Development of the Boundary Element Fast Multipole Method for Quasistatic Electromagnetic Modeling of the Brain

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Electrical and Computer Engineering

by

William Wartman

May 2021

APPROVED:

Professor Aapo Nummenmaa

Doctor Kyoko Fujimoto

Doctor Konstantin Weise

Professor Sergey Makarov, Advisor

Abstract

In this thesis, several algorithmic improvements of the Boundary Element Fast Multipole Method (BEM-FMM) for quasistatic electromagnetic modeling of multi-tissue anatomical human models have been suggested and implemented. These improvements include:

- Fast solid-angle approach for neighbor E-field integral calculations – FMM implementation
- Fast cubatures for neighbor potential and E-field integral calculations – FMM implementation

In addition, several pre/post-processing improvements of the modeling pipeline have been suggested and implemented. They include:

- Automated detection and removal of coincident faces for meshes with duplicated boundaries;
- Approach for automated volumetric labeling for BEM problems with large surface meshes;

The application examples discussed in this thesis include:

- Simulation of the MIDA head model (with 11 M triangular surface elements and 100+ tissue compartments)
- Transcranial magnetic stimulation, transcranial electrical stimulation, and electroencephalography/magnetoencephalography modeling toolkits with the BEM-FMM

Appropriate MATLAB scripts are given in the text. The corresponding BEM-FMM modeling toolkits, along with the documentation and application examples (MATLAB platform), are available at the following locations:

Transcranial Magnetic Stimulation:	https://tmscorelab.github.io/TMS-Modeling-Website/
Transcranial Electrical Stimulation:	https://tmscorelab.github.io/TES-Modeling-Website/
EEG/MEG Forward Solver:	https://tmscorelab.github.io/EEG_MEG-Modeling-Website/

The software should run as is on Windows systems running MATLAB r2019a or newer.

Acknowledgements

I would like to express my most heartfelt thanks to Prof. Sergey Makarov for his continuous support, encouragement, motivation, and insight over the past four years. He has taught and challenged and believed in me at every turn, and his attentive oversight has shaped me into a far more mature, capable, and dependable researcher.

I sincerely thank Prof. Aapo Nummenmaa, Dr. Kyoko Fujimoto, and Dr. Konstantin Weise for spending their time to review my work, this text, and its associated presentation. I am also grateful to Prof. Gregory Noetscher and Dr. Mathias Davids for their valuable contributions to the simulations, results, and data presented in this text. Additionally, I am thankful to William Schwartz for his technical expertise in conceptualizing and executing several projects yet to be disclosed.

I owe many thanks to my longtime friend Felix Sanchez for always standing by me in difficult times and constantly challenging me to improve, personally and professionally.

I would also like to thank Emma Pruitt and Katerina Angjeli for their unwavering support, both academic and personal, over the course of the many hours we have collaborated and the projects we have undertaken. I wish them both the utmost success in their future studies and careers.

I am grateful to Dr. Janakinadh Yanamadala, Dr. Vishwanath Iyer, and Dr. Mark Reichelt for repeatedly welcoming me as part of the RF Toolbox team at MathWorks, and I thank The MathWorks as a whole for financially supporting my pursuit of this degree.

Finally, I thank my parents, Albert and Melissa Wartman, as well as my siblings, Christina, Jessica, and Wyatt. Your constant enthusiasm for my work and support of my endeavors have carried me further than you can possibly know.

Table of Contents

I. Introduction and Background

This introduction has been largely excerpted from [1] (published in April 2019) and [2] (under review at the time of writing). It has been generalized to better contextualize the broader scope of this thesis.

I.A. Motivation

Neuropsychiatric disorders are a leading source of disability. Along with critical ailments related to the senior population, depression has been the leading cause of disability in the US among young people ages 15 to 44 [3],[4]. Since such disorders are thought to result from aberrant neuronal circuit activity, neuromodulation approaches are of increasing interest given their potential for manipulating circuits directly [5].

Brain stimulation therapies are important and effective treatments for people with depression (a fourfold US surge in 2020 [6]) and other mental disorders. 50-60% of people with depression who have failed to receive benefit from medications experience a clinically meaningful response with brain stimulation. About one-third of them experience a full remission [4]. Over the past fifteen years, the number of brain stimulation devices to undergo the US Food and Drug Administration (FDA) market authorization has grown exponentially in number and has shown significant sustained interest [7]. This behavior has been true for the most challenging implanted invasive devices: those targeting Parkinsonian symptoms and tremors [7]. Other demanding clinical applications include presurgical mapping in epileptic patients and accurate motor mapping prior to brain tumor surgery (cf. [8]-[10]), as well as brain-computer interfaces (cf. [11]-[16]). Major electric brain stimulation modalities to date include:

- Transcranial magnetic stimulation (TMS) – a noninvasive, noncontact method which uses magnetic induction to generate current internal to the brain remotely via a coil placed next to the subject's head [17], [18],[19]) . More recently, the same physical principles have been also employed in embedded microcoils targeting selected populations of neurons while avoiding problems associated with the tissue-electrode interface ([20], [21],[22]). Due to the non-invasive nature of TMS, computational modeling of the electric fields within a patient-specific head model is the major and often only way to foster spatial targeting and/or obtain a quantitative measure of the required stimulation intensity.
- Transcranial electrical stimulation (TES) – including transcranial direct current stimulation (tDCS) and transcranial alternating current stimulation (tACS) – a low-cost portable application technique with applied currents usually less than 1-2 mA. Its uses include therapy for chronic conditions, mostly depressive disorders, electroanesthesia, or better sleep and memory consolidations [5], [23]-[26].
- Cortical stimulation (CS) and intracortical microstimulation (ICMS) – invasive yet precise versions of TES with smaller injected currents. Small implanted electrodes may target/activate selected populations or nuclei of neurons and have applications in brain and motor mapping pertinent to epilepsy [27],[28], neuro-oncology [29],[30], vision [31],[32] as

well as in brain/machine and brain/computer interfaces [33]-[35]. Cortical and introacortical stimulation is often combined with direct recordings using subdural or intraparenchymal depth electrodes. A modern high-resolution intracranial recording technique – intracranial electroencephalography or iEEG – is blossoming in various fields of human neuroscience [36],[37].

- Deep brain stimulation (DBS) – an invasive technique with a permanently implanted neurostimulator targeting deep parts of the brain, such as the subthalamic nucleus and forebrain bundle, to reduce symptoms of treatment-resistant depression and Parkinson's disease [38]-[40]. The success of subthalamic deep brain stimulation for Parkinson's disease is highly dependent on knowledge of the anatomical extent of the electric field surrounding the active electrode contact [41],[42].

While several alternatives exist (Sim4Life, ANSYS Maxwell), the predominant FEM-based electromagnetic brain stimulation modeling software is currently SimNIBS v. 1-3 [43]-[48]. This software uses robust formulations of the finite element method (FEM). In switching from the open-source 1st order FEM solver getDP to a more robust 1st order FEM formulation enabled by SimNIBS 3.0, the software achieves a remarkable performance improvement: an iterative FEM solution computed in less than 30 sec using a head model with a nodal density of 0.5 nodes/mm^2, processed on an Intel i7-7500U laptop processor (2 cores) with a clock speed of 2.7-3.5 GHz [47],[48].

In this thesis, an alternative modeling approach for fast, high-resolution electromagnetic stimulation modeling is described. The mathematical algorithm is based on the direct formulation of the boundary element method (BEM) in terms of induced charge density at the interfaces naturally coupled with the fast multipole method (FMM), or BEM-FMM, originally described in [49],[50]. Some distinct features of the BEM-FMM based modeling approach developed herein include:

i. High numerical accuracy, which was recently shown to exceed that of the comparable FEM of the first order ([51]).

ii. Unconstrained numerical field resolution close to and across cortical surfaces, including both the outer cortical surface (the interface between gray matter (GM) and cerebrospinal fluid (CSF) following terminology of [52]) and the inner cortical surface (the interface between white matter (WM) and GM). Since the solution is fully determined by the conductivity boundaries, the BEM-FMM numerical field resolution within the cortex is not limited by the FEM volumetric mesh size and may reach a micron scale if desired.

iii. Zero post-processing time for the normal components of the electric field close to and across cortical interfaces, once the solution for the induced surface charge density is known.

iv. Comparable speed. For a head segmentation with approximately 1 M facets (default example *Ernie* of SimNIBS 3.x), the present implementation of the BEM-FMM algorithm computes the complete numerical solution in approximately 38 seconds (excluding preprocessing time, which occurs once per head model), while SimNIBS takes 32 seconds for the matrix solution step alone on the same 2.1 GHz multicore server. Future versions of the BEM-FMM algorithm will be even faster.

v. Scalability to large-scale / high-resolution models. A surface model with 70 M facets has been considered and computed within two hours, demonstrating the vast potential that the method has to solve large-scale and/or high-resolution problems.

vi. Precise coil modeling and optimization. By employing the FMM, it is possible to model and optimize off-the-shelf and/or custom-designed coil CAD models composed of hundreds of thousands of elementary current elements [53].

This thesis describes several key improvements and extensions of the BEM-FMM numerical algorithm. These range from direct speed improvements to extensions to different model types to extensions to other stimulation methods – notably transcranial electrical stimulation (TES), in which brain stimulation is achieved by placing electrodes directly on the scalp. Application examples are provided for several computationally-intensive TMS and TES stimulation modeling problems, especially regarding field strength and focality at given target regions.

Appendices B and C describe the TMS and TES BEM-FMM software packages and walk users through specific computation steps of pertinent examples. The complete computational toolkits, along with supporting documentation, are available for academic purposes via GitHub repositories [54]-[56].

I.B. BEM-FMM Formulation

In the past, the BEM-FMM algorithm was successfully applied to the modeling of high-frequency electromagnetic [57][58] and acoustic [59]-[63] scattering and radiation problems in non-medical fields with a focus on defense applications. Successful implementation of the method for quasistatic bioelectromagnetic problems, however, was lacking. One such implementation was suggested in [49], [50], based on accurate coupling of the canonic general-purpose FMM [64]-[66] and the direct (without using reciprocity) quasistatic BEM formulated in terms of induced surface charge density, also known as the adjoint double layer formulation [67],[68]. Below, we describe the complete BEM-FMM algorithm along with its most recent improvements and establish the method's convergence.

I.B.1. Direct charge-based BEM in a conducting medium

Induced charges with a surface charge density $\rho(\boldsymbol{r})$ in C/m^2 reside on macroscopic or microscopic tissue conductivity interface(s) S once an external electromagnetic stimulus (a primary electric field $\boldsymbol{E}^p(\boldsymbol{r})$, either conservative or solenoidal) is applied. The induced surface charges alter (typically block and/or redirect) the primary stimulus field. The total electric field anywhere in space except the charged interfaces themselves is governed by Coulomb's law

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^p(\boldsymbol{r}) + \boldsymbol{E}^s(\boldsymbol{r}) = \boldsymbol{E}^p(\boldsymbol{r}) + \int_S \frac{\rho(\boldsymbol{r}')}{4\pi\varepsilon_0} \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}', \quad \boldsymbol{r} \notin S \tag{1}$$

where ε_0 is dielectric permittivity of vacuum. The electric field is discontinuous at the interfaces. When approaching a charged interface S with a normal vector $\boldsymbol{n}$ from either direction (inside or outside with regard to the direction of the normal vector), the electric field is given by

$$\boldsymbol{E}_{in/out} = \boldsymbol{E}^p + \int_S \frac{1}{4\pi\varepsilon_0} \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3} \rho(\boldsymbol{r}') d\boldsymbol{r}' \mp \boldsymbol{n}(\boldsymbol{r}) \frac{\rho(\boldsymbol{r})}{2\varepsilon_0}, \qquad \boldsymbol{r} \in S \tag{2}$$

An integral equation for $\rho(\boldsymbol{r})$, is obtained after substitution of Eq. 2 into the quasistatic boundary condition, which enforces the continuity of the normal current component across the interface, that is

$$\sigma_{in} \boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{in} = \sigma_{out} \boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{out}, \qquad \boldsymbol{r} \in S \tag{3}$$

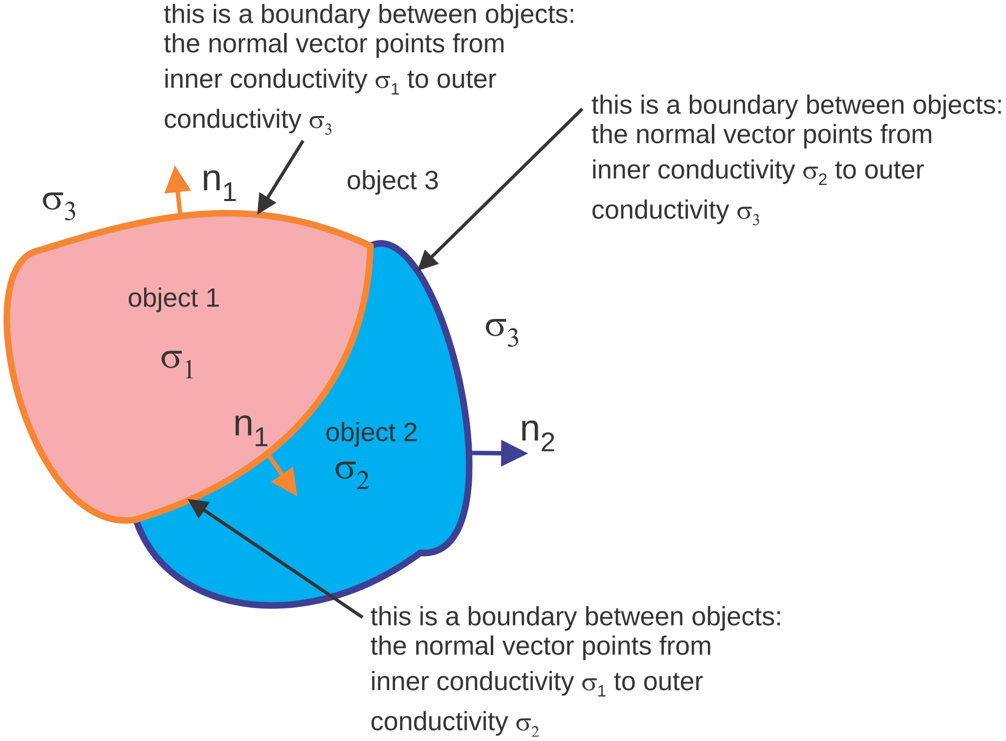
The result has the form (the adjoint double layer formulation, see [68])

$$\frac{\rho(\boldsymbol{r})}{2} - K\boldsymbol{n}(\boldsymbol{r}) \cdot \int_S \frac{1}{4\pi} \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3} \rho(\boldsymbol{r}') d\boldsymbol{r}' = K\boldsymbol{n}(\boldsymbol{r}) \cdot \varepsilon_0 \boldsymbol{E}^p(\boldsymbol{r}), \qquad \boldsymbol{r} \in S \tag{4}$$

where the electric conductivity contrast $K = \frac{\sigma_{in}-\sigma_{out}}{\sigma_{in}+\sigma_{out}}$ is defined at the interface(s). Here, $\sigma_{in}, \sigma_{out}$ are the conductivities inside and outside with regard to the direction of the normal vector, respectively. Note that if we solve Eq. 4 and then substitute the result for $\rho(\boldsymbol{r})$ in Eq. 1, the normalization constant ε_0 will cancel out. Therefore, its exact value does not matter for the subsequent analysis.

