# 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 SCIRun 5. 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 dataset containing multiple spheres forming a Mickey Mouse head. Besides the Mickey Mouse Toy Model, BrainStimulator contains a realistic human head model that was derived from the well-known Colin27 brain atlas in various levels of detail.**

## 1.1 Transcranial Direct Current Stimulation (tDCS)

Transcranial direct current stimulation (tDCS) is a non-invasive technique that affects 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 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 $\phi$ denotes linear basis functions.

The boundary problem formulation (complete electrode model) above can be efficiently solved with SCIRun 5 modules (see more below). That complete electrode model describes the 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 be sufficiently approximated by a number of magnetic dipoles. The 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 the 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:

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

with $\nabla\phi$ being the gradient of the electrical potential ($\phi$) and $\frac{dA}{dt}$ being the time derivative of the magnetic vector potential generated by the coil, with 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 time interval considered) 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 computer simulations have shown that its impact is rather 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. Please download the latest SCIRun5 release. Please follow up on the SCIRun5 requirements posted on our website SCIRun website. All models provided with BrainStimulator have different resolutions and therefore different memory and computing requirements. The Mickey Mouse Toy Model was designed to run on computers with limited computional capabilities (4 GB of RAM, 2 GHz Processors). Computationally more demanding models (e.g., Colin27 model) should be processed on multicore processor maschines having at least 16 GB of memory (RAM).

## Required Datasets

In order to perform non-invasive brain stimultion simulations with SCIRun5/BrainStimulator please download also the zipped data file containing data for BrainStimulator.

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 was implemented (Thielscher et al., 2004, Clinical Neurophysiology and later publications, www.simnibs.de).

## 3.1 Preparation of Finite Element Model

The finite element mesh (file **mesh.mat**) was generated using the meshing package (Cleaver) based on the label mask file (file **mickey4tDCS.nrrd**). The file **mickey4tDCS.nrrd** consists of the labels background; Mickey's body; right ear; left ear and two target regions for stimulation 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 SCIRun 5.

### 3.1.1 Tissue segmentation: Mickey Mouse Toy Model

The simulation setup of the tDCS Mickey Mouse example identifies 6 sites to place electrodes, which are parallel to the x-, y- and z-axis. There are two target regions usable for tDCS and TMS, one located superficially and one deeply inside Mickey's head (figure 3.1).

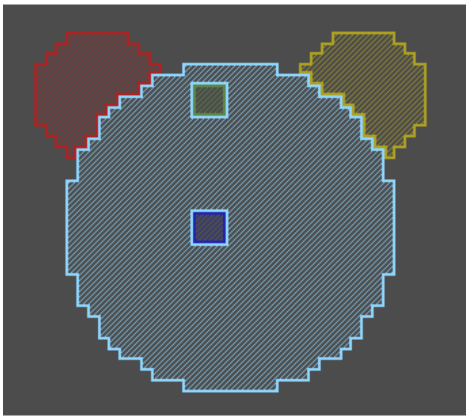


Figure 3.1. Toy example: Mickey Mouse segmentation (depicted tissue labels are red/light brown=left/right ear, light blue=body, dark blue/green=deep/superficial ROI)

### 3.1.2 Generating Finite Element Mesh (FEM) in SCIRun 5 from segmentations

The SCIRun 5 network shown in figure 3.2 implements a way to generate a tetrahedral mesh based on a segmentation (***.nrrd** file) that needs 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 through 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. These 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. The grid (module **CreateLatVol**) is created in the top part of this network and aligned with the segmentation (module **AlignMeshBoundingBoxes**) as well.

