Comparative Performance of the Finite Element Method and the Boundary Element Fast Multipole Method for a Canonic Problem Mimicking Transcranial Magnetic Stimulation (TMS)

Aung Thu Htet[1], Guilherme B. Saturnino[2,3], Edward H. Burnham[1], Aapo Nummenmaa[4], and Sergey N. Makarov[1,4]

[1]ECE Department, Worcester Polytechnic Inst., Worcester, MA 01609 USA
[2] Danish Research Centre for Magnetic Resonance, Centre for Functional and Diagnostic Imaging and Research, Copenhagen University Hospital Hvidovre, Hvidovre, DK-2650 Denmark
[3] Center for Magnetic Resonance, Department of Electrical Engineering, Technical University of Denmark, Kgs Lyngby, DK-2800 Denmark
[4]Athinoula A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Charlestown, MA 02129 USA

Abstract:

A study pertinent to the numerical modeling of cortical neurostimulation is conducted in an effort to compare the performance of the finite element method (FEM) and the boundary element fast multipole method (BEM-FMM) at exactly matched computational performance metrics. We consider a canonic multi-sphere problem and an external magnetic-dipole excitation where the analytical solution is available. The FEM algorithm tested is an open-source getDP solver running in SimNIBS 2.1.1 environment. The BEM-FMM method runs in MATLAB® 2018a environment.

We observe that the BEM-FMM algorithm gives a smaller solution error for all mesh resolutions and runs faster for high-resolution meshes when the number of triangular facets exceeds approximately 0.5 M. We present other relevant simulation results such as volumetric mesh generation times for the FEM, times necessary to compute the potential integrals for the BEM-FMM, and solution performance metrics for different hardware/operating system combinations. This study provides a mathematical justification for anticipated use of the BEM-FMM algorithm for high-resolution realistic transcranial magnetic stimulation scenarios.

Keywords: Neurostimulation, Transcranial Magnetic Stimulation, Numerical Modeling, Finite Element Method, Boundary Element Fast Multipole Method, Comparison

1. Introduction

For all three chief neurostimulation modalities – transcranial magnetic stimulation (TMS), transcranial electric stimulation (TES), and intracortical microstimulation (ICMS) – numerical computation of the electric fields within a patient-specific head model is the major and often only way to foster spatial targeting and obtain a quantitative measure of the required stimulation dose (Bikson et al., 2018).

At present, a large portion of the macroscopic electromagnetic simulations of the brain are done using finite element method (FEM). FEM is widely used across engineering, physics, and geosciences. There are many general-purpose, open-source environments for FEM modeling, from high-level environments such as getDP (Dular et al., 1988), Deal.II (Bangerth et al., 2007), and FEniCS (Logg et al., 2012), to lower-level environments such as PETSc (Balay et al., 2018). Those solvers provide a practical and well-tested choice for creating problem-specific software solutions, for example:

- The well-known, open-source transcranial brain stimulation modeling software SimNIBS (Thielscher et al., 2015; Opitz et al., 2015; Nielsen et al., 2018), whose most recent version, v2.1, currently uses the open-source FEM software getDP (see Reference Manual 2017), which originates from the previous century (Dular et al., 1988);
- ROAST, a recently introduced TES modeling pipeline (Huang et al., 2018), which again uses the open-source FEM software getDP;
- COMETS: A MATLAB® custom toolbox for simulating transcranial direct current stimulation (tDCS) (Jung et al., 2013; Lee et al., 2017), which is based on the first-order FEM from the textbook (Jin 2002).

On the other hand, the most popular commercial FEM solvers such as COMSOL Multiphysics® and ANSYS Maxwell 3D (ANSYS Electronics Desktop®) tend to be relatively slow and less accessible.

The boundary element method (BEM) is a basic mesh reduction technique. It is also widely used, mostly for EGG/MEG modeling (Geselowitz 1967; Meijs et al., 1989; Hämäläinen et al., 1993; Ferguson et al., 1994; Mosher et al., 1999; Gramfort et al., 2014; Tadel et al., 2011; Gramfort et al., 2010; Stenroos et al., 2007; Stenroos and Sarvas 2012; Stenroos and Nummenmaa 2016; Nummenmaa et al 2013).

This study aims to compare the performance of the popular open-source FEM solver getDP referenced above with an alternative computational engine which is the boundary element fast multipole method (BEM-FMM) (Makarov et al., 2018[26],[27],[28]). Both the FEM software and the BEM-FMM software use exactly the same and matched performance metrics: a multilayer sphere model, for which the analytical solution is available, the same server (Intel(R) Xeon(R) E5-26900 CPU 2.90 GHz), and the same operating system (Red Hat Enterprise Linux 7.5), with both running on a single processor. The getDP software runs within SimNIBS 2.1.1, while the BEM-FMM software runs within MATLAB® 2012-2018. Note that SimNIBS 2.1.1 and the BEM-FMM software also support both Linux and Windows operating systems simultaneously.

2. Materials and Methods

2.1. Finite Element Method versus Boundary Element Method

The FEM is reviewed in many textbooks (see Jin 2002); its application to brain electric field modeling has a long history (see Bertrand et al., 1991). The finite element matrix is sparse; its storage may require as low as $O(NlogN)$ memory complexity and the direct FEM solution (LU factorization) may require as low as $O(Nlog^2N)$ operations, where N is the number of volumetric elements, i.e. tetrahedra or hexahedra (Liu and Jiao 2010).

On the other hand, the boundary element matrix is dense; its storage requires N^2 memory complexity and the direct BEM solution (LU factorization) requires $O(N^3)$ operations, where N is the number of boundary facets (triangles or quadrilaterals) (Volakis and Sertel 2012).

Therefore, the FEM outperforms the original BEM for high-resolution cortical models with high N. The FEM is claimed to have a high numerical accuracy and it is additionally applicable to anisotropic conductivity distributions (Piastra et al., 2018).

2.2. Boundary Element Fast Multipole Method

2.2.1 Essence of the Fast Multipole Method

The fast multipole method introduced by Rokhlin and Greengard (Rokhlin 1985; Greengard and Rokhlin 1987) speeds up computation of a matrix-vector product by many orders of magnitude. Such a matrix-vector product naturally appears when a field from many point sources in space has to be computed at many observation or target points. The elementary sources can be either point electric charges or infinitesimally short electric current elements. In the first case, we compute the electric field and in the second case, the electric and magnetic fields. The same matrix-vector product appears when the system of equations of the BEM is solved iteratively. In terms of the BEM surface charge formulation used in this study, the iterative method is equivalent to repetitive computations of the electric field created by a large ensemble of surface charges (sources) at a large number of target points on the same surface. The fast multipole method eliminates the need for a large-size dense BEM system matrix and, in this sense, is the "matrix-free" method.

The idea of the fast multipole method is illustrated in Fig. 1 (Nabors and White 1991). In Fig. 1a (the multipole expansion), an obvious approach to determine the electric field vector at N target points due to M source charges requires $3N \cdot M$ operations. However, an accurate approximation can be computed in many fewer operations by exploiting the fact that $r \gg R$. The first step is to replace all charges within the inner circle by a single charge equal to their sum $\sum q_i$ and placed at the inner circle's center. This is the monopole approximation and is the first term of the general multipole ex pansion. It requires only $3N + M$ operations. The second step would be to find an equivalent dipole moment of the charge distribution, $\sum q_i \, (\mathbf{c}_i - \mathbf{c}_{center})$, within the inner circle with center $\mathbf{c}_{center}$, and then add the field of the equivalent dipole. Further, we compute a quadrupole contribution, etc. Since the sum of two very large numbers times a moderate factor is much smaller than the product of those numbers, the advantage of this approach is evident.

