

# A Trilinear Immersed Finite Element Method for Solving the Electroencephalography Forward Problem

Sylvain Vallaghé, Théodore Papadopoulo

► **To cite this version:**

Sylvain Vallaghé, Théodore Papadopoulo. A Trilinear Immersed Finite Element Method for Solving the Electroencephalography Forward Problem. SIAM Journal on Scientific Computing, Society for Industrial and Applied Mathematics, 2010, 32 (4), pp.2379-2394. <<http://link.aip.org/link/?SCE/32/2379/1>>. <10.1137/09075038X>. <inria-00536265>

**HAL Id: inria-00536265**

**<https://hal.inria.fr/inria-00536265>**

Submitted on 15 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A trilinear immersed finite element method for solving the EEG forward problem

Sylvain Vallaghé, Théodore Papadopoulo

This article describes a trilinear immersed finite element method for solving the EEG forward problem, which is a 3-D elliptic interface problem in the head geometry. The method uses hexahedral Cartesian meshes independent of the interfaces between head tissues, thus avoiding the sometimes difficult task of generating geometry fitting meshes. The interfaces are represented with levelsets and the finite element space is locally modified to better approximate the discontinuities of the solution. Numerical results show that this method achieves the same accuracy than the standard linear finite element method with geometry fitting mesh.

## 1 Introduction

The electroencephalography (EEG) forward problem is described by the following second order elliptic equation:

$$\begin{cases} \nabla \cdot (\sigma \nabla u) = f & \text{in } \Omega , \\ \sigma \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega , \end{cases} \quad (1)$$

where  $\Omega \subset \mathbb{R}^3$  is a domain corresponding to the human head,  $u$  is the electric potential,  $\sigma$  is the electrical conductivity and  $f$  is a function describing the brain electrical activity. Computing the solution of (1) is of interest in neurosciences for modeling the EEG, i.e measurements of the electric potential at the head surface due to the brain activity. In particular, it is a necessary step before doing EEG source localization, which consists in estimating the source  $f$  from the EEG measurements.

It is usually assumed that the head is composed of several subdomains  $\Omega_k$  with different electrical conductivities  $\sigma_k$ , hence the conductivity  $\sigma$  of the whole domain  $\Omega$  is modeled as a piecewise continuous (e.g. constant) function:

$$\sigma|_{\Omega_k} = \sigma_k \in C^0(\Omega_k) .$$

On an interface  $S$  between two subdomains, the following jump conditions hold:

$$\begin{cases} [u] = 0 , \\ [\sigma \nabla u \cdot \mathbf{n}] = 0 , \end{cases} \quad (2)$$

where  $[\cdot]$  denotes the jump of a function. These two conditions correspond to the continuity of the electric potential and the current density normal to the surface. As a consequence, the quantity  $\nabla u$  is discontinuous across an interface.

Several numerical methods can be used to solve (1): the boundary element method (BEM), the finite difference method (FDM), the finite volume method (FVM) and the finite element method (FEM). In their standard formulations, all these methods rely on a mesh of the domain  $\Omega$  under consideration, and for the best convergence with respect to the mesh resolution, the mesh must fit with the interfaces between tissues. With the standard FDM, the mesh is a Cartesian grid which allows only staircase descriptions of the interfaces, hence the convergence towards the real interfaces is poor. With the standard tetrahedral FEM, the mesh is unstructured and composed of tetrahedra, and the interfaces are represented by triangles which form a subset of the tetrahedral mesh. In this case the description of the interfaces is better, but the mesh generation can be a difficult task, especially for the human head (e.g. interface between grey matter and CSF).

In the last decades, several methods have been developed so that interface problems such as (1) can be solved accurately with meshes independent of the interfaces. Such approaches were first developed for FD methods, with the immersed boundary method [13] or the immersed interface method [7], and also for FE methods with the framework of Babuška et al. [2]. The immersed interface method has then been adapted to the FE framework [9], and led to a family of methods referred to as immersed finite element (IFE) methods [10]. The FE method presented in this paper falls in the framework of IFE methods.

IFE methods allow to use meshes which do not conform with the interfaces, like a Cartesian grid. In this case, there are two types of cells in the mesh: non-interface cells, which are totally included in a subdomain  $\Omega_k$ , and interface cells which are cut by an interface  $S_k$ . The main idea of IFE methods is to use standard FE basis functions in the non-interface cells, whereas in the interface cells, piecewise standard FE functions are used such that the interface jump conditions (2) are satisfied to a certain extent.

Most of the previous FE methods for the EEG forward problem are based on geometry fitting mesh and standard FE basis functions [11, 15, 19]. For 3D models of the human head, the geometry fitting meshes can be very difficult to generate due to the complex folding of certain interfaces (white, gray matter and CSF interfaces). An approach is presented in [16] to simplify the meshing step : starting from a regular hexahedral mesh, the nodes that are close to an interface are then moved to better approximate the geometry. In this paper, we present an IFE method for solving the EEG forward problem which allows to avoid completely the geometry fitting mesh generation. It is based on Cartesian meshes and trilinear FE basis functions, and is inspired from the bilinear IFE method for 2D problems presented in [6]. Our goal is to show that the method can achieve the same accuracy than a standard linear FEM, without the need to build a geometry fitting mesh. This is an important factor because many users of EEG forward computing codes are confronted with the meshing issue for realistic head models, which tends to prevent the use of FE methods in the

EEG community.

## 2 Approximation space

FE methods require a discretization  $\Omega_h$  of the domain  $\Omega$  on which the computation has to be made and a basis  $\{\phi_n\}$  for a discrete approximation  $V_h$  of the space  $V$  of sufficiently regular functions defined on  $\Omega$ . The most standard choice is to use tetrahedral meshes for  $\Omega_h$  and piecewise linear functions over the mesh (P1 elements) to approximate  $V$ . In this case, the basis  $\{\phi_n\}$  is defined as the set of piecewise linear functions that take the value 1 on one node of the mesh and zero on the other nodes.

Here we use another family: Q1 elements, which are piecewise multilinear polynomials. The reason is that the head models are built from magnetic resonance images (MRI), hence the pixels (voxels) of a MRI readily provide a Cartesian mesh and the standard Q1 function basis is exactly the one used with multilinear interpolation in images. This is thus an attractive choice to avoid the generation of a complicated mesh.