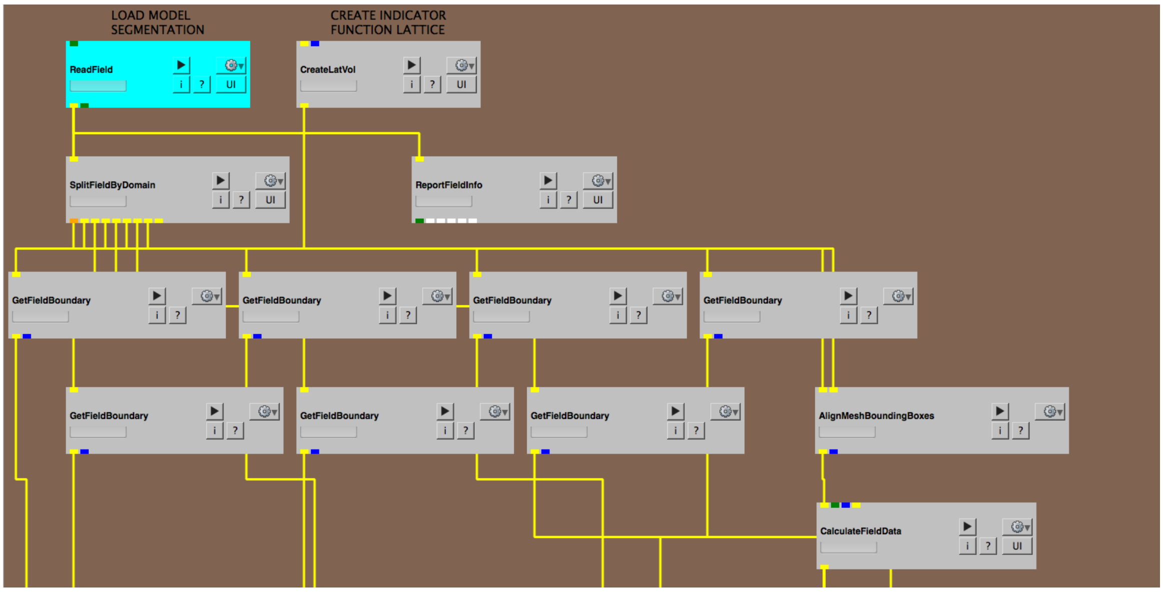


Figure 3.2. Mickey Mouse segmentation: Tetrahedral mesh generation, SCIRun 5 network (1/2)

The second part of the network (figure 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) is piped into a **CalculateFieldData** module to be converted to float numbers (from double). All signed distance fields are piped into Cleaver. Cleaver expects to have positive signed distance values inside and negative values outside. 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 SCIRun 5 network discussed in the following sections can be found at `src/ExampleNets/BrainStimulator/TDCS_TOY_EXAMPLE.srn5`.

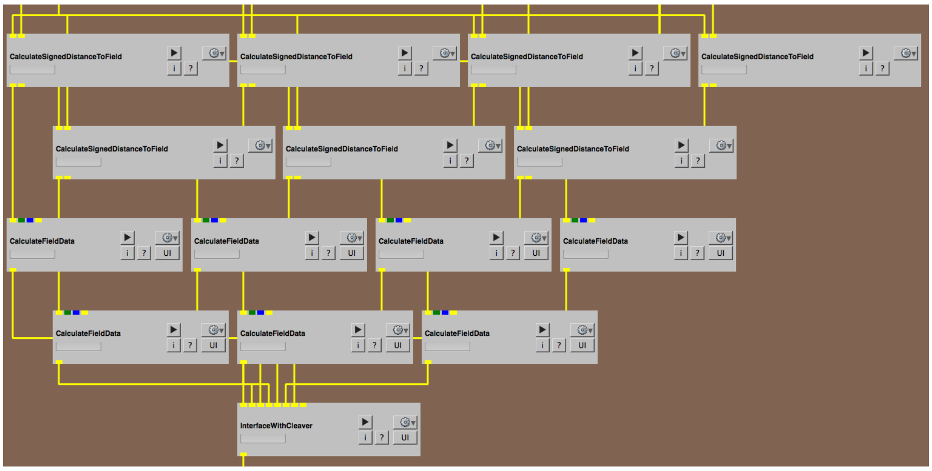

CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData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

Figure 3.3. Mickey Mouse segmentation: Tetrahedral mesh generation, SCIRun 5 network (2/2)

### 3.2.1 SCIRun 5 tDCS network

The only difference between figure 3.4 and figure 3.2 is the presence of two more modules in the top right corner. One module creates a matrix of electrode locations (x, y, z) and the other reads the shape of a tDCS electrode prototype.

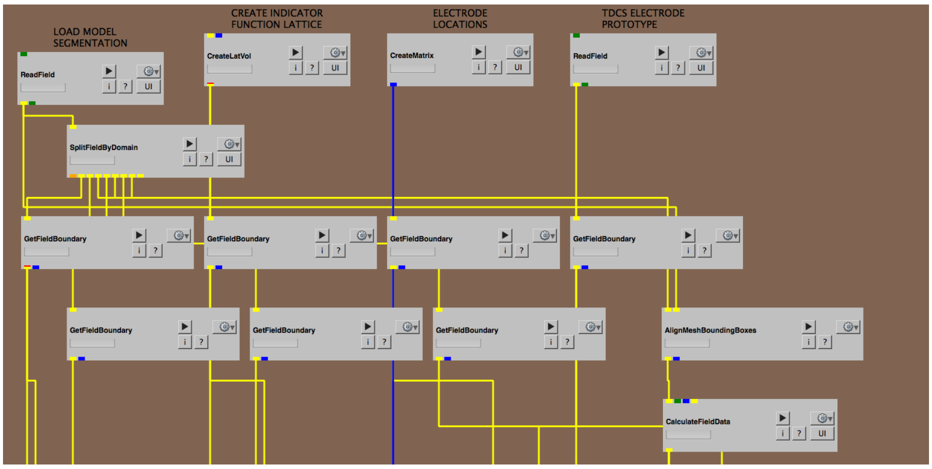


Figure 3.4. tDCS SCIRun 5 network (1/5)

In figure 3.5, the module **ElectrodeCoilSetup** highlights the position of 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 (Cleaver input).