A complement to the multipole expansion is the local expansion shown in Fig. 1b. Here, the multipole expansions cannot be used. However, it is still possible to speed up the problem by noting that the field at any of the target points is approximately the same as the field at the inner circle's center. To compute the field at the inner circle's center we thus need $3M$ operations plus another $3N$ operations to copy this result to all target points. Just as in the multipole case, it is possible to improve the accuracy of the above local expansion by including the effect of the distance between a target point and the circle's center, etc.

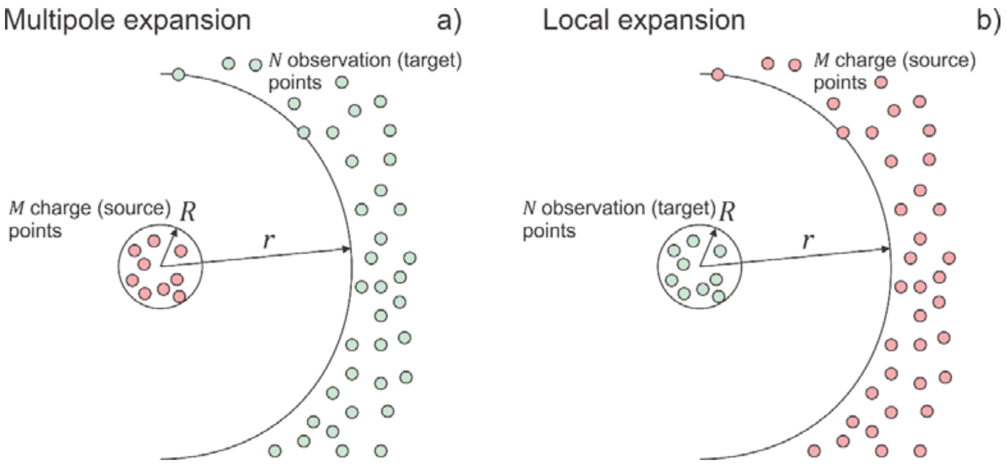

Fig. 1 Concept of the fast multipole method: multipole expansion and local expansion.

In order to achieve situations shown in Fig. 1, the entire computational domain has to be partitioned into a tree. A root or Level 0 cube is a cube that encloses the entire computational domain. The root cube is subdivided into eight equal smaller cubes, which correspond to Level 1. Each of the Level 1 cubes is then subdivided into eight smaller cubes, which correspond to Level 2, etc. The number of levels is usually higher than 4. Both multipole and local expansions in Fig. 1 are obtained in the form of a truncated series of spherical harmonics or multipoles, which include spherical Bessel functions and Legendre polynomials. These series are written in a local coordinate system. They need to be transformed to a global coordinate system while traversing the tree. Such a transformation requires series reassembly; it is a difficult mathematical operation. While it is relatively straightforward to design a demonstration version of the fast multipole method, its typical fast and accurate implementation is a complicated mathematical task.

2.2.2 Integration of fast multipole method and boundary element method

A proper integration of the fast multipole method (FMM) and the BEM is also a nontrivial operation. Our formulation is based on three new steps (Makarov et al., 2018[26],[27],[28]):

1. We write an integral equation of the BEM in terms of electric charges at the boundaries (Makarov et al., 2015). This is in contrast to the traditional BEM approach operating in terms of the electric potential (Geselowitz 1967; Meijs et al., 1989; Hämäläinen et al., 1993; Meijs et al., 1989; Mosher et al., 1999; Gramfort et al., 2014; Tadel et al., 2011; Gramfort et al., 2010; Stenroos et al., 2007; Stenroos and Sarvas 2012; Stenroos and Nummenmaa 2016; Nummenmaa et al 2013, etc.).

Our charge-based formulation allows us to use simple piecewise-constant or “pulse” basis functions. The electric potential of any piecewise-constant charge density is a continuous function everywhere in space. Therefore, we still achieve electric-potential continuity, which would otherwise require more complicated linear basis functions if the original potential formulation were used. Using the pulse bases greatly simplifies the following step.

2. We accurately compute all near-field interactions via precise analytical field integrals for neighbor triangular facets. For piecewise-constant bases, these integrals are well documented in the literature (see, for example, Wilton et al., 1984; Wang et al., 2003; Makarov et al., 2015). They have been tested, computed, and incorporated into the method. Neglecting accurate near-field interactions leads to a poor convergence of an iterative solution or to no convergence at all.

3. Last but not least, we adopt, integrate, and use an efficient and proven version of the FMM (Gimbutas and Greengard 2015) originating from its inventors. In this version, there is no a priori limit on the number of levels of the FMM tree, although after about thirty levels there may be floating point issues (L Greengard, private communication). The required number of levels is determined by a maximum permissible least-squares error, which is specified by the user. This FMM version allows for a straightforward inclusion of a controlled number of analytical neighbor integrals to be precisely evaluated, which appears to be a critical point for the convergence rate of our method.

These three steps combined make it possible to construct the algorithm which is able to compete with the finite element in terms of both accuracy and speed. This is in contrast to the previous unsuccessful attempt to implement the boundary element fast multipole method for brain modeling which was made over a decade ago (Kybic et al., 2005).

2.3 Comparison Testbed

2.3.1 Geometry

The comparisons are carried out in models consisting of four-layered spheres, as adopted from Engwer et al., 2017 and Piastra et al., 2018. Although both these references are concerned with MEG and EEG dipoles, the corresponding models are equally applicable to the present problem, which is also closely related to the MEG problem (Sarvas 1987).

Fig. 2a shows the problem geometry. The conductivity values are consistent with Engwer et al., 2017 and Piastra et al., 2018. To assure the test-grade surface triangulation, we first create *six* individual high-quality triangular base sphere meshes with the number of triangular facets ranging from approximately 0.011 M to 0.411 M (from lower to higher mesh density), using a high-quality surface mesh generator developed by Persson and Strang (Persson 2005; Persson and Strang 2004), and implemented in MATLAB®. The minimum triangle mesh quality (twice the ratio of inscribed to circumscribed circle radii for a triangular facet) is no less than 0.7, so that all the triangular facets are nearly equilateral. Also, all triangles have nearly the same size/area.

After that, we create six respective multi-sphere models by cloning and scaling every individual sphere mesh four times, as required by Fig. 2a. These “onion” models will be labeled #1 through #6. Additionally, every triangulated subsurface is also slightly scaled outwards so that its total area is exactly the sphere area with the prescribed radius. Table 1 lists the corresponding surface mesh resolution (or model resolution) and the mesh density (number of nodes per unit area) in the set of models. The mesh resolution is defined as the average edge length. The mesh density is given in nodes/mm^2, which is a common measure in SimNIBS.

Table. 1. Model resolution and mesh density in every four-layer sphere model.