Let us introduce some notations for the 3D case that will be used in the remainder of this article. Given a set of nodal values  $I_{ijk}$  on a cubic cell, we denote by  $\text{vec}(I_{ijk})$  the *vectorization* operation that orders the  $I_{ijk}$  values in a vector (of size 8 in this case). Given a monomial order (e.g.  $\mathbf{m} = (xyz, xy, xz, yz, x, y, z, 1)$ ), the coefficients of the Q1 polynomial corresponding to the values  $I_{ijk}$  is given by the invertible linear mapping  $\mathcal{C}$ , i.e. the Q1 polynomial  $p$  can be written as:

$$p(x, y, z) = \mathbf{m}^T \mathcal{C} \text{vec}(I_{ijk}) .$$

The opposite operation of evaluating the polynomial values at the nodes from its coefficients is obtained using the linear operator  $\mathcal{V} = \mathcal{C}^{-1}$ .

### 2.1 Interface representation with levelsets

As explained in the introduction, we will use Cartesian meshes which do not fit with the geometry, i.e. the interfaces between the different head tissues. As a consequence, the mesh does not represent explicitly the surfaces corresponding to the different interfaces. We will hence use another representation of these surfaces: levelsets. Let us first recall briefly what levelsets are. The levelsets technique was first mentioned in [4] and then developed by Osher and Sethian [12]. Since then it has received a lot of attention for representing and evolving closed shapes in space. The basic idea of levelsets is to represent a surface  $\mathcal{S}$  of codimension 1 (a curve in 2D or a surface in 3D) implicitly as the zero-level of a scalar function  $f_{\mathcal{S}}$  in its embedding space (e.g.  $\mathbb{R}^2$  for curves or  $\mathbb{R}^3$  for surfaces). Typically, this function  $f$  represents the signed distance to the surface (negative inside, positive outside). In practice, the embedding space is tessellated with a square grid (squares in 2D, cubes in 3D) and  $f_{\mathcal{S}}$  is discretized as a 2D or 3D image. One key of the success of the levelset method is

that complex shapes can be easily and conveniently manipulated as images. In this work, we will then simply represent the interfaces as levelset images on the Cartesian mesh used for FE computation. Furthermore, we adopt a multilinear interpolation of the levelset images, i.e. a representation with Q1 elements, as presented above. This allows for two things:

- The zero crossings within a pixel/voxel are well defined as the zero-crossings of the multilinear polynomial.
- Consequently, the topology of the levelset is always well defined.

Hence we can represent the head geometry with Q1 levelsets. We denote by  $B$  the bounding box of the head domain and assume that this domain can be depicted as a finite set of subdomains  $\Omega_k, k = 1..N$  delimited by closed non-intersecting interfaces  $S_k, k = 1..N - 1, S_j \cap S_k = \emptyset$  if  $j \neq k$ . Denoting by  $G_h$  a Cartesian mesh of  $B$ , each interface  $S_k$  is represented by a levelset as a set of nodal values on  $G_h$ . An example of a levelset in 2D is shown in figure 1a.

## 2.2 Immersed finite element space

As explained in the introduction, in IFE methods, standard basis functions are used in non-interface cells and piecewise standard basis functions are used for interface cells. Our IFE method is derived from standard Q1 FEM, so for non-interface cells, standard Q1 basis functions as described above are used. For interface cells, we use piecewise Q1 basis functions which satisfy the jump conditions (2) to a certain extent, as next explained.

### 2.2.1 Single interface cells

In 3D, standard Q1 functions are trilinear polynomials.

$$p(x, y, z) = p_{000} + p_{100}x + p_{010}y + p_{001}z + p_{110}xy + p_{101}xz + p_{011}yz + p_{111}xyz = \mathbf{m}^T \mathcal{C} \text{vec}(P_{ijk}) ,$$

where  $P_{ijk}$  corresponds to the nodal values of polynomial  $p$ . Consider an interface cell and denote by  $C$  the cell domain, by  $C_1$  the first subdomain with a conductivity  $\sigma_1$  and by  $C_2$  the second subdomain with a conductivity  $\sigma_2$ . The interface is given by a Q1 levelset, i.e. the zero-level of a trilinear polynomial  $l(x, y, z)$  with corresponding nodal values  $L_{ijk}$ . We consider a function  $\phi$  in  $C$  which is a piecewise trilinear polynomial, i.e. defined by:

$$\begin{cases} \phi(x, y, z) = p_1(x, y, z) & \text{in } C_1 , \\ \phi(x, y, z) = p_2(x, y, z) & \text{in } C_2 , \end{cases} \quad p_1, p_2 \in \text{Q1}(C) \quad (3)$$

and we want  $\phi$  to verify the jump conditions (2).  $\phi$  must be continuous at the interface: it means that  $p_1(x, y, z) - p_2(x, y, z) = 0$  on the surface  $l(x, y, z) = 0$ . Because  $p_1, p_2$  and  $l$  are all trilinear polynomials, necessarily:

$$p_1 - p_2 = \lambda l , \quad (4)$$

where  $\lambda$  is a real scalar. Also  $\sigma \nabla \phi \cdot \mathbf{n}$  must be continuous at the interface: it means that  $(\sigma_1 \nabla p_1 - \sigma_2 \nabla p_2) \cdot \mathbf{n} = 0$  on the surface  $l(x, y, z) = 0$ . As the quantity  $(\sigma_1 \nabla p_1 - \sigma_2 \nabla p_2) \cdot \mathbf{n}$  has fewer degrees of freedom than the trilinear polynomial  $l$ , this last condition is in most cases impossible to achieve. We thus impose a less constraining jump condition:

$$\int_{l(x,y,z)=0} \sigma_1 \nabla p_1 \cdot \mathbf{n} - \sigma_2 \nabla p_2 \cdot \mathbf{n} = 0 \quad , \quad (5)$$

which considers only the total flux through the whole interface of the cell. So finally, on an interface cell, we can define our finite element space as the space of piecewise trilinear polynomials which verify (4) and (5). One can verify that it is a vector space. We denote it  $V_h(C)$ .

**Lemma.** *Let  $C$  be a single interface cell with two subdomains of constant isotropic conductivities  $\sigma_1$  and  $\sigma_2$ . Then  $\phi \in V_h(C)$  is uniquely determined by its values at the nodes of  $C$ , except if  $\frac{\sigma_2}{\sigma_1} = K$ , where  $K$  is a constant which only depends on the interface.*

*Proof.* Without loss of generality, we consider  $C$  to be a cubic cell with nodes  $(i, j, k) \in \{0, 1\}^3$ . We denote  $Q1(C)$  the space of trilinear polynomials in  $C$ . Let  $l \in Q1(C)$  be the levelset of the interface cutting  $C = C_1 \cup C_2$ ,  $L_{ijk}$  its nodal values, and  $\lambda$  a real scalar as introduced in (4). We now fix a set of nodal values  $I_{ijk}$  and we want to show that there is a unique function  $\phi \in V_h(C)$  with nodal values  $I_{ijk}$ . Such a  $\phi$  is a piecewise trilinear polynomial and we denote  $p_1, p_2 \in Q1(C)$  its restrictions to  $C_1, C_2$ . The following conditions on  $p_1$  and  $p_2$  are satisfied:

$$\begin{cases} p_1(i, j, k) = I_{ijk} & L_{ijk} < 0 \quad , \\ p_2(i, j, k) = I_{ijk} & L_{ijk} > 0 \quad , \\ p_2(i, j, k) - p_1(i, j, k) = \lambda L_{ijk} \end{cases} \quad (6)$$

At each node  $p_2(i, j, k) - p_1(i, j, k) = \lambda L_{ijk}$  so  $p_2 - p_1 = \lambda l$  everywhere, which is exactly the jump condition (4). It is next shown that, assuming  $\lambda$  is known, this system completely defines the nodal values of  $p_1$  and  $p_2$ . Hence  $p_1$  and  $p_2$  are uniquely determined because they are standard Q1 functions.