Figure 3.6 demonstrates how 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.

Figure 3.7 presents the computational core of the network. The resulting tetrahedral

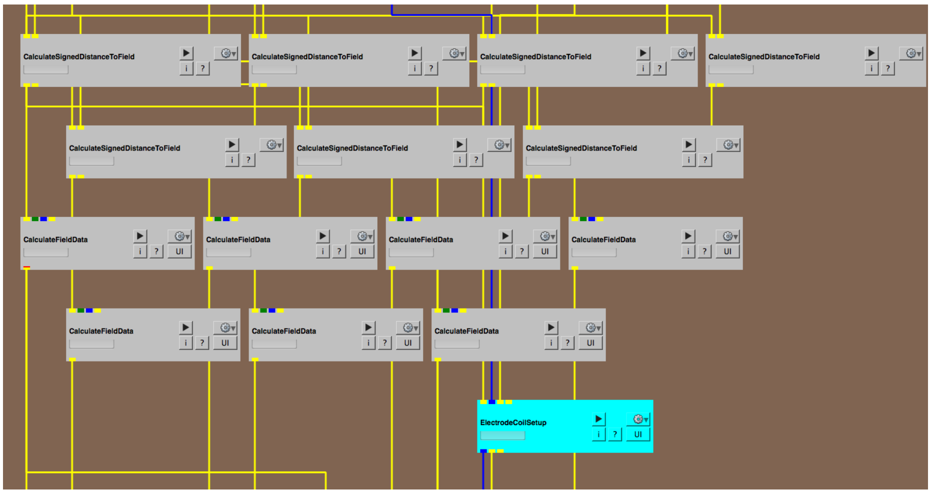

CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateSignedDistanceToField
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData
CalculateFieldData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

Figure 3.5. tDCS SCIRun 5 network (2/5)

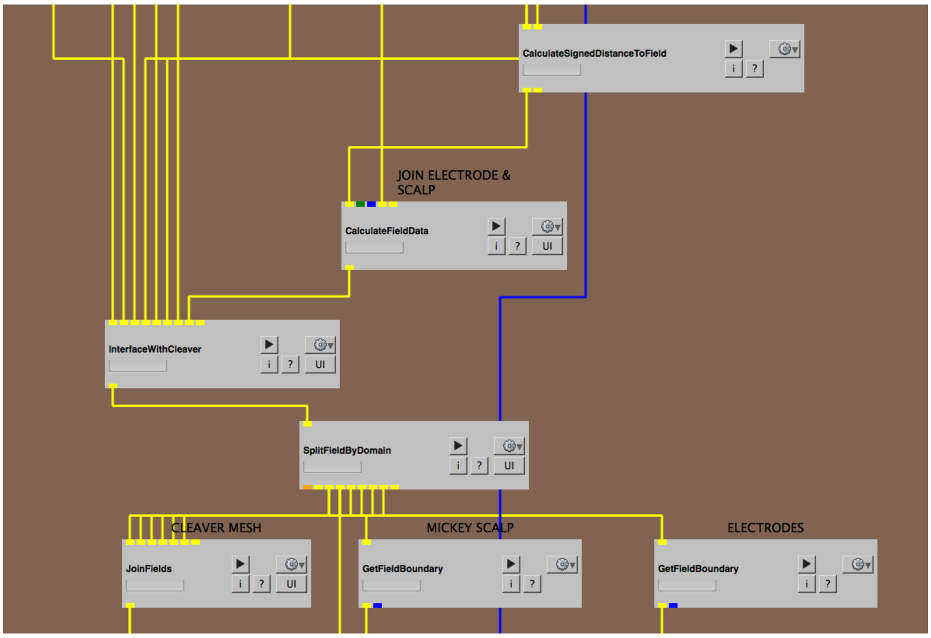


Figure 3.6. tDCS SCIRun 5 network (3/5)

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). These 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 node potential vector $U_n$, which is only needed for further computations (figure 3.8). Then, the node potentials are assigned to the pre-