I.B.2. Treatment of interfaces

If the surface is a 2-manifold object with no contact to other surfaces (a "nested" topology where each of the surfaces is associated with a single unique exterior compartment), $\boldsymbol{n}$ is simply the outer normal vector to the surface; σ_{in} is the conductivity inside the object; and σ_{out} is the conductivity of the surrounding medium. If two objects (1 and 2) are in contact with each other as shown in Fig. 1, the joint interface between them should be counted *only once*. In Fig. 1, this interface is counted only for object 1 with σ_1 being the inner conductivity and σ_2 being the outer conductivity (in the direction of the normal vector $\boldsymbol{n}_1$). Facets of object 2 at the interface are now ignored to avoid double-counting. Alternatively, the interface may belong only to object 2, with the direction of the normal vector and the conductivity values switched. In that case, facets of object 1 at the interface would be ignored.



this is a boundary between objects: the normal vector points from inner conductivity σ_1 to outer conductivity σ_3

this is a boundary between objects: the normal vector points from inner conductivity σ_2 to outer conductivity σ_3

this is a boundary between objects: the normal vector points from inner conductivity σ_1 to outer conductivity σ_2

Fig. 1. For two objects (1 and 2) in contact with each other, the joint interface between them should be counted only once. In the figure, this interface is counted only for object 1 with σ_1 being the inner conductivity and σ_2 being the outer conductivity (in the direction of the normal vector n_1). Facets of object 2 at the interface are ignored to avoid double-counting.

From the formal point of view, one needs a composite mesh *without* double coincident facets. Then, for an arbitrary triangular facet t_m of the mesh with a given unit normal vector $\boldsymbol{n}_m$, one needs to know the conductivity $\sigma_{m,out}$ "outside" (i.e. in the direction of $\boldsymbol{n}_m$) and the conductivity $\sigma_{m,in}$ "inside" (i.e. in the opposite direction of $\boldsymbol{n}_m$). This information is sufficient to completely describe the model.

I.B.3. Normal electric fields at the interfaces

A significant and previously unnoticed advantage of the above approach is an ability to precisely obtain electric fields normal to the cortical surfaces (or any other interfaces) without additional computational cost or postprocessing. Only the solution for the surface charge density is necessary. After taking the scalar product of Eq. 2 with the surface normal vector $\boldsymbol{n}$, Eq. 4 may then be substituted to explicitly find the normal electric field just inside the surface, $\boldsymbol{n}\cdot\boldsymbol{E}_{in}(\boldsymbol{r})$; the normal electric field just outside the surface, $\boldsymbol{n}\cdot\boldsymbol{E}_{out}(\boldsymbol{r})$; and the normal field discontinuity for any conducting interface, $d\boldsymbol{n}\cdot\boldsymbol{E}$. All three quantities are directly proportional to each other. One has, for any conducting interface S and any observation point $\boldsymbol{r}\in S$,

$$\boldsymbol{n}\cdot\boldsymbol{E}_{in}(\boldsymbol{r})=\frac{\sigma_{out}}{\sigma_{in}-\sigma_{out}}\frac{\rho(\boldsymbol{r})}{\varepsilon_0},\quad \boldsymbol{n}\cdot\boldsymbol{E}_{out}(\boldsymbol{r})=\frac{\sigma_{in}}{\sigma_{in}-\sigma_{out}}\frac{\rho(\boldsymbol{r})}{\varepsilon_0},\quad d\boldsymbol{n}\cdot\boldsymbol{E}=\frac{\rho(\boldsymbol{r})}{\varepsilon_0} \tag{5}$$

We should note that the normal component of the total E-field just inside/outside conductivity boundaries depends explicitly only on the induced surface charge density. For example, Eq. 5 immediately predicts that the normal component of the field just inside the outermost skin surface is always zero (there is indeed no current into air) since σ_{out} is equal to zero in the first expression of Eq. 5. However, both the surface charge density and the normal component of the field just outside the outermost surface are certainly different from zero.

I.B.5. Model discretization

The surface charge density is expanded into pulse bases (zeroth-order basis functions) on triangular facets t_m with area A_m. The charge density is thus constant for every facet. Defining the conductivity contrast

$$K = \frac{\sigma_{in} - \sigma_{out}}{\sigma_{in} + \sigma_{out}}, \tag{6}$$

the Galerkin method is then applied to Eq. 4 to obtain a system of *M* linear equations for unknown expansion coefficients c_m in the form

$$\frac{c_m}{2} - \frac{K}{A_m}\sum_{n=1}^{M}\left(\boldsymbol{n}_m \cdot \iint_{A_m A_n} \frac{1}{4\pi}\frac{(\boldsymbol{r}-\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|^3} d\boldsymbol{r'} d\boldsymbol{r}\right) c_n = \frac{K}{A_m}\varepsilon_0 \int_{A_m} \boldsymbol{n}_m \cdot \boldsymbol{E}^p(\boldsymbol{r}) d\boldsymbol{r}\,, m = 1{:}\,M \tag{7}$$

The double potential integrals present in Eq. 7 require care in their numerical evaluation. Facets which are spatially close to one another (i.e., not considered well separated on the lowest level of the FMM octree) cannot be treated with the FMM. The close facets are those whose center-to-center distances are on the order (i.e. 1-10 times) of a linear face size computed, for high-quality surface meshes, as the square root of the face area.

The corresponding nearfield potential integrals are instead directly calculated and stored in the sparse nearfield BEM matrix using analytical integration for the inner integral and a Gaussian quadrature of 10th degree of accuracy for the outer integrals [50]. The number of geometrical (based on Euclidian distance) neighbors in Eq. 7 may vary, but a relatively small number may be adequate. It must be noted that these geometrical neighbors may belong to *different* tissue compartments.

I.B.6. Fast multipole method

The general-purpose FMM and its most recent freeware distribution ([66], also updated in March 2020) is applied to compute the remainder of the integrals of type defined by Eq. 7 using the center-point approximation at face centers $\boldsymbol{r}_m$, yielding

$$\int_{t_m}\int_{t_n} \frac{\boldsymbol{r}-\boldsymbol{r'}}{|\boldsymbol{r}-\boldsymbol{r'}|^3} d\boldsymbol{r} d\boldsymbol{r'} \approx A_m A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3} \tag{8}$$

This problem is equivalent to finding the electric field at target points $\boldsymbol{r}_m$ generated by the point charges located at source points $\boldsymbol{r}_n$. The accuracy of the FMM (the number of levels) is conventionally estimated for arbitrary volumetric charge distributions. However, for surface-based charge distributions, a much better relative accuracy is observed. For example, with the intrinsic method accuracy set as $\varepsilon = 0.1$, the mean error for the pial cortical surface (GM shell) may be as low as 0.1% with respect to the electric field amplitude and 0.08 deg with respect to the field angle deviation as compared to the most accurate solution (i.e., the solution where FMM precision is set to maximum).

II. Numerical Development of BEM-FMM

This chapter describes several additions and improvements that have been made to the BEM-FMM engine as described in [1]. Most notably, the method has been made compatible with a new format of model, its preprocessing phase has been substantially accelerated, and support for additional postprocessing data has been added.

II.A. Solid-Angle Approach for Nearfield Neighbor Integrals

The solid-angle approach is a computationally-efficient approach for computing surface integrals of the form $\boldsymbol{n}_m \cdot \int_{A_n} \frac{(r_p - r')}{|r_p - r'|^3} d\boldsymbol{r'}$ (see Eq. 9 below), which naturally arise when solving the core BEM-FMM integral equation over discrete triangular surface elements (Eq. 7). Rewriting the method's neighbor integrals to take advantage of the solid-angle approach results in a substantial speed increase during the model preprocessing phase and increases the numerical stability of the integrals themselves.

II.A.1. Derivation of Discretized Neighbor Integrals

The Boundary Element Fast Multipole Method (BEM-FMM) is based on behavior and interactions of charges that accumulate at interfaces between media. A suitable model for BEM-FMM simulation consists of a collection of surfaces S, where each surface represents the outer boundary of a conductive material with given conductivity σ. For conductive materials, it is known that any free charges must accumulate at interfaces between materials of different conductivity. For any collection of surfaces S discretized into infinitesimal surface elements located at $\boldsymbol{r'}$ and carrying charge density $\rho(\boldsymbol{r'})$, the total electric field $\boldsymbol{E}$ at observation point $\boldsymbol{r}$ (equal to the sum of the primary field $\boldsymbol{E}^P$and the secondary field $\boldsymbol{E}^S$) is given by Eq. 1, reproduced below.

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^p(\boldsymbol{r}) + \boldsymbol{E}^s(\boldsymbol{r}) = \boldsymbol{E}^p(\boldsymbol{r}) + \int_S \frac{\rho(\boldsymbol{r'})}{4\pi\varepsilon_0} \frac{\boldsymbol{r} - \boldsymbol{r'}}{|\boldsymbol{r} - \boldsymbol{r'}|^3} d\boldsymbol{r'}, \quad \boldsymbol{r} \notin S \qquad (1)$$

When the observation point $\boldsymbol{r}$ approaches the surface S (where S has normal vector $\boldsymbol{n}(\boldsymbol{r})$), the secondary electric field takes on an additional term, as given in Eq. 2). Here, $\boldsymbol{E}_{in}$ is defined as the electric field just inside (in the opposite direction of the normal vector) the surface, and $\boldsymbol{E}_{out}$ is defined as the electric field just outside the surface.

$$\boldsymbol{E}_{in/out} = \boldsymbol{E}^p + \int_S \frac{1}{4\pi\varepsilon_0} \frac{\boldsymbol{r} - \boldsymbol{r}'}{|\boldsymbol{r} - \boldsymbol{r}'|^3} \rho(\boldsymbol{r}') d\boldsymbol{r}' \mp \boldsymbol{n}(\boldsymbol{r}) \frac{\rho(\boldsymbol{r})}{2\varepsilon_0}, \qquad \boldsymbol{r} \in S \tag{2}$$

Eq. 3 imposes a current conservation boundary condition on Eq. 2: the current density $\boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{J}(\boldsymbol{r})$ flowing from an inner material with conductivity σ_{in} to an outer material with conductivity σ_{out} across location $\boldsymbol{r} \in S$ must be equal in both materials. Recalling that $\boldsymbol{J}(\boldsymbol{r}) = \sigma \boldsymbol{E}(\boldsymbol{r})$**,**

$$\sigma_{in} \boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{in} = \sigma_{out} \boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{out}, \qquad \boldsymbol{r} \in S \tag{3}$$

Substituting Eq. 2 into Eq. 3, an expression is obtained for charge density $\rho(\boldsymbol{r})$ in terms of the primary field $\boldsymbol{E}_P$. Here, $K = \frac{\sigma_{in} - \sigma_{out}}{\sigma_{in} + \sigma_{out}}$.

$$\frac{\rho(\boldsymbol{r})}{2} - K\boldsymbol{n}(\boldsymbol{r}) \cdot \int_S \frac{1}{4\pi} \frac{\boldsymbol{r} - \boldsymbol{r}'}{|\boldsymbol{r} - \boldsymbol{r}'|^3} \rho(\boldsymbol{r}') d\boldsymbol{r}' = K\boldsymbol{n}(\boldsymbol{r}) \cdot \varepsilon_0 \boldsymbol{E}^p(\boldsymbol{r}), \qquad \boldsymbol{r} \in S \tag{4}$$

When the interfaces S are discretized into M triangular surface elements, the mth surface element has area A_m, piecewise-constant surface charge density c_m, and normal vector n_m. Converting Eq. 4 to act on discrete surface elements following these conventions, Eq. 7 is obtained. Note that $\boldsymbol{r}$ lies on the surface of the observation triangle (m) and $\boldsymbol{r}'$ lies on the surface of the source triangle (n).

$$\frac{c_m}{2} - \frac{K}{A_m} \boldsymbol{n}_m \cdot \sum_{n=1}^{M} \left(c_n \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r} - \boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r} \right) = \frac{K}{A_m} \varepsilon_0 \int_{A_m} \boldsymbol{E}^p(\boldsymbol{r}) d\boldsymbol{r}, m = 1{:}\,M \tag{7}$$

If triangles m and n are sufficiently far from each other, the triangles referenced in the double-integral can be approximated as point charges, yielding Eq. 8.

$$\iint_{A_m A_n} \frac{\boldsymbol{r} - \boldsymbol{r}'}{|\boldsymbol{r} - \boldsymbol{r}'|^3} d\boldsymbol{r} d\boldsymbol{r}' \approx A_m A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3} \tag{8}$$

Assuming all triangles are sufficiently distant for the approximation of Eq. 8 to hold, the summation in Eq. 7 can be rewritten as

$$\boldsymbol{n}_m \cdot \sum_{n=1}^{M} \left(c_n \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r} \right) \approx \frac{1}{4\pi} A_m \boldsymbol{n}_m \cdot \sum_{n=1}^{M} c_n A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3} \tag{9}$$

The right-hand-side of Eq. 9 is naturally well-suited for treatment using FMM.

