

# Chapter 1

## SimNIBS 2.1: A Comprehensive Pipeline for Individualized Electric Field Modelling for Transcranial Brain Stimulation



Guilherme B. Saturnino, Oula Puonti, Jesper D. Nielsen, Daria Antonenko, Kristoffer H. Madsen, and Axel Thielscher

### 1.1 Introduction

Non-invasive brain stimulation (NIBS) aims at modulating brain activity by inducing electric fields in the brain [1]. The electric fields are generated either by a magnetic coil, in the case of transcranial magnetic stimulation (TMS), or by a current source and electrodes placed directly on the scalp, in the case of transcranial electric stimulation (TES). In both cases, the induced electric fields in the brain have a complex and often counter-intuitive spatial distribution, which is dependent on the individual anatomy of a target subject. In recent years, there has been a growing interest in moving away from a one-size-fits-all stimulation approach in NIBS to more individually informed protocols [2]. The driving force behind this shift is the

---

Guilherme B. Saturnino and Oula Puonti contributed equally to this chapter.

---

G. B. Saturnino · A. Thielscher (✉)

Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

Department of Health Technology, Technical University of Denmark,

Kongens Lyngby, Denmark

e-mail: [axelt@drcmr.dk](mailto:axelt@drcmr.dk)

O. Puonti

Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

J. D. Nielsen · K. H. Madsen

Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark

D. Antonenko

Department of Neurology, Universitätsmedizin Greifswald, Greifswald, Germany

widely reported variation of NIBS effects within and between individuals [3], which could be explained in part by the interplay of the individual anatomy and the electric field propagation [4]. Although software tools have become available that generate realistic anatomical models of the head based on magnetic resonance imaging (MRI) scans and use those models to numerically estimate the electric field induced in the brain, they are still not predominantly used in NIBS studies. This is likely due to the lack of robustness and usability of the previous generation of tools, in turn hampering the individualized application of NIBS in both mapping the human brain function and as a rehabilitation tool in various neuropathologies [5, 6].

The aim of SimNIBS is to facilitate the use of individualized stimulation modeling by providing easy-to-use software tools for creating head models, setting up electric field simulations, and visualizing and post-processing the results both at individual and group levels. SimNIBS was first released in 2013 [7], had a major update in 2015, with the release of version 2 [2], and more recently another major update with the release of version 2.1, described in the current work. SimNIBS 2.1 is a free software, distributed under a GPL 3 license, and runs on all major operating systems (Windows, Linux and MacOS). In this tutorial, we will concentrate on **what** SimNIBS 2.1 can be used for and **how** the analyses are performed in practice with step-by-step examples. The chapter is structured as follows: First, we give a general overview of the simulation pipeline and of its building blocks. Next, we provide a step-by-step example of how to run a simulation in a single subject, and then we demonstrate a set of MATLAB tools developed for easy processing of multiple subjects. Finally, we conclude with an analysis of the accuracy of automated electrode positioning approaches. More information, as well as detailed tutorials and documentation can be found from the website [www.simnibs.org](http://www.simnibs.org).

## 1.2 Overview of the SimNIBS Workflow

Figure 1.1 shows an overview of the SimNIBS workflow for an individualized electric field simulation. The workflow starts with the subject's anatomical MRI images, and optionally diffusion-weighted MRI images. These images are segmented into major head tissues (white and grey matter, cerebrospinal fluid, skull and scalp). From the segmentations, a volume conductor model is created, and used for performing the electric field simulations. The simulations can be set up in a graphical user interface (GUI) or by scripting. Finally, the results can be mapped into standard spaces, such as the Montreal Neurological Institute (MNI) space or FreeSurfer's FsAverage.

### 1.2.1 Structural Magnetic Resonance Imaging Scans

The minimum requirement for running an individualized SimNIBS simulation is a T1-weighted structural scan of a subject's head anatomy. Although SimNIBS will run on almost all types of T1-weighted scans, we have found that setting the readout

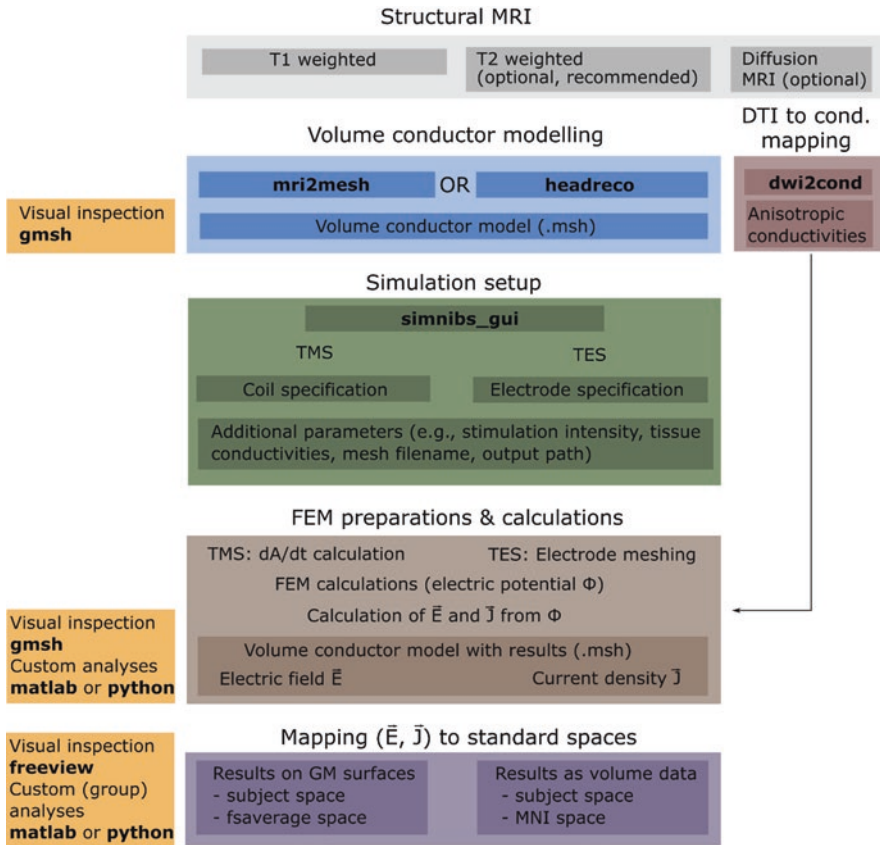
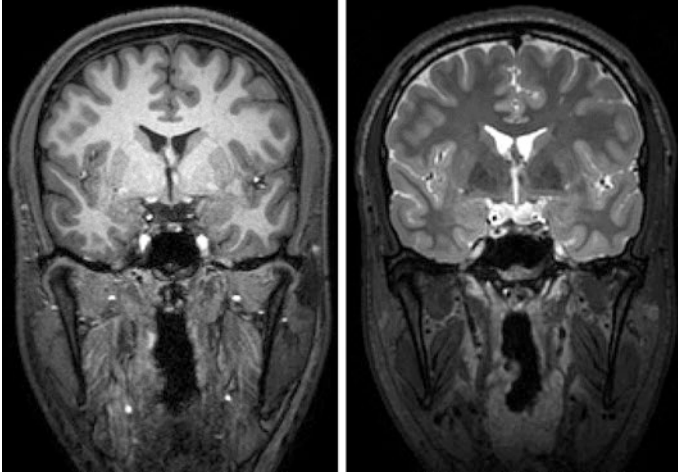


Fig. 1.1 Overview of the SimNIBS workflow

bandwidth low to ensure a good signal-to-noise ratio in the brain region and using a fat suppression method, such as selective water excitation, to minimize the signal from spongy bone, typically ensure a high quality of the resulting head models. See Fig. 1.2 for an example of good quality scans we found to work well with SimNIBS and [8] for the details of the sequences.