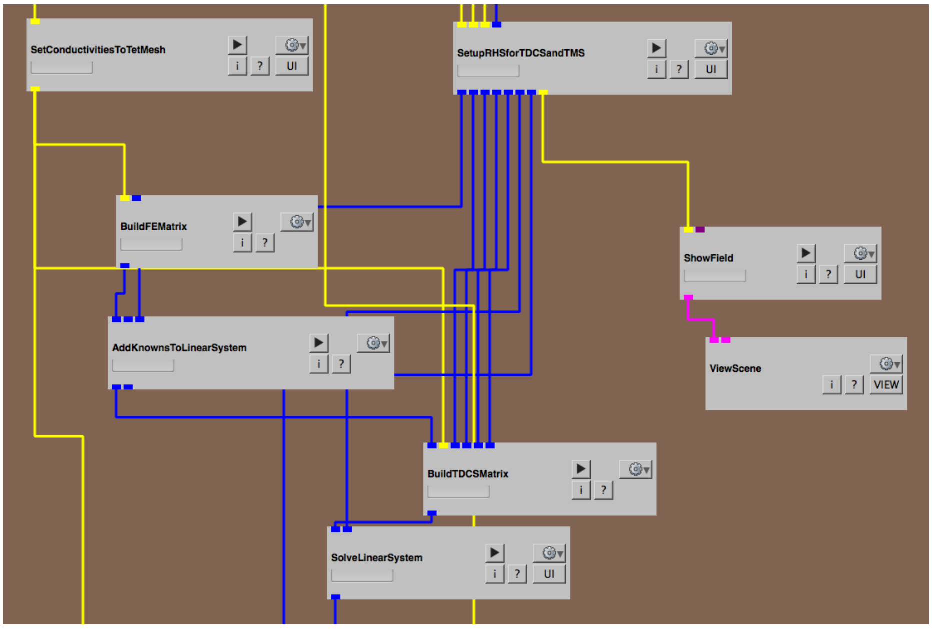


Figure 3.7. tDCS SCIRun 5 network (4/5)

viously generated tetrahedral mesh and mapped onto a triangle surface (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**.

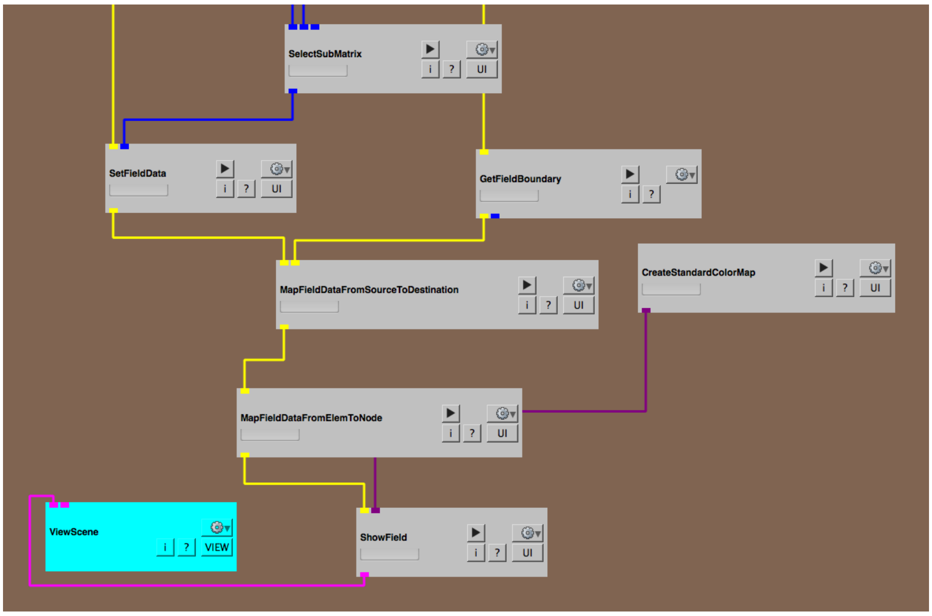


Figure 3.8. tDCS SCIRun 5 network (5/5)

## 3.3 Simulating Transcranial Magnetic Stimulation (TMS)

The SCIRun 5 network discussed in the following section can be found at `src/ExampleNets/BrainStimulator/TMS_TOY_EXAMPLE.srn5`. As previously described, the tetrahedral mesh is created in a similar fashion. 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.

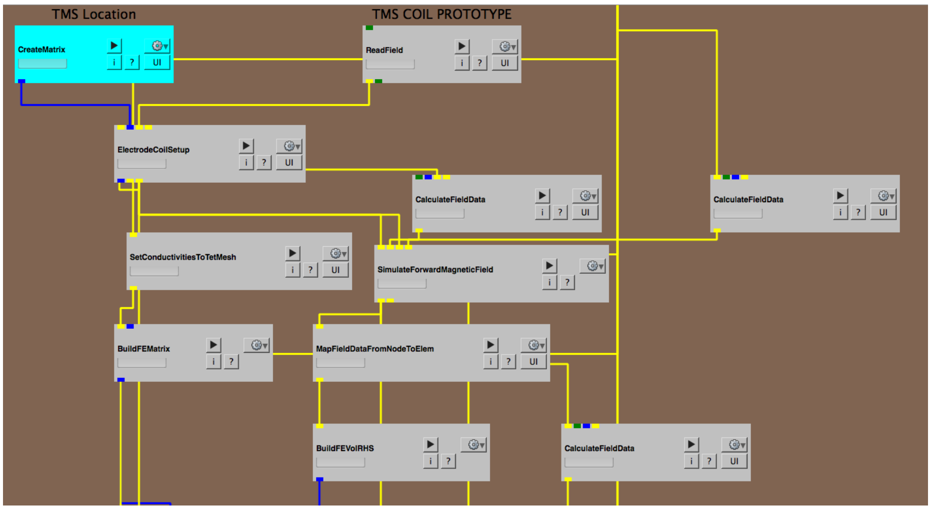


Figure 3.9. TMS SCIRun 5 network (1/4)

