Zeffiro user interface for electromagnetic brain imaging: a GPU accelerated FEM tool for forward and inverse computations in Matlab

Qin He\textsuperscript{a,b}, Atena Rezaei\textsuperscript{a}, Sampsa Pursiainen\textsuperscript{a}

\textsuperscript{a}Laboratory of Mathematics, Tampere University of Technology, P.O. Box 692, 33101 Tampere, Finland
\textsuperscript{b}Laboratory of Signal Processing, Tampere University of Technology, P.O. Box 692, 33101 Tampere, Finland

Abstract

Background and Objectives
This article introduces the Zeffiro interface (ZI) for brain imaging. Zeffiro is Italian for a ‘gentle breeze’. ZI is an attempt to provide a simple and accessible and multimodal open source platform for Finite Element Method based forward and inverse computations in the Matlab environment.

Methods
It allows one to (1) generate the finite element mesh for a given multi-layer head model, (2) form a basis for computing a lead field matrix and inverting a given set of measurements, and (3) to visualize the reconstructions on the finite element mesh. ZI uses graphics processing unit acceleration for each of the processing stages (1)–(3) to tackle the issue of the high computational cost. In its current configuration, ZI includes forward solvers for electro/magnetoencephalography and linearized electrical impedance tomography as well as a set of inverse solvers utilizing the hierarchical Bayesian model (HBM).

Results
GPU acceleration was found to be necessary, specifically, in the generation of the FE mesh and the evaluation of the LF matrix in order to achieve a feasibly short computation time. Computed examples of reconstructions and a guideline for parameter selection in HBM are given.

Conclusions
In this paper, we describe the mathematics behind the ZI, describe the principal operations and usage and introduce some central points for the developer.
perspective. We also give examples of computing times, and analyze numerically how the inversion parameters should be selected, when inverting brain activity with HBM.

Keywords: Matlab Interface, Brain Imaging, Computation tool, Finite Element Method (FEM), Forward and Inverse tool, Hierarchical Bayesian Method

1. Introduction

This article introduces the Zeffiro interface (ZI) [1] for electromagnetic brain imaging (Figure 1). The name Zeffiro is Italian for a 'gentle breeze'. ZI is aimed as an accessible and fast tool for utilizing the finite element method (FEM) in the electromagnetic (soft field) brain imaging applications in which the forward model follows from the Maxwell’s equations [2]. FEM is a volumetric modeling technique for solving equations in a bounded domain, such as the brain and the head [3, 4]. FEM allows discretizing the three-dimensional tissue parameter distributions, such as the conductivity, in an accurate way, taking into account anisotropic structures such as the fibrous white matter of the brain. In addition to the volumetric features, the finite element (FE) mesh can be accurately adapted to the complex internal surface layers of the head. The FEM can be applied to model an electromagnetic source within the grey matter (GM) of the brain in electro-/magnetoencephalography (EEG/MEG) [5, 6], where the neural activity of the brain is being sensed remotely by measuring the electromagnetic field evoked by the activity. Consequently, via FEM one can compute an EEG/MEG lead field matrix which maps a given candidate solution to the expected measurement outcome. Furthermore, the same quasi-static set of Maxwell’s equations that predicts the electric potential field of a neural source can be also approached via external magnetic field or current injections, where direct or alternating currents applied through electrodes act as the source of the electromagnetic field. Such applications include, for example, transcranial electric and magnetic stimulation (tES and tMS) [7, 8] and electrical impedance tomography (EIT) [9]. In tES, the brain activity is evoked through external stimuli, whereas in EIT the goal is to map the conductivity distribution within the computation domain. In contrast to EEG/MEG and tES/tMS, EIT constitutes a non-linear inverse problem
where generating a lead field (LF) matrix requires linearizing the forward model with respect to a given current distribution. FEM is a powerful tool in EIT, since it allows generating forward simulation for an arbitrary volumetric conductivity distribution, in contrast to the Boundary Element Method (BEM) [10] which the dominating method in the brain applications, but assumes that the conductivity is a layer-wise constant parameter. However, until recently, FEM computations have been considered as computationally expensive regarding the practical usage.

ZI is an attempt to provide a simple and accessible and multimodal open source platform for FEM based forward and inverse computations within a given brain and head geometry. It has been implemented for the Matlab (The MathWorks Inc.) environment. In a typical workflow, one first imports a set of triangular tissue surface layers and after that (1) generates the FE mesh for a given multi-layer head model, (2) forms a basis for computing a LF matrix and inverting a given set of measurements, and (3) visualizes the reconstructions on the FE mesh. To tackle the issue of the high computational cost, ZI uses graphics processing unit (GPU) acceleration for each of the processing stages (1)–(3). In its current configuration, it includes forward solvers for E/MEG and linearized EIT as well as a set of inverse solvers utilizing the hierarchical Bayesian model (HBM). The ZI platform and function library has been designed to be easily expandable and to allow implementing basically any FEM based linear or linearized forward simulation which can be formulated as a product between a LF matrix and a candidate solution vector.

In this paper, we describe the mathematics behind ZI, describe the principal operations and usage, concentrating especially on EEG/MEG, and introduce some central points for the developer perspective. We also show computed examples and analyze numerically how the inversion parameters should be selected, when inverting brain activity with HBM.

Figure 1: Zeffiro interface logo.
2. Methodology

2.1. Forward modeling

ZI employs the divergence conforming source model in which the primary current density belongs to the space $H(\text{div}) = \{ \vec{w} | \nabla \cdot \vec{w} \in L^2(\Omega) \}$. The model is current preserving, meaning that the source field does not have local sources or sinks. The total current density

$$\vec{J}^T = \vec{J}^P - \sigma \nabla u,$$  \hspace{1cm} (1)

i.e., the sum of the primary current and volume current terms, has zero divergence which implies that

$$\nabla \cdot (\sigma \nabla u) = \nabla \cdot \vec{J}^P.$$  \hspace{1cm} (2)

The boundary condition, $(\sigma \nabla u) \cdot \vec{n} = 0$, given on $\partial \Omega$, ensures that there is no current flow through the boundary of the domain. This equation multiplied by $v \in H^1(\Omega)$ and integrated by parts results in the weak form

$$\int_{\Omega} \nabla v \cdot (\sigma \nabla u) dV = - \int_{\Omega} v (\nabla \cdot \vec{J}^P) dV$$  \hspace{1cm} (3)

which has a unique solution, assuming that the ground level has been fixed \[11, 12\].