Including a T2-weighted scan is optional, but highly recommended as it facilitates accurate segmentation of the border between skull and cerebrospinal fluid (CSF). Both skull and CSF appear dark in T1-weighted scans, whereas in T2-weighted scans the CSF lights up, thus guiding the separation between the tissues. Skull has a low electric conductivity, while CSF is highly conducting, meaning that any segmentation errors in these two compartments can have a large effect on the resulting electric field distribution inside the head, especially when TES is applied [8]. If you are interested in modelling the neck region in detail, we recommend using neck coils if these are available at the imaging site.



**Fig. 1.2** Example of high-quality T1- and T2-weighted scans likely to work well with SimNIBS. Note that in the T1-weighted scan, the skull appears dark due to the fat suppression

Optionally, SimNIBS also supports modelling of anisotropic conductivities for grey (GM) and white matter (WM), which requires a diffusion-weighted MRI scan (dMRI). Only single shell data (i.e. with a single b-value in addition to some  $b = 0$  images) with a single phase encoding direction for the echo planar imaging (EPI) readout is supported.

### 1.2.2 Volume Conductor Modelling

The first step in the pipeline is the generation of a volume conductor model of the head, which is needed for simulating the induced electric fields. In order to create this finite element (FEM) mesh, we need to assign each voxel in the MRI scan(s) to a specific tissue class, i.e. to segment the scan into the different head tissues. Currently, SimNIBS offers two options for segmentation: **mri2mesh** [7] and **headreco** [8].

**mri2mesh** combines FSL [9] (version 5.0.5 or newer) and FreeSurfer [10] (version 5.3.0 or newer) to segment the head tissues. FSL is used to segment the extra-cerebral tissues, while FreeSurfer is used to segment the brain and to generate accurate surface reconstructions of the grey matter sheet. Note that **mri2mesh** is restricted only to the head and does not create models of the neck region.

**headreco** uses the SPM12 [11] toolbox for segmenting the MRI scan, and is now the recommended option in SimNIBS. It has been shown to be more accurate in segmenting the extra-cerebral structures, especially the skull, compared to **mri2mesh** [8], while also providing accurate segmentations of the brain tissues. The computational anatomy toolbox (CAT12, recommended) [12] provided with

SPM can be used to create surface reconstructions of the grey matter sheet which are on par with the accuracy of those generated by FreeSurfer [12]. In addition, **headreco** has an extended field of view, also modelling the neck region. For ease of use, both SPM12 and CAT12 are distributed together with SimNIBS.

Once the segmentation by either method has finished successfully, the tissue maps are cleaned by applying simple morphological operations, and used to create surface reconstructions. As a final step, the FEM mesh is generated by filling in tetrahedrons between the tissue surfaces using Gmsh [13].

Neither **mri2mesh** nor **headreco** have off-the-shelf support for pathologies such as tumours or lesions. These can however be included into the head models by manually editing the segmentation masks generated by the methods. When using **mri2mesh**, please consult the FreeSurfer website ([https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/WhiteMatterEdits\\_freeview](https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/WhiteMatterEdits_freeview)) on how to handle scans with pathologies. Manual edits using **headreco** should be done on the output segmentation masks in the *mask\_prep* folder located within the *m2m\_{subID}* folder. Once corrections have been made, the surface meshing step (“**headreco surfacemesh subID**”) and volume meshing step (“**headreco volumemesh subID**”) should be re-run to generate the edited head model. Note that when creating head models from scans with pathologies, the CAT12 toolbox should *not* be used.

**dwi2cond (optional)** uses FSL (version 5.0.5 or newer) to prepare diffusion tensors for GM and WM from dMRI data. The tensors are used by SimNIBS to estimate anisotropic conductivities in WM and GM during the FEM calculations.

### 1.2.3 Simulation Setup

Simulations can be set up using the graphical user interface (GUI), which provides an interactive view of the head model. This allows users to easily select parameters such as coil positions, electrode positions and shapes, as well as more advanced settings such as tissue conductivities and post-processing options.

It might also be of interest to do simulations of one or a few different setups across a group of subjects. With this in mind, version 2.1.1 introduced a new interface for setting up simulations using MATLAB or Python scripts.

The GUI as well as the scripts will be described in more detail in Sect. 1.3, as well as on the website [www.simnibs.org](http://www.simnibs.org).

### 1.2.4 Finite Element Method Calculations

Transcranial direct current stimulation (tDCS) simulations begin by adding electrodes to the head model. In this step, nodes in the skin surface are shifted to form the shape of the electrode, while keeping good quality elements. Afterwards, the body of the electrodes is constructed by filling in tetrahedra. As this step does

not require re-meshing the entire head, it can be done much more efficiently compared to other methods that require re-meshing, especially when only a few electrodes are used.

TMS simulations start by calculating the change in the magnetic vector potential  $\mathbf{A}$ , that is the  $d\mathbf{A}/dt$  field in the elements of the volume conductor mesh for the appropriate coil model, position and current. There are currently two types of coil models:

- .ccd files:** Created from geometric models of the coil and represented as a set of magnetic dipoles from which we can calculate the  $d\mathbf{A}/dt$  field using a simple formula [14].
- .nii files:** Created either from geometric models of the coils or direct measurement of the magnetic field [15]. Here, the  $d\mathbf{A}/dt$  field is defined over a large volume, and the calculation of the  $d\mathbf{A}/dt$  at the mesh elements is done via interpolation. This allows for faster simulation setup at little to no cost in simulation accuracy.

Both simulation problems are solved using the FEM with linear basis functions. This consists of constructing and solving a linear system of the type  $\mathbf{M}\mathbf{u} = \mathbf{b}$ , where  $\mathbf{M}$  is a large (in SimNIBS typically  $\sim 10^6 \times 10^6$ ) but sparse matrix, called the “stiffness matrix”,  $\mathbf{u}$  are the electric potentials at the nodes and the right-hand side  $\mathbf{b}$  contains information about boundary conditions (such as potentials in electrode surfaces in tDCS simulations), and source terms (such as the  $d\mathbf{A}/dt$  field in TMS simulations). SimNIBS solves the linear system using an iterative preconditioned conjugate gradient method [16]. SimNIBS 2.1 uses GetDP [17] to form the linear system, which in turn calls PETSc [18] to solve it.

