

openMEEG Tutorial
EEG and MEG forward problems
via the command-line interface.

Maureen Clerc
Alexandre Gramfort
Perrine Landreau
Théo Papadopoulos

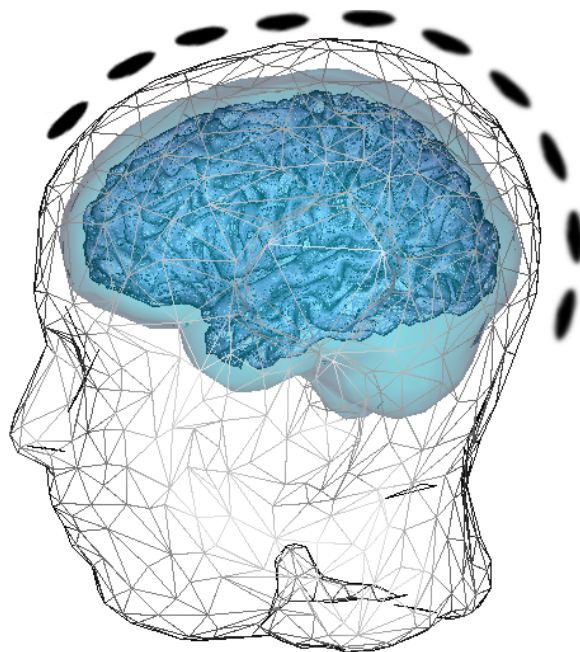
July 2008
Version 0.9

Contents

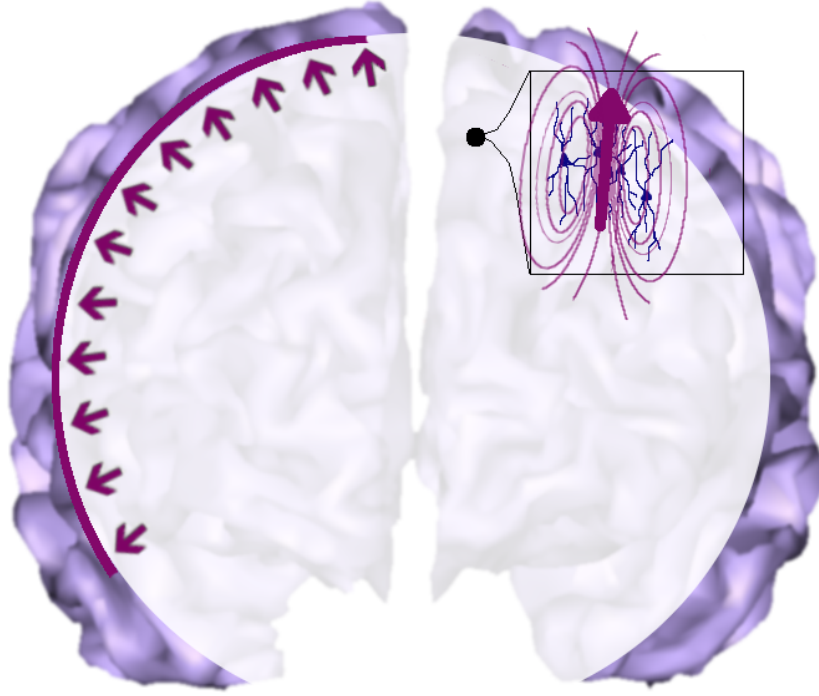
1	Context	5
2	Data	7
3	Commands	9
3.1	Head Matrix assembly HeadMat :	9
3.2	Source matrix assembly Source :	9
3.3	HeadMat matrix inversion :	10
3.4	Linear transformation from X to the sensor potential :	10
3.5	Gain matrix computation :	12
3.6	The forward problem :	12
A	Appendices	13
A.1	Geometry description file	13
A.2	Fichier de description des conductivités	14
A.3	Position et orientation des dipôles	14
A.4	Position (et orientation) des capteurs	15
A.5	Activation des sources	15

Chapter 1

Context



The forward problem consists in simulating the electric potential (EEG) and magnetic fields (MEG) on the sensors due to electrical sources within the brain.



