OpenMEEG

Biomag 2010 Satellite:

Analysis toolboxes for MEG data

Maureen Clerc, Alexandre Gramfort,
Emmanuel Olivi, Théo Papadopoulo

INRIA Sophia Antipolis - Méditerranée

March 27, 2010

What is OpenMEEG ?

- A general package for quasistatic electromagnetics

$$\begin{aligned}\nabla\cdot(\sigma\nabla V) &= f \quad \text{in domain}\\ \sigma\nabla V\cdot\mathbf{n} &= g \quad \text{on domain boundary}\end{aligned}$$

- Boundary Element discretization: σ piecewise constant
- Especially targeted at EEG and MEG
- Especially targeted at Forward Problems

What is OpenMEEG ?

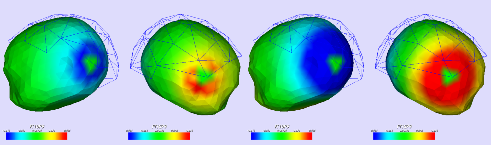
- A general package for quasistatic electromagnetics

$$\begin{aligned} \nabla \cdot (\sigma \nabla V) &= f \quad \text{in domain} \\ \sigma \nabla V \cdot \mathbf{n} &= g \quad \text{on domain boundary} \end{aligned}$$

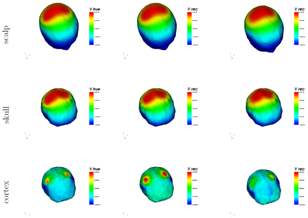
- Boundary Element discretization: σ piecewise constant
- Especially targeted at EEG and MEG
- Especially targeted at Forward Problems

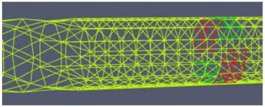
Our aim: make you want to use it !

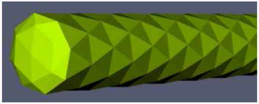
Not limited to EEG and MEG

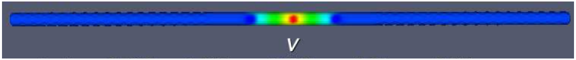


- Electrical Impedance Tomography
- Cortical Mapping
- ECoG
- Functional Electrical Stimulation
- Intracortical electrodes

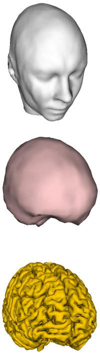








OpenMEEG may not solve all your needs



⊖ No geometry processing

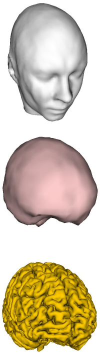
(Image segmentation, Mesh generation)

⊖ Does not handle anisotropic conductivities

⊖ Inverse problems: limited scope

⊖ No graphical functionalities

OpenMEEG may not solve all your needs



⊖ No geometry processing

(Image segmentation, Mesh generation)

⊖ Does not handle anisotropic conductivities

⊖ Inverse problems: limited scope

⊖ No graphical functionalities

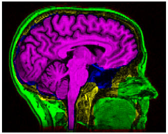
⇒ **Use OpenMEEG for what it's meant for ! (Forward Problem)**

Developers' goal:
make OpenMEEG easy to integrate into other packages.
(and goal of this Satellite: to foster communication !)

OpenMEEG facts

Methodology

- Galerkin Boundary element formulation
- Symmetric Boundary Element Method, involving both Potential V and normal current $\sigma\partial_{\mathbf{n}}V$ as unknowns