To simplify the notations, we will use  $P_{ijk}$  in place of  $\text{vec}(P_{ijk})$ .

Let us denote by  $P_{1ijk}$  and  $P_{2ijk}$  the nodal values of  $p_1$  and  $p_2$  respectively. Denoting by  $\mathbf{I}^-$  (resp.  $\mathbf{I}^+$ ) the sub-matrix of the  $8 \times 8$  identity matrix containing only the rows corresponding to the negative (resp. positive) elements of  $I_{ijk}$ , we define  $I_{ijk}^- = \mathbf{I}^- I_{ijk}$  (resp.  $I_{ijk}^+ = \mathbf{I}^+ I_{ijk}$ ) the vector containing only the negative (resp. positive) elements of  $I_{ijk}$ . Finally,  $\mathbf{P}$  is defined as  $\mathbf{P} = \begin{bmatrix} \mathbf{I}^- \\ \mathbf{I}^+ \end{bmatrix}$ .  $\mathbf{P}$

is just a permuted  $8 \times 8$  identity matrix, so  $|\det(\mathbf{P})| = 1$ . With these notations, the system (6) can be written as:

$$\begin{bmatrix} \mathbf{I}^- & \mathbf{0} \\ \mathbf{0} & \mathbf{I}^+ \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} P_{1ijk} \\ P_{2ijk} \end{bmatrix} = \begin{bmatrix} I_{ijk}^- \\ I_{ijk}^+ \\ \lambda L_{ijk} \end{bmatrix} \quad (7)$$

Denoting respectively by  $\mathbf{A}$  and  $\mathbf{x}$  the left-hand side matrix and vector of this equation, it is easy to show that  $|\det(\mathbf{A})| = 1$  (just add the second block of columns to the first to obtain an upper triangular block matrix; the two  $8 \times 8$  diagonal blocks are just  $\mathbf{P}$  and  $\mathbf{I}$ , hence the result). Noticing that:

$$\begin{bmatrix} I_{ijk}^- \\ I_{ijk}^+ \end{bmatrix} = \mathbf{P} I_{ijk} \ ,$$

the system (7) is decomposed as:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{P} I_{ijk} \\ \mathbf{0} \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{0} \\ L_{ijk} \end{bmatrix} \ .$$

If we only keep the first term of the right hand side, the solution of the linear system:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} I_{ijk}^- \\ I_{ijk}^+ \\ \mathbf{0} \end{bmatrix}$$

corresponds to  $p_1 = p_2 = \psi$  where  $\psi$  is exactly the standard Q1 function determined by the nodal values  $I_{ijk}$ . If we only keep the second term of the right hand side and drop  $\lambda$ , the solution of the linear system:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{0} \\ L_{ijk} \end{bmatrix}$$

is simply:

$$\begin{cases} P_{1ijk} = \max(L_{ijk}, 0) \\ P_{2ijk} = -\min(L_{ijk}, 0) \end{cases} \quad (8)$$

This solution corresponds to the piecewise Q1 function  $(p_1, p_2)$  which is zero at the nodes and  $p_1 - p_2 = l$ . We denote this particular function  $\zeta$ . It is now clear that the function  $\phi$  defined by (7) is

$$\phi = \psi + \lambda \zeta \ . \quad (9)$$

We now introduce the linear functional  $g(\phi)$  defined by

$$g(\phi) = \int_{l(x,y,z)=0} \sigma_1 \nabla \phi|_{C_1} \cdot \mathbf{n} - \sigma_2 \nabla \phi|_{C_2} \cdot \mathbf{n} \ .$$

$g(\phi) = 0$  is exactly the second jump condition (5). Using (9), this is equivalent to  $g(\psi) + \lambda g(\zeta) = 0$ . As a consequence, if  $g(\zeta) \neq 0$ , there is a unique  $\phi$  satisfying (6) and the jump condition (5), which is obtained for  $\lambda = \frac{-g(\psi)}{g(\zeta)}$ .

We now discuss the case  $g(\zeta) = 0$ . Denoting  $p_1, p_2$  the two restrictions of  $\zeta$  to  $C_1, C_2$ , we then have:

$$\begin{aligned} \int_{l(x,y,z)=0} \sigma_1 \nabla p_1 \cdot \mathbf{n} - \sigma_2 \nabla p_2 \cdot \mathbf{n} &= 0 \ , \\ \int_{l(x,y,z)=0} \nabla p_1 \cdot \mathbf{n} - \nabla p_2 \cdot \mathbf{n} &= \int_{l(x,y,z)=0} \nabla l \cdot \frac{\nabla l}{\|\nabla l\|} > 0 \ . \end{aligned}$$

From these two equalities, we necessarily have  $\int_{l(x,y,z)=0} \nabla p_2 \cdot \mathbf{n} \neq 0$  and  $\int_{l(x,y,z)=0} \nabla p_1 \cdot \mathbf{n} \neq 0$ , so we can define the following quantity:

$$K = \frac{\int_{l(x,y,z)=0} \nabla p_1 \cdot \mathbf{n}}{\int_{l(x,y,z)=0} \nabla p_2 \cdot \mathbf{n}} .$$

Then  $g(\zeta) = 0$  is equivalent to  $\frac{\sigma_2}{\sigma_1} = K$ . □

*Remark.* The condition  $\frac{\sigma_2}{\sigma_1} = K$  defines a semiline in the space of admissible conductivities  $\{(\sigma_1, \sigma_2) | \sigma_1 > 0, \sigma_2 > 0\}$ . This semiline is a measure-zero set so the probability of having  $\frac{\sigma_2}{\sigma_1} = K$  is zero. In practice, we do not encounter this situation, and it even appears that we always have  $K < 0$ , so the condition  $\frac{\sigma_2}{\sigma_1} = K$  is impossible to achieve.