Step 1 : the potential is computed on **all surfaces** of the head model (scalp, outer skull and inner skull for a three-layer model). Let \mathbf{X} contain the values of the potential on the discretized surfaces, as well as the values of the normal current. The Boundary Element Method leads to a linear system:

$$\mathbf{HeadMat} \cdot \mathbf{X} = \mathbf{SourceMat}$$

See sections 3.1, 3.2 and 3.3.

Step 2 : the matrices relating the head model surfaces and the sensor data must be computed. These are different for EEG and for MEG.

For EEG, the potential is interpolated from the surface discretization points to the sensor positions through a simple linear transformation :

$$[\text{potential at sensors}] = [\text{interpolation matrix}] \times [\text{potential at interfaces}] \text{ in the case of EEG.}$$

For MEG, the Biot and Savart equation allows to identify two contributions to the magnetic field: one which comes directly from the sources, and an ohmic contribution which comes from the volume conductor. Hence two linear transformations must be computed, one from the source locations to MEG sensors, the other from the head model interfaces to the MEG sensors.

See section 3.4.

Step 3 : the matrix relating the sources (at fixed positions and orientations) to the sensors is now ready to be computed (section 3.5). This matrix is called the **gain matrix** and is denoted \mathbf{G} . The gain matrix is then applied to the source activation to simulate the forward problem (section 3.6).

Chapter 2

Data

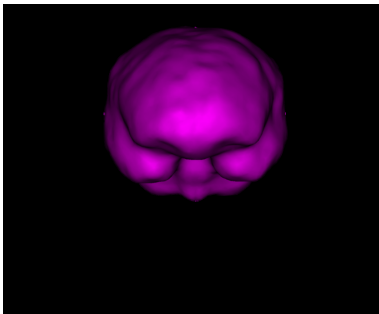
Initial requirements :

Meshes :

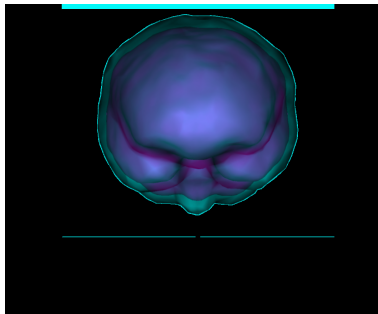
Meshes describing the interfaces between regions of homogeneous conductivity. These meshes generally represent:

- the inner skull surface
- the outer skull surface
- the outer scalp surface

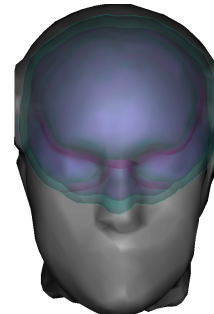
The recommended mesh size is approximately 600 to 800 points per surface.



Surface extérieure au cortex



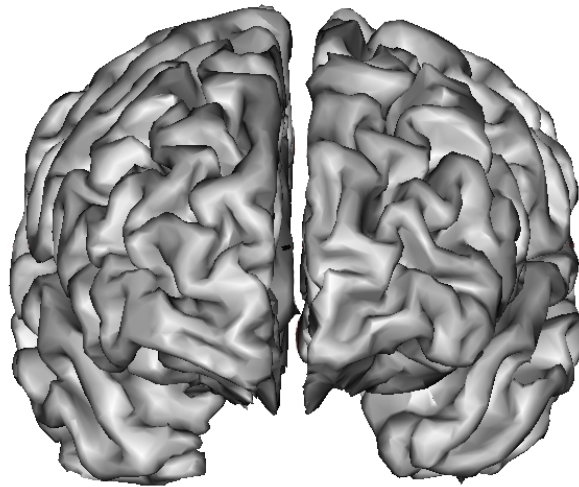
Surface extérieure au crâne en bleu et surface extérieure au cortex en fushia



Example with three surfaces :

- outer scalp (gray)
- outer skull (blue)
- inner skull (pink)

For distributed sources, a source mesh is required. This is a detailed mesh generally covering the whole cortex. The mesh size should not exceed 35 000 points.