TDCS simulations can also be easily extended to simulations of transcranial alternating current stimulation (tACS). In the frequency ranges used in tACS, a quasi-static approximation holds [19]. In the quasi-static approximation, the relationship between input currents  $I(t)$  and the electric field at the positions  $\mathbf{x}$ ,  $E(\mathbf{x})$  is linear:

$$E(\mathbf{x}, t) = \alpha(\mathbf{x})I(t)$$

where  $\alpha(\mathbf{x})$  is a proportionality constant, meaning that it does not vary during the oscillation. This constant can be obtained simply by running a simulation where we set the input current to unity.  $I(t)$  is the input current. For example, a sinusoidal current input can be written as

$$I(t) = I_o \sin\left(2\pi \frac{t}{f} + \phi\right)$$

where  $f$  is the stimulator frequency,  $\phi$  the stimulator phase and  $I_o$  the stimulator amplitude, which corresponds to half of the peak-to-peak current. Usually, we would visualize the electric field at the maximum or minimum of  $I(t)$ , which

corresponds to  $\pm I_o$ . In case several stimulators are used at different frequencies of phases, we have several pairs  $(\alpha_i(\mathbf{x})I_i(t))$ , one for each stimulator, and the total electric field at a given time point is given by the sum of their individual contributions

$$E(\mathbf{x},t) = \sum_{i=1}^n \alpha_i(\mathbf{x})I_i(t)$$

### 1.2.5 Mapping Fields

The result of the FEM calculation is the electric field at each tetrahedral element of the subject’s head mesh. However, visualization is often easier using cortical surfaces or NifTI volumes. Therefore, SimNIBS 2.1 can transform fields from the native mesh format to these formats via interpolation. Our interpolation algorithm is based on the superconvergent patch recovery method [20], which ensures interpolated electric field values that are consistent with tissue boundaries.

When performing simulations on multiple subjects, we often want to be able to directly compare the electric field across subjects to, for example, correlate the electric field with behavioural or physiological data on the stimulation effects [21]. For this purpose, SimNIBS can also transform simulation results to the MNI template, using linear and non-linear co-registrations, as well as to the FreeSurfer’s FsAverage surface.

## 1.3 Practical Examples and Use Cases

### 1.3.1 Hello SimNIBS: How to Process a Single Subject

Here we describe how to run a TMS and a tDCS simulation on a single example subject. The example subject “Ernie” can be downloaded from the SimNIBS website, and the steps below can be reproduced step by step to get familiar with SimNIBS.

#### Generating the Volume Conductor Model

Open a terminal and go to the directory “ernie” to access the example data set. Copy the content of the “org”-subfolder to another location in order to not overwrite the files of the original example dataset. Next, go to the folder where you copied the data, and call headreco to generate the volume conductor model:

### **headreco all --cat ernie ernie\_T1.nii.gz ernie\_T2.nii.gz**

In the command, the first argument, “**all**”, tells headreco to run all reconstruction steps including: segmentation, clean-up of tissue maps, surface meshing, and volume meshing. The second argument, “**--cat**” is a flag for using the CAT12 toolbox for accurate reconstruction of the cortical surface. The third argument, “**ernie**”, is a subject identifier (subID), which is used to name generated folders, e.g. m2m\_ernie, and output files, e.g. ernie.msh. The two final arguments are the paths to the T1- and T2-weighted structural scans.

A few extra input options are useful to know:

- d no-conform** Adding this option will prevent headreco from modifying, i.e. transforming and resampling, the original MRI scan. This might be desirable when a one-to-one correspondence between the head model coordinates and the neural navigation system coordinates is required.
- v < density >** This option allows you to set the resolution, or vertex density (nodes per mm<sup>2</sup>), of the FEM mesh surfaces. By default, SimNIBS uses 0.5 nodes/mm<sup>2</sup> as the **<density >** value.

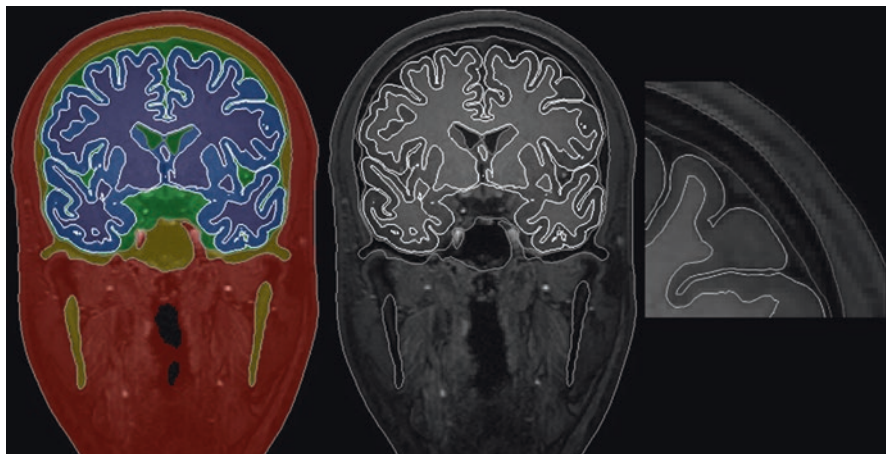
In general, we recommend using the **--cat** option; however, the execution time will be longer compared to omitting the option. In addition, if you want to process scans with pathologies, you should not use CAT12, as the cortical reconstruction is not designed to work with pathologies.

After **headreco** has finished, please check the quality of the head model by calling:

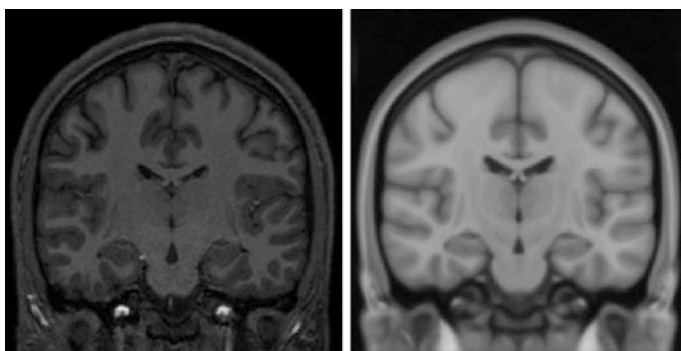
### **headreco check ernie**

If needed, open a new terminal for this operation and go into the folder in which you started headreco the first time. For our example case, the subject identifier is “ernie”, but please replace this one with whichever subID was used in the first call to **headreco**. Note that we recommend that you have installed freeview (provided by FreeSurfer, available on Linux and Mac OS X platforms) to visualize the results. The **check** function displays two windows for inspecting the output. The first window shows the T1-weighted scan with the segmentation and structure borders overlaid (Fig. 1.3, left). We recommend de-selecting the segmentation (**ernie\_final\_contr.nii**) in freeview, and checking that the segmentation borders follow the intensity gradients of different tissues (Fig. 1.3, middle). Fig. 1.4 shows the second freeview window, which displays the T1-weighted scan co-registered to the MNI template. We recommend checking if the T1-weighted scan overlaps well with the MNI template by de-selecting the T1-weighted scan (**T1fs\_nu\_nonlin\_MNI.nii**) in freeview (Fig. 1.4, right). Figure 1.5 shows an example of a segmentation error where the skull is erroneously labelled as skin. This can be seen in the front of the head, where the skin label protrudes into the skull. This example emphasizes the need for fat-suppressed data when only a T1-weighted scan is used. In the scan shown in Fig. 1.5, spongy bone is bright with intensities comparable to those of scalp, causing the segmentation method to mis-classify it as extra-cerebral soft tissue. Small segmen-