### 2.2.2 Double interface cell

The case of a cell being cut by two interfaces is very likely to happen for our application to the EEG forward problem. A typical case are the gray matter/CSF and CSF/skull interfaces which can come very close to each other. As a consequence, it is necessary to extend the immersed finite element space to the case of a double interface cell. Let  $C$  be such a cell, with three subdomains  $C_1, C_2, C_3$  of conductivity  $\sigma_1, \sigma_2, \sigma_3$ . The interfaces between  $C_1, C_2$  and  $C_2, C_3$  are given by levelsets  $l$  and  $m$  respectively. We assume that these two interfaces do not cut. Following the same model as for the single interface cell, we define the finite element space for the double interface cell  $C$ ,  $V_h(C)$ , as the space of functions  $\phi$  such that:

$$\begin{cases} \phi(x, y, z) = p_1(x, y, z) & \text{in } C_1 , \\ \phi(x, y, z) = p_2(x, y, z) & \text{in } C_2 , \\ \phi(x, y, z) = p_3(x, y, z) & \text{in } C_3 , \end{cases} \quad p_1, p_2, p_3 \in \mathbf{Q1}(C) , \quad (10)$$

$$p_1 - p_2 = \lambda l , \quad (11)$$

$$p_2 - p_3 = \mu m , \quad (12)$$

$$\int_{l(x,y,z)=0} \sigma_1 \nabla p_1 \cdot \mathbf{n} - \sigma_2 \nabla p_2 \cdot \mathbf{n} = 0 , \quad (13)$$

$$\int_{m(x,y,z)=0} \sigma_2 \nabla p_2 \cdot \mathbf{n} - \sigma_3 \nabla p_3 \cdot \mathbf{n} = 0 . \quad (14)$$

**Lemma.** *Let  $C$  be a double interface cell with three subdomains of constant conductivities  $\sigma_1, \sigma_2, \sigma_3$ . Then  $\phi \in V_h(C)$  is uniquely determined by its values at the nodes of  $C$ , except if  $(\sigma_1, \sigma_2, \sigma_3)$  belongs to a certain surface in  $\mathbb{R}^3$ , which depends only on the interface.*

*Proof.* We use the same ideas as in the proof of the single interface case. Denoting  $\mathbf{x}$  the vector containing the nodal values of  $p_1, p_2, p_3$ , we can build the following square linear system:

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{PI}_{ijk} \\ \lambda L_{ijk} \\ \mu M_{ijk} \end{bmatrix} , \quad (15)$$



with  $\det(\mathbf{A}) \neq 0$ . This linear system embeds the nodal values  $I_{ijk}$  and the continuity at the interfaces corresponding to (11) and (12). The solution to this system corresponds to a function  $\phi$  such that:

$$\phi = \psi + \lambda\zeta + \mu\eta . \quad (16)$$

$\psi$  is the standard Q1 function determined by the nodal values  $I_{ijk}$ .  $\zeta$  is a piecewise Q1 function ( $\zeta|_{C_1} = q_1, \zeta|_{C_2 \cup C_3} = q_2$ ) which is zero at the nodes and  $q_1 - q_2 = l$ .  $\eta$  is a piecewise Q1 function ( $\eta|_{C_1 \cup C_2} = r_1, \eta|_{C_3} = r_2$ ) which is zero at the nodes and  $r_1 - r_2 = m$ .

We now introduce the linear functionals  $g$  and  $h$ :

$$\begin{aligned} g(\phi) &= \int_{l(x,y,z)=0} \sigma_1 \nabla \phi|_{C_1} \cdot \mathbf{n} - \sigma_2 \nabla \phi|_{C_2} \cdot \mathbf{n} , \\ h(\phi) &= \int_{m(x,y,z)=0} \sigma_2 \nabla \phi|_{C_2} \cdot \mathbf{n} - \sigma_3 \nabla \phi|_{C_3} \cdot \mathbf{n} . \end{aligned}$$

$g(\phi) = 0$  and  $h(\phi) = 0$  are exactly the jump conditions (13) and (14). These are equivalent to:

$$\begin{aligned} g(\psi) + \lambda g(\zeta) + \mu g(\eta) &= 0 , \\ h(\psi) + \lambda h(\zeta) + \mu h(\eta) &= 0 . \end{aligned}$$

This defines a linear system for  $\lambda$  and  $\mu$ . If its determinant is non-zero, then  $\phi$  is uniquely defined.  $\zeta$  and  $\eta$  are fixed for a given cell  $C$ , so the determinant  $D$  is a continuous function of  $\sigma_1, \sigma_2, \sigma_3$ , and the condition  $D = 0$  defines a surface in  $\mathbb{R}^3$ . □

*Remark.* As for the single interface, the "problematic" surface is a measure-zero set in  $\mathbb{R}^3$ . In practice, we do not encounter the special case corresponding to a conductivity triplet  $\sigma_1, \sigma_2, \sigma_3$  which belongs to this surface.

### 2.2.3 Nodal basis

Following the previous results, we can assume that for an interface cell  $C$ ,  $\phi$  in  $V_h(C)$  is uniquely determined by its values at the nodes of  $C$ , as for Q1 functions in non-interface cells. Hence, we can define the usual nodal basis functions  $\phi_n$  which takes the value 1 at a node  $n$  and zero at the other nodes. Finally, the finite element space in the whole domain  $\Omega$  is defined as:

$$V_h(\Omega) = \text{span} \{ \phi_n \} .$$

We show on figure 1 the restriction of a nodal basis function to a non-interface cell and single interface cell respectively (in 2D).

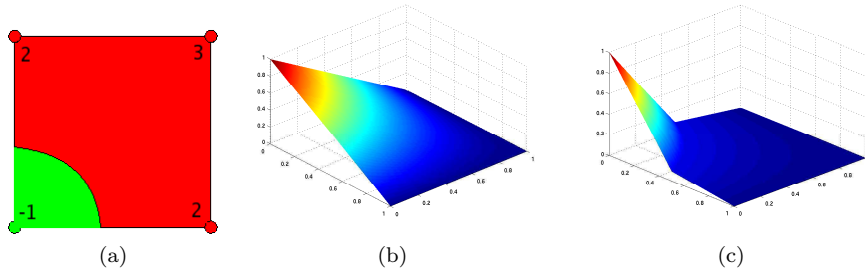


Figure 1: (a) A 2D levelset on a square cell. The nodal values are interpolated by a bilinear polynomial for which the zero-level corresponds to the interface curve. (b) A standard Q1 nodal basis function. (c) An IFE nodal basis function for the interface cell with a conductivity jump from 1 to 10.