$$
\begin{array}{cc}
\mathrm{nH} &
\begin{bmatrix}
(\sigma_1+\sigma_2)\mathsf{N}_{11} & -2\mathsf{D}^*_{11} & -\sigma_2\mathsf{N}_{12} & \mathsf{D}^*_{12} & & & \\
-2\mathsf{D}_{11} & (\sigma_1^{-1}+\sigma_2^{-1})\mathsf{S}_{11} & \mathsf{D}_{12} & -\sigma_2^{-1}\mathsf{S}_{12} & & & \\
-\sigma_2\mathsf{N}_{21} & \mathsf{D}^*_{21} & (\sigma_2+\sigma_3)\mathsf{N}_{22} & -2\mathsf{D}^*_{22} & -\sigma_3\mathsf{N}_{23} & \mathsf{D}^*_{23} & \\
\mathsf{D}_{21} & -\sigma_2^{-1}\mathsf{S}_{21} & -2\mathsf{D}_{22} & (\sigma_2^{-1}+\sigma_3^{-1})\mathsf{S}_{22} & \mathsf{D}_{23} & -\sigma_3^{-1}\mathsf{S}_{23} & \\
 & & -\sigma_3\mathsf{N}_{32} & \mathsf{D}^*_{32} & (\sigma_3+\sigma_4)\mathsf{N}_{33} & -2\mathsf{D}^*_{33} & \cdots \\
 & & \mathsf{D}_{32} & -\sigma_3^{-1}\mathsf{S}_{32} & -2\mathsf{D}_{33} & (\sigma_3^{-1}+\sigma_4^{-1})\mathsf{S}_{33} & \cdots \\
 & & & & \vdots & \vdots & \ddots
\end{bmatrix}
\cdot
\begin{bmatrix}
\mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{v}_2 \\ \mathbf{p}_2 \\ \mathbf{v}_3 \\ \mathbf{p}_3 \\ \vdots
\end{bmatrix}
=
\begin{bmatrix}
\sigma_1^{-1}\mathsf{N}_{1sources} \\ \mathsf{D}_{1sources} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots
\end{bmatrix}
.\mathbf{s} \quad \mathrm{nvS}
\\
 & \begin{array}{ccccccc} \mathrm{nv1} & \mathrm{nf1} & \mathrm{nv2} & \mathrm{nf2} & \mathrm{nv3} & \mathrm{nf3} & \qquad\qquad \mathrm{nvS} \end{array}
\end{array}
$$

[Kybic, Clerc et al, IEEE T Medical Imaging, 2005: *A Common Formalism for the integral formulations of the forward EEG problem*]

OpenMEEG facts

Software

- C++ Source code, started in 2006
- Multiplatform (binaries, cmake, Subversioning)
- Matlab i/o
- Python wrapping, Fieldtrip integration, etc.
- Open-source, CeCiLL-B license
  - Similar to, and compatible with GPL
  - **Citation duty**

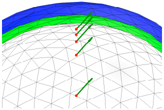
    [Gramfort, Papadopoulo, Olivi, Clerc, *OpenMEEG: opensource software for quasistatic bioelectromagnetics*, Biomedical Engineering Online, 2010, 9:45]

    [Kybic, Clerc et al., *A Common Formalism for the integral formulations of the forward EEG problem*, IEEE T Medical Imaging, 2005 Jan;24(1):12-28.]

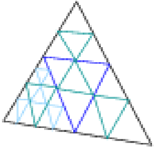
    More at: `http://openmeeg.gforge.inria.fr/`

Why use OpenMEEG ?

State of the art accuracy.



- more unknowns (V and $\sigma\partial_{\mathbf{n}}V$)
- domain-oriented
- adaptive numerical integration



Accuracy compared to analytical solutions using

$$RDM(g_{\mathrm{n}}, g_{\mathrm{a}}) = \left\| \frac{g_n}{\|g_n\|} - \frac{g_a}{\|g_a\|} \right\| \in [0,2] \ ,$$

and

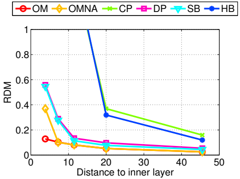
$$MAG(g_n, g_a) = \frac{\|g_n\|}{\|g_a\|} \ .$$