2.1.1. Discretization

The electrical potential field $u$ and the primary current density $\vec{J}^P$ are approximated as the finite sums $u_h = \sum_{i=1}^{N} z_i \psi_i$ and $\vec{J}^P_h = \sum_{j=1}^{K} x_j \vec{w}_j$, where $\psi_1, \psi_2, \ldots, \psi_N \in H^1(\Omega)$ and $\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_K \in H(\text{div}) \[13\]. The discretized fields $u_h$ and $\vec{J}^P_h$ correspond to vectors $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_K)$ which satisfy the linear system

$$\mathbf{A} \mathbf{z} = \mathbf{G} \mathbf{x}$$  \hspace{1cm} (4)

in which the system matrix $\mathbf{A}$ is of the form $\mathbf{A} \in \mathbb{R}^{(N \times N)}$, $A_{i,j} = \int_{\Omega} \nabla \psi_j \cdot (\sigma \nabla \psi_i) dV$ and the matrix $\mathbf{G}$ on the right hand side maps the primary current density on its divergence $\mathbf{G} \in \mathbb{R}^{(N \times K)}$, $G_{i,j} = \int_{\Omega} \psi_i (\nabla \cdot \vec{w}_j) dV$. 
2.1.2. LF matrix

The linear forward model is of the form

\[ y = Lx + n, \tag{5} \]

where \( L \) is the LF matrix and \( n \) is a noise vector. In EEG, the LF matrix is of the form \( L = RA^{-1}G \), where \( R \) is a restriction matrix. In the simplest case, \( R \) is a restriction to point values, i.e., the nodes in the FE mesh presenting the measurement point set. In addition to point electrodes, ZI allows one to use the lumped version of the complete electrode model (CEM) \([14]\) in which the entries of \( R \) follow from integral averages over the electrode patches. Computationally, the LF matrix can be obtained by evaluating a transfer matrix of the form \( T = A^{-1}R^T \). The LF and transfer matrix for MEG can be formed through the Biot-Savart integral formula for the total current field in a similar manner as for EEG as shown in \([15]\). The relationship between the EEG forward simulation, the CEM and linearized EIT has been briefly explained in Appendix A.

2.1.3. Implementation in ZI

ZI utilizes the second-order version of the H(div) source model \([13]\) in which both linear and quadratic basis functions are used for the primary current. In \([16, 13]\), this model has been shown to surpass the accuracy of the classical direct source modeling approaches based on the partial integration and St. Venant’s priciple and to be especially advantageous for thin cortices and inversion of distributions.

In the Cartesian orientation mode, each source position includes three sources, one for each Cartesian coordinate direction. The Cartesian orientations are obtained from the mesh-based ones using the Position Based Optimization (PBO) method \([17]\) with an adaptive \([18]\) 10-source stencil in which 4 face and 6 edge functions are applied for each tetrahedron containing a source \([13]\). Alternatively, the modeling can be restricted to the linear Whitney functions for which the 4-source stencil (4 face functions) can be applied to obtain the Cartesian set of orientations. Also, a plain set of Whitney functions can be used, that is, the LF matrix can be formed directly based on the mesh-based set of Whitney basis functions as suggested in \([18]\). Furthermore, the sources can be oriented normally to the cortex surface according to the physiological knowledge of neural activity \([19, 20]\). To accomplish this, one can either fix the (pre-inversion) source basis to the normal orientation,
or to find the positive or negative normal component of the neural activity after the inversion process via projection.

In ZI, the source positions are distributed randomly (uniformly) in the GM compartment and can be extended also in the white matter (WM), if necessary, e.g., if the deep brain structures are not included in GM.

2.2. HBM

The main inversion utility in ZI utilizes the HBM [21, 22] to find a focal reconstruction of the activity. In HBM, a posterior probability for \( x \) is defined via choosing the standard deviation (STD) of a Gaussian likelihood density, the hypermodel, i.e., the gamma (G) or inverse gamma (IG) hyperprior determining the actual prior, and the shape and scaling parameters \( \beta, \theta_0 \) of the hyperprior.

For a given measurement vector \( y \), the Bayes formula [23] for the posterior probability density of \( x \) is of the form

\[
p(x \mid y) = \frac{p(x) p(y \mid x)}{p(y)} \propto p(x) p(y \mid x),
\]

where \( p(x) \) is the prior density, and \( p(y \mid x) \) the likelihood function, following from the measurement noise model [24]. It is assumed that the noise term is a Gaussian zero mean random vector \( n = y - Lx \) with independent entries. Consequently, the likelihood is of the form \( p(y \mid x) \propto \exp\left(-\frac{1}{2\sigma^2}\|Lx - y\|^2\right) \), where \( \sigma \) is the noise standard deviation. In HBM, the prior can be expressed in the following hierarchical form

\[
p(x, h) \propto p(\theta) p(x \mid \theta),
\]

where \( \theta \) is the primary hyperparameter of the model. The conditional part \( p(x \mid \theta) \) of the prior is a zero mean Gaussian density, whose diagonal covariance matrix is predicted by the hyperprior \( p(\theta) \). The hyperprior is assumed to have a long-tailed density, implying that \( x \) is likely to be a sparse vector corresponding focal activity.

In ZI, it is either G or IG density [21], whose shape and scaling are controlled by the parameters \( \beta \) and \( \theta_0 \), respectively. ZI allows approximating both the posterior maximizer, i.e., maximum a posteriori (MAP), and conditional mean (CM) of the posterior density. The MAP estimate can be obtained via the iterative alternating sequential (IAS) method [21] for the whole domain at once, for a region of interest (ROI), or multiple different
resolution levels. Of these, the third strategy can be applied to enhance the visibility of the deep activity. The basic version of the IAS method can be found in Appendix B. The CM estimate can be computed within a ROI using the Markov chain Monte Carlo (MCMC) sampling.

2.3. Hardware requirements

ZI is principally designed to be used with a workstation or a high-end desktop computer with tens of gigabytes of RAM, a multi-core CPU and one or more GPUs. A project after generating the FE mesh and LF matrix is supposed consume several gigabytes of RAM. Achieving a one-millimeter accuracy in forward modeling will presumably require ≥ 64 GB of RAM during computations with the total project size, when saved in a file, being around 0.5–1 GB.

2.4. GPU operations

ZI utilizes a GPU to accelerate the FE mesh generation process, forward and inverse computations, source interpolation and decompositions, as well as to speed up visualizing the reconstructions and the computation geometry in 3D. This is vital in order to achieve a feasibly short, around one-hour, computation time for a one-millimeter FE mesh resolution which has been shown to be essential in order to achieve physiologically accurate inverse estimates [25]. A GPU is a parallel processing unit which is somewhat limited with respect to the RAM in comparison to the motherboard. It can handle computation intensive operations very effectively, while memory intensive operations need to be avoided. The operations related to forward and inverse computations can be accelerated due to the fast processing of matrix-vector products in a GPU, and the other operations are mainly based on the acceleration of find and sort operations and evaluating those as blocks instead of singular entries. A script example of a loop performed in blocks with an adjustable size for optimal GPU-specific performance is illustrated in Figure 2.