Source mesh

Sensors : For EEG, the sensors are defined by the list of the x-y-z coordinates of the electrode positions. The electrodes are considered punctual and are called *patches*.

Chapter 3

Commands

In the following, the binaries in **red**, the options in **green**, the inputs are in **black** and the output in **blue**.

3.1 Head Matrix assembly **HeadMat** :

Inputs :

- **subject.geom** : geometry description file (see Appendix **A.1**)
- **subject.cond** : conductivity description file (see Appendix **A.2**)

Output :

- **HeadMat.bin** : binary file containing the matrix **HeadMat** (symmetric format).

The symmetric format only stores the lower half of a matrix.

```
$ om.assemble -HeadMat subject.geom subject.cond HeadMat.bin
```

Note: the abbreviated option names **-HM** or **-hm** can be used instead of **-HeadMat**.

3.2 Source matrix assembly **Source** :

Inputs :

- **subject.geom** : geometry description file (see Appendix **A.1**)
- **subject.cond** : conductivity description file (see Appendix **A.2**)
- the source(s) :
 - [**dipolar case**] **dipolePosition.dip** : dipole description file (list of coordinates and orientations) (see Appendix **A.3**)
 - [**case of distributed sources**] **sourcemes**h : source mesh (accepted formats: *.tri or *.mesh of BrainVisa, or *.vtk)

Output :

- **SourceMat.bin** : binary file containing **SourceMat**

For dipolar sources :

```
$ om_assemble -DipSourceMat subject.geom subject.cond dipolePosition.dip SourceMat.bin
```

Note: the abbreviated option names `-DSM` or `-dsm` can be used instead of `-DipSourceMat`.

For distributed sources :

```
$ om_assemble -SurfSourceMat subject.geom subject.cond sourcemesh SourceMat.bin
```

Note: the abbreviated option names `-SSM` or `-ssm` can be used instead of `-SurfSourceMat`.

3.3 HeadMat matrix inversion :

Inputs :

- `HeadMat.bin` : binary file containing matrix **HeadMat** (symmetric format)

Sortie :

- `HeadMatInv.bin` : binary file containing matrix **HeadMat**⁻¹ (symmetric format)

```
$ om_minverser HeadMat.bin HeadMatInv.bin
```

3.4 Linear transformation from X to the sensor potential :

✓ For EEG :

A linear interpolation is computed which relates **X** to the electrode potential through the linear transformation :

$$\mathbf{V}_{\text{electrode}} = \mathbf{Head2EEG} \cdot \mathbf{X}$$

where :

- **V_{electrode}** is the column-vector of potential values at the sensors (output of EEG forward problem),
- **X** is the column-vector containing the values of the potential and the normal current on all the surface of the model,
- **Head2EEGMat** is the linear transformation to be computed.

Inputs :

- `subject.geom` : geometry description file (see Appendix A.1)
- `subject.cond` : conductivity description file (see Appendix A.2)
- `patchespositions.txt` : file containing the positions of the EEG electrodes (see Appendix A.4)

Sortie :

- `Head2EEGMat.bin` : file containing the matrix **Head2EEGMat** (sparse format)

The sparse format allows to store efficiently matrices containing a small proportion of non-zero values.

```
$ om_assemble -Head2EEGMat subject.geom subject.cond patchespositions.txt Head2EEGMat.bin
```

Note: the abbreviated option names `-H2EM` or `-h2em` can be used instead of `-Head2EEGMat`.

✓ For MEG :

In the case of MEG there are more matrices to assemble, as explained in section ??.