Model #	Facets total	Mesh resolution, mm	Mesh density, nodes/mm^2
1	0.06 M	4.2	0.07
2	0.12 M	2.9	0.14
3	0.24 M	2.0	0.28
4	0.47 M	1.4	0.55
5	1.03 M	0.98	1.21
6	2.06 M	0.69	2.41

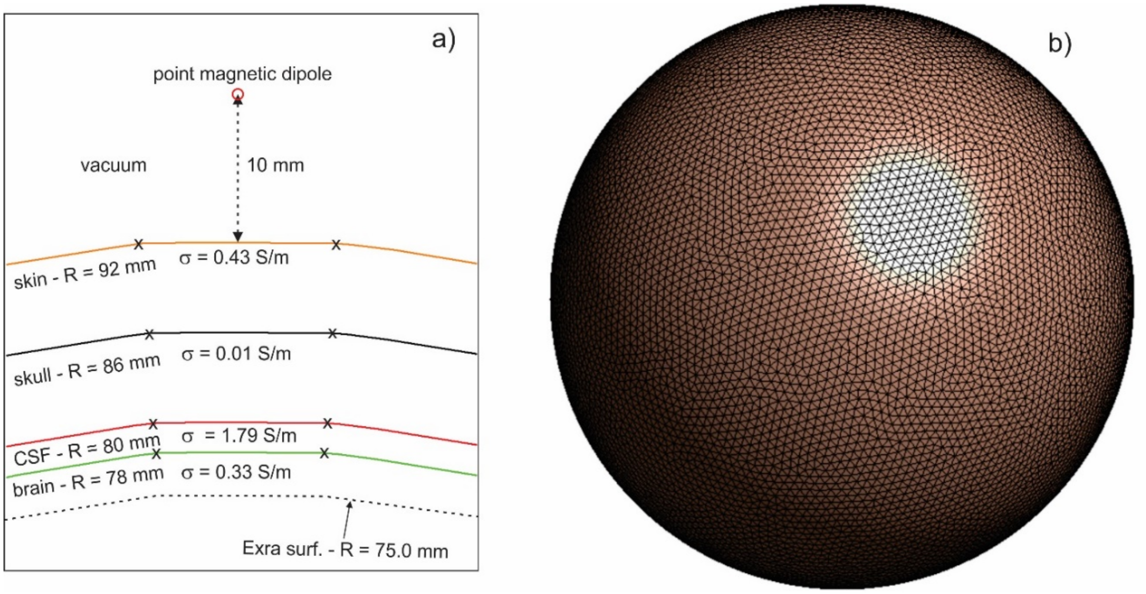

Fig. 2. a) – Model geometry; b) – surface mesh topology for sphere #2 with the mesh resolution of 2.9 mm and the mesh density of 0.14 nodes/mm^2.

The field error is measured on two observation sphere surfaces. One of them is located 0.5 mm below the brain surface in Fig. 2a and has the radius of 77.5 mm. Another is located 1.5 mm below the brain surface in Fig. 2a and has the radius of 76.5 mm.

Note that the FEM may have an insufficient resolution in regions where volumetric mesh density is low. In order to provide a fair comparison, and assure the proper and sufficient FEM volumetric meshing in this observation domain, we introduce a fifth sphere with the radius of 75 mm into the model as shown in Fig. 2a. This sphere is dummy: its conductivity is equal to the brain conductivity of 0.33 S/m, so that the corresponding conductivity contrast,

$$\Sigma = \frac{\sigma_e - \sigma_i}{\sigma_e + \sigma_i} \tag{1}$$

with σ_e/σ_i being external/internal medium conductivities, is equal to zero. However, this dummy sphere is explicitly present in both FEM and BEM-FMM models in Table 1. The mesh size of the combined model (four nontrivial brain compartments plus one dummy sphere) ranges from 0.06 M to 2.06 M facets in Table 1.

2.3.2 Excitation

The excitation is given by a point magnetic dipole (a small loop of current), schematically shown in Fig. 2a, and located 10 mm above the skin surface. A magnetic dipole with the moment $\boldsymbol{m}(t)$ located at point $\boldsymbol{r}_2$ generates the magnetic vector potential given by,

$$\boldsymbol{A}^{primary}(\boldsymbol{r}_1, t) = \frac{\mu_0 \boldsymbol{m}(t) \times (\boldsymbol{r}_1 - \boldsymbol{r}_2)}{4\pi|\boldsymbol{r}_1 - \boldsymbol{r}_2|^3} \tag{2}$$

where $\boldsymbol{r}_1$ is an arbitrary observation point and μ_0 is the magnetic permeability of vacuum. From Eq. (2), the solenoidal electric field of the dipole in free space becomes

$$\boldsymbol{E}^{primary}(\boldsymbol{r}_1,t) = -\frac{\partial \boldsymbol{A}^{primary}}{\partial t} = -\frac{\mu_0 \partial \boldsymbol{m}/\partial t \times (\boldsymbol{r}_1 - \boldsymbol{r}_2)}{4\pi|\boldsymbol{r}_1 - \boldsymbol{r}_2|^3} \tag{3}$$

Further, we assume harmonic excitation of the form $\boldsymbol{m}(t) = \boldsymbol{m}_0\exp(+j\omega t)$, convert to phasors, and eliminate the redundant constant phase factor of j using multiplication by j. This gives us the "static" real-valued excitation field

$$\boldsymbol{E}^{primary}(\boldsymbol{r}_1) = \frac{\mu_0 \omega \boldsymbol{m}_0 \times (\boldsymbol{r}_1 - \boldsymbol{r}_2)}{4\pi|\boldsymbol{r}_1 - \boldsymbol{r}_2|^3} \tag{4}$$

which could indeed be treated as a result of the separation of the time dependence and the spatial dependence, respectively. For demonstration purposes, Fig. 3 illustrates the dipole electric field distribution in a plane without (Fig. 3a), and with (Fig. 3b), a single conducting sphere due to the dipole excitation, as well as the corresponding electric potential. From Fig. 3, we see that the induced charge distribution on the sphere surface substantially reduces the field within the sphere.