Accuracy comparison for EEG

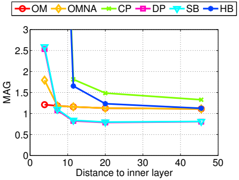
BEM solvers tested:

OM	OpenMEEG
OMNA	OM non adaptive
CP	Fieldtrip / CP
DP	Fieldtrip / Dipoli
SB	Simbio
HB	Helsinki BEM

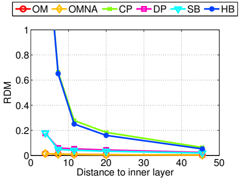
EEG
(regular meshes)



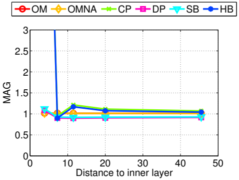
RDM 162 points/interface



MAG 162 points/interface



RDM 642 points/interface



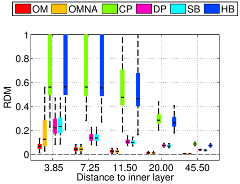
MAG 642 points/interface

Accuracy comparison for EEG

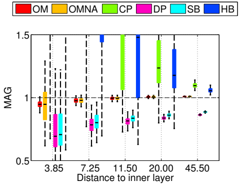
BEM solvers tested:

OM	OpenMEEG
OMNA	OM non adaptive
CP	Fieldtrip / CP
DP	Fieldtrip / Dipoli
SB	Simbio
HB	Helsinki BEM

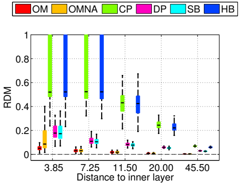
EEG
(100 random meshes)



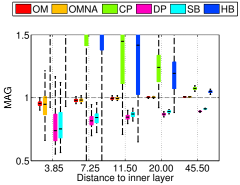
RDM 600 points/interface



MAG 600 points/interface



RDM 800 points/interface



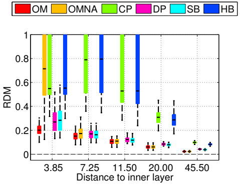
MAG 800 points/interface

Accuracy comparison for EEG

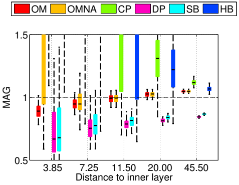
BEM solvers tested:

OM	OpenMEEG
OMNA	OM non adaptive
CP	Fieldtrip / CP
DP	Fieldtrip / Dipoli
SB	Simbio
HB	Helsinki BEM

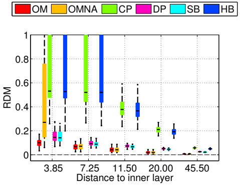
EEG
(100 random meshes)



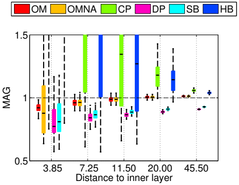
RDM with 1500 unknowns



MAG with 1500 unknowns



RDM with 3000 unknowns



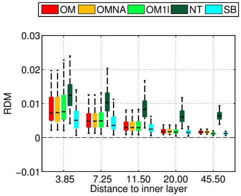
MAG with 3000 unknowns

Accuracy comparison for MEG

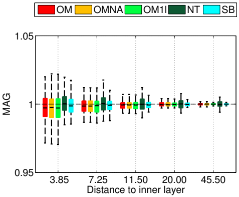
BEM solvers tested:

OM	OpenMEEG
OMNA	OM non adaptive
OM1l	OM one layer
NT	Nolte
SB	Simbio

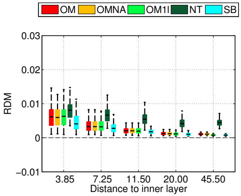
MEG, radial gradiometers
(100 random meshes)