2.4.1. Forward simulation

In the Matlab environment, the most essential speed up gain is related to the sparse FE matrix-vector products which need to be evaluated iteratively in the forward simulation stage. The GPU parallelization of the forward simulation is especially important because Matlab currently handles the sparse matrix products in a single processor thread. Evaluating a single column
aux_vec_1 = (1/3) *(boundary_p(boundary_t(:,1,:),:) + boundary_p(boundary_t(:,2,:),:) + boundary_p(boundary_t(:,3,:),:))';
aux_vec_2 = boundary_p(boundary_t(:,2,:),:) - boundary_p(boundary_t(:,1,:),:);
aux_vec_3 = boundary_p(boundary_t(:,3,:),:) - boundary_p(boundary_t(:,1,:),:);
aux_vec_4 = cross(aux_vec_2,aux_vec_3)/2;

ind_vec = zeros(size(nodes,1),1);
ones_vec = ones(length(aux_vec_1),1);
ind_vec_aux = zeros(length_I,1);
nodes_aux = nodes(1,:);

par_num = evalin('base','zef.parallel_vectors');
bar_ind = ceil(length_I/(50*par_num));
i_ind = 0;

for i = 1 : par_num : length_I
i_ind = i_ind + 1;
block_ind = [i:min(i+par_num-1,length_I)];
aux_vec = nodes_aux(:,block_ind);
aux_vec = reshape(aux_vec,3,1,length(block_ind));
aux_vec_5 = aux_vec_1(:,ones(1,length(block_ind))) - aux_vec(:,ones_vec,:);
aux_vec_2 = sum(aux_vec_5.*aux_vec_4(:,ones(1,length(block_ind))));
aux_vec_3 = sqrt(sum(aux_vec_5.*aux_vec_5));
aux_vec_3 = (aux_vec_5.*aux_vec_3).*aux_vec_3;
aux_vec_6 = sum(aux_vec_2./aux_vec_3)/(4*pi);
ind_vec_aux(block_ind) = aux_vec_6(:);
end

Figure 2: A script example of finding elements inside a given closed triangulated tissue
layer surface based on determining the solid angle for a given point. The solid angle is zero
for an exterior point and 4π for a point inside. Here the for loop has been implemented
in blocks for fast processing in a GPU. The block size, i.e., the number of vectors that are
processed in parallel, can be controlled in order to obtain the best possible GPU-specific
trade-off between parallel processing and memory consumption.

t of the transfer matrix T necessitates solving a linear system of the form
At = rT, where r is a single row of the restriction matrix R. To find the solution,
ZI uses the preconditioned conjugate gradient (PCG) [26] method with
a lumped diagonal preconditioner LDP in which the each diagonal entry is
obtained as a row sum of the absolute entry values. LDP is an advantageous
preconditioner regarding the limited GPU memory. While LDP is not opti-
mal with respect to minimizing the iteration steps needed for convergence, it
enables establishing a fast forward solver due to a GPU’s high computational
capacity.
2.4.2. IAS MAP iteration

The IAS MAP estimation algorithm can be found in Appendix B. In the iteration, the most time consuming step is the third one (B.2) in which the size of the matrix to be inverted is determined by the length of the data vector. If a high number of time steps will need to be processed, the fastest inversion is obtained by evaluating the matrix-vector product of (B.2) in a GPU.

3. Interface structure and function

3.1. Data structures

When started, ZI creates a single data structure (struct) \texttt{zef} in Matlab’s base workspace. All the parameters and variables, such as the lead field matrix, measurement data and reconstruction, can be accessed via the \texttt{zef} structure. It (current project) can be also saved or loaded on a hard disk. Consequently, the LF and other variables will be easily accessible for the user. The export functions enable saving separate variables, for example, the LF matrix which, among other things, enables merging EEG and MEG LF matrices for parallel EEG/MEG. The main window contains the essential functions needed for mesh and LF generation as well as for visualizing the results. Additional adjustments can be made via the \textit{Miscellaneous options} dialog box (Edit menu) which allows controlling, e.g., smoothing, meshing and plotting parameters. The items of the \textit{Inverse tools} menu enable inversion of the data.

For external inverse procedure development, the list of most important variables is the following:

1. The lead field matrix \texttt{zef.L},
2. The measurement data \texttt{zef.measurements} (a matrix or a cell array with the number of rows and equal to that of \texttt{zef.L} and the time steps in the dataset, respectively),
3. source positions \texttt{zef.source.positions} (source positions corresponding to the columns of \texttt{zef.L} in the respective order),
4. Source directions \texttt{zef.source_directions} (an empty array, if Cartesian orientations are used, meaning that the source location/position for the columns of \texttt{zef.L} is given by the following pattern: position 1, xyz; position 2, xyz; position 3, xyz, ...),
5. The reconstruction is stored in \texttt{zef.reconstruction}. 

9
The reconstruction can be visualized using the interface. For creating a new inversion tools menu item, one needs a .fig file and two .m files for initializing and updating the parameters.

3.2. Workflow

![Diagram of workflow](image3.png)

Figure 3: A diagram for the workflow in ZI.

![Main window of ZI](image4.png)

Figure 4: The main window of ZI. Illustrated is the 5.2 node and 30 M element FE mesh generated using the population head model (PHM).

The workflow to produce a reconstruction with ZI is depicted as a diagram in Figure 3. The user first (1) imports the triangular surface model
of each tissue type by loading the .dat file of the points and triangles in the main window (Figure 4). In the current version, up to nine different tissue types can be included in a single model. The default conductivity values for WM, GM, cerebrospinal fluid (CSF), skull, and scalp are 0.14, 0.33, 0.0064, and 0.43 S/m, respectively, according to [27, 28]. The conductivity distribution produced by the mesh generator is currently isotropic. However, the function for computing the LF matrix is compatible also with an anisotropic conductivity tensor.

Then, (2) a uniform tetrahedral mesh is generated based on the surface segmentation. The process is robust and allows the tissue boundaries to intersect each other which is a typical situation with a real segmentation based on magnetic resonance imaging (MRI) data. The user can set the meshing priority (number) of each tissue compartment. A tetrahedron with nodes in several compartments will be associated with the one with the highest priority (smallest number). The ability to change the priority is essential in order to avoid holes in the thin skin, skull and GM compartments which is important regarding the modeling and inversion accuracy. The FE mesh can be also smoothed using the bi-Laplacian smoothing flow [29, 30].