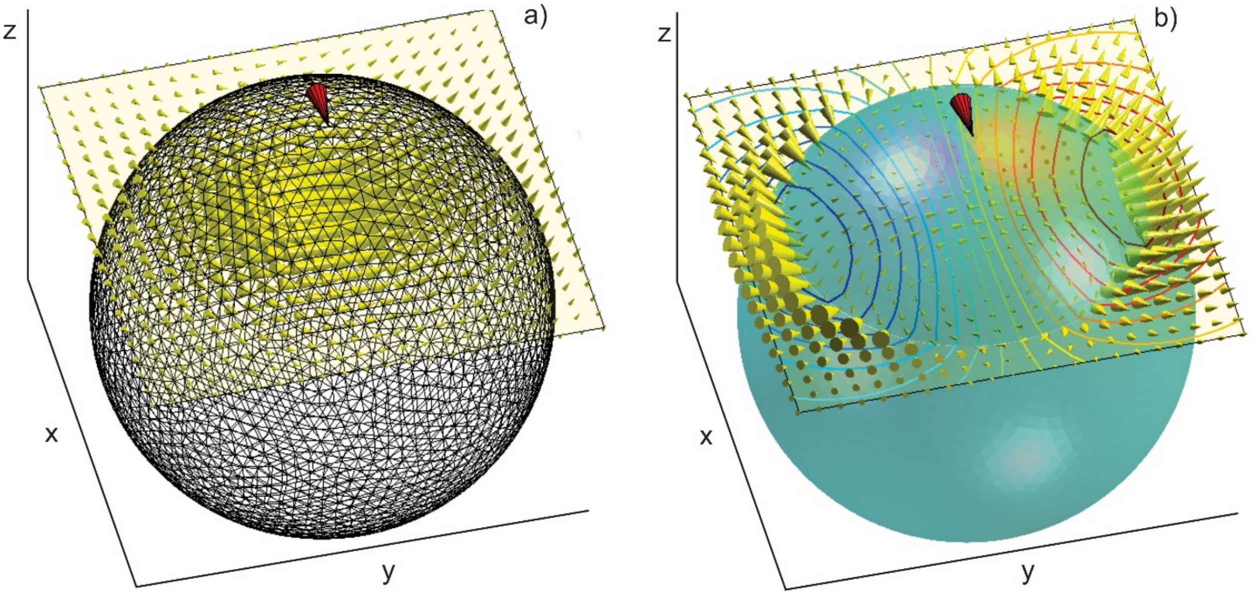

Fig. 3. Total electric field without (a) and with (b) a conducting sphere due to the dipole excitation and equipotential lines in a plane. The surface charge distribution is also seen in (b).

2.3.3 Analytical solution

For a dipole outside a spherical model with a spherically-symmetric conductivity distribution, the corresponding analytical solution neither depends on the individual sphere radii nor on the specific conductivity values (Sarvas 1987). The same manipulations that lead to Eq. (4) allow us to obtain from Ref. Sarvas 1987 an expression for the total field $\boldsymbol{E}$ in the form:

$$\begin{aligned} &\boldsymbol{E}(\boldsymbol{r}_1) = \frac{\omega\mu_0}{4\pi F^2}[F(\boldsymbol{r}_1 \times \boldsymbol{m}_0) - (\boldsymbol{m}_0 \cdot \nabla_2 F)(\boldsymbol{r}_1 \times \boldsymbol{r}_2)] \\ &F = |\boldsymbol{a}|(|\boldsymbol{r}_2||\boldsymbol{a}| + \boldsymbol{r}_2 \cdot \boldsymbol{a}), \qquad \boldsymbol{a} = \boldsymbol{r}_2 - \boldsymbol{r}_2 \\ &\nabla_2 F = \left(\frac{|\boldsymbol{a}|^2}{|\boldsymbol{r}_2|} + 2|\boldsymbol{a}| + \boldsymbol{2}|\boldsymbol{r}_2| + \frac{\boldsymbol{r}_2 \cdot \boldsymbol{a}}{|\boldsymbol{a}|}\right)\boldsymbol{r}_2 - \left(|\boldsymbol{a}| + \boldsymbol{2}|\boldsymbol{r}_2| + \frac{\boldsymbol{r}_2 \cdot \boldsymbol{a}}{|\boldsymbol{a}|}\right)\boldsymbol{r}_1 \end{aligned} \tag{5}$$

where $\boldsymbol{r}_1$ is now an arbitrary observation point *within* the sphere model. The following short MATLAB script implements Eqs. (5):

```
function E = analytmagndipole(f, m0, r2, R1)
%   Inputs:
% f - frequency, Hz
% m0 - magnetic dipole moment, A*m^2
% r2 - dipole position, 1x3, m
% R1 - Mx3 array of positions within the conducting sphere(s) where the field
%   is plotted
%   Output(s):
% E - electric field within the sphere
    mu0         = 1.25663706e-006;
    M = size(R1, 1); E = zeros(M, 3);
    omega = 2*pi*f;
    for m = 1:M
        r1 = R1(m, :);  %   Single observation point
        a   = r2  - r1;
        as  = sqrt(dot(a, a));
        r2s = sqrt(dot(r2, r2));
        F   = as*(r2s*as + dot(r2, a));
        G2F = (as^2/r2s + 2*as + 2*r2s + dot(r2, a)/as)*r2 - ...
              (as + 2*r2s + dot(r2, a)/as)*r1;
        E(m, :) = +omega*mu0/(4*pi*F^2)*(F*cross(r1, m0) ...
                  - dot(m0, G2F)*cross(r1, r2));
    end
end
```

2.3.4 Error measure

Once the analytical solution is available, we compute the error in the total electric field using the relative least squares difference, that is

$$RDM\left(\boldsymbol{E}_{num}, \boldsymbol{E}_{analyt}\right) = \frac{\left\|\boldsymbol{E}_{num} - \boldsymbol{E}_{analyt}\right\|}{\left\|\boldsymbol{E}_{analyt}\right\|} \tag{6}$$

In all 6 models, we observed the electric field at 47,500 triangle barycenters of the observation sphere, with the radius of either 77.5 mm or 76.5 mm, respectively, generated as explained above in the text. This observation sphere has the surface mesh resolution of 2 mm and the mesh density of 0.28 nodes/mm^2.

3. Results

3.1. Finite Element Method Performance (`getDP` solver used in SimNIBS 2.1.1)

All results reported in this section are obtained using the same server: Intel(R) Xeon(R) E5-2690 CPU at 2.90 GHz, and using the same operation system, Red Hat Enterprise Linux 7.5, with simulations running on a single processor. SimNIBS 2.1.1 employs the default FEM solver: the open sourceFEM software package called `getDP` (see Reference manual 2017). In SimNIBS, getDP is configured to use the PETSc conjugate gradient (CG) solver with a relative residual of 1e-9 and the incomplete Cholesky (ICC) preconditioner with 2 factor levels.

After the FEM solution is completed, the field interpolation for arbitrary points in space is done using “super-convergent approach” (or SCA) recently implemented in SimNIBS 2.1.1. In this approach, the original tessellation is preserved, and the electric field at the nodes is interpolated from the electric field values at the tetrahedra centers. Afterwards, further linear interpolation for arbitrary observation points is performed using the original tessellation (Zienkiewicz and Zhu 1992).

The BEM-FMM approach performs precise analytical integration over the 12 closest neighbor facets. For postprocessing computations of the volumetric E-field from the known charge distribution, we do not use any analytical integration. The tolerance level `iprec` of the FMM algorithm is set at 0 (the relative least-squares error is guaranteed no to exceed 0.5%). The FMM is a FORTAN 90/95 program compiled for MATLAB. In this work, we use a native MATLAB GMRES (generalized minimum residual) iterative algorithm written by Drs. P. Quillen and Z. Hoffnung of MathWorks, Inc. Although this method may be somewhat slower than a simplified in-house version of the GMRES, its overall performance and convergence are excellent, especially for complicated head geometries. The relative residual of the BEM-FMM iterative method is set as 1e-4.

3.1.1 Solution performance

Table 2 presents the run times for the FEM solution and the corresponding relative average error in the electric field computation, respectively. We consider two distinct observation spheres located 0.5 mm and 1.5 mm below the brain surface in Fig. 2a, respectively. The `getDP` FEM software was unable to process the largest problem with 2.06 M triangles on the server used in this study. Therefore, the last row of Table 2 is left blank.