If C_m is defined as the set of triangles too close to triangle m to be treated using the approximation of Eq. 8, then a correction can be applied to Eq. 9 as follows:

$$\begin{aligned} &\boldsymbol{n}_m \cdot \sum_{n=1}^{M} \left(c_n \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r} \right) \\ &\approx \frac{1}{4\pi} A_m \boldsymbol{n}_m \cdot \sum_{n=1}^{M} c_n A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3} - \frac{1}{4\pi} A_m \boldsymbol{n}_m \cdot \sum_{n \in C_m} c_n A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3} \\ &+ \sum_{n \in C_m} \left(c_n \boldsymbol{n}_m \cdot \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r} \right) \end{aligned} \tag{10}$$

Here, the first term on the right-hand side is the initial FMM result for all triangles in the model, the second term on the right-hand side removes the FMM contribution of triangles too close to be accurately treated with FMM, and the third term adds back in an exact contribution from those triangles.

By introducing a set of P discrete integration points $\boldsymbol{r}_p$ with weights w_p at the surface of triangle m, the double-integral in the third term of Eq. 10 can be reduced to

$$\boldsymbol{n}_m \cdot \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r} \approx \sum_{p=1}^{P} \frac{w_p}{4\pi} \left(\boldsymbol{n}_m \cdot \int_{A_n} \frac{(\boldsymbol{r}_p - \boldsymbol{r}')}{\left|\boldsymbol{r}_p - \boldsymbol{r}'\right|^3} d\boldsymbol{r}' \right), \boldsymbol{r}_p \in A_m \ \forall \ p = 1{:}P \tag{11}$$

A naïve approach to solving the expression in parentheses in Eq. 10 is to compute the full integral over triangle n, and then take the dot product of the result with normal vector n_n, an algorithm for which is given in [69]. However, this computational task can be simplified by recognizing the expression in parentheses as the solid angle $\boldsymbol{\Omega}$ subtended by triangle A_n at field point r_p. An efficient method for computing such solid angles, which results in 32 multiplications, 20 additions, 3 square roots, and 1 ATAN2, is given in [70]:

$$\tan\left(\Omega/\boldsymbol{2}\right) = \frac{\boldsymbol{R}_1 \cdot (\boldsymbol{R}_2 \times \boldsymbol{R}_3)}{R_1 R_2 R_3 + (\boldsymbol{R}_1 \cdot \boldsymbol{R}_2) R_3 + (\boldsymbol{R}_1 \cdot \boldsymbol{R}_3) R_2 + (\boldsymbol{R}_2 \cdot \boldsymbol{R}_3) R_1} \tag{12}$$

where $\boldsymbol{R}_{1,2,3}$ are vectors drawn from integration point r_p to the vertices of triangle n, and $R_{1,2,3}$ denote the magnitudes of those respective vectors.

All MATLAB code developed for the algorithms described in the following sections can be found in Appendix A, section AA.1.

II.A.2. Accelerating triangle-to-point integrals via the solid angle approach

II.A.2.a. Functions under consideration

The three functions under test are potint2, potint4, and potint4b. The function potint2 computes the full integral over one triangle at multiple observation points using the method of [69]. The functions potint4 and potint4b both compute the normal components of this integral using the solid angle approach of [70]. The core difference between these methods is that potint4 is vectorized for N triangles and iterates over M observation points, where potint4b is vectorized for N triangles and M observation points simultaneously.

II.A.2.b. Estimated Memory Usage

The function potint4 must store three three-dimensional vertices per triangle, plus one three-dimensional position per observation point. With N triangles and M observation points, initial memory consumption (just for input argument storage) is 9N + 3M doubles (double-precision floating point numbers), and the storage for the output argument is MN. Within the loop over the observation points, the temporary variables require combined storage space of 15N doubles. The approximate number of doubles required by potint4 is 24N + 3M + MN, yielding (assuming 8 bytes in a double) 192N + 24M + 8MN bytes of storage. During the main loop, only 16N + 3 doubles (128N + 24 bytes) are accessed per iteration, excluding memory requirements of functions acting on those doubles.

The function potint4b, being fully vectorized, must store all intermediate calculations as 3-dimensional matrices. The input and output arguments once again require 9N + 3M doubles and MN doubles, respectively. The intermediate variables require 7(3MN) + 6(1MN) + 3M doubles. The approximate number of doubles required by potint4b is 9N + 6M + 28MN, corresponding to 72N + 48M + 224MN bytes, again excluding memory requirements of functions acting on this data.

Since these memory estimates do not attempt to account for intermediate memory requirements of mathematical operations or function calls (e.g. atan2), and the actual memory used is likely to be substantially larger.

II.A.2.c. Test 1: Accuracy for one triangle

In Test 1, potint4 and potint4b were tested against potint2 using a test setup comprising a single triangle of area 50 cm^2 and approximately 360 observation points placed in a circle around the triangle, as shown in Fig. 2. The radius of the circle was varied from 1 mm to 1 km, and relative error in $\Omega(\boldsymbol{r})$ was computed at each observation point as $e = \frac{\Omega_{test}-\Omega_{base}}{\Omega_{base}}$, where Ω_{test} is the solid angle computed by potint4 or potint4b and Ω_{base} is the solid angle computed by taking the dot product of the triangle's normal vector with the full field integral computed by potint2.

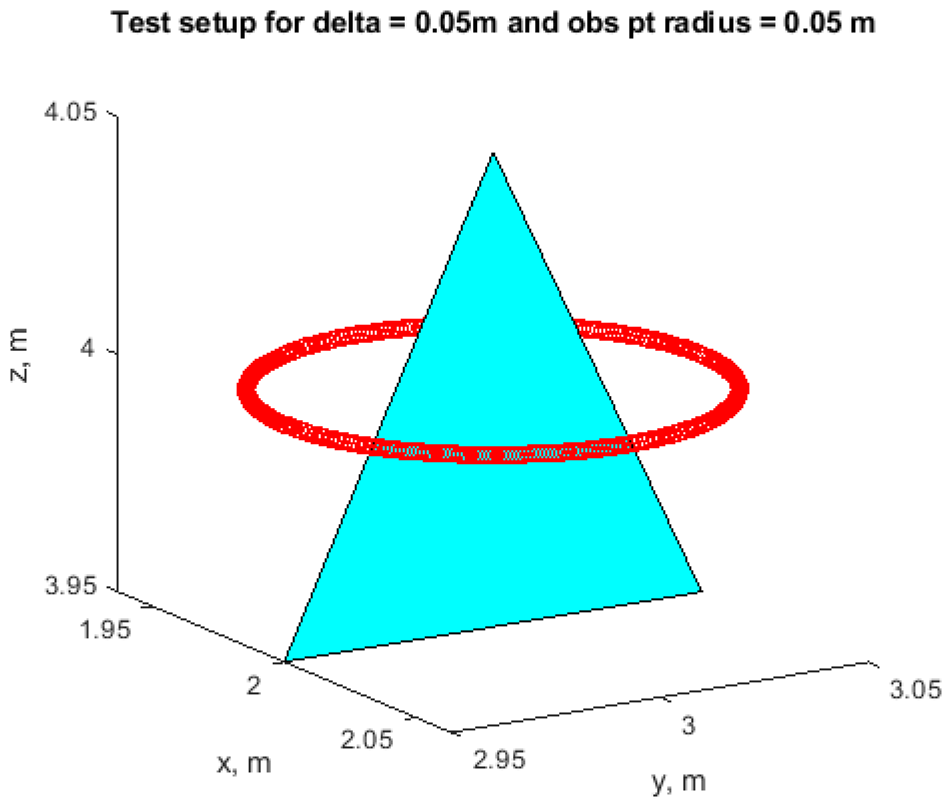


Fig. 2. Test setup showing observation points (red) in a 0.05-m-radius circle around the test triangle of area 0.005 m^2. 'delta' in the title refers to a parameter that determines the triangle size.

Figures 3 and 4 below show the maximum, mean, and minimum (absolute value of) error in $\Omega(\boldsymbol{r})$ with respect to distance from the triangle for potint4 and potint4b, respectively. The minimum relative error is zero until all observation points are more than 100 meters from the triangle. The maximum relative error is on the order of machine precision until the observation points are approximately 0.5 m from the triangle, and reaches 1e-8 when the observation points are approximately 1 km from the triangle. This result was additionally found to be almost completely independent of triangle size.

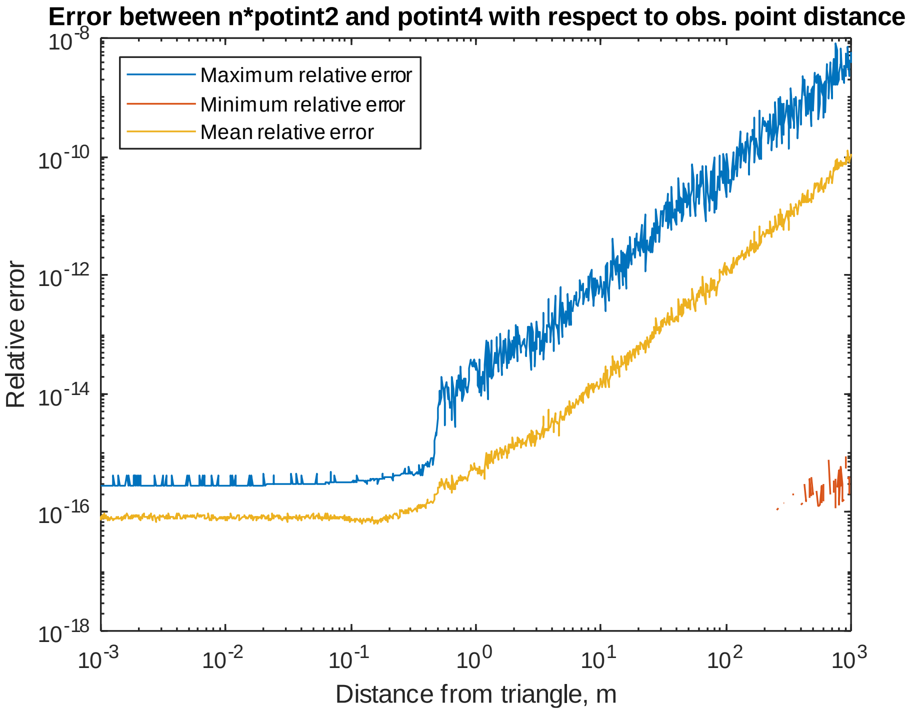


Fig. 3. Maximum, minimum, and mean relative error between the triangle-vectorized solid angle approach (potint4) and the normal-of-full-integral approach (potint2) as the radius of the circle of observation points increases.

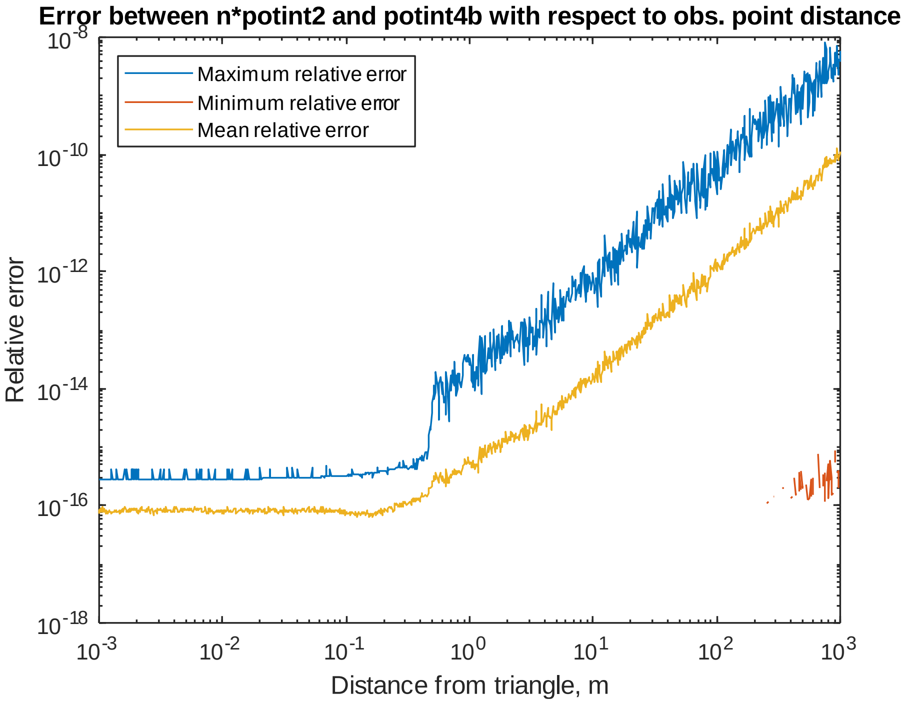


Fig. 4. Maximum, minimum, and mean relative error between the fully-vectorized solid angle approach (potint4b) and the normal-of-full-integral approach (potint2) as the radius of the circle of observation points increases.

Even as the radius of the circle of observation points approaches 1000 meters, the maximum error observed by any individual observation point is below 10^-8. The partially-vectorized and fully-vectorized solid-angle approaches produce virtually identical results to the normal-of-full-integral approach for this test configuration.