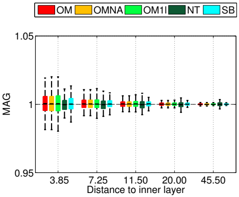
RDM with 600 unknowns



MAG with 600 unknowns



RDM 800 points/interface



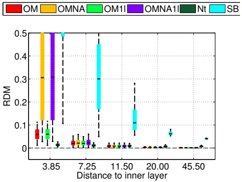
MAG 800 points/interface

Accuracy comparison for MEG

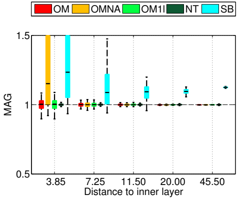
MEG, non-radial gradiometers
(100 random meshes)

BEM solvers tested:

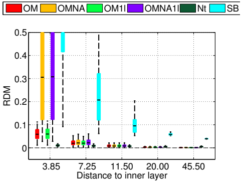
OM	OpenMEEG
OMNA	OM non adaptive
OM1l	OM one layer
NT	Nolte
SB	Simbio



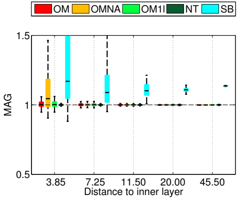
RDM with 600 unknowns



MAG with 600 unknowns



RDM 800 points/interface



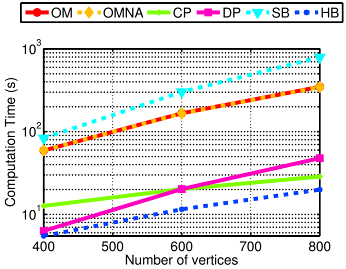
MAG 800 points/interface

Computation time for EEG

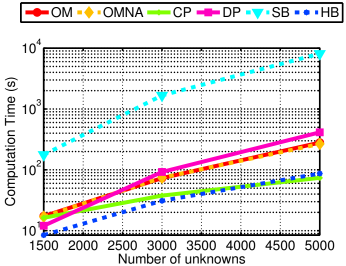
BEM solvers tested:

OM	OpenMEEG
OMNA	OM non adaptive
CP	Fieldtrip / CP
DP	Fieldtrip / Dipoli
SB	Simbio
HB	Helsinki BEM

as a function of number of vertices



as a function of number of unknowns



How to use OpenMEEG

First define:

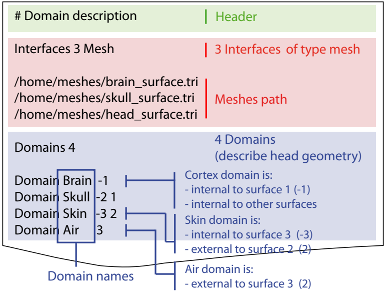
- **Head model**
  - closed nested meshes (any number)
  - conductivities
- **Sensor model**
  - EEG: positions projected to scalp surface
  - MEG: positions and weights (→ mag. or grad.)
- **Source model**
  - list of dipole positions and moments
  - surface describing dipole positions, normal to surface.

OpenMEEG commands available through

- **Command-line** interface
- **Python** scripts
- **Matlab/Fieldtrip** integration

Data structure

*.geom - Geometry description file



```
# Domain description

Interfaces 3 Mesh

/home/meshes/brain_surface.tri
/home/meshes/skull_surface.tri
/home/meshes/head_surface.tri

Domains 4

Domain Brain -1
Domain Skull -2 1
Domain Skin -3 2
Domain Air 3
```

Header

3 Interfaces of type mesh

Meshes path

4 Domains (describe head geometry)

Cortex domain is:
- internal to surface 1 (-1)
- internal to other surfaces

Skin domain is:
- internal to surface 3 (-3)
- external to surface 2 (2)

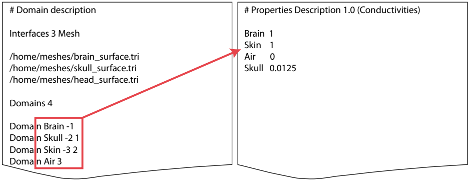
Air domain is:
- external to surface 3 (2)