Table. 2. Speed and accuracy of the `getDP` solver within the SimNIBS 2.1.1 environment. The observation sphere is located 0.5 mm and 1.5 mm below the brain surface in Fig. 2a, respectively and has the radius of either 77.5 mm or 76.5 mm.

Model #	Facets total	FEM solution time, sec	FEM least squares error (using SCA) 0.5 mm below the brain surface, %	FEM least squares error (using SCA) 1.5 mm below the brain surface, %
1	0.06 M	8.13	6.5	2.9
2	0.12 M	34.35	6.1	2.5
3	0.24 M	94.87	5.8	2.0
4	0.47 M	458.2	3.0	0.58
5	1.03 M	1765.85	2.0	0.43
6	2.06 M	NA	NA	NA

3.1.2 Mesh generation times

The FEM method requires volumetric mesh generation. This operation has to be done only once for each model, but it may require a significant amount of time. Table 3 reports volumetric mesh generation times in SimNIBS 2.1.1 for the six multi-sphere models. The mesh generation process is not parallelized. In SimNIBS 2.1.1, the volume meshing is performed using Gmsh (Geuzaine et al., 2009) using the frontal algorithm implemented in Tetgen (Si 2015).

Table. 3. Volumetric mesh generation in SimNIBS 2.1.1.

Model #	Facets total	Tetrahedra total (in the volumetric mesh)	Meshing time (volumetric mesh generation time), sec
1	0.06 M	0.26M	7.05
2	0.12 M	0.72M	21.15
3	0.24 M	1.8M	66.04
4	0.47 M	5.05M	219.32
5	1.03 M	16.43M	988.65
6	2.06 M	44.64M	2678.06

3.2. BEM-FMM Performance (MATLAB 2018a Platform)

All results reported in this section and the following sections are also obtained using the same server: Intel(R) Xeon(R) E5-2690 CPU at 2.90 GHz, and using the same operation system: Red Hat Enterprise Linux 7.5, with simulations running on a single processor. We run MATLAB version 2018a Linux.

3.2.1 Solution performance and number of iterations

The accuracy of the iterative BEM-FMM solution depends on type of the iterative solver and the number of iterations used. The number of iterations is controlled by the prescribed value of the relative residual (a relative deviation from zero of the corresponding integral equation, after substitution of the approximate solution). We emphasize that the relative residual error is not the solution error itself, although there is a correlation (sometimes weak) between both of them. A larger number of iterations results in a better accuracy but requires more CPU time.

Table 4 reports BEM-FMM solution times for two prescribed values of the relative residual: 1e-4 and 1e-3, respectively, and using a single processor. It is seen that the former solution is more accurate, although both solutions still provide a better accuracy than the FEM method in Table 2. The observation sphere is again located 0.5 mm below the brain surface in Fig. 2a, and has a radius of 77.5 mm.

Table. 4. Effect of the number of iterations (which is determined by the limiting value of the relative residual) on the BEM-FMM accuracy and solution time. Server: Intel(R) Xeon(R) E5-2690 CPU (2.90 GHz), running on a single processor. The observation sphere is located 0.5 mm below the brain surface in Fig. 2a, and has the radius of 77.5 mm.

Model #	Facets total	BEM-FMM solution time for 1e-4 residual, sec	BEM-FMM lest squares error for 1e-4 residual, %	BEM-FMM solution time for 1e-3 residual, sec	BEM-FMM least squares error for 1e-3 residual, %
1	0.06 M	17	2.7	11	3.5
2	0.12 M	38	1.6	26	2.6
3	0.24 M	67	0.76	44	1.32
4	0.47 M	145	0.33	97	1.24
5	1.03 M	299	0.24	202	1.23
6	2.06 M	517	0.18	338	1.23

3.2.2 Times necessary for computations of potential integrals

While the FEM requires volumetric mesh generation, the BEM-FMM requires precomputing and storing potential integrals for the neighbor triangles. The number of neighbors is typically 4-16. This operation has to be done only once. It is based on a `for`-loop over all triangular facets and is trivially parallelizable in MATLAB using the `parfor` syntax. Table 5 reports execution times for potential-integral computations and writing data to file, given 12 neighbors, and using the `parfor`-loop with 16 cores (`parpool(16)`) in MATLAB.

Table. 5. Times necessary for precomputing and storing potential integrals for the BEM-FMM algorithm, given 12 neighbors, and using the the `parfor`-loop with 16 cores (`parpool(16)`) in MATLAB.

Model #	Facets total	Potential integrals computation time, sec	Saving `*.mat` file, sec
1	0.06 M	35	1
2	0.12 M	42	2
3	0.24 M	55	3
4	0.47 M	82	6
5	1.03 M	143	11
6	2.06 M	253	19

3.3. Error close to the surface vs. error at larger distances from the surface

Table 6 compares the error of the FEM and the BEM-FMM approaches when the observation sphere is located farther away, namely at 1.5 mm below the brain surface in Fig. 2a, and has a radius of 76.5 mm. It is seen that the FEM error very significantly improves, while the BEM-FMM error remains nearly unchanged, when compared in the case of the observation points located closer to the brain surface.

Table. 6. Accuracy of the `getDP` solver within SimNIBS 2.1.1 environment and the BEM-FEM engine when the observation sphere is located at 1.5 mm below the brain surface in Fig. 2a.

Model #	Facets total	FEM least squares error (using SCA), %	BEM-FMM lest squares error for 1e-4 residual, %
1	0.06 M	2.91	2.8
2	0.12 M	2.52	1.7
3	0.24 M	1.98	0.74
4	0.47 M	0.58	0.33
5	1.03 M	0.43	0.24
6	2.06 M	NA	0.18

3.4. Post-processing times

Here, we compare the speed of the super-convergent approach (or SCA) recently implemented in SimNIBS 2.1.1 and the BEM-FMM field restoration algorithm for the observation sphere located at 1.5 mm below the brain surface in Fig. 2a. Table 7 presents the corresponding run-times for a single processor. Since the potential integrals are not computed at this stage, the BEM-FEM is reduced to the plain FMM and is therefore very fast. If the potential field integrals were included, the post-processing time would increase by about 1 min without a significant effect on the solution accuracy.

We emphasize that the SCA algorithm is not yet optimized, and that it is possible to construct an interpolant, and store it to very significantly speed up future computations.

Table. 7. Post-processing run times of the SCA and the BEM-FEM engine when the observation sphere is located at 1.5 mm below the brain surface in Fig. 2a.

Model #	Facets total	FEM SCA speed, sec	BEM-FMM field restoration speed, sec
1	0.06 M	3.9	1.4
2	0.12 M	8.4	1.4
3	0.24 M	12	1.6
4	0.47 M	40	2.2
5	1.03 M	165	4.2
6	2.06 M	NA	4.5

3.4. Operation system and hardware performance for the BEM-FMM engine

In this last section, we compare the BEM-FMM performance using Linux- and Windows-based machines:

A. Intel(R) Xeon(R) E5-2690 CPU at 2.90GHz, Red Hat Enterprise Linux 7.5; MATLAB 2018a Linux

B. Intel Xeon E5-2698 v4 CPU at 2.20GHz, Windows Server 2016 Standard; MATLAB 2018a Windows

Table 8 presents BEM-FMM solution run times for server A (Linux) and server B (Windows) for the prescribed value of the relative residual equal to 1e-4. Server B significantly outperforms Server A, most likely due to the increased L1 cache size and multithreading capabilities of the E5-2698 vs the E5-2690.