### 3 The Finite Element Method

We now consider the function space  $V$ , defined as:

$$V = \{v \in L^2(\Omega), \nabla v \in L^2(\Omega)^3, \} .$$

The weak solution  $u \in V$  of (1) is such that

$$\forall v \in V \quad a(u, v) = l(v) , \quad (17)$$

where  $a$  is the bilinear form

$$a(u, v) = - \int_{\Omega} \sigma \nabla u \cdot \nabla v \quad (18)$$

and  $l$  is the linear functional

$$l(v) = \int_{\Omega} f v . \quad (19)$$

It can be shown that problem (17) has a unique solution in the space  $V^* = \{v \in V, \int_{\Omega} v = 0\}$ , we refer for instance to [18]. For numerical computation, (17) is then discretized using the finite element space  $V_h(\Omega)$  described in the previous section. It is important to note that  $V_h$  is nonconforming, i.e.  $V_h \not\subset V$ . Although standard Q1 basis functions are continuous, this is no more true for our basis functions on interface cells: they can be discontinuous on a face between two interface cells, so the conformity is lost. As a consequence, the derivation of a priori error estimates is not straightforward. For the approximation error, the following estimates have been derived in [6] and can be directly extended to our FE space:

$$\|u - \Pi_h u\|_0 + h \|u - \Pi_h u\|_1 \leq Ch^2 \|u\|_2 ,$$

where  $\Pi_h u$  is the interpolant of  $u$  in  $V_h$ ,  $\|\cdot\|_0$  is the  $L^2$  norm,  $\|\cdot\|_1$  is the  $H^1$  norm, and  $\|\cdot\|_2$  is a broken  $H^2$  norm. For the consistency error (due to nonconformity), we are not yet able to derive an error estimate, and there is no such result in the literature to the authors' knowledge.

## 4 Numerical results

### 4.1 Spherical Model.

The accuracy of the method is evaluated by computing the electric potential generated by a current dipole in a 3-sphere model. In this configuration, there is an analytic expression of the potential [20] which can be used as a ground truth to compute the error. The current dipole is given by its position  $\mathbf{p} \in \mathbb{R}^3$  and its moment  $\mathbf{q} \in \mathbb{R}^3$ . The corresponding source term in (1) is then

$$f = \nabla \delta_{\mathbf{p}} \mathbf{q} , \quad (20)$$

where  $\delta_{\mathbf{p}}$  is the Dirac delta distribution at  $\mathbf{p}$ , and the corresponding linear functional in (17) is

$$l(v) = \nabla v(\mathbf{p}) \cdot \mathbf{q} . \quad (21)$$

This is quite abusive since normally  $f$  needs to be in  $L^2(\Omega)$  for (17) to be well-posed. The problem can be reformulated using the subtraction approach [1, 5] : in this case the singularity is treated analytically, and a new correction potential is considered with an equation similar to (1) with  $f \in L^2(\Omega)$ . Nevertheless, the formulation (20) is often used to model current dipoles in the finite element method due to its reasonable accuracy and low complexity. The spherical model is composed of 3 spheres of radii 87, 92, 100 *mm* and the conductivity values are 0.33, 0.0042, 0.33 *S.m*<sup>-1</sup> from inside to outside.

To quantify the accuracy of the method, we use two classical quantities in the field of EEG forward computation: the Relative Difference Measure (RDM) and the MAGnification factor (MAG). These two quantities are used to compare two potential distributions at the scalp surface of the model, and are defined by:

$$RDM(u_{num}, u_{ref}) = \left\| \frac{u_{num}}{\|u_{num}\|} - \frac{u_{ref}}{\|u_{ref}\|} \right\| , \quad (22)$$

$$MAG(u_{num}, u_{ref}) = \frac{\|u_{num}\|}{\|u_{ref}\|} , \quad (23)$$

where  $\|\cdot\|$  is the  $L^2$  norm on the scalp surface  $\partial\Omega$ . In the following, we compare the potential distribution of the numerical solution given by the finite element method  $u_{num}$  with the reference solution  $u_{ref}$  computed analytically. This is done by sampling the solutions at 642 points uniformly distributed on the outermost sphere of radius 1.

Different dipole configurations are considered: one single dipole is moved along the  $z$ -axis inside the innermost domain (representing the brain) such that it comes close to the interface between brain and skull, which has the usual effect to increase the numerical errors. Both radial and tangential orientations are considered for the dipole moment.

As an introduction, we first show a comparison between our method and a standard regular hexahedral approach using trilinear basis functions, i.e. a

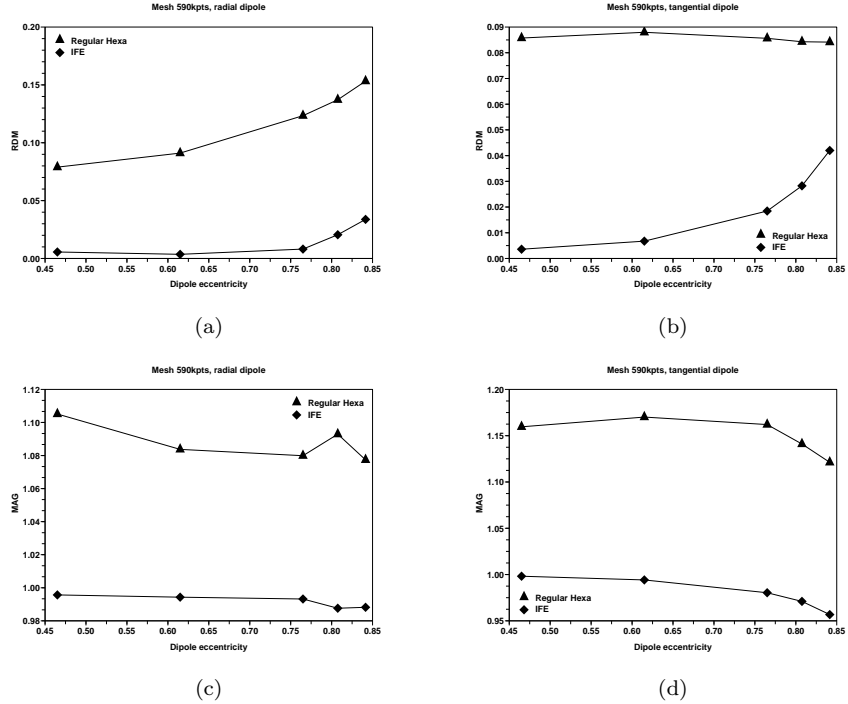


Figure 2: In a 3-sphere model, the RDM and MAG of the method (Immersed Finite Element - IFE) are compared to the RDM and MAG of the standard regular hexahedral approach.

staircase description of the interfaces between domains, allowing one domain per cell. We embedded the spherical model in a Cartesian grid such that the outermost sphere contains a number of voxels equal to 590000. The results of both methods were compared to the analytical solution by computing the RDM and MAG quantities, and are shown in figure 2. The results of our method are far better, due to the better approximation of the interfaces.