After the FE mesh generation phase, (3) the LF matrix can be computed for a given number of source positions. Additionally, in order to enable visualizing the reconstructions, (4) an interpolation process connecting the
source positions and the FE mesh nodes will need to be run. Finally, after importing data, (5) the actual reconstruction can be computed and visualized with the inversion tools and visualization functions included in ZI.

3.3. IAS MAP estimation process

The window of the IAS MAP estimation tool is shown in Figure 5. A reconstruction can be obtained as follows:

1. Import data. This can be done either via the Import menu or by directly associating zef.measurements with a dataset. The number of rows in the measurement array should coincide with that of the sensors. The number of columns is considered as the number of the recorded time steps. If the dataset is a cell array, then the cell entries are considered as different segments.

2. Open the IAS MAP Estimation dialog box.

3. Set the parameters (Sampling frequency, Time interval start, Low-cut frequency, High-cut frequency, FFT time window and Time step) to match with the dataset and the investigated frequency range. Choose the desired Data segment.

4. Set the Hyperprior (either G or IG), i.e., the outlier model (hyperprior) for the prior variance. Choose the Shape and Scaling parameter for the hyperprior. The former one of these controls the shape of the hyperprior (the strength of the outliers) and the latter one sets the initial prior variance. If the shape parameter $\beta$ is 1.5 and the G hyperprior is used, the reconstructions will correspond to the classical minimum norm estimate [6] and minimum current estimate [31], when the number of the iteration steps is 1 and $> 1$, respectively. Similarly, with IG there is a correspondence to the minimum support estimate [32].

5. Set the Likelihood STD to match the estimated noise level. The value is relative to the Data normalization.

6. Choose the desired number of iterations (the more steps the more focal solution).

7. Press start. The reconstruction will be computed for each time step in the dataset.

8. Visualize the reconstruction either on the surface segmentation or in the volumetric mesh. The type of the visualization can be chosen in the Visualization drop-down menu of the main window (Figure 4). The
Snapshot/Movie button lets you print the image/time-lapse of the reconstruction into a file. The video codec to be applied can be defined in the file zeffiro_interface.ini.

3.4. Computed examples

| ID | Method | $\theta_0$ | Hyperp. | Noise (%) |
|----|--------|------------|---------|-----------|
| (1) | EEG | 1E-5 | G | 2 |
| (2) | MEG | | | |
| (3) | EEG | 1E-5 | G | 5 |
| (4) | MEG | | | |
| (5) | EEG | 1E-5 | IG | 2 |
| (6) | MEG | | | |
| (7) | EEG | 1E-5 | IG | 5 |
| (8) | MEG | | | |
| (9) | EEG | 1E-9 | G | 2 |
| (10) | MEG | | | |
| (11) | EEG | 1E-9 | G | 5 |
| (12) | MEG | | | |
| (13) | EEG | 1E-9 | IG | 2 |
| (14) | MEG | | | |
| (15) | EEG | 1E-9 | IG | 5 |
| (16) | MEG | | | |

Table 1: The parameters for the reconstructions computed in the EEG/MEG source localization experiment.

3.5. Inversion of distributions

To demonstrate the forward and inversion modeling capability of ZI, a 0.85 mm resolution FE mesh was generated for the WM, GM, CSF, skull, and scalp surfaces of the population head model\footnote{https://itis.swiss/virtual-population/regional-human-models/phm-repository/} using ZI’s default conductivity value set. A total of 72 ring electrodes with 1 kOhm impedance and outer and inner diameter of 10 mm were attached on the scalp.

The tetrahedral FE mesh obtained is shown in Figure 6. The total number of the elements and nodes in the FE mesh generated was 30 M and 5.2 M,
### Table 2: The hypermodel and scaling parameter corresponding to the best median accuracy obtained in the EEG/MEG localization of a superficial and deep source.

| Case | Noise | Source | Meas. | Hyp. | Scaling | Value |
|------|-------|--------|-------|------|---------|-------|
| EEG  | 2 %   | Superf. | Pos.  | G    | 1E-5    | 4 mm  |
|      |       | Superf. | Angle | IG   | 1E-9    | 31 deg|
|      |       | Deep    | Pos.  | IG   | 1E-9    | 11 mm |
|      |       | Deep    | Angle | IG   | 1E-9    | 8 deg |
|      | 5 %   | Superf. | Pos.  | G    | 1E-5    | 6 mm  |
|      |       | Superf. | Angle | IG   | 1E-9    | 38 deg|
|      |       | Deep    | Pos.  | IG   | 1E-9    | 10 deg|
|      |       | Deep    | Angle | IG   | 1E-9    | 8 deg |
| MEG  | 2 %   | Superf. | Pos.  | G    | 1E-9    | 2 mm  |
|      |       | Superf. | Angle | IG   | 1E-9    | 4 deg |
|      |       | Deep    | Pos.  | G    | 1E-9    | 14 mm |
|      |       | Deep    | Angle | IG   | 1E-9    | 14 deg|
|      | 5 %   | Superf. | Pos.  | G    | 1E-9    | 3 mm  |
|      |       | Superf. | Angle | IG   | 1E-9    | 4 deg |
|      |       | Deep    | Pos.  | G    | 1E-5    | 12 mm |
|      |       | Deep    | Angle | G    | 1E-9    | 18 deg|

Figure 6: A comparison of between the surface model (left) of the brain and volumetric FE mesh (right) generated for the PHM using the meshing resolution of 0.85mm.

respectively. Using the FE mesh, an LF matrix was produced for 10000 source locations and Cartesian source orientations, and the sources were connected to FE mesh nodes via interpolation.
Figure 7: Six examples of how a (multiresolution IAS) reconstruction can be visualized in ZI. The distribution can be visualized interpolated on the surface of the cortex or in the volume. It can be constrained to the positive or negative normal direction, or restricted to the tangential one. Here, normal and tangential refer to the orientation with respect to the visualized surface (either cut or not). In this case, two synthetic sources are simultaneously active: one (P20/N20) in area 3b in the posterior wall of the central sulcus and a deep (P14/N14) upward-pointed source. The first one of these is visible in the sulcus wall in normally constrained distribution and a deep component corresponding to the second one can be observed in the tangentially restricted volumetric cut.
3.6. Hypermodel and parameter selection

To find a guideline for selecting the hypermodel and scaling parameter for the IAS MAP estimation process in EEG/MEG, we compared the localization of a simultaneously active pair of deep and superficial 10 nA/m sources. The reconstruction was found as a center of mass of the primary current distribution within two 30 mm ROIs centered at the source locations. The accuracy was measured via evaluating the positioning and orientation (angle) difference with respect to the exact sources.