Table 8. BEM-FMM solution time for servers A (Linux) and B (Windows).

Model #	Facets total	BEM-FMM solution time for server A, sec	BEM-FMM solution time for server B, sec
1	0.06 M	17	9
2	0.12 M	38	25
3	0.24 M	67	44
4	0.47 M	145	97
5	1.03 M	299	219
6	2.06 M	517	378

4. Discussion and Conclusion

Fig. 4 below summarizes the results from Tables 2, 4, and 6 for both methods, and for the exactly matched computational performance metrics. For the FEM solution, the superconvergent interpolation is used. For the BEM-FMM solution, the prescribed value of the relative residual is equal to 1e-4. Fig. 4a shows the corresponding simulation times of the main algorithm. The BEM-FMM algorithm starts to outperforms the FEM algorithm when the number of surface facets exceeds approximately 100,000. We observe that the BEM-FMM method runs much faster for high-resolution models.

Fig. 4b shows the least-squares errors in the electric field for the observation surface located 1.5 mm under the “brain” surface in Fig. 2a. We observe that the BEM-FMM method gives a smaller solution error for all mesh resolutions. The result does not change significantly when the observation surface is moved farther away from the brain interface.

Fig. 4c shows the least-squares errors in the electric field for the observation surface located 0.5 mm under the “brain” surface in Fig. 2a. We observe that the BEM-FMM method gives a much smaller solution error for all mesh resolutions.

The speed advantage of the BEM-FMM algorithm also holds for the pre- and post-processing steps as evidenced by Tables 3, 5, and 7. The speed of the BEM-FMM algorithm can further be improved by switching from default complex arithmetic to real arithmetic (a factor of 2), and speeding up the MATLAB-based `for`-loops for the inclusion of the potential integrals.

However, the FMM part of the BEM-FMM algorithm is quite nontrivial in implementation. The BEM part, on the other hand, relies upon tuning several parameters in order to obtain a good convergence.

In contrast to this, the FEM algorithm has been extensively studied and applied across various engineering disciplines for decades; there are many highly reliable solvers available. Also, an application-specific implementation of FEM, coupled with novel solvers such as algebraic multigrid (AMG) preconditioners (Henson et al., 2002), can significantly speed up calculations when compared with the general FEM environment and a classic multipurpose solver, with no loss of accuracy or stability. Another issue is that of accessibility. The present implementation of the BEM-FMM relies upon proprietary software (MATLAB®), while SimNIBS relies on the open-source software only.

Finally, it is not yet clear how both methods will compare to each other for realistic head models. In this case, a large node density is required to appropriately capture the complex geometry of patient-specific intracranial surfaces. Relatively small curvature radii of the characteristic folds of the cerebral cortex may constitute difficulties for both methods. A detailed and separate comparison study may therefore be necessary.

Acknowledgements

The authors wish to thank Dr. Leslie Greengard of the Courant Institute of Mathematical Sciences, New York, NY for useful remarks. The authors are thankful to Dr. Angel Peterchev of Duke University, Durham, NC for his constructive criticism. The authors are also very thankful to Dr. Gregory Noetscher of Worcester Polytechnic Institute, Worcester MA for his continuous support. Dr. Axel Thielscher of Technical University of Denmark and the Danish Research Centre for Magnetic Resonance greatly helped in initiating and shaping this study. This work has been partially supported by the National Institutes of Health Grant R01MH111829, Novo Nordisk Fonden (grant Nr. NNF14OC0011413), and Lundbeckfonden (grant Nr. R118-A11308).

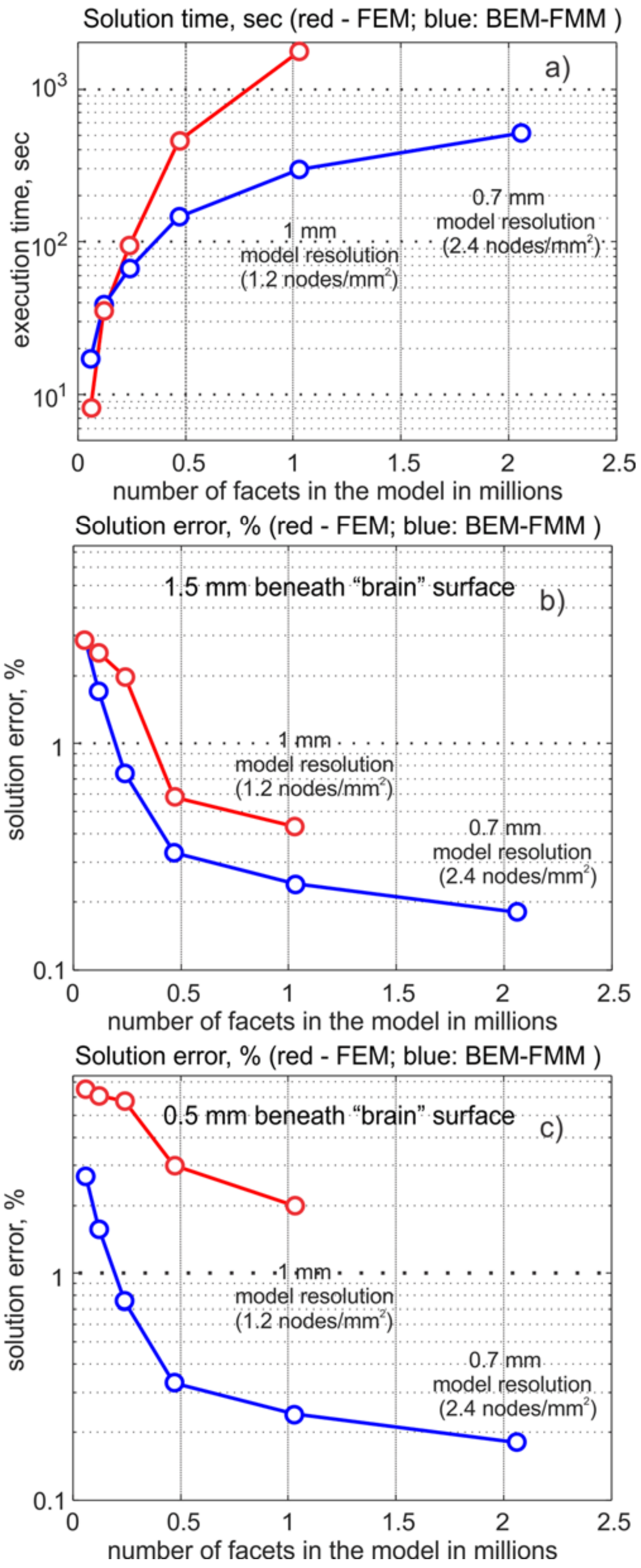

Fig. 4. a) – Solution time; b) – solution error of the FEM and BEM-FMM algorithms, respectively, both as functions of the number of facets in the model (model resolution and/or mesh density) at 1.5 mm beneath the "brain" surface; c) – the same result at 0.5 mm beneath the "brain" surface.

References

[1] Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, *PETSc Users Manual*. Argonne National Laboratory. 2018. ANL-95/11 - Revision 3.9. Online: http://www.mcs.anl.gov/petsc.