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

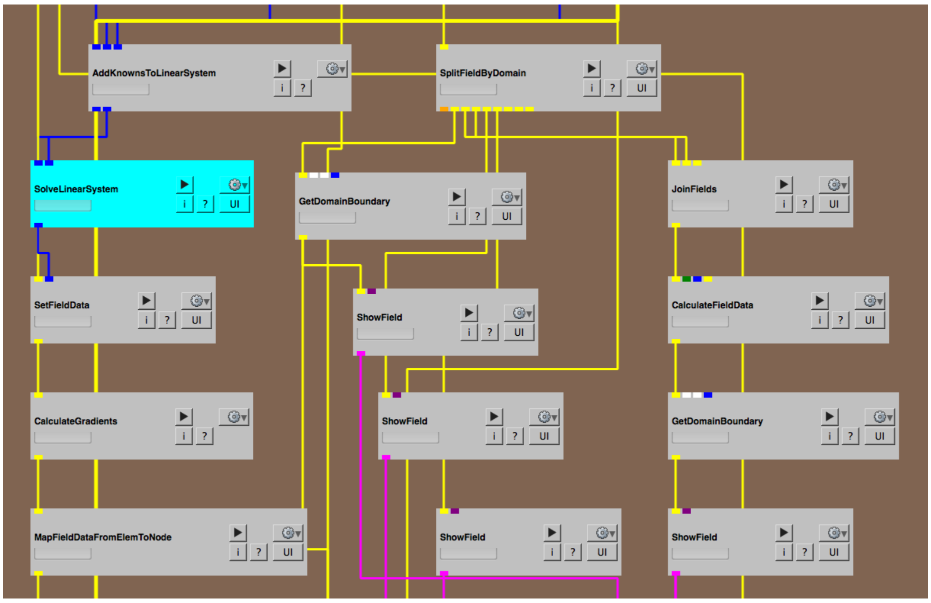


Figure 3.10. TMS SCIRun 5 network (2/4)

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

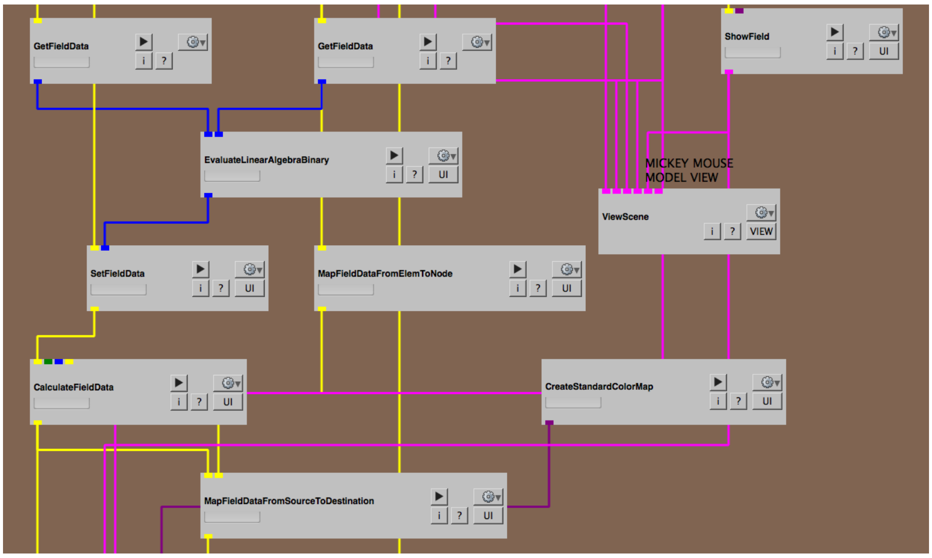


Figure 3.11. TMS SCIRun 5 network (3/4)