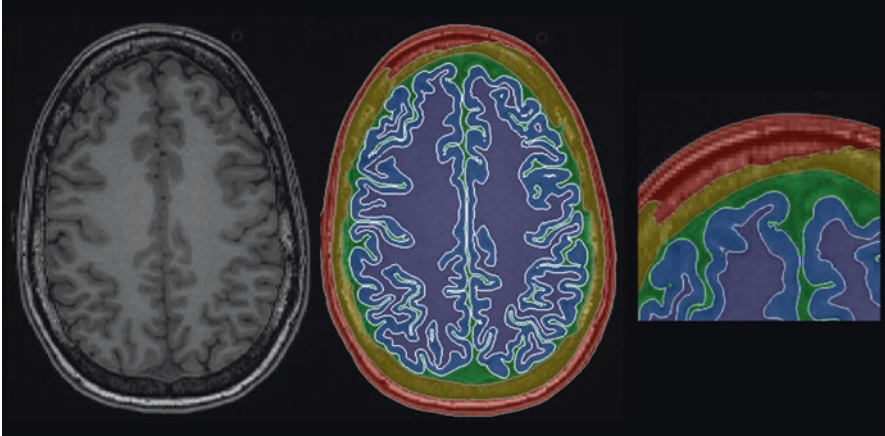


**Fig. 1.3** Data displayed after calling the **check** option. Left: T1-weighted scan with the segmentation and structure borders overlaid. Middle: structure borders overlaid on the T1-weighted scan after de-selecting the segmentation in freeview. Right: zoom-in of the cortex. Note that the segmentation borders nicely follow the intensity borders between the tissues



**Fig. 1.4** Data displayed after calling the **check** option. Left: T1-weighted scan co-registered to the MNI template. Right: MNI template shown after de-selecting the T1-weighted scan in freeview. Note that the scans seem to be well registered

tation errors like this can be corrected by manually re-labelling the segmentation masks in the “*mask\_prep*” folder located in the *m2m\_{subID}* folder, and re-running the surface and volume meshing steps. If you are not familiar with using freeview, please refer to the tutorial on the SimNIBS website ([http://www.simnibs.org/\\_media/docu2.1.1/tutorial\\_2.1.pdf](http://www.simnibs.org/_media/docu2.1.1/tutorial_2.1.pdf)). If you do not have access to freeview, the visualizations will be displayed using SPM. However, these are very primitive and are not recommended for checking the output from **headreco**.



**Fig. 1.5** Example of a segmentation error after headreco processing. The spongy bone is erroneously labelled as skin. This example emphasizes the need for fat-suppression when using only a T1-weighted scan

Finally, you should inspect the volume conductor mesh for any obvious errors. This can be done by calling:

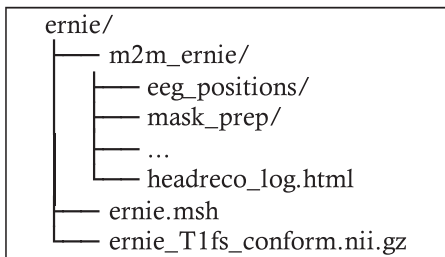
```
gmsch ernie.msh
```

in the subject folder. This call opens a gmsch window displaying the generated head model; please see the tutorial on the website if you are not familiar with gmsch ([http://www.simnibs.org/\\_media/docu2.1.1/tutorial\\_2.1.pdf](http://www.simnibs.org/_media/docu2.1.1/tutorial_2.1.pdf)).

The folder structure and most important files are shown in Table 1.1.

- *eeg\_positions/* Folder containing the 10-10 electrode positions for the subject both as a “.csv”, used for acquiring electrode positions, and a “.geo” file, used for visualization of the positions in Gmsch. If you have custom electrode positions, they should be added here as a .csv file.
- *mask\_prep/* Folder containing the cleaned tissue maps along with the white matter and pial surface files if CAT12 was used. In case there are errors in the segmentation, the masks can be manually corrected and a new head model can subsequently be generated. Note that the CAT12 WM and GM surfaces can currently not be modified.
- *headreco\_log.html*, a log-file with output from the **headreco** run. If something goes wrong, the log-file helps with troubleshooting, and should be sent as an attachment when contacting the SimNIBS support email list (support@simnibs.org).
- *ernie.msh*, the FEM head model used for the simulations.
- *ernie\_T1fs\_conform.nii.gz*, the input scan in the conform space defined by the `-d` option. This scan has the same millimetre space as the head model, and can be used to annotate landmarks which can then be directly transformed onto the head model.

**Table 1.1** The folder structure after **headreco** has finished. In this table, only the most important folders and files are listed



## Setting Up a Simulation

Once the head model is ready, we can set up tDCS and TMS simulations interactively using the GUI. The GUI can be started on the command line by calling:

**simnibs\_gui**

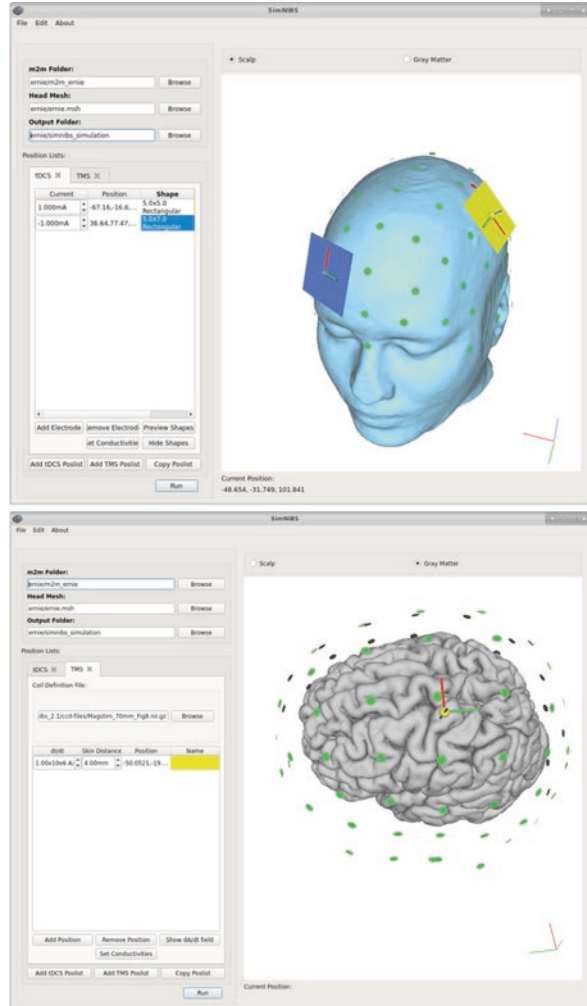
In the GUI, the user can:

- Visualize and interact with head models.
- Define electrode and coil positions by clicking in the model or selecting a position from the EEG 10-10 system.
- Visually define electrode shapes and sizes.
- Select from the available coil models.
- Change tissue conductivity parameters and set up simulations with anisotropic conductivity distributions.
- Run simulations.

In the GUI, there are two types of tabs, one for tDCS simulations, and another for TMS simulations, shown respectively in the top and bottom of Fig. 1.6. The tDCS tabs define a single tDCS field simulation with an arbitrary number of electrodes. On the other hand, TMS tabs can define several TMS field simulations using the same coil. For this example, we will set up a tDCS simulation with a  $5 \times 5$  cm anode placed over C3 and a  $7 \times 5$  cm cathode placed over AF4, and a TMS simulation with the coil placed over the motor cortex, pointing posteriorly. Details on how to use the graphical interface can be found on the website ([http://www.simnibs.org/\\_media/docu2.1.1/tutorial\\_2.1.pdf](http://www.simnibs.org/_media/docu2.1.1/tutorial_2.1.pdf)).