Domain names

Sample geometry file

*.geom - Geometry description file

*.cond - Conductivity description file

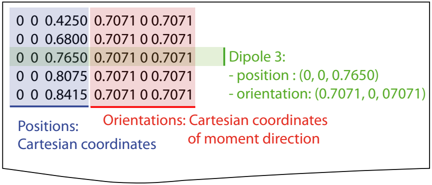


```
# Domain description

Interfaces 3 Mesh

/home/meshes/brain_surface.tri
/home/meshes/skull_surface.tri
/home/meshes/head_surface.tri

Domains 4

Domain Brain -1
Domain Skull -2 1
Domain Skin -3 2
Domain Air 3
```

```
# Properties Description 1.0 (Conductivities)

Brain  1
Skin   1
Air    0
Skull  0.0125
```

Sample conductivity file

Dipoles positions and orientations description file



```
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
```

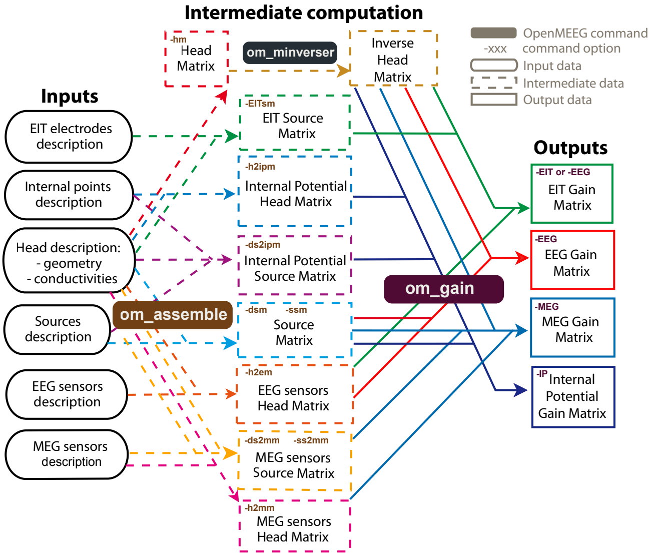
Dipole 3:
- position : (0, 0, 0.7650)
- orientation: (0.7071, 0, 07071)

Positions: Cartesian coordinates

Orientations: Cartesian coordinates of moment direction

Sample source file

Generating Lead-Fields



Intermediate computation

-hm Head Matrix

om_minverser

Inverse Head Matrix

OpenMEEG command

-xxx command option

Input data

Intermediate data

Output data

Inputs

EIT electrodes description

Internal points description

Head description:
- geometry
- conductivities

Sources description

EEG sensors description

MEG sensors description

om_assemble

-EITsm EIT Source Matrix

-h2ipm Internal Potential Head Matrix

-ds2ipm Internal Potential Source Matrix

-dsm -ssm Source Matrix

-h2em EEG sensors Head Matrix

-ds2mm -ss2mm MEG sensors Source Matrix

-h2mm MEG sensors Head Matrix

om_gain

Outputs

-EIT or -EEG EIT Gain Matrix

-EEG EEG Gain Matrix

-MEG MEG Gain Matrix

-IP Internal Potential Gain Matrix

Example scripts

EEG gain matrix:

```
om_assemble -HeadMat head.geom head.cond head.hm
om_assemble -SSM head.geom head.cond sources.tri head.ssm
om_assemble -h2em head.geom head.cond head.eegsensors head.h2em
om_minverser head.hm head.hm_inv
om_gain -EEG head.hm_inv head.ssm head.h2em head.gain
```

MEG gain matrix

```
om_assemble -HeadMat head.geom head.cond head.hm
om_assemble -DSM head.geom head.cond sources.dip head.dsm
om_assemble -h2mm head.geom head.cond head.squids head.h2mm
om_assemble -ds2mm sources.dip head.squids head.ds2mm
om_minverser head.hm head.hm_inv
om_gain -MEG head.hm_inv head.dsm head.h2mm head.ds2mm head.gain
```

