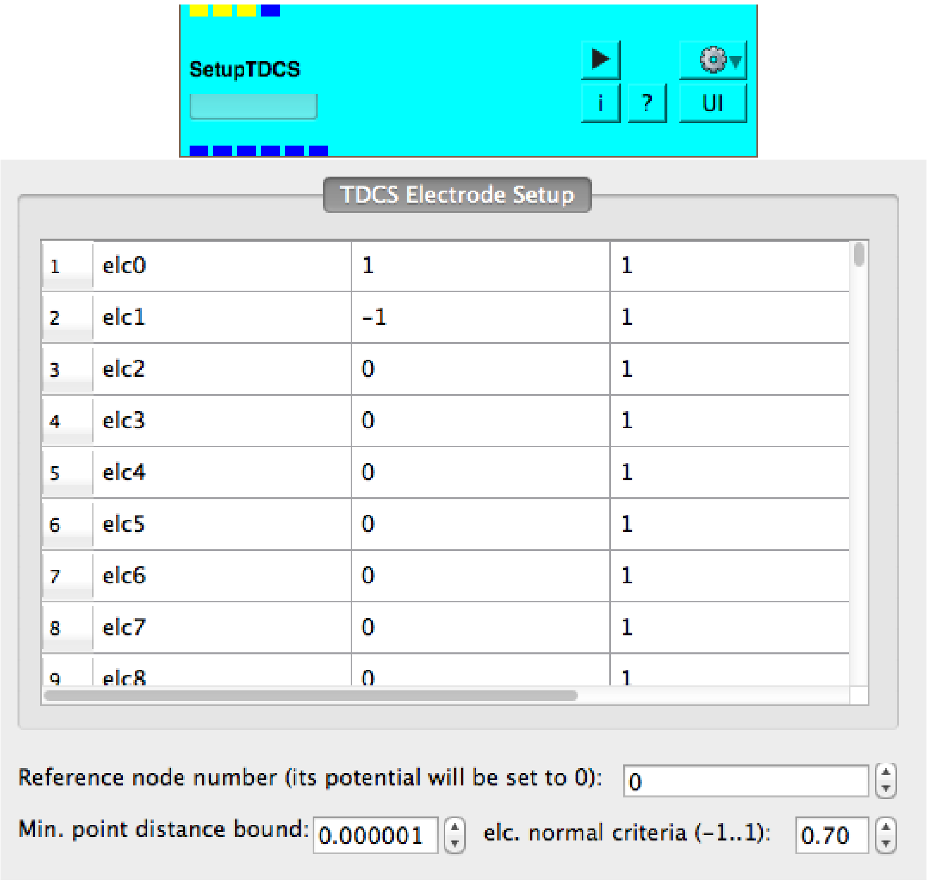
	Input #	Type	X	Y	Z	Angle	NX
1	USE_MODU	tDCS	31.000	31.000	15.653	???	???
2	USE_MODU	tDCS	31.000	15.653	31.000	???	???
3	USE_MODU	tDCS	15.653	31.000	31.000	???	???
4	USE_MODU	tDCS	46.347	31.000	31.000	???	???
5	USE_MODU	tDCS	31.000	46.347	31.000	???	???
6	USE_MODU	tDCS	31.000	31.000	46.347	???	???

☐ All inputs are tDCS electrodes

☑ Only use this prototype input: USE_MODUL

Figure 4.1. Electrode / coil setup module allows for electrode or magnetic coils properties to be manually inputed, such as the height, width, and length.

4.1.2 SetupTDCS



TDCS Electrode Setup

1	elc0	1	1
2	elc1	-1	1
3	elc2	0	1
4	elc3	0	1
5	elc4	0	1
6	elc5	0	1
7	elc6	0	1
8	elc7	0	1
9	elc8	0	1