As the computation domain we used a six-compartment (WM, GM, CSF, Compact Skull, Spongious Skull, Skin) head model of a 49-year old male subject with default conductivity values. For the spongious part of the skull 0.028 S/m was selected [28]. The EEG LF matrix was formed for a cap of 72 electrodes and the one for MEG was based on a helmet of 271 radial magnetometers. As the scaling parameter, we tested both 1E-5 and 1E-9. The shape parameter $\beta$ was set to be 1.5 in each case. The relative STD of the (white) noise term was either 2 or 5 %. The details of the reconstructions computed can be found in Table 3.4. Each reconstruction was evaluated for 50 different realizations of the noise vector. The resulting samples were analyzed via box-plots.

4. Results

4.1. Inversion of distributions

The total time of the mesh generation, LF matrix evaluation and interpolation processes were 21, 39 and 3.5 min, respectively, with NVIDIA Quadro P6000 [31] GPU. Using GPU was also found to be necessary to obtain a
feasible computation time as GPU accelerated these routines by more than a factor of ten.

An example a multiresolution IAS reconstruction obtained using synthetic data with 3 % relative (white) noise STD corresponding to a simultaneously active dipolar thalamic (e.g. N14) and somatosensory (e.g. N20/P20) component is illustrated in Figure 7. Since the thalamus is not included in the GM compartment of PHM, the sources positions cover both GM and WM. A reconstruction for somatosensory activity obtained when restricting the

Figure 9: Examples of the center of mass (red) found for the exact (cyan) synthetic deep and superficial source in the case of the synthetic EEG data.
Figure 10: Examples of the center of mass (red) found for the exact (cyan) synthetic deep and superficial source in the case of the synthetic MEG data.

Sources into the GM can be found in Figure 8 which includes, additionally, also a IAS MAP reconstruction of a conductivity anomaly produced with ZI using a linearized EIT model and synthetic data (Appendix A) within a spherical geometry.
Figure 11: The distributions for position and angle differences in the source localization experiments in the case of synthetic EEG data. In each image the box-plot for the deep source is on the left and for the superficial one on the right.
Figure 12: The distributions for position and angle differences in the source localization experiments in the case of synthetic MEG data. In each image the box-plot for the deep source is on the left and for the superficial one on the right.
4.2. Hypermodel and parameter selection

Figures 9 and 10 show comparisons between the exact and reconstructed sources for EEG and MEG. The box-plot analysis of the reconstruction accuracy can be found in Figures 11 and 12, respectively. Table 3.4 lists the hypermodel and scaling parameter corresponding to the smallest obtained median position and angle difference for the deep and superficial position. The obvious tendency is that the superficial source can be localized more accurately than the deep one. In EEG, the best results were obtained with the value 1E-5 and 1E-9 for hyperpriors G and IG, respectively. Regarding the superficial source, G was observed to yield the superior results for the position, whereas IG was found to be advantageous with respect to the angle. For the deep part, IG with \( \theta_0 = 1E-9 \) was observed to be preferable regarding both quantities. In MEG, G yielded the most accurate position estimate in each case. IG performed appropriately for angle in the superficial part, whereas for deep part, the best orientation accuracy was obtained with IG and G for the noise levels 2 \% and 5 \%, respectively. The general observation for both EEG and MEG was that the low scaling parameter value \( \theta_0 = 1E-9 \) was preferable for localizing the deep source, especially, with the lower 2 \% noise level.

5. Discussion

This paper has introduced the Zeffiro interface (ZI) [1] for brain imaging. ZI provides a GPU accelerated approach to using the FEM [3, 35] in applications involving forward modeling and inversion of electromagnetic fields [2]. The central aim of ZI is to provide an accessible and fast modeling tool for the FEM. When aided by a GPU, ZI allows one to invert a given set of EEG data for a physiologically accurate [25] sub-one-millimeter volumetric multi-compartment headmodel within a reasonable one hour’s time.

GPU acceleration was found to be necessary, specifically, in the generation of the FE mesh and the evaluation of the LF matrix in order to achieve a feasibly short computation time. This is especially important, since Matlab does not currently parallelize the sparse matrix operations in a CPU which is why the performance difference between CPU and GPU computations, both applicable in ZI, is especially pronounced. A standard laptop GPU was observed to be generally not sufficient for high-resolution forward simulation regarding its speed and memory, whereas a workstation GPU performed decently. The workstation GPUs utilized in this study (Nvidia
Quadro P6000, P4000 and M40000 \cite{34}) had at least 8 GB of RAM. The minimum requirement is 2 GB based on the maximal memory consumption during the computations.

Consequently, for generating the FE mesh and the LF matrix, the office is suggested to be equipped with at least one workstation or powerful desktop as mentioned above. A laptop was, however, found to be sufficient for the inversion process. An important property, in this regard, is the ability to save the \texttt{zef} struct, including all the variables, the LF matrix, and FE mesh of ZI, as a single .mat file, so that the project can be shared and utilized for further calculation easily. Thereby, multiple neuroscientists will be able to use and make further calculation or analyze on the single file on different computers for various purposes. Handling the variables as a single struct data type within Matlab was also found to be advantageous from the viewpoint of having access to the all the variables not only via the ZI’s front-end, but also using the command prompt which allows the scientists and developers to use external functions easily together with ZI.

The central advantage of ZI’s tetrahedral mesh generator is that, unlike many widely used software, e.g., TetGen \cite{36} and Netgen \cite{37}, it allows the tissue boundaries to intersect each other without collapsing. Whereas an ordinary FE mesh generator attempts to optimize the elements with respect to both volumetric and surface geometry, and will, therefore, easily crash when meshing a set of thin or intersecting surface layers, ZI utilizes a robust uniform mesh structure and priority based compartment labeling which enables, via control of the priority parameter, optimizing the layer structures also after the actual meshing process. This is essential in practice, since the tissue segmentation routines utilizing MRI images often do not distinguish the different compartments smoothly.

The present computed examples show that ZI enables robust inversion of the primary current distribution, and that the resulting reconstruction can be visualized as a volumetric field, restricting the activity according the physiological normal constraint \cite{19, 20} if needed. Since the current is inverted and visualized using the FE mesh, the volumetric structure of the domain has a strong influence on the final reconstruction. For example, if the deep structures of the head model are absent which was the the case with the PHM \cite{33} used this study. In ZI, the solution is to extend the sources to white matter which works appropriately based on the present results. Differences between source spaces extended over GM and WM, especially, regarding localization of deep sources will be an important future topic.
As shown by the results, the hypermodel and scaling parameter $\theta_0$ (initial prior variance) selection for the IAS MAP estimation process will affect on the source localization accuracy in EEG/MEG. An advantageous choice for depth localization seems to be IG coupled with the value $\theta_0 = 1E-9$. For the superficial areas, G with a slightly larger value, e.g., $\theta_0 = 1E-5$ appears beneficial. The guidelines found in this study are in line with previous findings [21] and might be optimized later on in an application specific context.