OpenMEEG with Python (EEG leadfield)

```python
import openmeeg as om
# Load data
cond_file = ’om_demo.cond’
geom_file = ’om_demo.geom’
dipole_file = ’cortex.dip’
electrodes_file = ’eeg_electrodes.txt’

geom = om.Geometry()
geom.read(geom_file,cond_file)
dipoles = om.Matrix()
dipoles.load(dipole_file)
electrodes = om.Sensors()
electrodes.load(electrodes_file)

# Compute forward problem (Build Gain Matrices)
gauss_order = 3 # Numerical integration order

hm        = om.HeadMat(geom, gauss_order)
hminv     = hm.inverse()
dsm       = om.DipSourceMat(geom, dipoles, gauss_order)
h2em      = om.Head2EEGMat(geom, electrodes)
gain_eeg = om.GainEEG(hminv, dsm, h2em)
```

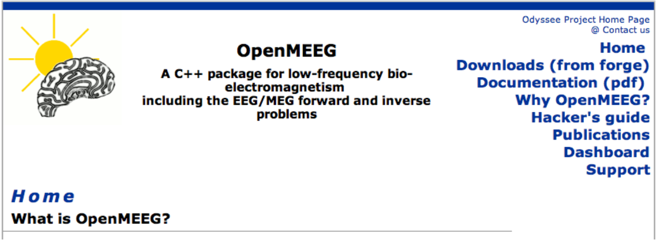
OpenMEEG with Fieldtrip (EEG leadfield)

```
%% The structure for the BEM volume conduction model
%% Each layer mesh is indexed by k
% vol.bnd(k).pnt : contains vertices for mesh "k"
% vol.bnd(k).tri : contains triangles for mesh "k"
%% Set the conductivities of each domain
% vol.cond       : contains conductivities

%% EEG electrodes
% sens.pnt        : contains locations of electrodes

%% Positions of the dipoles
% pos             : contains locations of dipoles

%% Compute the BEM
% choose BEM method (OpenMEEG, BEMCP or Dipoli)
cfg.method = 'openmeeg';
% Compute the BEM matrix
vol = ft_prepare_bemmodel(cfg, vol);
cfg.vol = vol;
cfg.grid.pos = pos;
cfg.elec = sens;
% Compute leadfield (no orientation constraint)
lf_openmeeg = ft_prepare_leadfield(cfg);
```

How to download OpenMEEG

Latest release (March 25, 2010): **OpenMEEG 2.0**

- From the forge:
  `http://openmeeg.gforge.inria.fr/`



Odyssee Project Home Page
@ Contact us

OpenMEEG

A C++ package for low-frequency bio-electromagnetism including the EEG/MEG forward and inverse problems

Home

What is OpenMEEG?

- anonymous download of source code:
  svn checkout svn://scm.gforge.inria.fr/svn/openmeeg
- or download binaries for your OS
  Supported OS: Windows, Linux, Mac OS X

For more info

- Talks (Wednesday):

W-4.1	*The symmetric BEM: bringing in more variables for better accuracy*
W-4.3	*The adjoint method for general EEG and MEG sensor-based lead field*

- Posters (Wednesday):

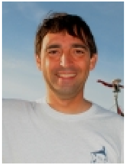
W-I T3-11	*Domain decomposition for coupling finite and boundary element methods*
W-I T3-6	*Evaluation of free BEM solvers for accurate M/EEG forward modeling*

- Subscribe to `openmeeg-info@lists.gforge.inria.fr` at `http://lists.gforge.inria.fr/mailman/listinfo/openmeeg-info`
- Contact the developers

Maureen Clerc

Alexandre Gramfort

Emmanuel Olivi



Théo Papadopoulo