After the simulation setup, click on the **Run** button to start the simulations. Running both simulations takes 10–15 minutes, depending on the computer, and uses around 6 GB of memory. As a note, before starting the simulations, you can set additional options (in the menu Edit→Simulation Options) to let SimNIBS write out the results as surface data or NifTI volume data. This is not further covered in this basic example, but the output files created in these cases are described in the next example. The results of the simulation will be written in the output folder specified in the GUI, in this case “*simnibs\_simulation/*”. The folder has the files shown below in Table 1.2.

**Fig. 1.6** Set-up of a tDCS (top) and a TMS (bottom) simulation in the graphical user interface



**Table 1.2** The output folder of a simple tDCS and TMS simulation

```

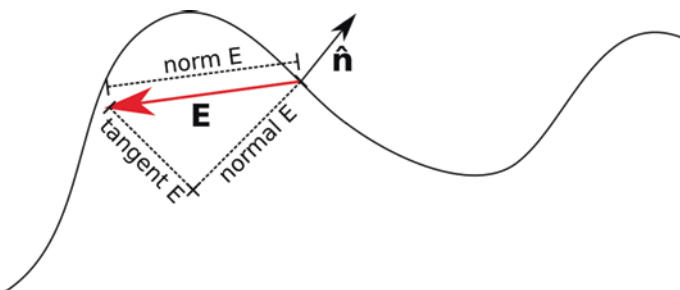
simnibs_simulation/
├── ernie_TDCS_1_scalar.msh
├── ernie_TMS_2-0001_Magstim_70mm_Fig8_nii_scalar.msh
├── ernie_TMS_2-0001_Magstim_70mm_Fig8_nii_coil_pos.geo
├── simnibs_simulation_20180920-130401.log
└── simnibs_simulation_20180920-130401.mat

```

- “*ernie\_TDCS\_1\_scalar.msh*” is the output from the tDCS simulation, in Gmsh “.msh” format. The first part of the file name, “*ernie*”, is the subID. The second part, “*TDCS*”, informs us that this is a tDCS simulation. The third part, “*1*”, denotes that this was the first simulation we have defined in the GUI, and finally, “*scalar*” tells us we have used scalar (as opposed to anisotropic) conductivities for the simulations.
- “*ernie\_TMS\_2-0001\_Magstim\_70mm\_Fig8\_nii\_scalar.msh*” is the output of the second simulation, also in gmsh “.msh” format. As is the case for the tDCS output, the first part of the file name is the subID, and the second is the number of the simulation in the simulation list. We next see the number of the TMS position, as it might happen that several TMS positions are defined in a single TMS list. Following this, “*Magstim\_40mm\_Fig8\_nii*” gives us the name of the coil used for the simulation, and “*scalar*” the type of conductivity.
- “*ernie\_TMS\_2-0001\_Magstim\_70mm\_Fig8\_nii\_coil\_pos.geo*” is a Gmsh “.geo” file which shows the coil position for the corresponding simulation.
- “*simnibs\_simulation\_20180920-13041.log*” is a text file with a detailed log of the simulation steps. This file can be used for troubleshooting. Here, the second part of the file is date and time information of when the simulation started.
- “*simnibs\_simulation\_20180920-13041.mat*” is a MATLAB data file with the simulation setups. This file can be loaded into the GUI or MATLAB at a later time to check the simulation parameters, or to change them and re-run the simulation.

## Visualizing Fields

The electric field  $\mathbf{E}$  is a vector field meaning that the electric field has both a norm (i.e. vector length or magnitude) and a direction in space, as shown in Fig. 1.7. As visualizations of the entire vector are challenging and often unclear, in SimNIBS we

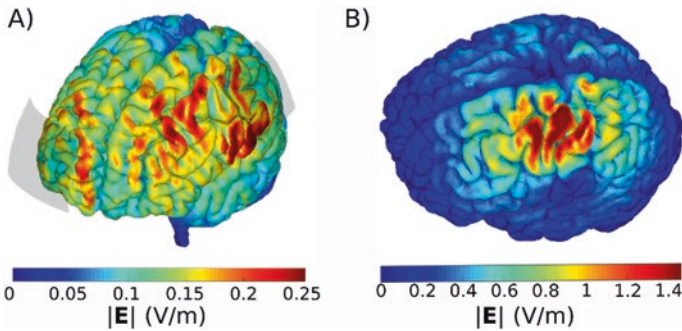


**Fig. 1.7** Decomposition of a vector  $\mathbf{E}$  in relation to a surface. The norm corresponds to the length of the vector. At each point, the surface defines a normal vector  $\hat{\mathbf{n}}$ , and this vector is perpendicular to the tangent plane to the surface at that point. Given the normal vector, we can decompose the vector  $\mathbf{E}$  into normal and tangent components. The normal component is the part of  $\mathbf{E}$  in the same line as the normal vector, and the tangent component is perpendicular to it. The normal component also has a sign, indicating if the field is entering or leaving the surface. In SimNIBS, a positive normal indicates that the field is entering the surface, and a negative normal indicate the field is leaving the surface

usually visualize the **norm** (or strength) of the electric field instead. The norm of the electric field corresponds to the size of the electric field vector, and therefore is always positive and does not contain any information about the direction of the electric field.

One way we can quickly visualize the simulation results is to use the **mesh\_show\_results** MATLAB function. This function comes as a part of SimNIBS version 2.1.2, and provides visualizations of the output fields using MATLAB plotting tools, as well as some summary values for the field strength and focality. For example, when running the function on the output tDCS mesh, we obtain the plot shown in Fig. 1.8a, and the values below in Table 1.3.

The first lines in Table 1.3 show that the displayed data is the field “norm E”, that is the norm or strength of the electric field, calculated in the region number 2, which corresponds to the GM volume. Afterwards, we have information on the peak electric fields. We see that the value of 0.161 V/m corresponds to the 95th percentile of the norm of the electric field, the value of 0.201 V/m to the 99th percentile and 0.249 to the 99.9th percentile. We also have information about the focality of the



**Fig. 1.8** Visualization of (a) tDCS and (b) TMS electric field norms in MATLAB

**Table 1.3** Output of **mesh\_show\_results** for the tDCS simulation

-----			
SUMMARY			
field name: normE			
region indices: 2			
peak fields			
percentiles:	95	99	99.9
values:	0.161	0.201	0.249 (in [V/m])
focality			
cutoffs:	50	75 (in % of 99.9 percentile)	
values:	1.4e+05	1.29e+04 (in cubic mm)	
-----			

electric field. Here, focality is measured as the GM volume with an electric field greater or equal to 50% or 75% of the peak value. To avoid the effect of outliers, the peak value is defined as the 99.9th percentile.

Running the same function on the TMS result file, we obtain the plot shown in Fig. 1.8b, as well as the peak fields and focality measures shown below in Table 1.4.