The goal of this paper is to show that our method can achieve the same accuracy than the standard P1 tetrahedral FEM with geometry fitting mesh. We chose to compare these two methods by considering the number of mesh nodes. We built three tetrahedral fitting meshes of the spherical model with 118000, 310000 and 590000 mesh nodes, and regular Cartesian discretizations with the same number of nodes. The tetrahedral meshes were built using CGAL [21]. The mesh generation algorithm is a Delaunay refinement process followed by an optimization phase, such that all kind of quasi degenerate tetrahedra are eliminated. Simulations were run using the standard P1 FEM on tetrahedral meshes and using our method on the Cartesian grids. RDM and MAG results are shown in figures 3 and 4.

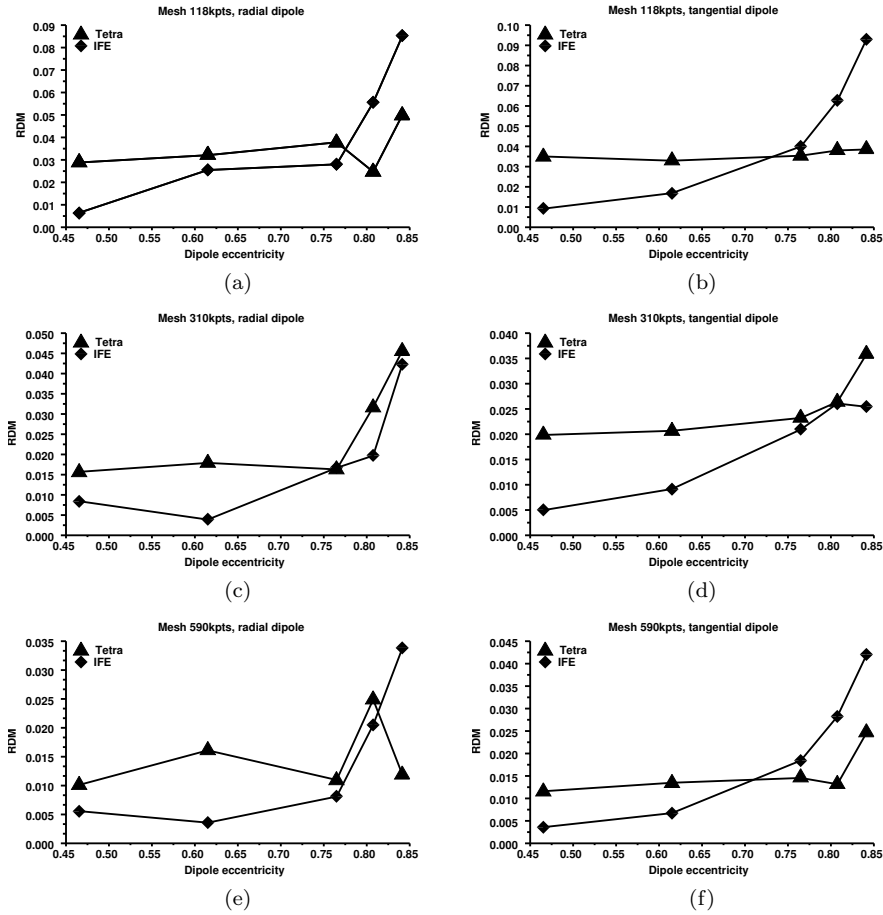


Figure 3: In a 3-sphere model, the RDM of the method (Immersed Finite Element - IFE) is compared to the RDM of the standard P1 tetrahedral FEM with geometry fitting mesh.

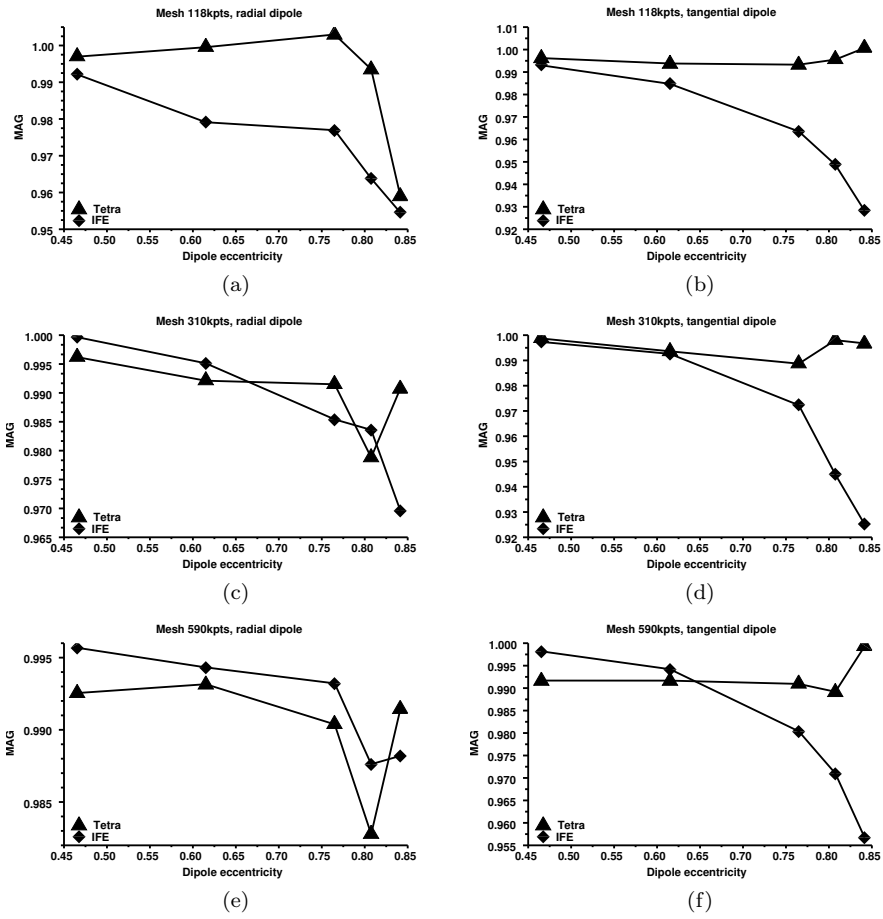


Figure 4: In a 3-sphere model, the MAG of the method (Immersed Finite Element - IFE) is compared to the MAG of the standard P1 tetrahedral FEM with geometry fitting mesh.