The magnetic field is related both to the sources directly, as well as to the electric potential, according to :

$$\mathbf{M}_{\text{sensor}} = \mathbf{Source2MEGMat} \cdot \mathbf{S} + \mathbf{Head2MEGMat} \cdot \mathbf{X}.$$

Contribution to MEG from the potential (**Head2MEGMat**) :

Inputs :

- `subject.geom` : geometry description file (see Appendix A.1)
- `subject.cond` : conductivity description file (see Appendix A.2)
- `sensorpositions.txt` : positions and orientations of MEG sensors (see Appendix A.4)

Output :

- `Head2MegMat.bin` : binary file containing **Head2MEGMat**

```
$ om_assemble -Head2MEGMat subject.geom subject.cond sensorpositions.txt Head2MEGMat.bin
```

Note: the abbreviated option names `-H2MM` or `-h2mm` can be used instead of `-Head2MEGMat`.

Contribution to MEG from the sources (**Source2MEGMat**) :

Inputs :

- the source(s) :
 [dipolar sources] `dipolePosition.dip` : dipole description file (list of coordinates and orientations)
 (see Appendix A.3)
 [distributed sources] `sourcemes` : source mesh (accepted formats: *.tri or *.mesh of BrainVisa, or *.vtk)
- `sensorpositions.txt` : positions and orientations of MEG sensors (see Appendix A.4)

Output :

- `DipSource2MEGMat.bin` : binary file containing **DipSource2MEGMat**
- or `SurfSource2MEGMat.bin` : binary file containing **SurfSource2MEGMat**

For dipolar sources :

```
$ om_assemble -DipSource2MEGMat dipolePosition.dip sensorpositions.txt DipSource2MEGMat.bin
```

Note: the abbreviated option names `-DS2MM` or `-ds2mm` can be used instead of `-DipSource2MEGMat`.

For distributed sources :

```
$ om_assemble -SurfSource2MEGMat sourcemes sensorpositions.txt SurfSource2MEGMat.bin
```

Note: the abbreviated option names `-SS2MM` or `-ss2mm` can be used instead of `-SurfSource2MEGMat`.

3.5 Gain matrix computation :

The gain matrix represents the linear transformation relating the activation of sources, at **predefined positions and orientations** to the values of the fields of interest (electric potential or magnetic field) at predefined sensor positions (and orientations for MEG).

✓ For EEG :

Inputs :

- `HeadMatInv.bin` : binary file containing **HeadMat**⁻¹ (symmetric format)
- `SourceMat.bin` : binary file containing either **SurfSourceMat** or **DipSourceMat**
- `Head2EEGMat.bin` : binary file containing **Head2EEGMat** (sparse format)

Output :

- `GainEEGMat.bin` : binary file containing the gain matrix

```
$ om_gain -EEG HeadMatInv.bin SourceMat.bin Head2EEGMat.bin GainEEGMat.bin
```

✓ For MEG :

Inputs :

- `HeadMatInv.bin` : binary file containing **HeadMat**⁻¹ (symmetric format)
- `SourceMat.bin` : binary file containing either **SurfSourceMat** or **DipSourceMat**
- `Head2MEGMat.bin` : binary file containing **Head2MEGMat**
- `Source2MEGMat.bin` : binary file containing either **DipSource2MEGMat** or **SurfSource2MEGMat**

Output :

- `GainMEGMat.bin` : binary file containing the gain matrix

```
$ om_gain -MEG HeadMatInv.bin SourceMat.bin Head2MEGMat.bin Source2MEGMat.bin GainMEGMat.bin
```

3.6 The forward problem :

Once the gain matrix is computed, the forward problem amounts to defining the source activation, and applying the gain matrix to this activation.