We can see that the peak fields for TMS are much higher than for tDCS, even though we simulated with a current of  $10^6$  A/s, very low for TMS. In the focality measures, we see that the TMS electric fields are much more focal than the tDCS electric fields, with around five times less GM volume exceeding 75% of the peak value than tDCS.

Additionally, the “.msh” files can be opened with the Gmsh viewer, producing 3D visualizations as shown in Fig. 1.9. Gmsh has a vast range of functionalities, such as clipping planes, but can be harder to use than `mesh_show_results`.

**Table 1.4** Output of `mesh_show_results` for the TMS simulation

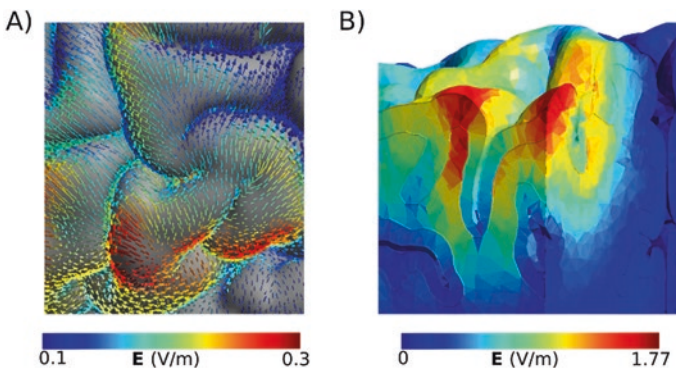
```

-----
SUMMARY
field name: normE
region indices: 2

peak fields
percentiles:   95    99    99.9
values:        0.446  0.849  1.41 (in [V/m])

focality
cutoffs:       50    75 (in % of 99.9 percentile)
values:        1.28e+04  3.32e+03 (in cubic mm)
-----

```



**Fig. 1.9** Visualization in Gmsh of (a) electric field vectors around central gyrus for the tDCS simulation and (b) TMS electric field depth profile in the hotspot

### 1.3.2 *Advanced Usage: Group Analysis*

Now, we want to simulate one tDCS montage, with a  $5 \times 5$  cm electrode over C3 and a  $5 \times 7$  cm electrode over AF4 in five subjects, called “sub01”, “sub09”, “sub10”, “sub12”, “sub15” and visualize the results in a common space, namely the FsAverage surface. The subjects and example scripts can be downloaded from: <https://osf.io/ah5eu/>

#### Head Meshing

For each subject, follow the steps in section “[Generating the Volume Conductor Model](#)”.

#### Write a Python or MATLAB Script

We can set up the simulation of each subject using the GUI, as described in the first example. However, when working with multiple subjects, it can be advantageous to script the simulations for efficiency. SimNIBS provides both MATLAB and Python interfaces to set up simulations. Script 1.1 shows how to set up and run a simulation with a  $5 \times 5$ cm anode placed over C3 and a  $7 \times 5$ cm cathode placed over AF4 for all subjects. The output of Script 1.1 for sub01 is shown in Table 1.5.

To define the rectangular electrodes, we need two coordinates. The “*centre*” defines where the electrode will be centred, and “*pos\_ydir*” how the electrode will be rotated. More precisely, the electrode’s “*y*” axis is defined as a unit vector starting at “*centre*” and pointing towards “*pos\_ydir*”. Fig. 1.10 shows one of the cathodes (return electrode) defined using the script above, with the coordinate system and EEG positions overlaid. We can see that the electrode is centred in AF4, and its Y axis points towards F6. “*pos\_ydir*” does not need to be set when the electrodes are round.

When the *map\_to\_fsavg* option is set to *true*, SimNIBS computes the electric fields in a surface located in the middle of the GM layer. This cortical surface, along with the norm, normal and tangent components of the electric field at the cortical surface and the angle between the electric field and the cortical surface can be found in the *subject\_overlays* folder, for both the left hemisphere (*lh*) and for the right hemisphere (*rh*) as shown in Table 1.5. Afterwards, these quantities are transformed into the FsAverage space. The transformed quantities can be found in the *fsavg\_overlays* folder, as shown in Table 1.5. Additionally, we have the electric field and its norm in MNI space in the *mni\_volumes* folder.

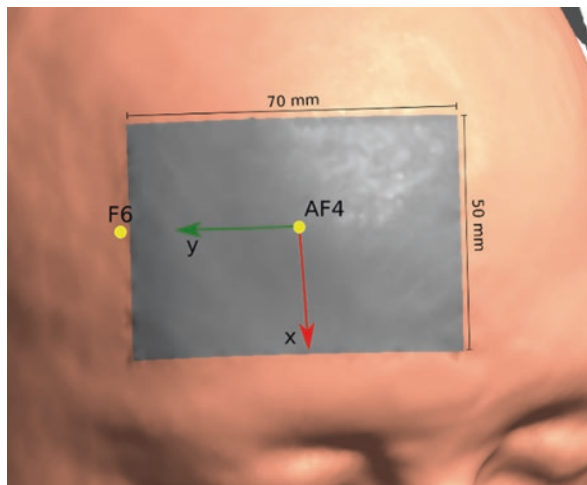


**Table 1.5** Output files and folders of Script 1 for sub01. The “.angle”, “.norm”,... files are FreeSurfer overlay files and the “.central” files are FreeSurfer surface files

```

bipolar/
├── fsavg_overlays
│   ├── lh.sub01_TDCS_scalar.fsavg.E.angle
│   ├── lh.sub01_TDCS_scalar.fsavg.E.norm
│   ├── lh.sub01_TDCS_scalar.fsavg.E.normal
│   ├── lh.sub01_TDCS_scalar.fsavg.E.tangent
│   ├── rh.sub01_TDCS_scalar.fsavg.E.angle
│   └── ...
├── mni_volumes
│   ├── sub01_TDCS_scalar_MNI_E.nii.gz
│   └── sub01_TDCS_scalar_MNI_normE.nii.gz
├── simnibs_simulation_20181019-100530.log
├── simnibs_simulation_20181019-100530.mat
└── subject_overlays
    ├── lh.central
    ├── lh.sub01_TDCS_scalar.central.E.angle
    ├── lh.sub01_TDCS_scalar.central.E.norm
    ├── lh.sub01_TDCS_scalar.central.E.normal
    ├── ...
    ├── rh.central
    ├── rh.sub01_TDCS_scalar.fsavg.E.angle
    └── ...
    
```

**Fig. 1.10** 50 × 70 mm electrode defined with a “centre” in AF4 and a “pos\_ydir” in F6



```

path_to_headmodels = "/path/to/head/models/";
subjects = ["sub01", "sub09", "sub10", "sub12", "sub15"];
results_folder = "bipolar/fsavg_overlays";

normals = {};
for i = 1:length(subjects)
    sub = subjects(i);
    % Load normal field data
    normal_surf = sprintf('lh.%s_TDCS_1_scalar.fsavg.E.normal', sub);
    m = mesh_load_fsresults(char(...
        fullfile(path_to_headmodels, sub, results_folder, normal_surf)));
    % Add to cell
    normals{i} = m.node_data{1}.data;
end
% Calculate average and standard deviation of the normal at each node
normals = cell2mat(normals);
avg_normal = mean(normals, 2);
std_normal = std(normals, 0, 2);
% Place the fields in the mesh structure
m.node_data{1}.data = avg_normal;
m.node_data{1}.name = 'E.normal.avg';
m.node_data{2}.data = std_normal;
m.node_data{2}.name = 'E.normal.std';
% Plot the fields
mesh_show_surface(m, 'field_idx', 'E.normal.avg')
mesh_show_surface(m, 'field_idx', 'E.normal.std')