With respect to the RDM, the errors of both methods are equivalent, except for the lowest resolution mesh (118 kpts). For this mesh, the RDM for very eccentric dipoles (close to the brain-skull interface) is about twice as big with our method than with the standard P1 FEM. This can be explained by the fact that in this low resolution, many voxels of the Cartesian grid are cut both by the brain/skull and skull/scalp interfaces, so that our method has to handle many double interface cells. Intuitively, the increase of numerical errors due to the dipole singularity coming close to the interface of conductivity discontinuity becomes even bigger because the dipole is close to many double interface cells, which handle twice more discontinuities. For the MAG, the errors of both methods are equivalent for radial dipoles but the MAG of our method tends to decrease for tangential dipoles with a high eccentricity. It is important to note that the error curves do not have very smooth variations: this is due to the fact that depending on the dipole position in the mesh cells, the FE formulation of the singular source term (21) gives a variable numerical error. Typically, the best results are obtained when the dipole is at the center of a mesh cell, and the accuracy decreases when the dipole position gets close to a cell boundary, see for instance [17]. For our experiments, we fixed the dipole positions in the geometry of the model without caring for their positions in the mesh, so the variability of the dipole positions in the mesh cells introduce some variability in the numerical errors.

Overall we see that for the same number of nodes, the standard P1 FEM performs slightly better than our method. Nevertheless, apart from the fact that our method does not require a geometry fitting mesh, this last point is counterbalanced by the memory usage. Because of the regular Cartesian grid, our method can share the memory for all non-interface cells, whereas with unstructured tetrahedral meshes, the standard FEM has to store data for every cell. For instance, the standard FEM requires 1 Gb of RAM to run on the mesh with 590000 points, whereas our method needs only 200 Mb to run on an equivalent grid resolution. Hence our method can handle higher resolutions more easily. However, if anisotropy of most of the tissues (like skull and the whole brain) has to be modeled, these advantages are weakened.

To compare both methods with respect to the memory cost, we can consider the mesh with 118000 points for the standard P1 FEM and the mesh with 590000 points for our method, which both have a memory cost of about 200 Mb. In this case, the RDM and MAG quantities are compared in figure 5. We see that our method has better results, except for the MAG of the tangential dipole.

In section 2.2, we assumed isotropic conductivities for the sake of simplicity, but the method can also be used with anisotropic conductivities. We tested it on the 3-layer spherical model, with an anisotropic conductivity in the middle layer (the skull): the conductivity has two different values in the radial and tangential directions with respect to the spherical geometry. The radial skull conductivity is  $0.0042 \text{ S.m}^{-1}$  and the tangential skull conductivity is  $0.042 \text{ S.m}^{-1}$ , corresponding to an anisotropy ratio of 10. We show in figure 6 the RDM and MAG of the method on the mesh with 310000 points. We do not compare it to the

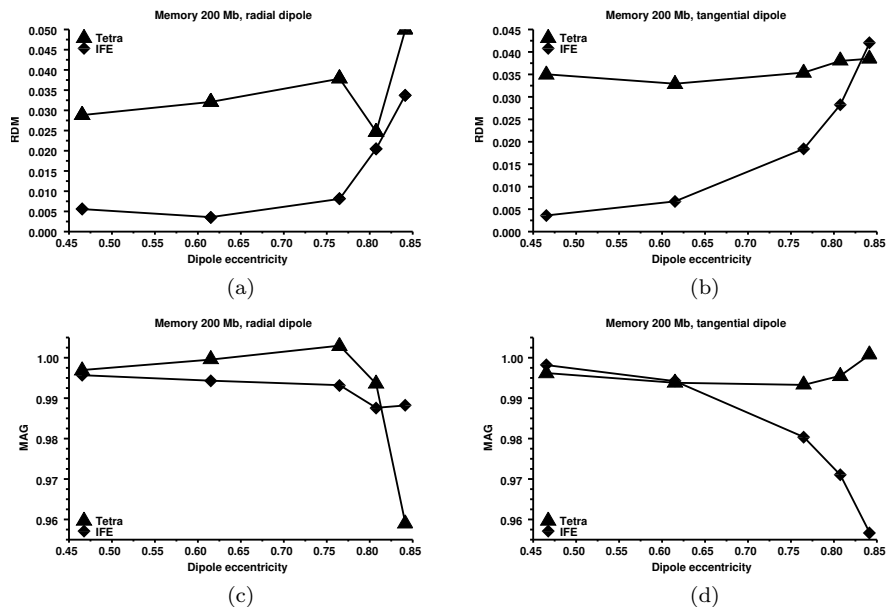


Figure 5: In a 3-sphere model, the method is compared to the standard P1 tetrahedral FEM with geometry fitting mesh, with respect to the memory cost.

(11)

Figure 6: MAG and RDM of the method in a 3-sphere model with anisotropy in the middle layer.

tetrahedral P1 FEM because the code we used does not allow anisotropic conductivity. For the RDM, the results are as good as for the isotropic case, except at high eccentricities for the tangential dipole. For the MAG, the values are overall higher than in the isotropic case.

## 4.2 Real Head Model.

To model a real head, an MRI image of size  $256 \times 256 \times 160$  has been taken and segmented using four levelsets (also within a  $256 \times 256 \times 160$  discretized domain). The levelset images have then been downsampled by a factor 2 to reduce the computational load for the FE method, so the final head model is embedded in a Cartesian grid of size  $128 \times 128 \times 80$ . To give an idea of the discretization size, the number of voxels inside the scalp surface is 550000. Figure 7 shows some surfaces and cuts through this model. It should be noted that in this model, the graymatter/CSF and CSF/skull interfaces are sometimes very close, as shown in figure 7, so that many double interface cells are present.

A dipole was positioned inside the left hemisphere of the brain, with tan-



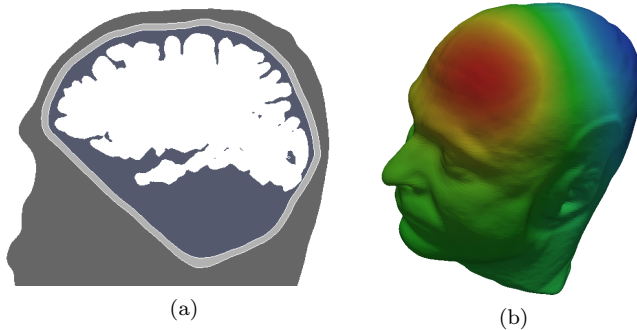


Figure 7: (a) The realistic head model : a sagittal cut (the cerebellum is not included in the model). (b) The electric potential was computed for a tangential dipole located in the left hemisphere of the brain. The potential values are shown on the scalp surface: red for positive, blue for negative.

gential orientation. Figure 7 shows the computed potential over the skin. For this simulation, isotropic constant conductivities have been used for each of the subdomains:  $0.33 \text{ S.m}^{-1}$  for scalp and brain,  $0.0042 \text{ S.m}^{-1}$  for the skull and  $1.79 \text{ S.m}^{-1}$  for the CSF [3].