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

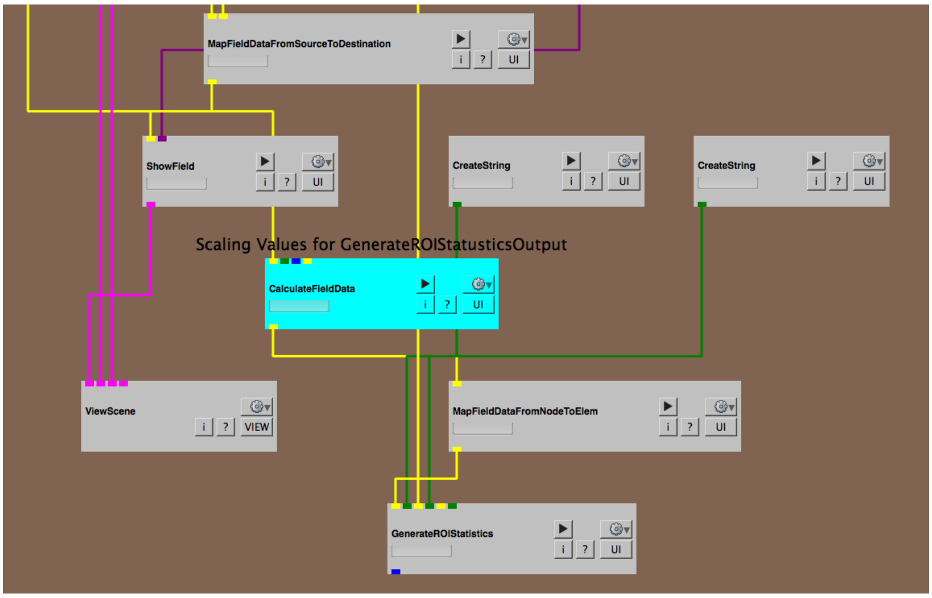

MapFieldDataFromSourceToDestination
ShowField
CreateString
CreateString
Scaling Values for GenerateROIStatusticsOutput
CalculateFieldData
ViewScene
VIEW
MapFieldDataFromNodeToElem
GenerateROIStatistics
UI

Figure 3.12. TMS SCIRun 5 network (4/4)