```

**Script 1.1** Script for running a tDCS simulations with an anode over C3 and a cathode over AF4 in five subjects and transforming the results to FSAverage and MNI spaces.

```

path_to_headmodels = "/path/to/head/models/";
subjects = ["sub01", "sub09", "sub10", "sub12", "sub15"];
results_folder = "bipolar/fsavg_overlays";

normals = {};
for i = 1:length(subjects)
    sub = subjects(i);
    % Load normal field data
    normal_surf = sprintf('lh.%s_TDCS_1_scalar.fsavg.E.normal', sub);
    m = mesh_load_fsresults(char(...
        fullfile(path_to_headmodels, sub, results_folder, normal_surf)));
    % Add to cell
    normals{i} = m.node_data{1}.data;
end
% Calculate average and standard deviation of the normal at each node
normals = cell2mat(normals);
avg_normal = mean(normals, 2);
std_normal = std(normals, 0, 2);
% Place the fields in the mesh structure
m.node_data{1}.data = avg_normal;
m.node_data{1}.name = 'E.normal.avg';
m.node_data{2}.data = std_normal;
m.node_data{2}.name = 'E.normal.std';
% Plot the fields
mesh_show_surface(m, 'field_idx', 'E.normal.avg')
mesh_show_surface(m, 'field_idx', 'E.normal.std')

```

**Script 1.2** Analysis of simulation results in FSAverage space.

## Visualizing Results

We can also make use of the MATLAB library within SimNIBS to analyze the results from the simulations. Here, we are interested in the average and standard deviation of the normal component of the electric field in the cortex. The normal component, as shown in Fig. 1.7, is the part of the electric field which is either entering or leaving the cortex.

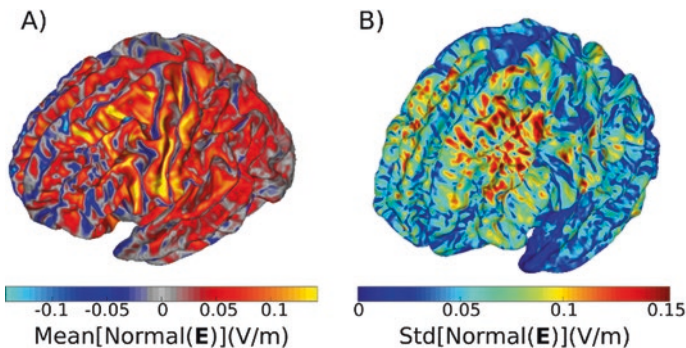
Script 1.2 loads the normal field component data for each subject and calculates the mean and the standard deviation across subjects at each position of the FsAverage template. The fields are then visualized using MATLAB visualization tools. The results are shown in Fig. 1.11. We can, for example, see strong current in-flow in the central gyrus, and large variations in the normal component in frontal regions.

### 1.4 The Accuracy of Automatic EEG Positioning

Here, we compare EEG 10-10 positions obtained either from:

- Transforming EEG 10-10 electrode positions defined in MNI space to the subject space using a non-linear transform, and then projecting the positions to the scalp. This is done for both **mri2mesh** and **headreco** head models.
- Manually locating the fiducials: left pre-auricular point (LPA), right pre-auricular point (RPA), nasion (Nz) andinion (Iz) on MRI images, and afterwards calculating the EEG positions using the definitions in [22].

Calculations using method A require no user input and are automatically performed in both **mri2mesh** and **headreco** head modelling pipelines, while calculations using method B require the user to manually select the fiducial positions.



**Fig. 1.11** (a) Mean and (b) Standard deviation of the normal field component across 5 subjects. The fields were caused by tDCS with an anode over C3 and a cathode over AF4. Positive values in (a) denote inflowing currents, and negative values outflowing currents

To compare the methods A and B to position the electrodes, we calculated the EEG 10-10 positions using both ways for MR data of 17 subjects. The data was acquired as part of a larger study. The subjects gave written informed consent before the scan, and the study was approved by the local ethics committee of the University of Greifswald (Germany). The 17 datasets were acquired on a 3-Tesla Siemens Verio scanner (Siemens Healthcare, Erlangen, Germany) using a 32-channel head coil (T1:  $1 \times 1 \times 1 \text{ mm}^3$ , TR 2300 ms, TE 900 ms, flip angle  $9^\circ$ , with selective water excitation for fat suppression; T2:  $1 \times 1 \times 1 \text{ mm}^3$ , TR 12770 ms, TE 86 ms, flip angle  $111^\circ$ ). For method B, the fiducials were manually located for each subject by a trained investigator on the T1- and T2-weighted images. The later had no knowledge of the automatically determined positions. The fiducials Nz, Iz, LPA and RPA were set in freeview, following the procedure described in [22] and additionally verified using the SimNIBS GUI. The subject-specific coordinates of the fiducials were extracted, and these manually set positions were then compared with those calculated by the automatic algorithm in each individual.

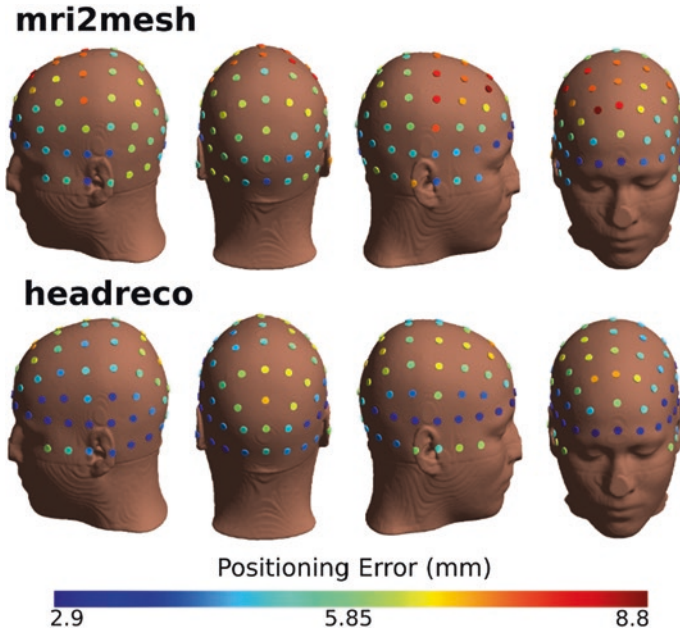
Table 1.6 shows the maximal distance across all subjects between the fiducials obtained using method A and manually selected fiducials (B). We see that Nz is the most consistent fiducial, where we have the least deviation, whereas Iz is where we have the highest deviation. Also, the maximal difference in position across the two methods is  $\sim 1 \text{ cm}$ , indicating that method A works well to approximate the positions of the fiducials.

Furthermore, in Fig. 1.12, we compare the two methods for all electrode positions in the EEG 10-10 system. The deviation in positioning each electrode was calculated as the mean of the distance between the positions obtained with either `headreco` or `mri2mesh` to the manually located fiducial positions, across all 17 subjects and for each electrode.