II.A.2.d. Test 2: Accuracy for multiple triangles

Fig. 5 below shows the test setup employed to verify that potint4 and potint4b produce results equivalent to potint2 for multiple triangles and multiple observation points. This test case uses four triangles and a circle of 360 observation points with radius 0.05 m. The triangles are positioned and oriented in a manner that prevents meaningful symmetry from occurring for most observation points. For this test case, the maximum error between either solid-angle approach and the normal-of-full-integral approach was 2.61e-11.

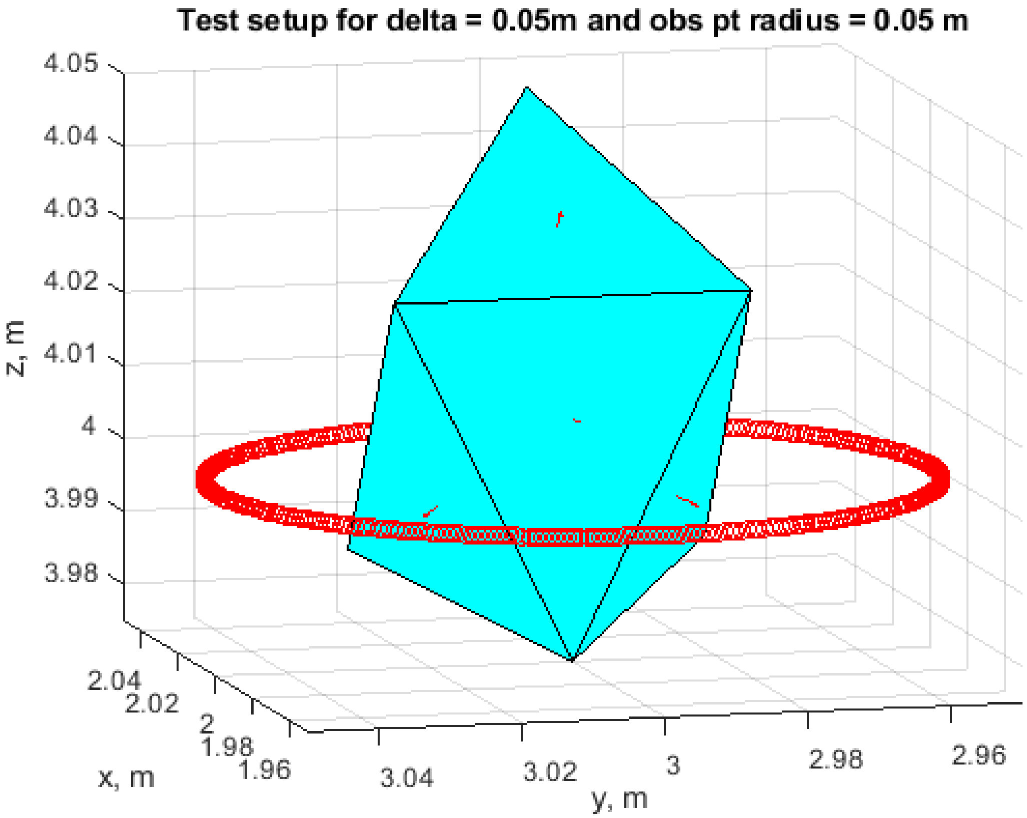


Fig. 5. Test setup showing observation points (red) in a 0.05-m-radius circle around four test triangles. The triangles' normal vectors are denoted by small red arrows.

II.A.2.e. Test 3: Performance comparison

The performance of potint2, potint4, and potint4b was tested by running each function on an increasing number of triangles and observation points. The number of observation points was varied from 1 to 1000 in steps of 10, and the number of triangles was varied from 4 to 1000 in steps of 4. The triangles used were duplicates of the test setup for Test 2 (see Fig. 5), occupying identical spaces. The observation points were evenly spaced on a circle of radius 0.05 meters in the XY plane centered on [x, y, z] = [2, 3, 4], as also shown in Fig. 5. For each combination of number of observation points and number of triangles, the functions potint2, potint4, and potint4b were run and timed 30 times each, outlier times were removed, and the remaining (non-outlier) times were averaged for each function.

Figs. 6 through 8 below show the dependence of execution time on number of observation points and triangles for potint2, potint4, and potint4b, respectively. Figs. 9 through 11 show the relative speedup obtained by using, respectively: potint4 in place of potint2, potint4b in place of potint2, and potint4b in place of potint4. In each of these figures, a red plane has been drawn at the level where relative speedup is equal to 1 (i.e., the break-even point between the compared methods). Finally, Figs. 12 through 17 show the same results, but focus on the region bounded by 100 observation points and 100 triangles to demonstrate performance in more typical cases. Note that the speedup plots have opposite x and y axis directions from the execution time plots to better capture the shape of the data.

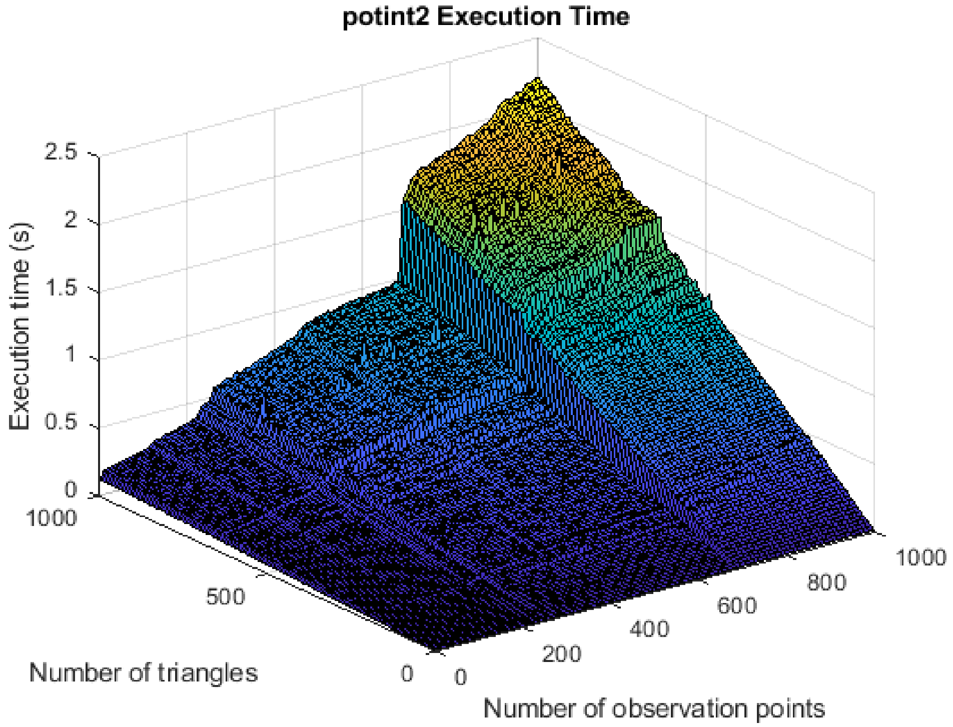


Fig. 6. Execution time for the normal-of-full-integral approach (potint2) as a function of number of observation points and number of triangles.

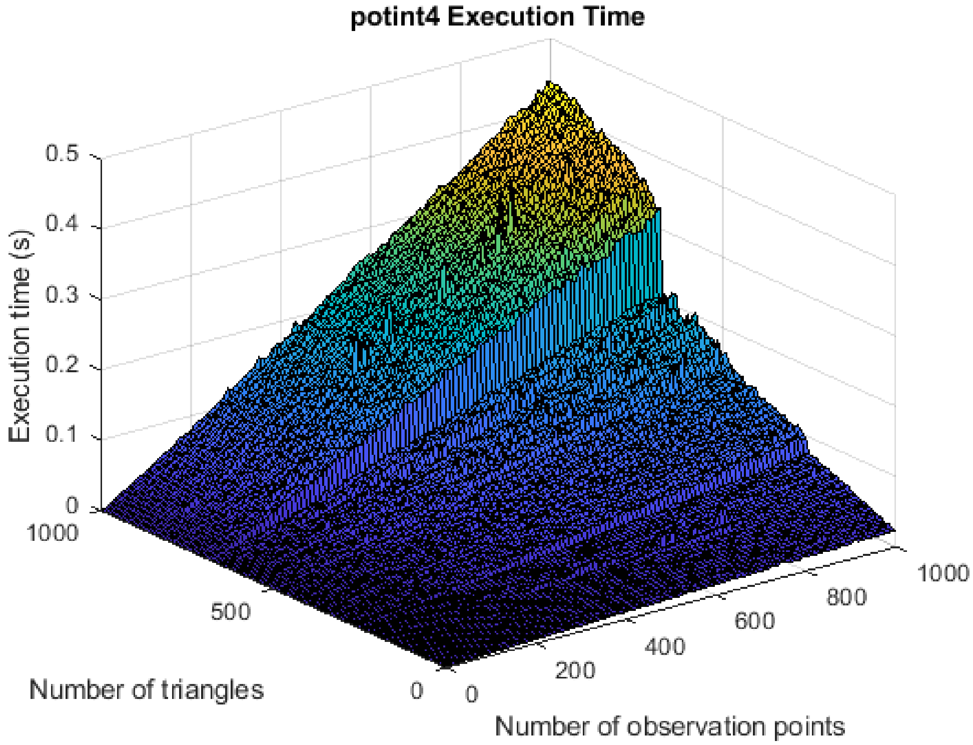


Fig. 7. Execution time for the triangle-vectorized solid-angle method (potint4) as a function of number of triangles and number of observation points. Note that the vertical scale is one-fifth that of Fig. 5.

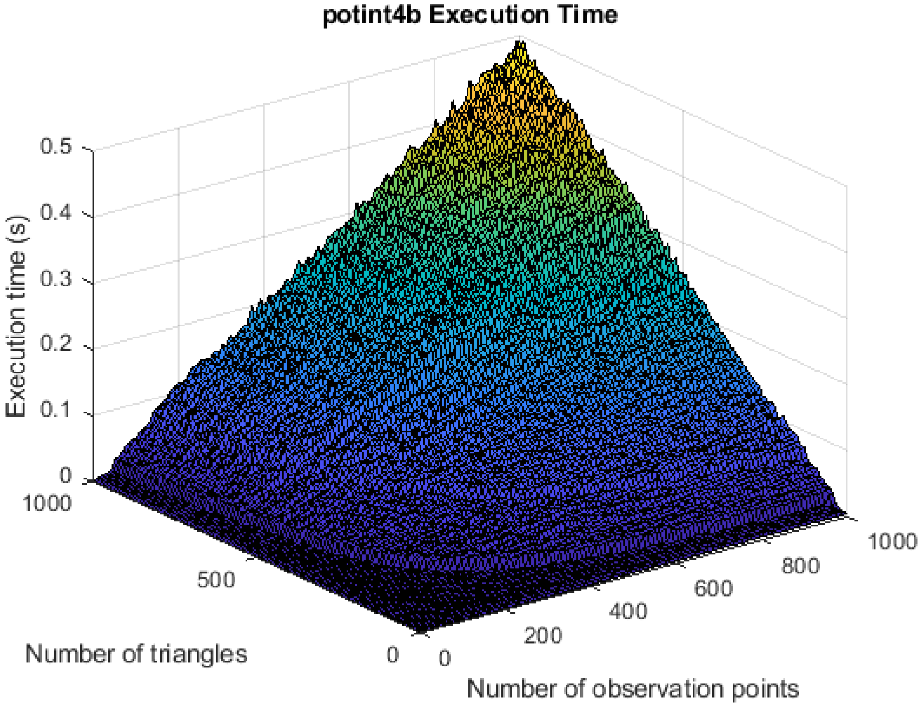
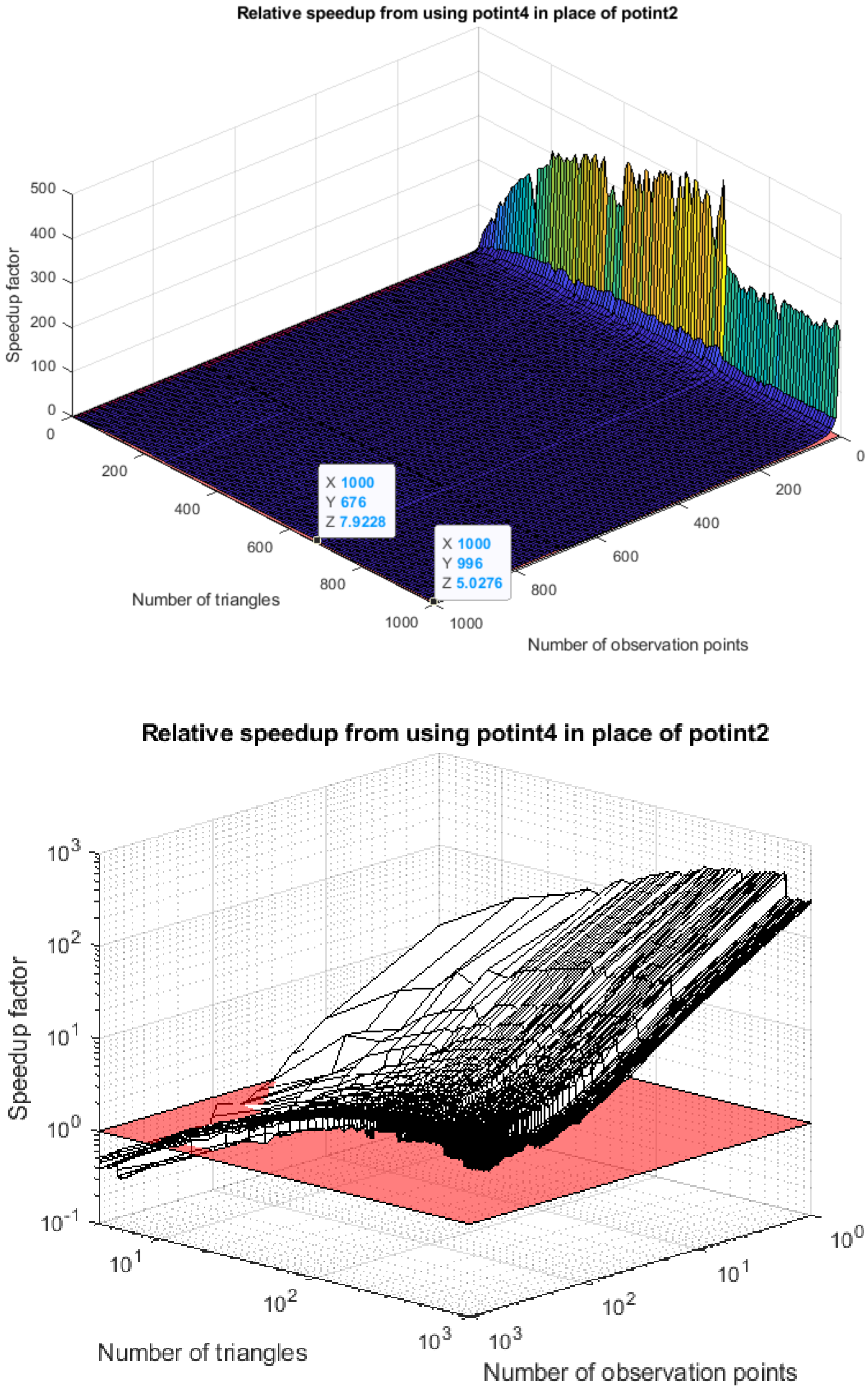