Inputs :

- `GainMat.bin` : binary file containing EEG or MEG gain matrix
- `activationSources.txt` : file describing the source activation (see Appendix A.5)
- `noise` : noise (zero, or positive real number)

Output:

- `sensorResults.txt` : file containing the simulated sensor data.

```
$ om_forward gainMat.bin activationSources.txt sensorResults.txt noise
```

Appendix A

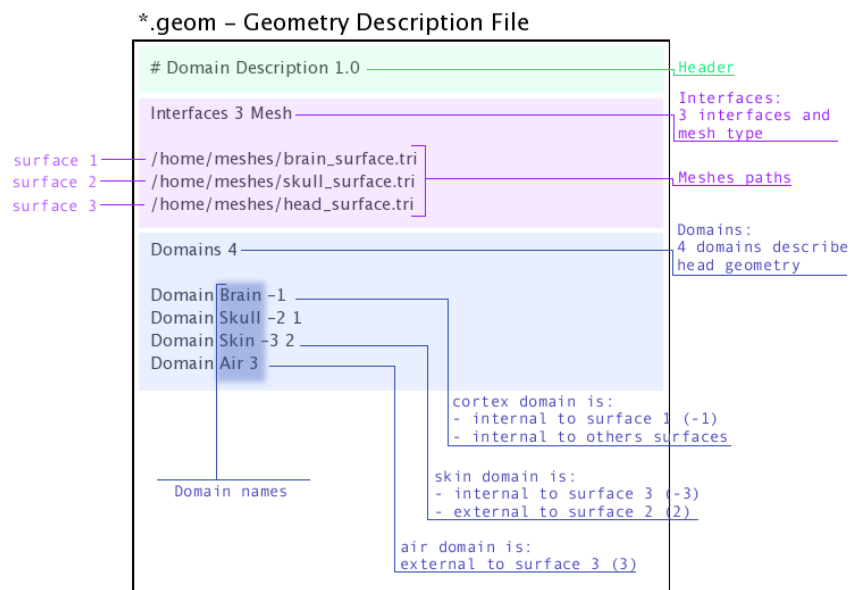
Appendices

A.1 Geometry description file

TO BE TRANSLATED

Ce fichier décrit la géométrie utilisée : le nombre et le nom des surfaces maillées séparant les différents domaines de conductivité ainsi que le nombre et le nom de chacun de ces domaines et leur disposition par rapport aux surfaces précédemment citées.

Extension du fichier : *.geom par convention.



Note 1 : nous avons un bug qui peut survenir sur l'ordre des domaines. Il est alors préférable de noter les domaines de la manière suivante (noter en premier la référence aux surfaces externes et en deuxième position la surface interne) :

Domain Brain -1

Domain Skull **1 -2** *au lieu de Domain Skull -2 1*

Domain Skin **2 -3** *au lieu de Domain Skin -3 2*

Domain Air 3

Note 2 : les "Meshes paths" sont

- soit absolus (comme le montre le dessin)
- soit relatifs à l'endroit où on exécute les lignes de commandes

Seuls les formats suivants sont lus pour les surfaces maillées :