The preliminary inversion results obtained with EIT, suggest that ZI can also be extended to non-linear problems and inversion of scalar-valued fields. Since the CEM and the ability to inject current patterns into the domain, which is necessary in EIT [14, 9], have already been implemented in ZI, it would be relatively easy to add in solvers for other applications involving current patterns, such as tES.

Currently, Zeffiro is the only open source Matlab toolbox to use GPU assisted FEM for EEG/MEG, but there are many other software tools aimed at EEG/MEG neuroimaging. For instance, the Duneuro [38, 39] and SimBio [40] are open source libraries with similar functions as Zeffiro but utilizing C++ language. Brainstorm [41] and Fieldtrip [42] are both implemented for Matlab as well. However, the core forward modeling approach of Brainstorm is the BEM method, and the Fieldtrip does not have an advanced forward and inverse modeling functions. Moreover, none of these are currently capable of advanced FEM or GPU computations. The MNE-Python toolbox [43] is the leading option for Python which also allows using a GPU. However, MNE Python is also limited to the BEM forward simulation. We emphasize that the ability to utilize the FEM in ZI is essential, to allow extending it to other inverse imaging modalities.

In the future work, we will investigate further the forward and inversion accuracy that is obtainable with ZI. An interesting direction will be to examine improved techniques to find a reconstruction with the HBM approach, e.g., the MCMC method restricted into a ROI. Implementing more inverse modalities for ZI and developing machine learning based analysis methods for it will also be potential directions. The data structure of the current Zeffiro can also be enhanced so as to make the computation faster and occupying less memory.
Acknowledgments

QH, AR and SP were supported by the Academy of Finland Centre of Excellence in Inverse Modelling and Imaging 2018–2025.

Appendix A. Complete electrode model

The quasi-static electric potential equation (2) can be equipped with the following (lumped) CEM boundary conditions [14]:

\[ \sigma \nabla u \cdot \vec{n} |_{\partial \Omega \cup e_{\ell}} = 0, \quad (A.1) \]
\[ \int_{e_{\ell}} \sigma \nabla u \cdot \vec{n} \, dS = I_{\ell} \quad (A.2) \]
\[ (u + Z_{\ell} A_{\ell} \sigma \nabla u \cdot \vec{n}) |_{e_{\ell}} = U_{\ell}, \quad (A.3) \]

for \( k = 1, 2, \ldots, L \). According to the first one the normal current \( \sigma \nabla u \cdot \vec{n} \) on \( \partial \Omega \) can flow out of or into the domain only through electrodes. The second one sets the net current flowing through each electrode is \( I_{\ell} \), and the third one corresponds to the potential jump on the skin-electrode contact boundary. The voltage of the \( k \)-th electrode is denoted by \( U_{\ell} \). \( Z_{\ell} \) is the average contact impedance or resistance and \( A_{\ell} \) is the contact area of the \( \ell \)-th electrode.

An additional condition is the the equation \( \sum_{\ell=1}^{L} I_{\ell} = 0 \), meaning that the subject is grounded appropriately so that there is no current flowing out of the head through the subject’s neck.

Integrating by parts yields the CEM weak form for the potential field is given by [14]:

\[ -\int_{\Omega} (\nabla \cdot \vec{J}^p) v \, dV = \int_{\Omega} \sigma \nabla u \cdot \nabla v \, dV \]
\[ + \sum_{\ell=1}^{L} \frac{1}{Z_{\ell} A_{\ell}} \int_{e_{\ell}} u \, v \, dS \]
\[ - \sum_{\ell=1}^{L} \frac{1}{Z_{\ell} A_{\ell}^2} \int_{e_{\ell}} u \, dS \int_{e_{\ell}} v \, dS \]
\[ - \sum_{\ell=1}^{L} Z_{\ell} I_{\ell} \quad (A.4) \]
If the divergence of \( \vec{J}^p \) is square integrable, i.e., if \( \vec{J}^p \in \{ \vec{w} | \nabla \cdot \vec{w} \in L^2(\Omega) \} \), then (A.4) has a unique solution \( u \in H^1(\Omega) = \{ w \in L^2(\Omega) : \partial w/\partial x_i \in L^2(\Omega), i = 1, 2, 3 \} \) satisfying (A.4) for all \( v \in H^1(\Omega) \).

A straightforward calculation shows that the weak form (A.4) within the subspace of the basis functions is given by

\[
\left( \begin{array}{cc}
\tilde{A} & -B \\
-B^T & C
\end{array} \right) \left( \begin{array}{c}
z \\
v
\end{array} \right) = \left( \begin{array}{c}
-Gx \\
I
\end{array} \right). 
\] (A.5)

Matrix \( \tilde{A} \) is of the form

\[
\tilde{a}_{i,j} = \int_{\Omega} \sigma \nabla \psi_i \cdot \nabla \psi_j dV + \sum_{\ell=1}^{L} \frac{1}{Z_{\ell} \tilde{A}_{\ell}} \int_{e_{\ell}} \psi_i \psi_j dS, 
\] (A.6)

where additionally \( \tilde{a}_{i',i} = 1 \) and \( \tilde{a}_{i',j} = 0 \) for \( j \neq i' \) for a single nodal basis function \( \psi'_i \) attaining its maximum on the part of the boundary not covered by the electrodes. The entries of \( B \in \mathbb{R}^{N \times L} \), \( C \in \mathbb{R}^{L \times L} \) and \( G \in \mathbb{R}^{N \times M} \) are given by

\[
b_{i,\ell} = \frac{1}{Z_{\ell}} \int_{e_{\ell}} \psi_i dS, 
\] (A.7)

\[
c_{\ell,\ell} = \frac{1}{Z_{\ell}} \int_{e_{\ell}} dS \quad \text{and} \quad c_{i,\ell} = 0, \quad \text{if} \quad i \neq \ell, 
\] (A.8)

The current vector \( I = (I_1, I_2, \ldots, I_L) \) is nonzero, if the electrodes are actively injecting currents. Electrode voltages \( y = (U_1, U_2, \ldots, U_L) \) predicted by (A.5) can be obtained via \( y = Rv \), in which multiplication by the real \( L \times L \) matrix \( R \), \( r_{j,j} = 1 - 1/L \) for \( j = 1, 2, \ldots, L \), and \( r_{i,j} = -1/L \) for \( i \neq j \), fixes the sum of the electrode voltages to zero.