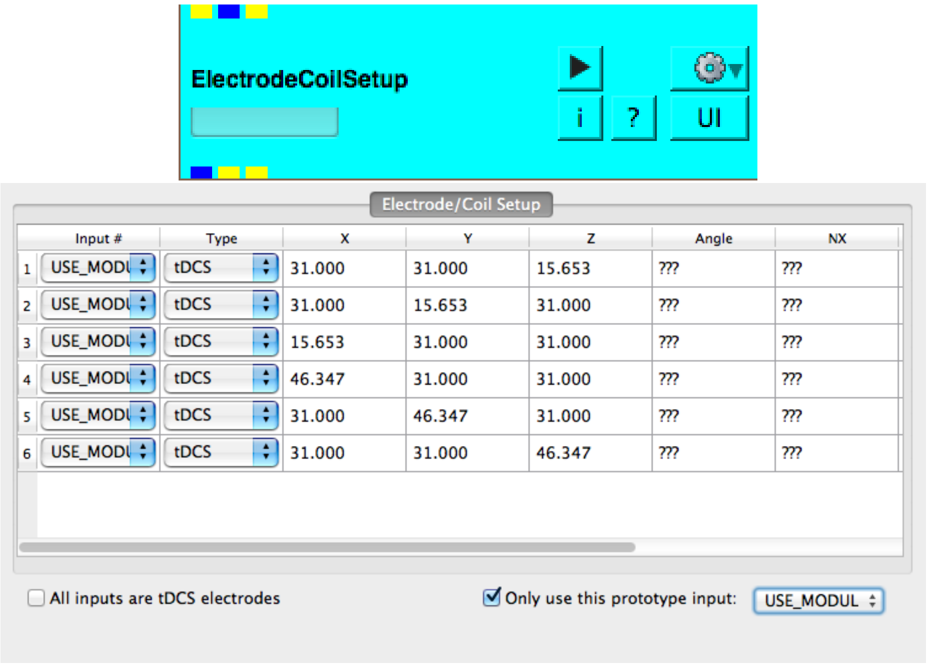
Chapter 4

# Appendix

## 4.1 Modules

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

### 4.1.1 ElectrodeCoilSetup



Electrode/Coil Setup

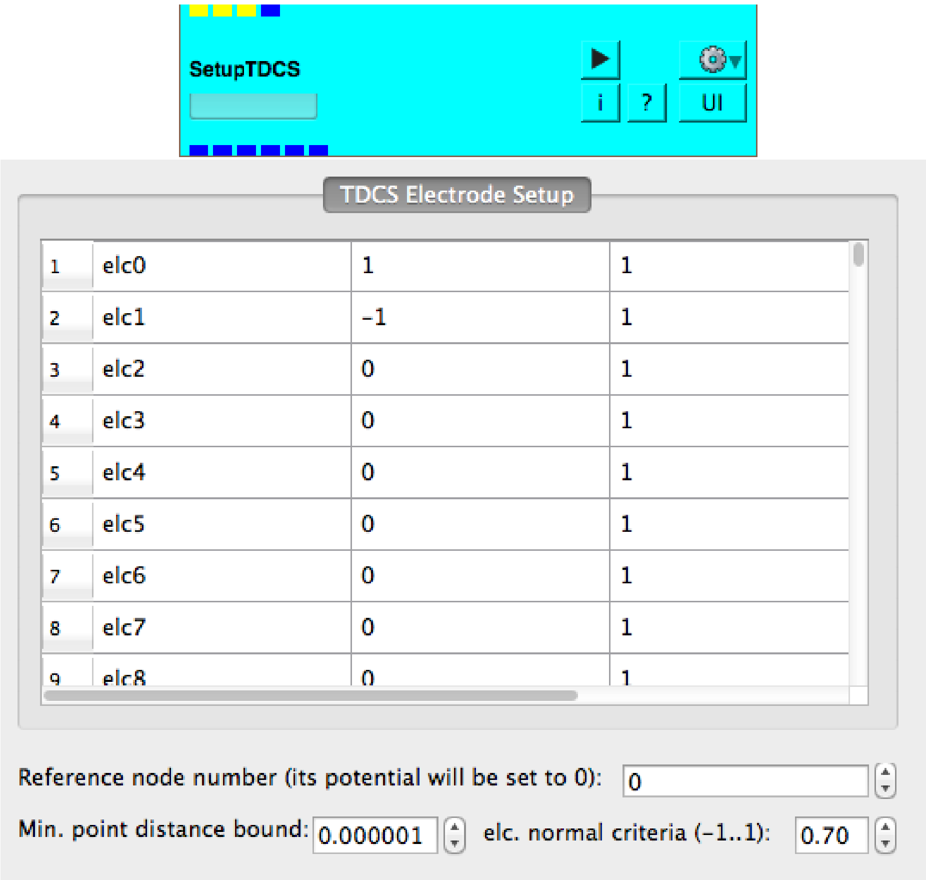
| | Input # | Type | X | Y | Z | Angle | NX |
|---|---|---|---|---|---|---|---|
| 1 | USE_MODU | tDCS | 31.000 | 31.000 | 15.653 | ??? | ??? |
| 2 | USE_MODU | tDCS | 31.000 | 15.653 | 31.000 | ??? | ??? |
| 3 | USE_MODU | tDCS | 15.653 | 31.000 | 31.000 | ??? | ??? |
| 4 | USE_MODU | tDCS | 46.347 | 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.

**ElectrodeCoilSetup** is designed to position tDCS electrodes and TMS coils prototypes to their final positions. A tDCS electrode protoype is a triangular mesh of any shape that is symetrically positioned at the origin of the coordinate system. A tDCS electrode prototype can be created e.g., using CreateLatVol. A TMS coil prototype consists of a point cloud of magnetic dipole positions. For each magnetic dipole position a vector is stored defining its contribution to the magnetic output of the coil. The module translates and rotates all defined TMS coils to their final positions and rotates them as provided by User input (GUI). In a similar way, tDCS electrode sponges can be moved to their final destinations. Furthermore, at their final destinations tDCS electrode can be mapped onto the scalp that was provided. This modules has to be excuted once to make the inputs available in the GUI. After input of GUI information a second output will generate useful output. Hint: In the case that an tDCS electrode sponge has been attached to the scalp from the head inside the user can invert the normal definition (multiplying Nx,Ny,Nz by -1) and generate a tDCS electrode sponge on the scalp surface after reexecution of the module.

### 4.1.2 SetupTDCS



| | | | |
|---|---|---|---|
| 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 |