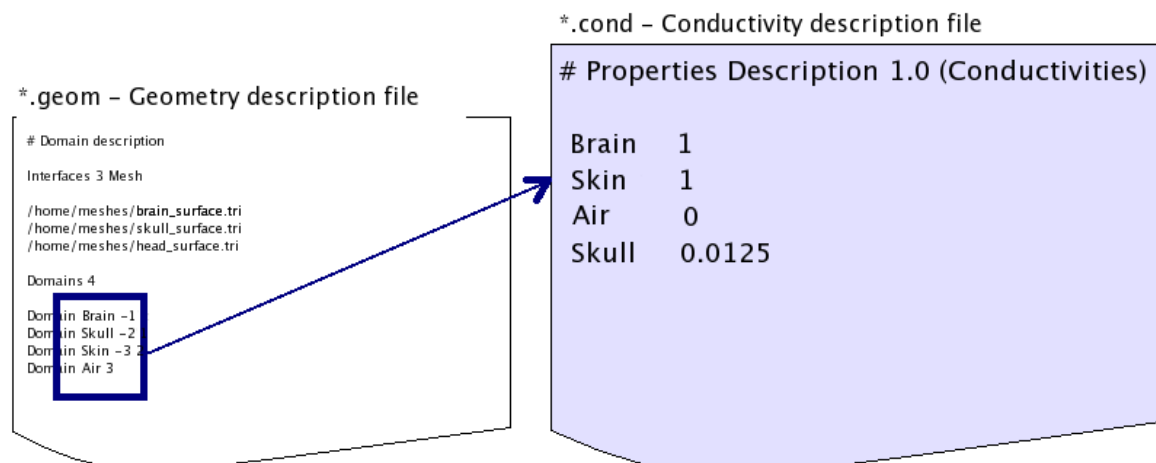
- *.tri : format TRI issu des premières versions de BrainVisa. Aussi lu par Anatomist.
- *.mesh : format MESH issu des versions 3.0.2 et plus de BrainVisa. Aussi lu par Anatomist.
- *.vtk : format de maillage VTK.

A.2 Fichier de description des conductivités

Ce fichier décrit les valeurs des conductivités associées aux différents domaines déclarés dans le fichier de description de la géométrie.

Extension du fichier : *.cond par convention.

Attention : Attention : bien respecter la casse entre les noms de domaines donnés dans le fichier de description de la géométrie et ceux notés dans le fichier de description des conductivités !



Note 3 : ce sont des conductivités relatives : l'air n'est pas conducteur, donc a une conductivité nulle, le cerveau est considéré comme étant entièrement conducteur tout comme la peau et le crâne a une conductivité égale à 1/80 de celle du cerveau.

A.3 Position et orientation des dipôles

Fichier texte. A une ligne donnée correspond un dipôle :

- les trois premiers nombres (séparés par un espace) sont les coordonnées cartésiennes de la position du dipôle,

- les trois derniers nombres (séparés par un espace) sont les coordonnées cartésiennes du vecteur normale donnant l'orientation du dipôle.

Extension de fichier : *.dip ou *.txt.

Dans l'exemple ci dessous, sont décrits 5 dipôles :

***.dip – Dipole position**

0 0 0.4250	0.7071 0 0.7071
0 0 0.6800	0.7071 0 0.7071
0 0 0.7650	0.7071 0 0.7071
0 0 0.8075	0.7071 0 0.7071
0 0 0.8415	0.7071 0 0.7071

Dipôle n°3 :
 - position : (0, 0, 0.7650)
 - orientation : (0.7071, 0, 0.7071)

Orientations : coordonnées cartésiennes des normales

Coordonnées cartésiennes des dipôles

Note 4 : les coordonnées sont prises dans le même repère que celui des maillages (repère de l'IRM en général)

A.4 Position (et orientation) des capteurs

Dans le cas de l'EEG, il s'agit d'un fichier de description de la position des capteurs en coordonnées cartésiennes et dans le même repère que celui des maillages (en général, le repère de l'IRM). A une ligne correspond les coordonnées d'un capteur séparées par un espace.

Dans le cas de la MEG, il s'agit d'un fichier de description de la position et de l'orientation des capteurs en coordonnées cartésiennes et dans le même repère que celui des maillages (en général, le repère de l'IRM). A une ligne correspond les coordonnées d'un capteur (séparées par un espace) suivies des coordonnées du vecteur orientation normé (séparés par un espace).

A.5 Activation des sources

Les fichiers d'activation des sources sont des fichiers texte. A une ligne correspond une source :

- dans le cas dipolaire, sont écrits sur la ième ligne les états d'activation du ième dipôle,
- dans le cas des sources distribuées, sont écrits sur la ième ligne les états d'activation du ième point du maillage des sources.

On trouve en colonne les temps auxquels sont associés les activations.

Exemple dans le cas dipolaire :