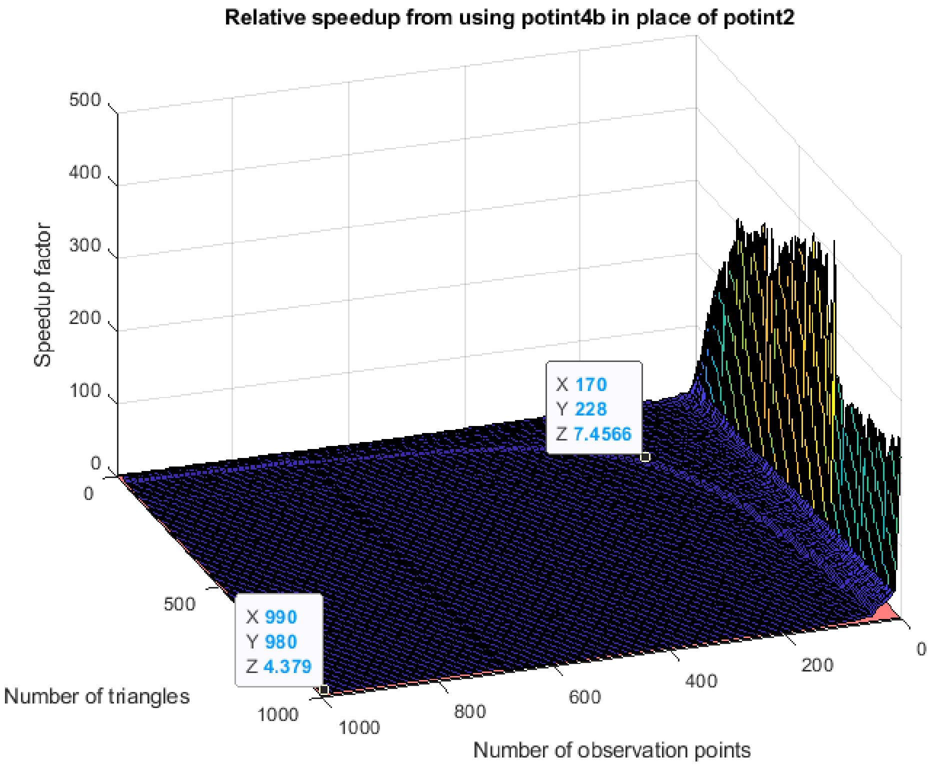
Fig. 8. Execution time for the fully-vectorized solid-angle method (potint4b) as a function of number of triangles and number of observation points. Note that the scale is again one-fifth that of Fig. 6.



Relative speedup from using potint4 in place of potint2

Relative speedup from using potint4 in place of potint2

Fig. 9. Relative speedup obtained by using the triangle-vectorized solid-angle method (potint4) in place of the normal-of-full-integral method (potint2). Function potint4 is many times faster for small numbers of observation points, and is still five times faster even in the worst case with 1000 triangles and 1000 observation points.



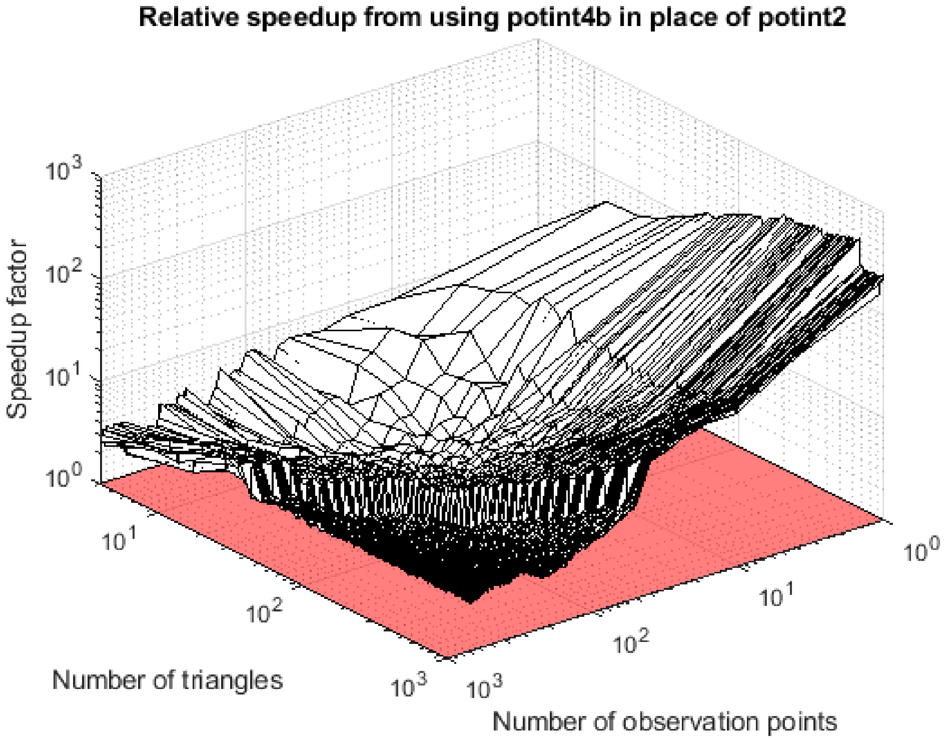
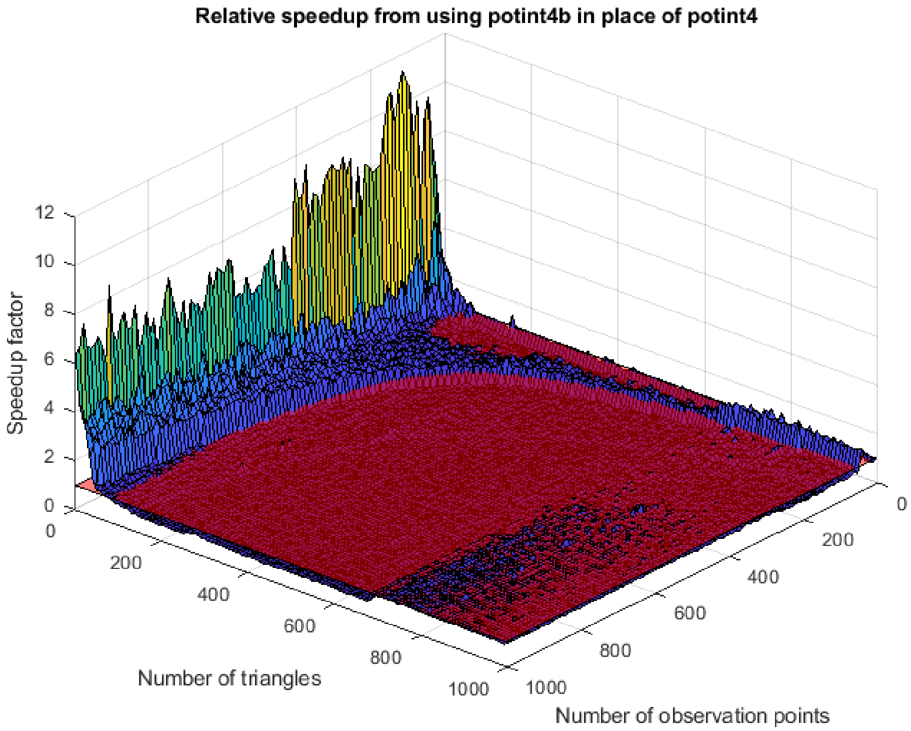


Fig. 10. Relative speedup obtained by using the fully-vectorized solid-angle method (potint4b) in place of the normal-of-full-integral method (potint2). Even in the worst case with 1000 triangles and 1000 observation points, function potint4b is 4.4 times faster than potint2.



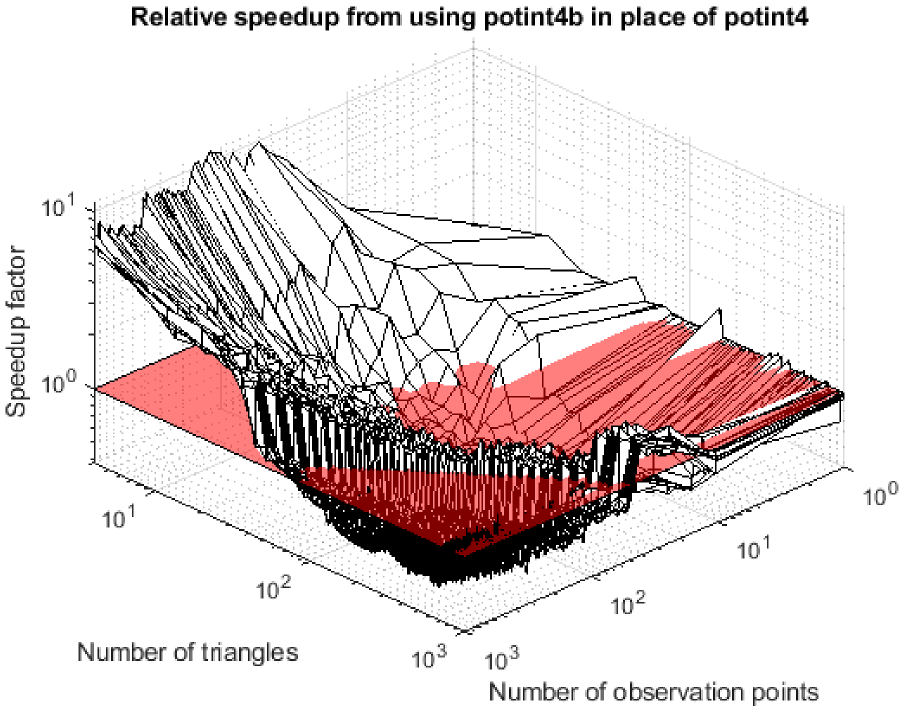


Fig. 11. Relative speedup obtained by using the fully-vectorized solid-angle method (potint4b) in place of the triangle-vectorized solid-angle method (potint4). A translucent red plane is drawn where the speedup factor is equal to 1, so in a region where this plane is visible, potint4 is faster than potint4b. For small numbers of triangles, function potint4b is several times faster than potint4, especially as the number of observation points increases.

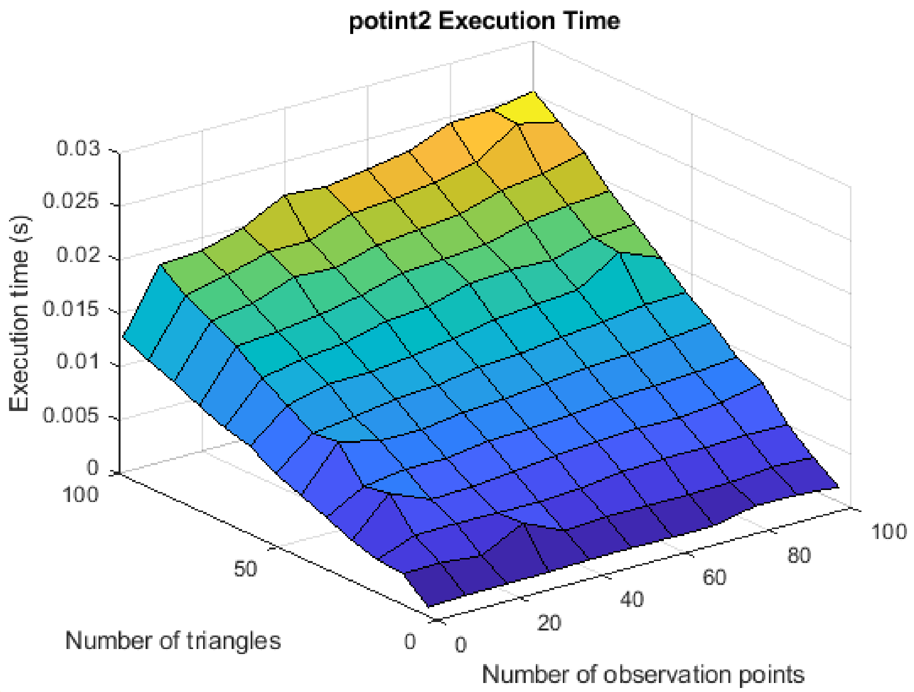


Fig. 12. Execution time for the normal-of-full-integral approach (potint2) as a function of number of observation points and number of triangles, bounded to 100 observation points and 100 triangles to present a more typical use case.