**Appendix A.1. EEG lead field**

In EEG, the current vector \( I \) is zero, as the electrodes only measure the voltage on the skin. Thus, vector \( v \) can be explicitly solved from (A.5) which leads to

\[
y = R(B^T \tilde{A}^{-1} B - C)^{-1}B^T \tilde{A}^{-1} Gx, 
\] (A.9)

and, further, to the following EEG LF matrix of for the CEM

\[
L_{\text{CEM}} = R(B^T \tilde{A}^{-1} B - C)^{-1}B^T \tilde{A}^{-1} G. 
\] (A.10)

To determine \( L_{\text{CEM}} \) efficiently, the matrix \( H = B^T \tilde{A}^{-1} \) is first computed using iterative solvers for \( AH^T = B^T \).
Appendix A.2. Linearized EIT lead field

In EIT, the primary current density can be assumed to be zero, i.e., \( x = 0 \), as the magnitude of the injected currents is far superior to the brain activity. The unknown of the EIT inverse problem is the conductivity distribution \( \sigma \).

The voltage measurements \( y = Rv \) generated by the current injections \( I \) are used as the data. The forward model that follows is given by

\[
y = RM^{-1}I,
\]

where \( M = (C - B^T A^{-1}B) \). The conductivity distribution is assumed to be piecewise (elementwise) constant, i.e., of the form \( \sigma = \sum_{m=1}^{M} s_m \chi_m \), where \( \chi_m \) is the indicator function of the element \( m \) in the FE mesh. The LF for linearized EIT can be derived by differentiating both sides of the equation \( Mz = I \) as follows:

\[
0 = (C - B^T \frac{\partial}{\partial s_m} A^{-1}B)z + M \frac{\partial}{\partial s_m} z.
\]

Moreover, a straightforward differentiation of the equation \( AA^{-1} = I \) shows that

\[
\frac{\partial}{\partial s_m} (AA^{-1}) = \frac{\partial A}{\partial s_m} A^{-1} + A \frac{\partial A^{-1}}{\partial s_m} = 0,
\]

and, further, that \( \frac{\partial A^{-1}}{\partial s_m} = A^{-1} \frac{\partial A}{\partial s_m} A^{-1} \). Taking into account that \( \frac{\partial y}{\partial s_m} = R \frac{\partial z}{\partial s_m} \), the linearized lead field can be written as

\[
\frac{\partial y}{\partial s_m} = RM^{-1}(C - B^T \frac{\partial A}{\partial s_m} A^{-1}B)z.
\]

Thus, the differential is of the form

\[
\frac{\partial y}{\partial s_m} = RM^{-1}(C - B^T A^{-1}A \frac{\partial A}{\partial s_m} A^{-1}B)M^{-1}I
\]

and the linearized forward model is of the form \( y \approx Lx + y^{(bg)} \).

Appendix B. IAS MAP inversion

The iterative alternating sequential (IAS) inversion approach [45, 21, 46] to find a maximum a posteriori estimate for the posterior density as is given by:
1. Choose parameters $\beta$ and $\theta_0$, and set the initial guess as follows: $\theta^{(0)} = (\theta_0, \theta_0, \ldots, \theta_0)$ and $k = 1$.
2. Find $x^{(k)} = \arg \max_x p(x \mid y, \theta^{(k-1)})$.
3. Find $\theta^{(k)} = \arg \max_{\theta} p(\theta \mid y, x^{(k)})$.
4. If $k$ is less than the total number of iterations chosen by the user, then go to 2. and set $k = k + 1$, else set $x_{MAP} = x^{(k)}$.

That is, the optimization is done alternatingly for the unknown vector $x$ and the hyperparameter $\theta$. The algorithm can be, further, written as

1. Set $k = 0$ and $\theta^{(0)} = (\theta_0, \theta_0, \ldots, \theta_0)$.
2. Set $L^{(k)} = LD_{\theta^{(k)}}^{1/2}$ with
   \[
   D_{\theta^{(k)}}^{1/2} = \text{diag}\left(\sqrt{|\theta_1^{(k)}|}, \sqrt{|\theta_2^{(k)}|}, \ldots, \sqrt{|\theta_n^{(k)}|}\right). \tag{B.1}
   \]
3. Evaluate
   \[
   x^{(k+1)} = D_{\theta^{(k)}}^{1/2} L^{(k)} (L^{(k)} L^{(k)T} \sigma^2 I + \sigma^2 I)^{-1} y. \tag{B.2}
   \]
4. Update the hyperparameter based on the hypermodel.
   - If the hypermodel is G, set
     \[
     \theta_i = \frac{1}{2} \theta_0 \left( \eta + \sqrt{\eta^2 + 2x_i^{(k)} / \theta_0} \right) \tag{B.3}
     \]
     with $\eta = \beta - 3/2, i = 1, 2, \ldots, n$.
   - If IG is used, set
     \[
     \theta_i^{(k+1)} = \left( \theta_0 + \frac{x_i^{(k)} / 2}{\kappa} \right) / \kappa \tag{B.4}
     \]
     with $\kappa = \beta + 3/2, i = 1, 2, \ldots, n$.
5. Set $k = k + 1$ and go back to 2., if $k$ is less than the total number of iterations defined by the user.

References

[1] Zeffiro forward and inverse interface for brain imaging, [https://github.com/sampsapursiainen/zeffiro_interface](https://github.com/sampsapursiainen/zeffiro_interface), 2018. Accessed: 2018-11-18.
[2] P. Monk, Finite Element Methods for Maxwell’s Equations, Clarendon Press, Oxford, UK, 2003.

[3] D. Braess, Finite Elements, Cambridge University Press, Cambridge, 2001.

[4] J. de Munck, C. H. Wolters, M. Clerc, EEG & MEG forward modeling., in: R. Brette, A. Destexhe (Eds.), Handbook of Neural Activity Measurement, Cambridge University Press, New York, 2012.

[5] E. Niedermeyer, F. L. da Silva, Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, Fifth Edition, Lippincott Williams & Wilkins, Philadelphia, 2004.

[6] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuttila, O. V. Lounasmaa, Magnetoecephalography — theory, instrumentation, and applications to invasive studies of the working human brain, Reviews of Modern Physics 65 (1993) 413–498.

[7] C. S. Herrmann, S. Rach, T. Neuling, D. Strüber, Transcranial alternating current stimulation: a review of the underlying mechanisms and modulation of cognitive processes, Frontiers in human neuroscience 7 (2013) 279.

[8] M. Hallett, Transcranial magnetic stimulation and the human brain, Nature 406 (2000) 147.

[9] M. Cheney, D. Isaacson, J. C. Newell, Electrical impedance tomography, SIAM review 41 (1999) 85–101.

[10] B. He, T. Musha, Y. Okamoto, S. Homma, Y. Nakajima, T. Sato, Electric dipole tracing in the brain by means of the boundary element method and its accuracy, IEEE Transactions on Biomedical Engineering (1987) 406–414.