This simulation shows the advantage of our method: starting directly from the levelsets of the surfaces, the electric potential can be computed. On the contrary, building a tetrahedral mesh for this model would require much more work, and such a mesh can easily have defects which produce numerical errors, because of complex surfaces like the graymatter/CSF interface.

## 5 Discussion and conclusion

We presented a method to compute the EEG forward solution without generating geometry fitting meshes. This is a practical advantage as it can be very time consuming to generate a geometry fitting mesh for the head and its subdomains, especially when it has to be done for many subjects.

We can compare the achieved accuracy of our method to the one presented in [16], which also aims at avoiding the tetrahedral mesh generation. In [16], the numerical results are obtained for a four-sphere model with radii of 78, 80, 86, 92  $mm$ , discretized with 2  $mm$  and 3  $mm$  hexahedral meshes. In our paper, we use a three-sphere model with radii of 87, 92, 100  $mm$ , the meshes with 310000 and 590000 points correspond to 2.4 and 1.95  $mm$  hexahedral cells, we hence have similar discretizations than in [16]. In our numerical experiments as well as in [16], the most eccentric dipole was positioned at 97% of the radius of the innermost sphere. For this dipole position, we achieved RDM values between 0.03 and 0.04, which is the same range of values than in [16]. The advantage of our method is that the EEG forward solution is directly computed from the regular hexahedral mesh and the levelset segmentations, whereas in [16] the

regular hexahedral mesh still has to be modified into a non-regular hexahedral mesh which better approximates the geometry, so the mesh generation is not as straightforward. Our method needs no processing for mesh generation as it uses a Cartesian grid, which can be directly adapted from the MRI images used for segmentations.

As stated in the beginning of section 4.1, we used a model for current dipoles which has intrinsic error. A rigorous approach for dipole modeling is the subtraction method [1], which has been shown to give very good results : in [5], using the same model than in [16] and a standard FEM with a tetrahedral fitting mesh of 360000 points, the RDM values are inferior to 0.01, even for a dipole positioned at 98.7% of the radius of the innermost sphere. So we can expect to reduce the numerical errors by using the subtraction approach in our method.

Nevertheless, with the current dipole model we used in this paper, we showed on numerical experiments that the method achieves similar errors than the standard linear FEM with geometry fitting mesh, so there is no loss in accuracy because of avoiding the mesh generation. Further benefits are obtained thanks to the regular discretization structure : the memory cost is smaller and many parts of the code were easily parallelized using OpenMP.

## References

- [1] KA Awada, DR Jackson, JT Williams, DR Wilton, SB Baumann, and AC Papanicolaou. Computational aspects of finite element modeling in EEG source localization. *IEEE Transactions on Biomedical Engineering*, 44(8):736–752, 1997.
- [2] I. Babuska, G. Caloz, and J.E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31:945, 1994.
- [3] SB Baumann, DR Wozny, SK Kelly, and FM Meno. The electrical conductivity of human cerebrospinal fluid at body temperature. *IEEE Transactions on Biomedical Engineering*, 44(3):220–223, 1997.
- [4] A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh-Taylor instability. *Lecture Notes in Mathematics*, 771:145–159, 1979.
- [5] F. Drechsler, CH Wolters, T. Dierkes, H. Si, and L. Grasedyck. A full subtraction approach for finite element method based source analysis using constrained Delaunay tetrahedralisation. *NeuroImage*, 46(4):1055–1065, 2009.
- [6] X. He, T. Lin, and Y. Lin. Approximation capability of a bilinear immersed finite element space. *Numerical Methods for Partial Differential Equations*, 24(5):1265–1300, 2008.

- [7] R.J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1019, 1994.
- [8] S. Lew, CH Wolters, T. Dierkes, C. R. ”oer, and RS MacLeod. Accuracy and run-time comparison for different potential approaches and iterative solvers in finite element method based EEG source analysis. *Applied Numerical Mathematics*, 59(8):1970–1988, 2009.
- [9] Z. Li. The immersed interface method using a finite element formulation. *Applied Numerical Mathematics*, 27(3):253–267, 1998.
- [10] Z. Li, T. Lin, and X. Wu. New cartesian grid methods for interface problems using the finite element formulation. *Numerische Mathematik*, 96(1):61–98, 2003.
- [11] Gildas Marin, Christophe Guerin, Sylvain Baillet, Line Garnero, and Gérard Meunier. Influence of skull anisotropy for the forward and inverse problems in EEG: simulation studies using FEM on realistic head models. *Human Brain Mapping*, 6:250–269, 1998.
- [12] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
- [13] C.S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220, 1977.
- [14] P.H. Schimpf, C. Ramon, and J. Haueisen. Dipole models for the eeg and meg. *IEEE Transactions on Biomedical Engineering*, 49(5):409–418, 2002.
- [15] R. Van Uitert, D. Weinstein, C. Johnson, and L. Zhukov. Finite element eeg and meg simulations for realistic head models: quadratic vs. linear approximations. *J. Biomedizinische Technik*, 46:32–34, 2001.
- [16] CH Wolters, A. Anwander, G. Berti, and U. Hartmann. Geometry-Adapted Hexahedral Meshes Improve Accuracy of Finite-Element-Method-Based EEG Source Analysis. *IEEE Transactions on Biomedical Engineering*, 54(8):1446–1453, 2007.
- [17] CH Wolters, H. Köstler, C. Möller, J. Härdtlein, and A. Anwander. Numerical approaches for dipole modeling in finite element method based source analysis. In *International Congress Series*, volume 1300, pages 189–192. Elsevier, 2007.
- [18] CH Wolters, H. Köstler, C. Möller, J. Härdtlein, L. Grasedyck, and W. Hackbusch. Numerical mathematics of the subtraction method for the modeling of a current dipole in eeg source reconstruction using finite element head models. *SIAM Journal on Scientific Computing*, 30:24, 2007.

- [19] CH Wolters, M. Kuhn, A. Anwander, and S. Reitzinger. A parallel algebraic multigrid solver for finite element method based source localization in the human brain. *Computing and Visualization in Science*, 5(3):165–177, 2002.
- [20] Zhi Zhang. A fast method to compute surface potentials generated by dipoles within multilayer anisotropic spheres. *Phys. Med. Biol.*, 40:335–349, 1995.
- [21] CGAL. Computational geometry algorithms library. <http://www.cgal.org>.