The errors for all electrodes are below 1 cm, indicating that the two algorithms for placing EEG electrodes are in agreement. We can also see that the errors in the EEG positions obtained from `headreco` are on average lower than the ones obtained from `mri2mesh`. It also seems that the anterior electrodes have less errors than the posterior electrodes. Interestingly, the location of the errors is different across the two pipelines, with `mri2mesh` being more inaccurate in superior regions and `headreco` more inaccurate in posterior regions. This might be caused by differences in the way FSL (`mri2mesh`) and SPM (`headreco`) calculate non-linear MNI transfor-

**Table 1.6** Maximum and mean distance between the fiducial positions selected by hand and obtained from the MNI transformations across 17 subjects, for the two head modelling pipelines

Fiducial	mri2mesh		headreco	
	Max distance (mm)	Mean distance $\pm$ standard deviation (mm)	Max distance (mm)	Mean distance $\pm$ standard deviation (mm)
LPA	6.4	$3.2 \pm 1.5$	8.7	$5.4 \pm 2.0$
RPA	8.9	$3.0 \pm 1.6$	10.6	$5.9 \pm 1.7$
Nz	3.9	$2.1 \pm 1.0$	6.0	$3.9 \pm 1.6$
Iz	14.3	$4.0 \pm 3.5$	13.2	$5.2 \pm 3.3$



**Fig. 1.12** Positioning error for electrodes in the EEG 10-10 system. The error is calculated by comparing the positions calculated based on manually selected fiducials to positions calculated based on non-linear MNI transformations

mations is different. The average error across all positions was 5.6 mm for **mri2mesh** head models and 4.9 mm for **headreco** head models indicating good accuracy.

## 1.5 Conclusion

We presented SimNIBS 2.1 ([www.simmibs.org](http://www.simmibs.org)), a software for individualized modelling of electric fields caused by non-invasive brain stimulation. SimNIBS is free software and available for all major platforms. SimNIBS does not require the installation of any additional software in order to run simulations on the example dataset. To construct head models, SimNIBS relies either on MATLAB, SPM12 and CAT12 (**headreco**) or on FSL and FreeSurfer (**mri2mesh**).

We also presented two examples of workflows in SimNIBS. In the first example, we started by using **headreco** to construct a head model. Following this, we used the GUI to set up a tDCS and a TMS simulation in an interactive way, and finally visualized the results. In the second example, we constructed several head models and used a MATLAB script to run simulations for each subject. We then calculated the mean and the standard deviation of the electric field norm across all subjects, using the FreeSurfer's FsAverage brain template. Finally, we show results validating our automatic procedure to obtain electrode positions for the EEG 10-10 system.

SimNIBS is still being actively developed, and we expect further updates to be implemented in the future.

**Acknowledgements** Lundbeckfonden (grant Nr. R118-A11308), and NovoNordisk fonden (grant Nr. NNF14OC0011413). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731827 "STIPED". The results and conclusions in this chapter present the authors' own views, and do not reflect those of the EU Commission.

## References

1. Fregni, F., & Pascual-Leone, A. (2007). Technology insight: Noninvasive brain stimulation in neurology – Perspectives on the therapeutic potential of rTMS and tDCS. *Nature Clinical Practice Neurology*, 3(7): 282–393
2. Thielscher, A., Antunes, A., and Saturnino, G. B.. (2015). Field modeling for transcranial magnetic stimulation: A useful tool to understand the physiological effects of TMS? In *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS* (pp. 222–225).
3. Guerra, A., López-Alonso, V., Cheeran, B., & Suppa, A. (2018). Solutions for managing variability in non-invasive brain stimulation studies. *Neuroscience Letters*.
4. Li, L. M., Uehara, K., & Hanakawa, T. (2015). The contribution of interindividual factors to variability of response in transcranial direct current stimulation studies. *Frontiers in Cellular Neuroscience*, 9, 181.
5. Kubis, N. (2016). Non-invasive brain stimulation to enhance post-stroke recovery. *Front. Neural Circuits*, 10, 56.
6. Morishita, T., & Hummel, F. C. (2017). Non-invasive Brain Stimulation (NIBS) in motor recovery after stroke: Concepts to increase efficacy. *Curr. Behav. Neurosci. Reports*, 4(3), 280–289.
7. Windhoff, M., Opitz, A., & Thielscher, A. (2013). Electric field calculations in brain stimulation based on finite elements: An optimized processing pipeline for the generation and usage of accurate individual head models. *Human Brain Mapping*, 34(4), 923–935.
8. Nielsen, J. D., Madsen, K. H., Puonti, O., Siebner, H. R., Bauer, C., Madsen, C. G., Saturnino, G. B., & Thielscher, A. (2018). Automatic skull segmentation from MR images for realistic volume conductor models of the head: Assessment of the state-of-the-art. *NeuroImage*, 174, 587–598.
9. Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E. J., Johansen-Berg, H., Bannister, P. R., De Luca, M., Drobnjak, I., Flitney, D. E., Niazy, R. K., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, J. M., & Matthews, P. M. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23, S208–S219.
10. Fischl, B. (2012). FreeSurfer. *NeuroImage*, 62(2), 774–781.
11. W. Penny, K. Friston, J. Ashburner, S. Kiebel, and T. Nichols, *Statistical parametric mapping: The analysis of functional brain images*. 2007. Elsevier; London, UK.
12. Dahnke, R., Yotter, R. A., & Gaser, C. (2013). Cortical thickness and central surface estimation. *NeuroImage*, 65, 336–348.
13. Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331.
14. Thielscher, A., & Kammer, T. (2004). Electric field properties of two commercial figure-8 coils in TMS: Calculation of focality and efficiency. *Clinical Neurophysiology*, 115(7), 1697–1708.

15. Madsen, K. H., Ewald, L., Siebner, H. R., & Thielscher, A. (2015). Transcranial magnetic stimulation: An automated procedure to obtain coil-specific models for field calculations. *Brain Stimulation* 8(6), 1205–1208.
16. Saad, Y. (2003). *Iterative methods for sparse linear systems* (2nd ed.). SIAM. Philadelphia, PA, USA.
17. Geuzaine, C. (2007). GetDP: A general finite-element solver for the de Rham complex. *PAMM*, 7(1), 1010603–1010604.
18. Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Grop, W., Kaushik, D., Knepley, M., May, D., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B., Zampini, S., Zhang, H., and Zhang, H. (2018). {PETS}c {W}eb page.
19. Opitz, A., Falchier, A., Yan, C. G., Yeagle, E. M., Linn, G. S., Megevand, P., Thielscher, A., Deborah, R. A., Milham, M. P., Mehta, A. D., & Schroeder, C. E. (2016). Spatiotemporal structure of intracranial electric fields induced by transcranial electric stimulation in humans and nonhuman primates. *Scientific Reports*, 6, 1–11.
20. Zienkiewicz, O. C., & Zhu, J. Z. (1992). The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7), 1331–1364.
21. Bungert, A., Antunes, A., Espenhahn, S., & Thielscher, A. (2017). Where does TMS stimulate the motor cortex? Combining electrophysiological measurements and realistic field estimates to reveal the affected cortex position. *Cerebral Cortex*, 27(11), 5083–5094.
22. Jurcak, V., Tsuzuki, D., & Dan, I. (2007). 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4), 1600–1611.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