[2] Bangerth W, Hartmann R, Kanschat, G. Deal.II—A general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)*. 2007 Aug;33(4). Article #24. doi: 10.1145/1268776.1268779.

[3] Bertrand O, Thevenet M, Perrin F. 3-D finite element method in brain electrical activity studies. *Biomagnetic Localization and 3D Modeling*. 1991. Report TKK-F-A689, eds J. Nenonen, H.-M. Rajala, and T. Katila (Espoo: Helsinki University of Technology, Department of Technical Physics, Laboratory of Biomedical Engineering).

[4] Bikson M, Brunoni AR, Charvet LE, Clark VP, Cohen LG, Deng ZD, Dmochowski J, Edward DJ, Frohlich F, Kappenman ES, Lim KO, Loo C, Mantovani A, McMullen DP, Parra LC, Pearson M, Richardson JD, Rumsey JM, Sehatpour P, Sommers D, Unal G, Wassermann EM, Woods AJ, Lisanby SH. Rigor and reproducibility in research with transcranial electrical stimulation: An NIMH-sponsored workshop. *Brain Stimul*. 2018 May - Jun;11(3):465-480. doi: 10.1016/j.brs.2017.12.008. Epub 2017 Dec 29.

[5] Dular, P, Geuzaine, C, Henrotte, F, Legros, W. A general environment for the treatment of discrete problems and its application to the finite element method. *IEEE Transactions on Magnetics*. 1988 Sep;34(5):3395–3398. doi: 10.1109/20.717799.

[6] Engwer C, Vorwerk J, Ludewig J, Wolters CH. A Discontinuous Galerkin Method to Solve the EEG Forward Problem Using the Subtraction Approach. *SIAM J. Sci. Comput*. 2017;39(1):B138–B164. doi: 10.1137/15M1048392.

[7] Ferguson AS, Zhang X, Stroink G. A Complete Linear Discretization for Calculating the Magnetic Field Using the Boundary Element Method. *IEEE Trans Biomed Eng*. 1994;41(5): 455–460. doi: 10.1109/10.293220. PMID: 8070805.

[8] Geselowitz DB. On bioelectric potentials in an inhomogeneous volume conductor. *Biophys. J*. 1967 Jan;7(1):1-11. doi: 10.1016/S0006-3495(67)86571-8. PMID: 19210978.

[9] GetDP. Reference Manual. Nov. 5th 2017. Online: http://getdp.info/doc/texinfo/getdp.pdf.

[10] Geuzaine C, Remacle JF. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Num. Methods Eng*. 2009 May 7; 79(11):1309–1331. doi: 10.1002/nme.2579.

[11] Gimbutas Z, Greengard L. Simple FMM Libraries for Electrostatics, Slow Viscous Flow, and Frequency-Domain Wave Propagation. *Communications in Comput. Phys*. 2015;18(2):516-528. Online: https://dx.doi.org/10.4208/cicp.150215.260615sw

[12] Gramfort A, Luessi M, Larson E, Engemann DA, Strohmeier D, Brodbeck C, Parkkonen L, Hämäläinen MS. MNE software for processing MEG and EEG data. *NeuroImage*. 2014 Feb 1; 86:446-60. doi: 10.1016/j.neuroimage.2013.10.027.

[13] Gramfort A, Papadopoulo T, Olivi E, Clerc M. OpenMEEG: open source software for quasistatic bioelectromagnetics. *BioMedical Engineering Online*.2010 Sep 6;9, article 45. doi: 10.1186/1475-925X-9-45.

[14] Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J. Comput. Phys.* 1997 August; 135(2):280-292. doi: 10.1006/jcph.1997.5706.

[15] Hämäläinen MS, Hari R, Ilmoniemi RJ, Knuutila J, Lounasmaa OV. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of Modern Physics*. 1993 April 1; 65:413. doi: 10.1103/RevModPhys.65.413.

[16] Hämäläinen MS, Sarvas J. Realistic conductivity geometry model of the human head for interpretation of neuromagnetic data. *IEEE Trans. Biomed. Eng.* 1989 Feb;36(2):165-171. doi: 10.1109/10.16463. PMID: 2917762.

[17] Henson VE, Yang UM. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *J Appl. Numerical Math.* 2002 April; 41(1):155-177. doi: 10.1016/S0168-9274(01)00115-5.

[18] Huang Y, Datta A, Bikson M, Parra LC. Realistic Volumetric-Approach to Simulate Transcranial Electric Stimulation - ROAST - a fully automated open-source pipeline. *bioRxiv Preprint.* 2017 Nov 10; doi: 10.1101/217331.

[19] Jin J. *The finite element method in electromagnetics*. 2nd Ed. New York: John Wiley & Sons; 2002.

[20] Jung YJ, Kim JH, Im CH. COMETS: A MATLAB toolbox for simulating local electric fields generated by transcranial direct current stimulation (tDCS). *Biomed. Eng. Lett*. 2013 Mar;3(1):39–46. doi: 10.1007/s13534-013-0087-x.

[21] Kybic J, Clerc M, Faugeras O, Keriven R, Papadopoulo T. Fast multipole acceleration of the MEG/EEG boundary element method. *Phys. Med. Biol*. 2005 Sep 21; 50:4695–4710. PMID: 16177498 doi: 10.1088/0031-9155/50/19/018.

[22] Lee C, Jung YJ, Lee SJ, Im CH. COMETS2: an advanced MATLAB toolbox for the numerical analysis of electric fields generated by transcranial direct current stimulation. *J Neurosci Meth*. 2017 Feb 1; 277:56–62. doi: 10.1016/j.jneumeth.2016.12.008.

[23] Lee EG, Duffy W, Hadimani RL, Waris M, Siddiqui W, Islam F, Rajamani M, Nathan R, Jiles DC. Investigational Effect of Brain-Scalp Distance on the Efficacy of Transcranial Magnetic Stimulation Treatment in Depression. *IEEE Trans. Magn.* 2016 July;52(7):1-4. doi: 10.1109/TMAG.2015.2514158.

[24] Liu H, Jiao D. H-Matrix-Based Fast Direct Finite Element Solver for Large-Scale Electromagnetic Analysis. *Purdue e-Pubs*. 2010. Purdue University. ECE Technical Reports. 2-1-2010. Online: https://docs.lib.purdue.edu/ecetr/396/

[25] Logg A, Mardal K, Wells, G. (Eds.). *Automated Solution of Differential Equations by the Finite Element Method. The FEniCS Book*. 2012. Springer. New York. ISBN: 3642230989.

[26] Makarov SN, Noetscher GM, Raij T, Nummenmaa A. A Quasi-Static Boundary Element Approach with Fast Multipole Acceleration for High-Resolution Bioelectromagnetic Models.

IEEE Trans. Biomed. Eng. 2018 March 07. doi: 10.1109/TBME.2018.2813261. Early Access: http://ieeexplore.ieee.org/document/8307445/.

[27] Makarov SN, Noetscher GM, Raij TA, Nummenmaa A, "Fast Multipole Method for Rapid Modeling of Transcranial Brain Stimulation Problems," *40th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society* (EMBC 2018), Honolulu, HI, July 17-21, 2018.