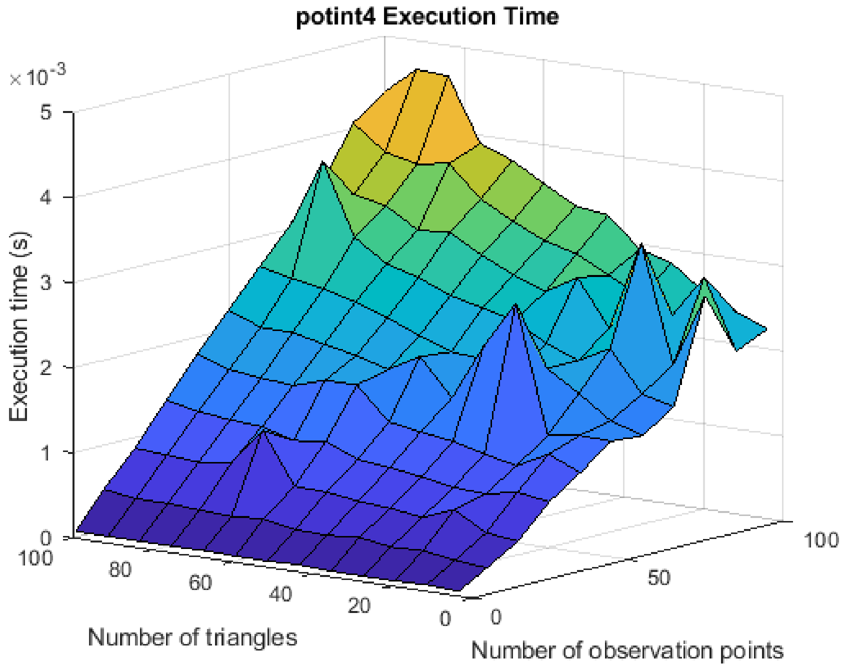


Fig. 13. Execution time for the triangle-vectorized solid angle approach (potint4) as a function of number of observation points and number of triangles, bounded to 100 observation points and 100 triangles to present a more typical use case. Note that the vertical scale is one-sixth that of Fig. 12.

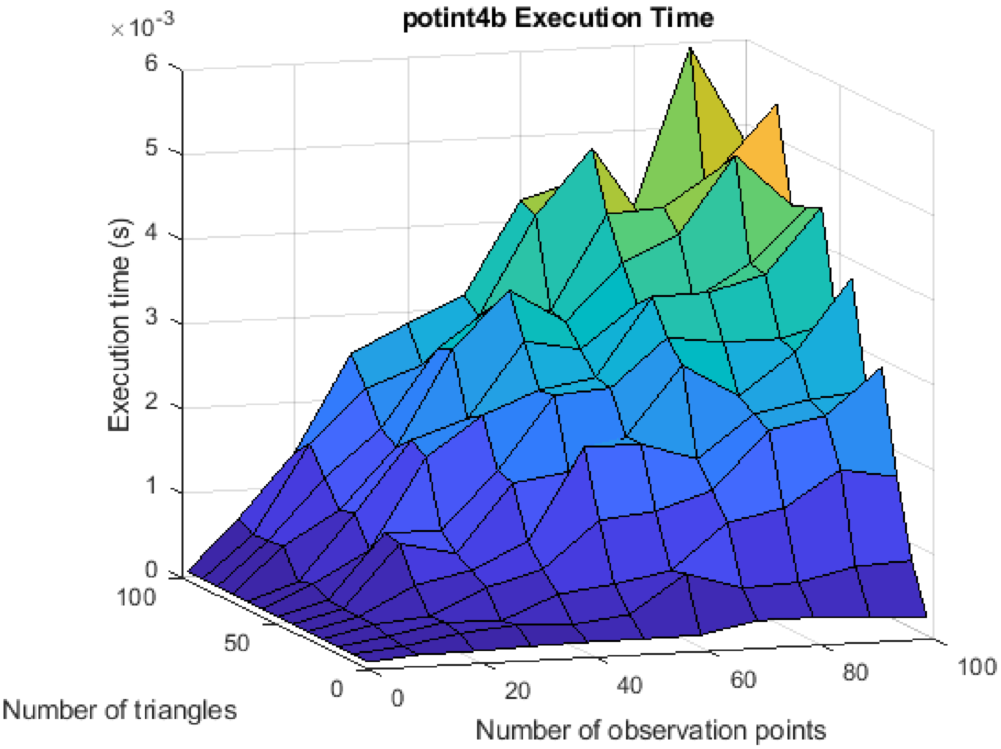


Fig. 14. Execution time for the fully-vectorized solid-angle approach (potint4b) as a function of number of observation points and number of triangles, bounded to 100 observation points and 100 triangles to present a more typical use case. Note that the vertical scale is one-fifth that of Fig.12.

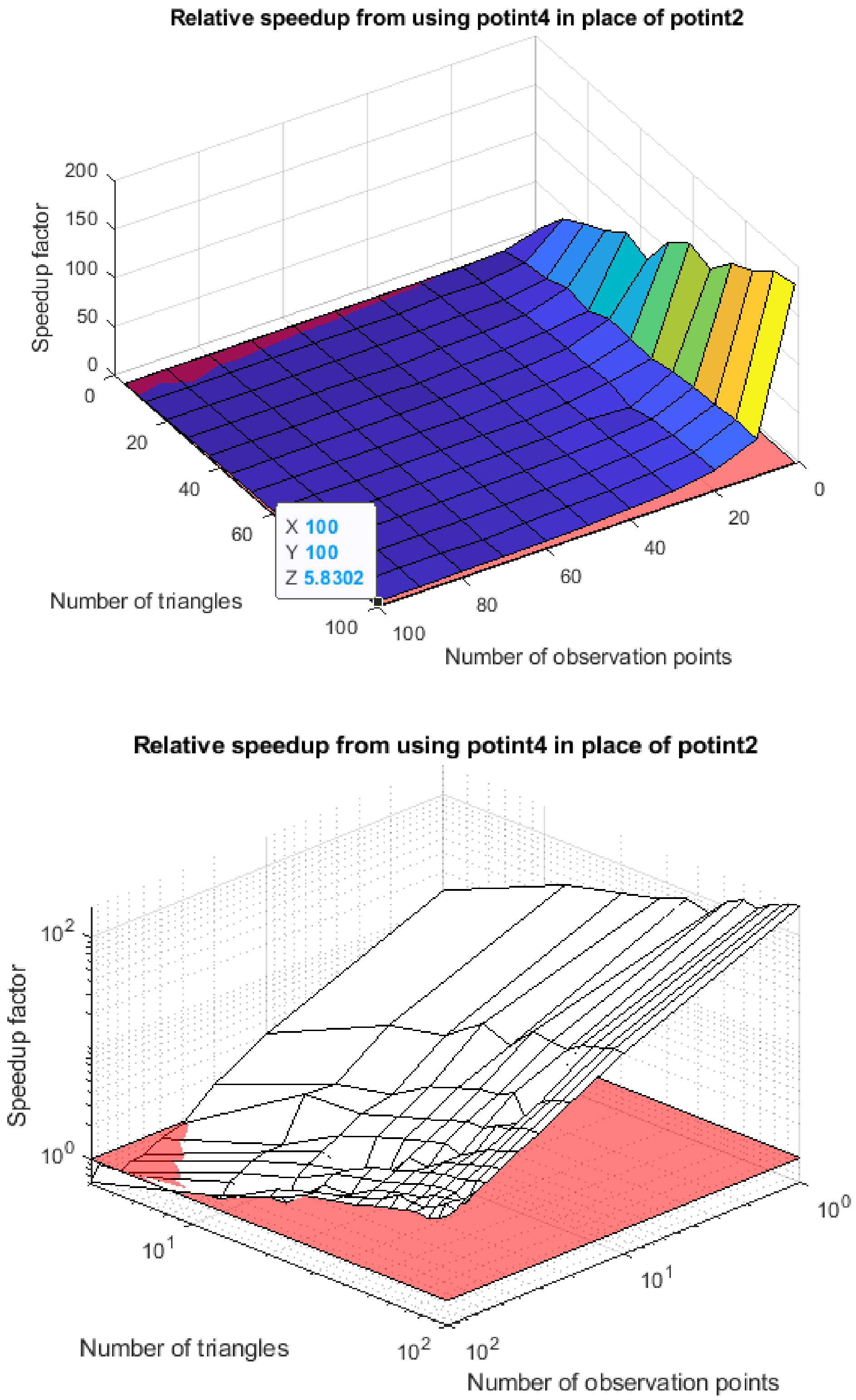


Fig. 15. Relative speedup obtained by using the triangle-vectorized solid-angle approach in place of the normal-of-full-integral approach, bounded to 100 triangles by 100 observation points to present a more typical use case.

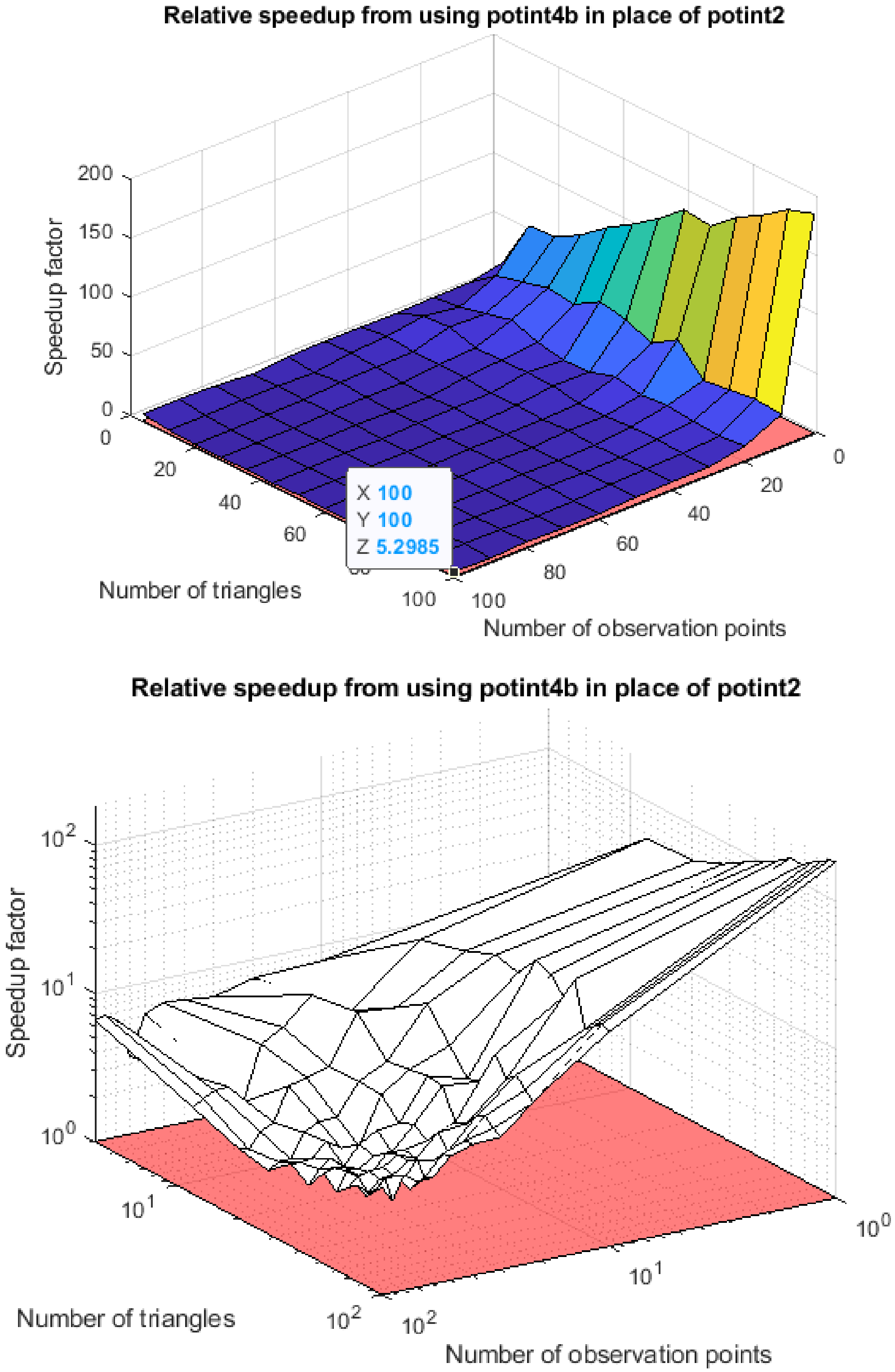
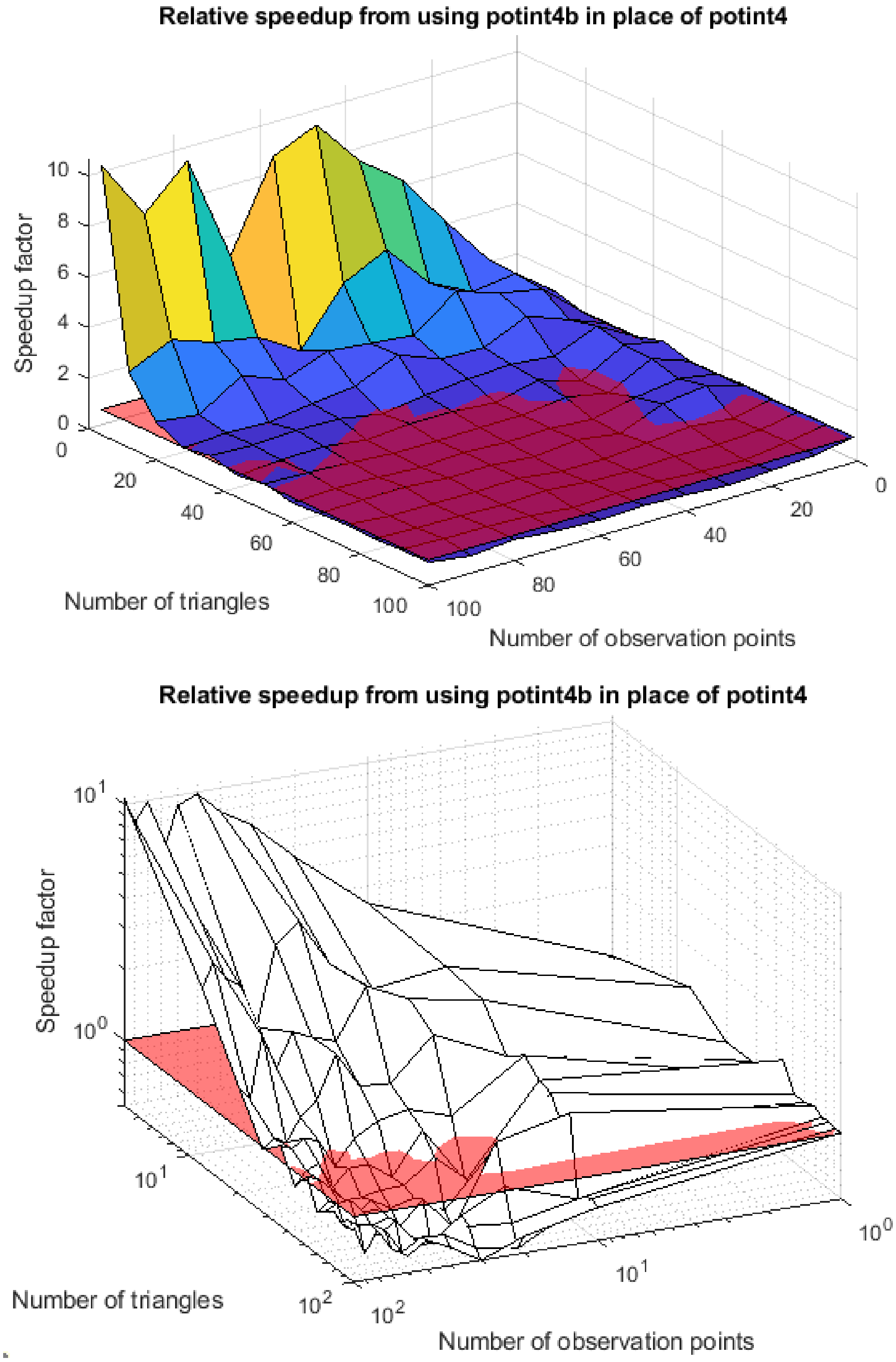