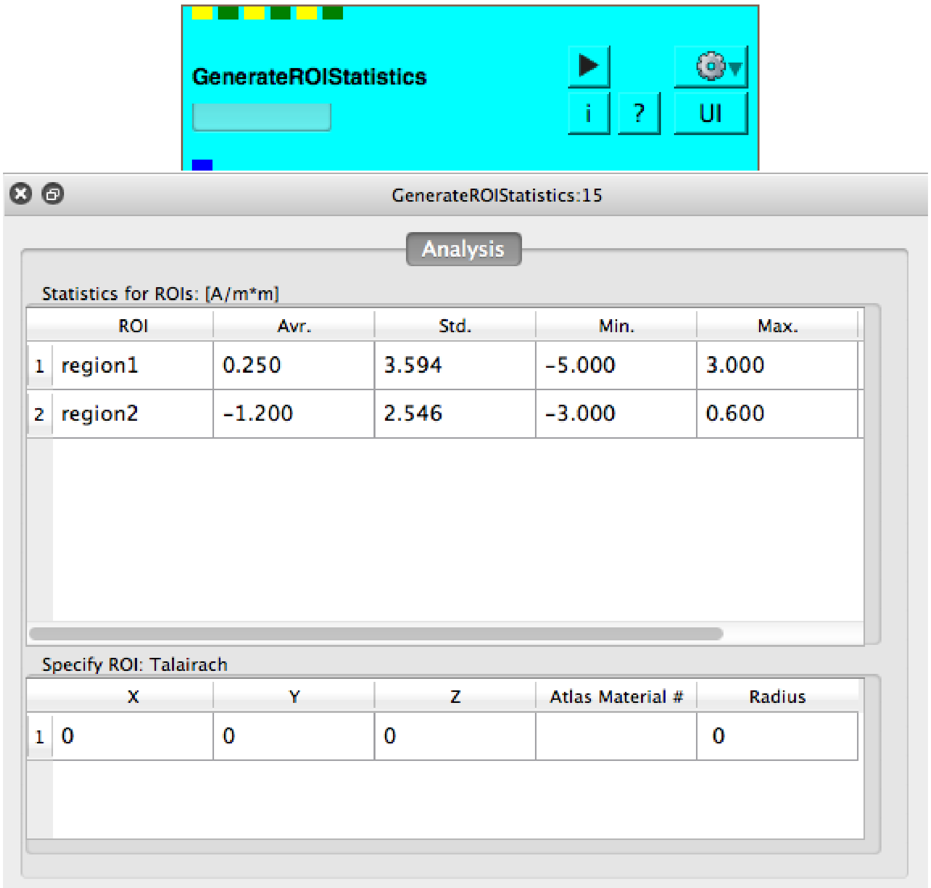
Reference node number (its potential will be set to 0): 0

Min. point distance bound: 0.000001 elc. normal criteria (-1..1): 0.70

Figure 4.2. SetupRHSforTDCSandTMS module allows the desired current intensity and impedance for individual electrodes to be set manually. The reference electrode is also able to be specified by the row number in the table.

4.1.3 GenerateROIStatistics

4.1.4 SetMeshConductivities



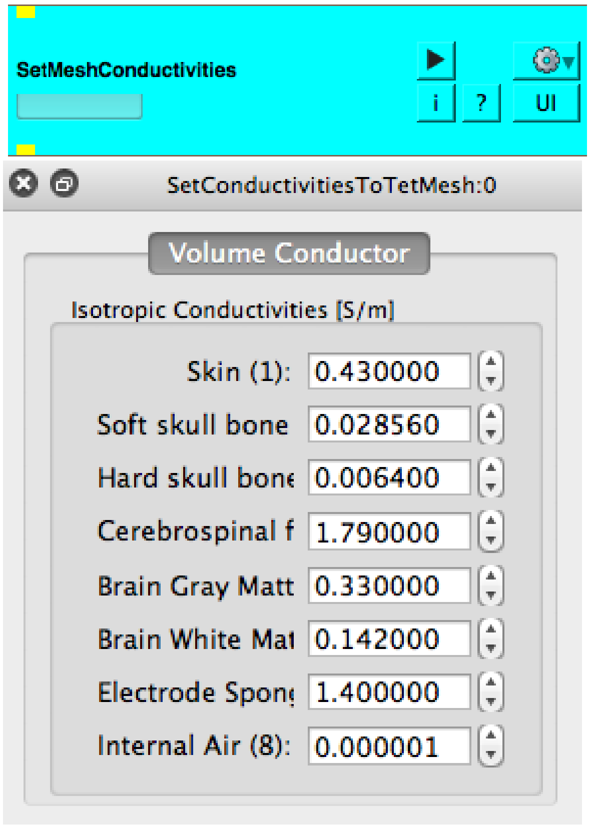
Statistics for ROIs: [A/m*m]

	ROI	Avr.	Std.	Min.	Max.
1	region1	0.250	3.594	-5.000	3.000
2	region2	-1.200	2.546	-3.000	0.600

Specify ROI: Talairach

	X	Y	Z	Atlas Material #	Radius
1	0	0	0		0

Figure 4.3. GenerateROIStatistics module calculates the average, standard deviation, maximum, and minimum of the current density at desired ROIs. ROIs are specified by x, y, z marker and a radius which is used to create a sphere from the point specified.



SetConductivitiesToTetMesh:0

Volume Conductor

Isotropic Conductivities [S/m]

Skin (1): 0.430000

Soft skull bone 0.028560

Hard skull bone 0.006400

Cerebrospinal f 1.790000

Brain Gray Matt 0.330000

Brain White Mat 0.142000

Electrode Spong 1.400000

Internal Air (8): 0.000001

Figure 4.4. This module allows you to set conductivity values for different fields contained within the mesh. The fields that are available for changing are: skin, soft skull bone, hard skull bone, cerebral spinal fluid, brain gray matter, brain white matter, electrode sponge, and internal air.