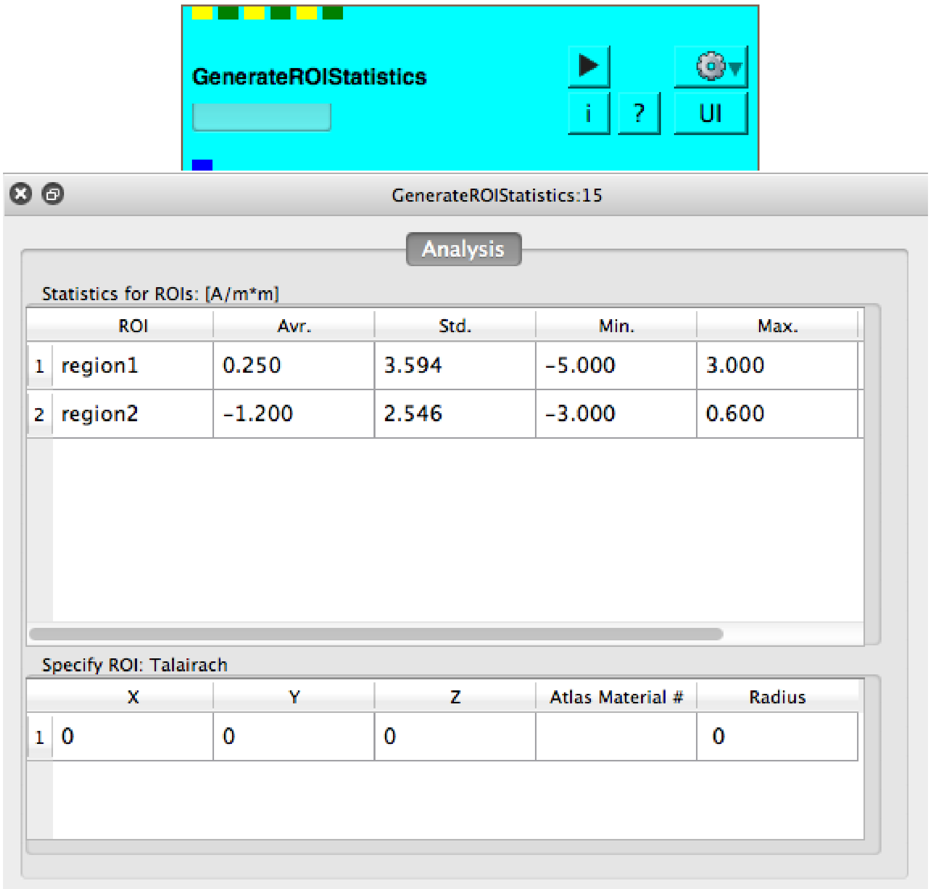
Reference node number (its potential will be set to 0): 0

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.

**SetupTDCS** gets information (module input) on which triangle elements are part of the tDCS electrode sponge. It uses these information to determine the top part of the electrode sponges. Once the electrode sponge tops are determined it creates outputs to be fed into the module **BuildTDCSMatrix** that creates the tDCS forward Matrix $M$. In the module GUI, the user need to specify a ground potential on a node somewhere in the tetrahedral input mesh (first module input). That mesh node should be far away from the current injection sites and the ROI (preferentially its on the bottom of the head model). For each defined electrode (sponge) the user can define a current intensity (in milli Amphere $[mA]$) and a real impedance ($[Ohm * m^2]$) in the GUI. After the first execution of the module the GUI is adjusted based on the outputs. There GUI options to control the size of the electrode sponge top (elc. normal criteria) and minimal distance to find points in the input mesh (first input).

### 4.1.3 GenerateROIStatistics



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 |

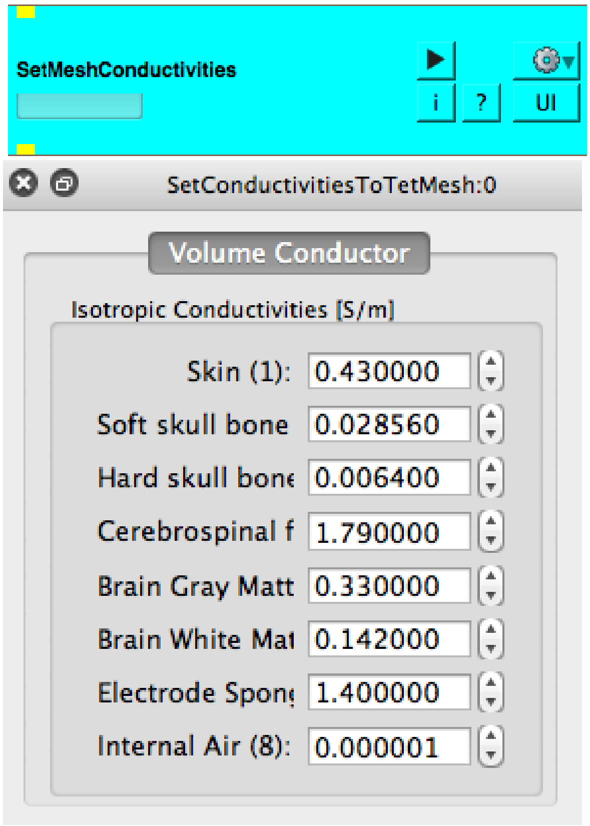
Specify ROI: Talairach

| | 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.

**GenerateROIStatistics** creates descriptive statistics (average, minimum, maximum, standard deviation) from brain stimulation simulations (here tDCS and TMS). It is meant to work in conjunction with predefined atlases where each mesh element is assigned to one material or anatomical regions. After distinct anatomical regions are determined (from the atlas) statistics values (see above) are computed (from the actual simulation input) and presented in the GUI table. The values of the GUI table are piped to the only output of the module as a dense matrix.

### 4.1.4 SetMeshConductivities



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.

**SetMeshConductivities** provides simple GUI options to assign electrical isotropic conductivities to the tetrahedral meshes (labels: 1,2,..8) for typical head tissues known to have very distinct electrical properties.