Fig. 16. Relative speedup obtained by using the fully-vectorized solid-angle approach in place of the normal-of-full-integral approach, bounded to 100 triangles by 100 observation points to present a more typical use case.



Relative speedup from using potint4b in place of potint4

Relative speedup from using potint4b in place of potint4

Fig. 17. Relative speedup obtained by using the fully-vectorized solid-angle approach in place of the triangle-vectorized solid-angle approach, bounded to 100 triangles by 100 observation points to present a more typical use case.

II.A.2.f. Discussion: triangle-to-point integral acceleration

Test 1 and Test 2 show that both solid-angle-based methods return equivalent results to the normal-of-full-integral method for both a single triangle and for multiple triangles for distances ranging from 1 mm to 1 km. Test 3 shows that both potint4 and potint4b are always faster than potint2, as expected. The execution time of potint2 varies strongly with the number of triangles, the execution time of potint4 varies strongly with number of observation points, and the execution time of potint4b varies strongly with both. For this machine, Figs. 11 and 17 show that potint4b is faster than potint4 when the number of triangles is smaller than 30.

Large discontinuities in execution time are clearly visible in Figs. 6, 7, and 8. These likely occur when the amount of working data exceeds that which can be stored solely in a given level of the processor cache, thus causing the processor to stall while it awaits data from the next lower (and slower) level. If this is the case, then the call to atan2 is likely the largest contributor to memory usage (when evaluating memory usage instruction by instruction).

II.A.3. Updating triangle-to-triangle integrals to use the solid-angle based triangle-to-point integrals

II.A.3.a. Triangle-to-triangle integrals

For charged triangles sufficiently far from each other, the integral of one triangle's electric field over the surface of the other triangle (and vice versa) can be approximated by Coulomb's Law, and the FMM can be applied to compute trillions of such interactions within seconds. For triangles not far enough from each other for this approximation to hold, a correction must be applied to the FMM-computed result. For the full model, this correction takes the form of a sparse matrix $EC.$ EC is the sum of two other sparse matrices: $IC,$ which subtracts out the Coulomb's Law contribution from neighboring triangles, and $IN,$ which adds back in the exact integral contribution from those triangles. Following from Eq. 10 and Eq. 11, these matrices are given by:

$$EC = IN - IC \tag{13}$$

$$in_{m,n} = \frac{1}{4\pi}\left(\boldsymbol{n}_m \cdot \iint_{A_m A_n} \frac{1}{4\pi} \frac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3} d\boldsymbol{r}' d\boldsymbol{r}\right), n \in\ C_m, m = 1\!:\!M \tag{14}$$

$$ic_{m,n} = \frac{1}{4\pi} A_m \boldsymbol{n}_m \cdot \sum_{n \in C} A_n \frac{\boldsymbol{r}_m - \boldsymbol{r}_n}{|\boldsymbol{r}_m - \boldsymbol{r}_n|^3}, n \in\ C_m, m = 1\!:\!M, \tag{15}$$

The functions that can be used to compute EC are meshneighborints, meshneighborints_1b, and meshneighborints_2. The function meshneighborints computes both potential integrals and electric-field integrals using the full integral method, the function meshneighborints_1b computes only electric-field integrals using the same method as meshneighborints, and meshneighborints_2 computes only electric-field integrals using the solid-angle approach. The function meshneighborints_1b exists mainly for timing comparisons between the solid-angle neighbor integral method and the normal-of-full-integral neighbor integral method.

II.A.3.b. Stability of the solid-angle approach for triangle-triangle integrals

To test the stability of the solid angle approach compared to the full integral approach, an adverse test setup involving two perpendicular triangles was employed as shown in Fig. 18. The matrix IN for this system was computed using both methods. Note that symmetry of the problem ensures that the contribution of the first triangle to the second is equal to the contribution of the second triangle to the first, so only one entry of the 2x2 matrix IN needs to be considered. Both methods employed barycentric triangle subdivision to find integration points on the triangle surfaces, and subdivision parameters increased from 10 to 1000 in steps of 10 (corresponding to 100 to 1M subdivided triangles). Figs. 19 and 20 below show the results of this test.

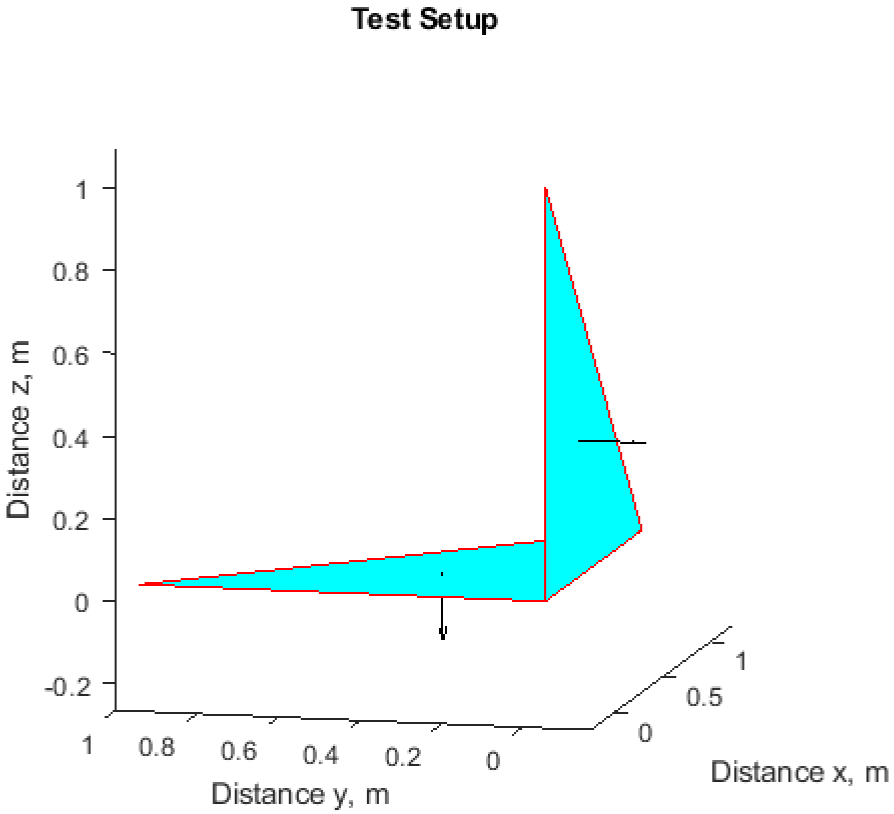


Fig. 18. Test setup for stability of the solid-angle method versus the normal-of-full-integral method

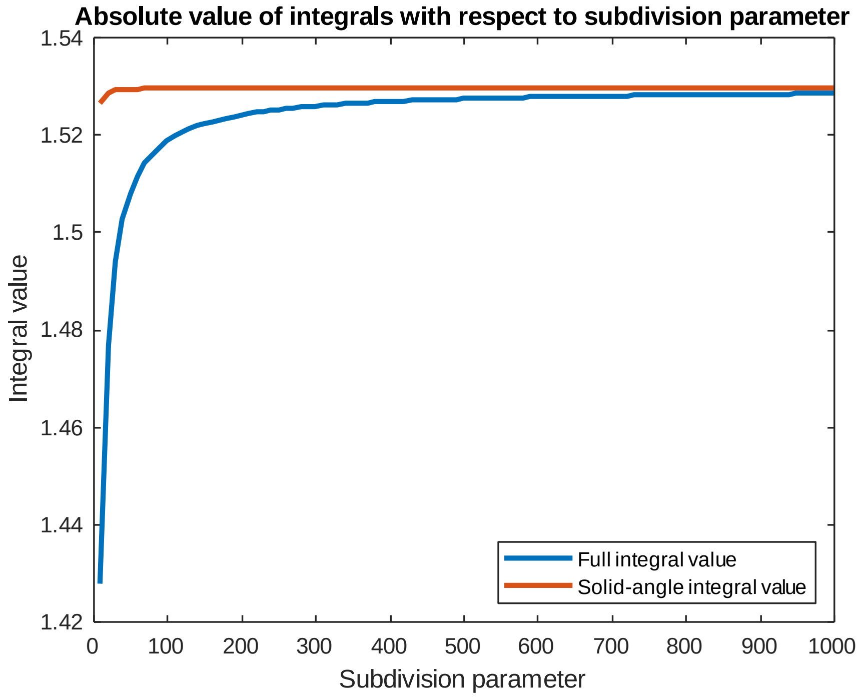


Fig. 19. Absolute convergence of the solid-angle approach and the normal-of-full-integral approach as the subdivision parameter used for integration is varied from 10 to 1000, corresponding to 100 to 1000000 subdivisions.

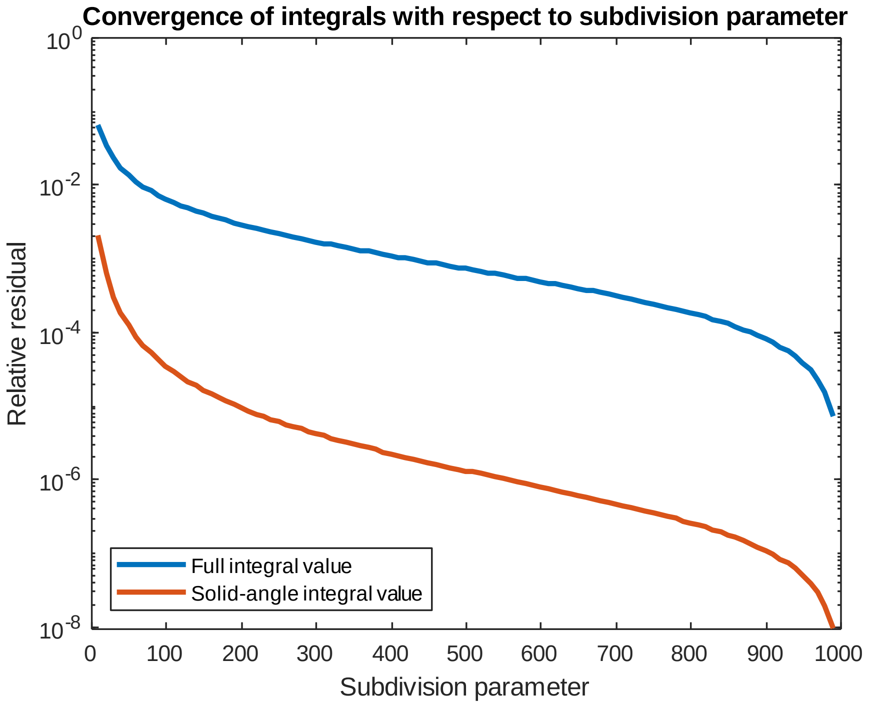


Fig. 20. Relative convergence of the solid-angle approach and the normal-of-full-integral approach as the subdivision parameter used for integration is varied from 10 to 1000.

As shown in Figs. 19 and 20, the integral computed by the solid-angle approach converges far more rapidly than the normal-of-full-integral approach, and in terms of the relative residual, the initial state of the solid-angle approach (with a subdivision parameter of 10, corresponding to 100 subdivided triangles) is approximately 30 times more accurate than the initial state of the full integral.