[28] Makarov SN, Noetscher GM, Sundaram P. Microscopic and Macroscopic Response of a Cortical Neuron to an External Electric Field Computed with the Boundary Element Fast Multipole Method. *bioRxiv Preprint*. 2018 Aug 13. doi: 10.1101/391060.

[29] Makarov SN, Noetscher GM, Nazarian A. *Low-Frequency Electromagnetic Modeling of Electrical and Biological Systems Using MATLAB*. 2015. Wiley, New York. ISBN: 978-1-119-05256-2.

[30] Meijs JW, Weier OW, Peters MJ, Van Oosterom A. On the numerical accuracy of the boundary element method. *IEEE Trans Biomed Eng*. 1989 Oct;36(10):1038-1049. doi: 10.1109/10.40805. PMID: 2793196.

[31] Mosher JC, Leahy RM, Lewis PS. EEG and MEG: forward solutions for inverse methods. *IEEE Trans Biomed Eng*. 1999 Mar;46(3):245-259. doi: 10.1109/10.748978. PMID: 10097460.

[32] Nabors K, White J. FastCap: A Multipole Accelerated 3-D Capacitance Extraction Program. IEEE Trans. Computer-Aided Design. 1009. 1991 Nov;10(11):1447-1459. doi: 10.1109/43.97624.

[33] Nielsen JD, Madsen KH, Puonti O, Siebner HR, Bauer C, Madsen CG, Saturnino GB, Thielscher A. Automatic skull segmentation from MR images for realistic volume conductor models of the head: Assessment of the state-of-the-art. *Neuroimage*. 2018 Jul 1;174:587-598. doi: 10.1016/j.neuroimage.2018.03.001. Epub 2018 Mar 12.

[34] Nummenmaa A, Stenroos M, Ilmoniemi RJ, Okada YC, Hämäläinen MS, and Raij T. Comparison of spherical and realistically shaped boundary element head models for transcranial magnetic stimulation navigation. *Clinical Neurophysiology*. 2013 Oct;124(10):1995–2007. doi: 10.1016/j.clinph.2013.04.019. Epub 2013 Jul 25.

[35] Opitz A, Paulus W, Will S, Antunes A, Thielscher A. Determinants of the electric field during transcranial direct current stimulation. *Neuroimage*. 2015 Apr 1;109:140-50. doi: 10.1016/j.neuroimage.2015.01.033. Epub 2015 Jan 19.

[36] Persson PO, Strang G. A Simple mesh generator in MATLAB. *SIAM Review*. 2004 June;46(2):329-345. doi: 10.1137/S0036144503429121.

[37] Persson PO. *Mesh Generation for Implicit Geometries*. 2005. PhD Thesis, MIT, 126 p. Online: http://persson.berkeley.edu/thesis/persson-thesis-color.pdf

[38] Piastra MC, Nüßing A, Vorwerk J, Bornfleth H, Oostenveld R, Engwer C, Wolters CH. The Discontinuous Galerkin Finite Element Method for Solving the MEG and the Combined MEG/EEG Forward Problem. *Frontiers in Neuroscience*. 2018 Feb 2;12:30. doi: 10.3389/fnins.2018.00030.

[39] Rokhlin V. Rapid Solution of Integral Equations of Classical Potential Theory. *J. Computational Physics*. 1985 Sep 15; 60(2):187–207. doi: 10.1016/0021-9991(85)90002-6.

[40] Saad Y. *Iterative Methods for Sparse Linear Systems*. 2nd edition, Society for Industrial and Applied Mathematics. 2003. ISBN 978-0-89871-534-7.

[41] Sarvas J. Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. *Phys. Med. Biol.* 1987 Jan; 32(1):11-22. PMID: 3823129.

[42] Schöberl J. NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Comput Visual Sci*. 1997 July. 1(1);41-52. doi: 10.1007/s007910050004.

[43] Si H. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. on Mathematical Software*. 41 (2), Article 11 (Feb. 2015), 36 pages. DOI=10.1145/2629697 http://doi.acm.org/10.1145/2629697.

[44] Stenroos M, Mäntynen V, Nenonen J. A MATLAB library for solving quasi-static volume conduction problems using the boundary element method. *Computer Methods and Programs in Biomedicine*, 2007 Dec;88(3):256-263. PMID 18022274. doi: 10.1016/j.cmpb.2007.09.004.

[45] Stenroos M, Nummenmaa A. Incorporating and Compensating Cerebrospinal Fluid in Surface-Based Forward Models of Magneto- and Electroencephalography. *PLoS One*, 2016 Jul 29;11(7):e0159595. doi: 10.1371/journal.pone.0159595.

[46] Stenroos M, Sarvas J. Bioelectromagnetic forward problem: isolated source approach revis(it)ed, *Phys Med Biol*. 2012 Jun 7;57(11):3517-35. doi: 10.1088/0031-9155/57/11/3517. Epub 2012 May 11.

[47] Tadel F, Baillet S, Mosher JC, Pantazis D, Leahy RM. Brainstorm: A User-Friendly Application for MEG/EEG Analysis. *Computational Intelligence and Neuroscience*. 2011; 2011:ID 879716:1-13. doi: 10.1155/2011/879716.

[48] *The Population Head Model Repository*. 2017. IT'IS Foundation website. doi: 10.13099/VIP-PHM-V1.0. Online: https://www.itis.ethz.ch/virtual-population/regional-human-models/phm-repository/

[49] Thielscher A, Antunes A, Saturnino GB. Field modeling for transcranial magnetic stimulation: A useful tool to understand the physiological effects of TMS?. *Conf Proc IEEE Eng Med Biol Soc*. 2015;222-5. doi: 10.1109/EMBC.2015.7318340.

[50] Van Essen DC, Ugurbil K, Auerbach E, Barch D, Behrens TE, Bucholz R, Chang A, Chen L, Corbetta M, Curtiss SW, Della Penna S, Feinberg D, Glasser MF, Harel N, Heath AC, Larson-Prior L, Marcus D, Michalareas G, Moeller S, Oostenveld R, Petersen SE, Prior F, Schlaggar BL, Smith SM, Snyder AZ, Xu J, Yacoub E. The Human Connectome Project: A data acquisition perspective. *NeuroImage*. 2012 Oct 1; 62(4):2222–31. doi: 10.1016/j.neuroimage.2012.02.018. PMID: 22366334. Epub 2012 Feb 17.

[51] Volakis JL, Sertel K. *Integral Equation Methods for Electromagnetics*, 2012 SciTech Publishing, Raleigh, NC, USA. ISBN-10: 1891121936.

[52] Wang Z, Volakis J, Saitou K, Kurabayashi K. Comparison of semi-analytical formulations and Gaussian-quadrature rules for quasi-static double-surface potential

integrals. *IEEE Antennas and Propagation Magazine.* 2003 Dec;45(6):96-102. doi: 10.1109/MAP.2003.1282185.

[53] Wilton DR, Rao SM, Glisson AW, Schaubert DH, Al-Bundak OM, Butler CM. Potential integrals for uniform and linear source distribution on polygonal and polyhedral domains. *IEEE Trans. Antennas and Propagation*. 1984 March;32(3):276-281. doi: 10.1109/TAP.1984.1143304.

[54] Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. 1992. *Int. J. for Num. Methods in Eng.* 1992 May 30;33(7):1331-1364. doi: 10.1002/nme.1620330702.