[11] L. Evans, Partial Differential Equations, Graduate studies in mathematics, American Mathematical Society, 1998.

[12] F. Drechsler, C. Wolters, T. Dierkes, H. Si, L. Grasedyck, A full subtraction approach for finite element method based source analysis using constrained delaunay tetrahedralisation., NeuroImage 46 (2009) 1055–1065.
[13] S. Pursiainen, J. Vorwerk, C. Wolters, Electroencephalography (EEG) forward modeling via H(div) finite element sources with focal interpolation, Physics in Medicine and Biology 61 (2016) 8502–8520.

[14] K.-S. Cheng, D. Isaacson, J. Newell, D. G. Gisser, Electrode models for electric current computed tomography, IEEE Transactions on Biomedical Engineering 36 (1989) 918–924.

[15] S. Pursiainen, F. Lucka, C. H. Wolters, Complete electrode model in EEG: relationship and differences to the point electrode model, Phys. Med. Biol. 57 (2012) 999–1017.

[16] T. Miinalainen, A. Rezaei, D. Us, A. Nüßing, C. Engwer, C. H. Wolters, S. Pursiainen, A realistic, accurate and fast source modeling approach for the eeg forward problem, NeuroImage 184 (2019) 56–67.

[17] M. Bauer, S. Pursiainen, J. Vorwerk, H. Köstler, C. H. Wolters, Comparison Study for Whitney (Raviart-Thomas) Type Source Models in Finite Element Method Based EEG Forward Modeling, IEEE Transactions on Biomedical Engineering 62 (2015) 2648–56.

[18] T. Miinalainen, S. Pursiainen, A case study of focal bayesian eeg inversion for whitney element source spaces: Mesh-based vs. cartesian orientations, in: EMBEC & NBC 2017, Springer, 2017, pp. 1065–1068.

[19] O. D. Creutzfeldt, G. H. Fromm, H. Kapp, Influence of transcortical dc currents on cortical neuronal activity, Experimental neurology 5 (1962) 436–452.

[20] R. Hari, S. Baillet, G. Barnes, R. Burgess, N. Forss, J. Gross, M. Hämäläinen, O. Jensen, R. Kakigi, F. Mauguère, et al., Ifc-endorsed practical guidelines for clinical magnetoencephalography (meg), Clinical Neurophysiology (2018).

[21] D. Calvetti, H. Hakula, S. Pursiainen, E. Somersalo, Conditionally Gaussian hypermodels for cerebral source localization, SIAM J. Imaging Sci. 2 (2009) 879–909.

[22] A. O’Hagan, J. J. Forster, Kendall’s advanced theory of statistics, volume 2B: Bayesian inference, volume 2, Arnold, 2004.
[23] J. P. Kaipio, E. Somersalo, Statistical and Computational Methods for Inverse Problems, Springer, Berlin, 2004.

[24] D. M. Schmidt, J. S. George, C. C. Wood, Bayesian inference applied to the electromagnetic inverse problem, Human Brain Mapping 7 (1999) 195–212.

[25] M. Rullmann, A. Anwander, M. Dannhauer, S. K. Warfield, F. H. Duffy, C. H. Wolters, Eeg source analysis of epileptiform activity using a 1 mm anisotropic hexahedra finite element head model, NeuroImage 44 (2009) 399–410.

[26] G. Golub, C. van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, 1989.

[27] M. Dannhauer, B. Lanfer, C. H. Wolters, T. R. Knösche, Modeling of the human skull in EEG source analysis, Human Brain Mapping 32 (2011) 1383–1399.

[28] J. Vorwerk, J.-H. Cho, S. Rampp, H. Hamer, T. R. Knösche, C. H. Wolters, A guideline for head volume conductor modeling in eeg and meg, NeuroImage 100 (2014) 590–607.

[29] Y. Ohtake, A. Belyaev, I. Bogaevski, Mesh regularization and adaptive smoothing, Computer-Aided Design 33 (2001) 789–800.

[30] S. Pursiainen, Raviart–thomas-type sources adapted to applied eeg and meg: implementation and results, Inverse Problems 28 (2012) 065013.

[31] K. Uutela, M. Hämäläinen, E. Somersalo, Visualization of magnetoencephalographic data using minimum current estimates, NeuroImage 10 (1999) 173–180.

[32] S. S. Nagarajan, O. Portniaguine, D. Hwang, C. Johnson, K. Sekihara, Controlled support meg imaging, NeuroImage 33 (2006) 878–885.

[33] E. Lee, W. Duffy, R. Hadimani, M. Waris, W. Siddiqui, F. Islam, M. Rajamani, R. Nathan, D. Jiles, Investigational effect of brain-scalp distance on the efficacy of transcranial magnetic stimulation treatment in depression, IEEE Transactions on Magnetics 52 (2016) 1–4.
[34] List of nvidia graphics processing units. https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units, 2018. Accessed: 2018-11-18.

[35] J. de Munck, B. van Dijk, H. Spekreijse, Mathematical dipoles are adequate to describe realistic generators of human brain activity 35 (1988) 960–966.

[36] Tetgen, http://wias-berlin.de/software/tetgen/, 2018. Accessed: 2018-11-18.

[37] Netgen/ngsolve, https://ngsolve.org/, 2018. Accessed: 2018-11-18.

[38] duneuro: An open source c++ library for forward modeling in bioelectromagnetism, ???. Feb, 2018.

[39] DUNE: Distributed and Unified Numerics Environment, a modular toolbox for solving partial differential equations with grid-based methods, ???. Feb, 2018.

[40] SimBio Development Group, SimBio: A generic environment for biomimetical simulations, 2018. Retrieved Feb 2, 2018.

[41] BrainStorm: A collaborative, open-source application dedicated to the analysis of brain recordings: MEG, EEG, fNIRS, ECoG, depth electrodes and animal electrophysiology, 2018. Retrieved Feb 2, 2018.

[42] Fieldtrip toolbox: Dipole fitting of combined MEG/EEG data - tutorial, 2017. Retrieved 17 August, 2017.

[43] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, et al., Meg and eeg data analysis with mne-python, Frontiers in neuroscience 7 (2013) 267.

[44] S. Pursiainen, S. Lew, C. H. Wolters, Forward and inverse effects of the complete electrode model in neonatal EEG, Journal of Neurophysiology (2016).

[45] D. Calvetti, E. Somersalo, A gaussian hypermodel to recover blocky objects, Inverse problems 23 (2007) 733.
[46] D. Calvetti, F. Pitolli, E. Somersalo, B. Vantaggi, Bayes meets krylov: Statistically inspired preconditioners for cglss, SIAM Review 60 (2018) 429–461.