II.A.3.c. Validation using Connectome Subject 110411

The solid-angle based neighbor integral method was tested in a realistic application by computing the correction matrix EC and the neighbor integrals IN for Connectome Subject 110411, processed using the SimNIBS pipeline [43]-[48]. For every triangle, four neighbor triangles were selected based on proximity. The error between the solid-angle approach and the normal-of-full-integral approach was quantified as

$$\Delta IN = \frac{||IN_{test} - IN_{base}||}{||IN_{base}||} \tag{16}$$

$$\Delta EC = \frac{||EC_{test} - EC_{base}||}{||EC_{base}||} \tag{17}$$

Where IN_{test} is IN as computed by the solid-angle approach, IN_{base} is IN as computed by the normal-of-full-integral approach, and EC_{test} and EC_{base} follow the same convention.
For meshneighborints_2 versus meshneighborints, ΔIN is 0.0365 and ΔEC is 0.0091.

II.A.3.d. Timing for solid-angle method for computing neighbor integrals

To compare execution times between the solid-angle approach and normal-of-full-integral approach, timing commands were inserted directly into the functions under test. The function meshneighborints_1b is a version of meshneighborints from which the potential integrals have been removed for equivalent comparison with meshneighborints_2. The timing results for these three functions are summarized in Tables 1 through 3 below.

Table 1: Timing of the neighbor integral methods for 15 threads and 4 neighbor triangles

Operation	meshneighborints	meshneighborints_1b	meshneighborints_2
Parallel Pool Initialization	26.8779 s	21.5561 s	17.0392 s
Neighbor Integral Calculation	36.9135 s	19.6503 s	9.5485 s
Parallel Pool Shutdown	4.5772 s	4.5504 s	4.4606 s
EC Assembly	7.6812 s	9.6602 s	3.8905 s

Table 2: Timing of the neighbor integral methods for 15 threads and 25 neighbor triangles

Operation	meshneighborints	meshneighborints_1b	meshneighborints_2
Parallel Pool Initialization	21.3189 s	21.3625 s	17.1404 s
Neighbor Integral Calculation	102.7413 s	54.9546 s	17.5118 s
Parallel Pool Shutdown	4.2269 s	4.2748 s	4.2201 s
EC Assembly	13.5727 s	24.0857 s	7.4383 s

Table 3: Timing of the neighbor integral methods for 15 threads and 100 neighbor triangles

Operation	meshneighborints	meshneighborints_1b	meshneighborints_2
Parallel Pool Initialization	21.3765 s	21.3910 s	16.9151 s
Neighbor Integral Calculation	356.3959 s	182.3036 s	55.1158 s
Parallel Pool Shutdown	4.2482 s	4.4992 s	4.1493 s
EC Assembly	87.2093 s	72.8173 s	20.5396 s

II.A.4. Considerations for higher-level code

II.A.4.a. Removing dependence on full neighbor integral solutions

Initially, the full neighbor integrals calculated by the function meshneighborints had been saved along with the correction matrix EC and thus were available to high-level code. Their primary use was in calculating and displaying fields that occurred exactly at surfaces of the model. However, switching to the solid-angle approach eliminated the intermediate step of calculating the full neighbor integrals required for this operation. Instead, triangle subdivision (Gaussian or barycentric) coupled with FMM can be used to obtain the electric field at surfaces.

The subdivision methods tested were Gaussian (with 4, 7, 13, and 25 subdivisions) and barycentric (with parameters of 2, 3, 4, 5, and 6, corresponding to 4, 9, 16, 25, and 36 subdivisions respectively). After finding the charge distribution on Connectome Subject 110411 due to a coil placed above the motor hand area, the secondary electric field was calculated at each facet of the model. Solutions were first obtained using full neighbor integrals with 4, 64, and 128 neighbors for each triangle. Error was calculated as $\frac{\|E_{subdiv}-E_{analyt}\|}{\|E_{analyt}\|}$ in each case, where E_{analyt} is the electric field calculated using corrections from the exact neighbor integrals, E_{subdiv} is the electric field calculated through subdivision, and $\|*\|$ denotes the matrix 2-norm.

Table 4 below shows the computation time and error for several subdivision methods when applied to Connectome Subject 110411. Error is shown for the 4-neighbor case (requiring 7 seconds if exact neighbor integrals are known), the 64-neighbor case (requiring 10 seconds if exact neighbor integrals are known), and the 128-neighbor case (requiring 14 seconds if exact neighbor integrals are known).

Table 4: Subdivision method accuracy with respect to analytical neighbor integrals

Subdivision Method	Subdivision Parameter	Runtime	Error (4 nbrs.)	Error (64 nbrs.)	Error (128 nbrs.)
Gaussian	4	11.1866 s	0.0432	0.0415	0.0415
Gaussian	7	14.1866 s	0.0410	0.0385	0.0385
Gaussian	13	35.6007 s	0.0349	0.0322	0.0322
Gaussian	25	56.9180 s	0.0252	0.0219	0.0219
Barycentric	2	11.2613 s	0.0190	0.0187	0.0187
Barycentric	3	19.7735 s	0.0136	0.0123	0.0123
Barycentric	4	43.1691 s	0.0114	0.0091	0.0091
Barycentric	5	55.7387 s	0.0104	0.0072	0.0071
Barycentric	6	73.6604 s	0.0099	0.0059	0.0059

Based on the data in Table 4, barycentric subdivision with subdivision parameter 3 (corresponding to 9 subdivided triangles) appears to provide the best balance of accuracy and speed.

II.A.4.b. Validation using ANSYS Maxwell and a simple test setup

This test setup was a model of a homogeneous block of dimensions 0.5 m by 0.5 m by 0.5 m with two imprinted voltage electrodes held at +1 V and -1 V (see Fig. 21). This system was simulated using ANSYS® Maxwell Electronics Desktop 2019 R1 and the electric field was calculated at two sets of points: one 100-by-100 point uniform grid in the Z = 300 mm plane, and in a 50-by-50-by-50 point uniform grid throughout the cube's volume. The electric field at these points was also calculated using BEM-FMM, relying first on the solid-angle neighbor integrals method and second on the normal-of-full-integral method. Two error metrics were calculated. The first error metric, the 2-norm error, is calculated as

$$\Delta E_{2norm} = \frac{\left\| E_{bemfmm} - E_{ansys} \right\|}{\left\| E_{ansys} \right\|} \tag{18}$$

where $\|*\|$ indicates the matrix 2-norm.

The second error metric, the R2 error, is calculated as

$$\Delta E_{R2} = \frac{\left\| \left| E_{bemfmm} - E_{ansys} \right| \right\|}{\left\| \left| E_{ansys} \right| \right\|} \tag{19}$$

where $|*|$ indicates a row-wise vector 2-norm and $\|*\|$ indicates a column-wise vector 2-norm.

The results from this test are summarized in Table 5.

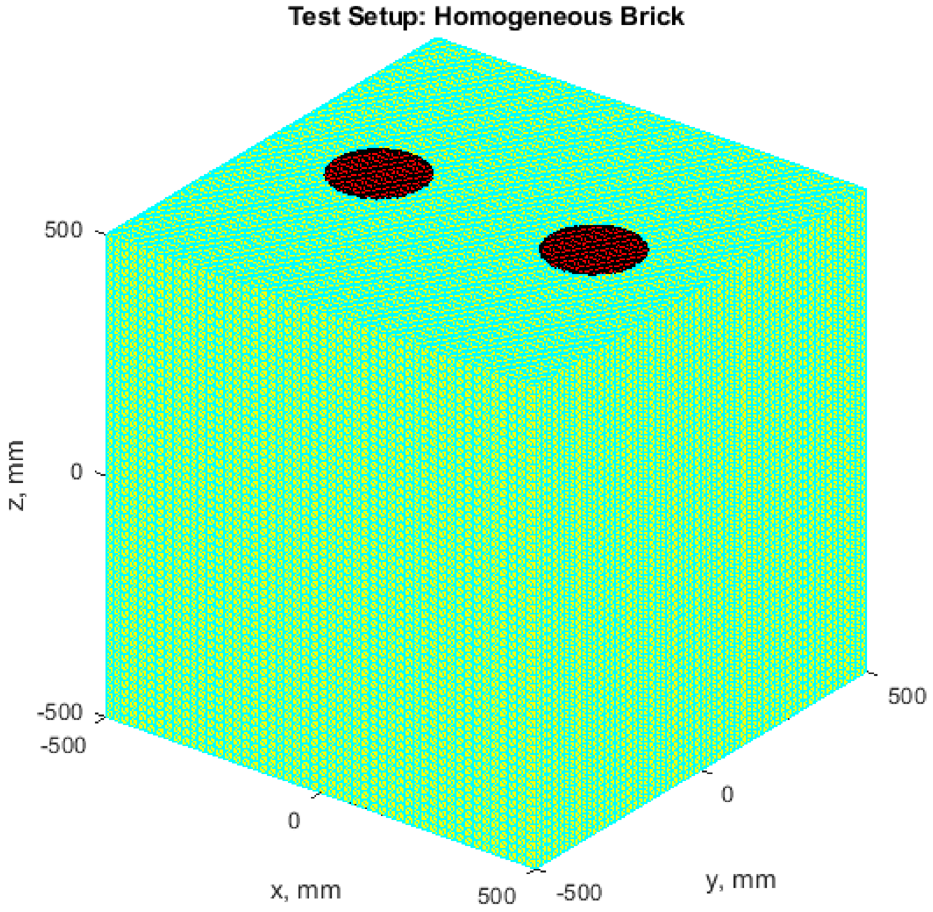


Figure 21: Homogeneous block with dimensions 0.5 m by 0.5 m by 0.5 m and two imprinted voltage electrodes

Table 5: Comparison with ANSYS Maxwell for a homogeneous brick

Integral Method	Observation Point Set	R2 Error	2-Norm Error
Solid-Angle	z = 300 mm plane	0.0053	0.0063
Normal of Full Integral	z = 300 mm plane	0.0054	0.0063
Solid-Angle	Full volume	0.0057	0.0057
Normal of Full Integral	Full volume	0.0058	0.0058

II.A.4.c. Validation between charge solutions for the solid-angle approach and normal-of-full-integral approach

For this test, an MRi-B91 MagVenture coil was positioned above the motor hand area of Connectome Subject 110411. The coil was driven with 5 kA at 3 kHz, resulting in a maximum current time-derivative of 94 A/µs. The charge solution (charge density at every triangle, normalized by ϵ_0) for this test setup was computed using both the solid-angle approach and the normal-of-full-integral approach, and error between the charge solutions was computed as follows (where c_{test} is the charge solution that arises from the solid-angle approach, c_{base} is the charge solution that arises from the normal-of-full-integral approach, and $\|*\|$ denotes the vector 2-norm).

$$\Delta c = \frac{\|c_{test} - c_{base}\|}{\|c_{base}\|} \tag{20}$$

The error in the charge solution was found to be 0.0039.

II.B. Coincident Facet Detection and Resolution

II.B.1. Motivation

When the BEM-FMM is employed to compute the surface charge density solution for a given model, the total electric field incident on a given facet is computed in two steps. First, an initial approximation is generated using the FMM, which treats all other facets as point charges and vastly accelerates the solution. Next, the point-charge-approximated contributions from several (typically between 4 and 32 depending on model complexity) nearby facets are subtracted out, and precise integral contributions from those facets are computed and added back in.

If the model has at least one facet that shares a coincident centroid with the current facet, then the distance between the facets' centroids will be zero, and the FMM-approximated contribution from the coincident facet to the current facet will be NaN. Since the result of any arithmetic operation performed on a NaN is also NaN, the NaN will propagate to all other facets on the next solver iteration and corrupt simulation results. If a model contains coincident facets, they must be detected and resolved prior to simulation.

One common reason for the occurrence of coincident facets is that a model's components are fully self-contained; in other words, each component of the model specifies all of its boundaries and does not depend on other components to supply missing information. When two such components share a boundary, each component contributes its own segmentation of that boundary. These two segmentations typically contain facets that are exact duplicates of each other. An example of a model in which each component is fully self-contained is the MIDA model [71]. Some models in which components are not self-contained are the Connectome subjects [72], [78]. In these models, only the outer boundaries of components are segmented, and information about a given component's inner boundary is contributed by the outer boundary of the next nested component.

II.B.2. Resolution

Coincident facets can be detected as follows. First, each facet's nearest neighboring facet is found (by a fast method such as, for example, the *knnsearch* function of MATLAB's Statistics and Machine Learning Toolbox) based on centroid-centroid distance. If the distance between the centroids of a given facet and its nearest neighbor is smaller than a certain threshold, the facets are flagged as having coincident centroids. Of the facets that are flagged to have coincident centroids, those whose vertices are also closer than a certain threshold are additionally flagged as fully coincident.

Facets that share coincident centroids but which are not fully coincident can be resolved by a small perturbation of their centroids along their respective normal vectors. These instances are usually rare, since the probability that two non-identical triangles share the same centroid is very small. The process for resolving fully coincident facets is more involved.

The BEM-FMM boundary conditions are based on the numerical contrast between material properties (e.g. electrical conductivity or dielectric constant) across model boundaries. Each object is initially assigned a known inside conductivity and an assumed outside conductivity. As previously mentioned, coincident facets typically occur when two components (Object 1 and Object 2) both segment their mutual boundary. This can be taken to mean that the outer material for the coincident facet of Object 1 should be the inner material of Object 2, instead of the assumed outer material. Likewise, the outer material for the coincident facet of Object 2 should be the inner material of Object 1.

For each pair of fully coincident facets, one facet is designated as the facet to be kept, and the other is designated as the facet to be deleted. The outside material of the facet to be kept is updated to match the inside material of the facet to be deleted. The other facet and associated data (centroid, vertices, area, etc.) are then removed from the model.

The MATLAB code implemented for coincident facet resolution can be found in Appendix A, section AA.2.

II.B.3. Example: Coincident facets in the MIDA model

The MIDA model is one of the most detailed human head models currently available, comprising 115 distinct tissue compartments and roughly 11 M triangular surface elements. As previously mentioned, every tissue compartment is fully enclosed by its associated mesh. Where any two compartments touch, one boundary is contributed by each compartment. Fig. 22 below illustrates a tiny fraction of the coincident boundaries in this model.

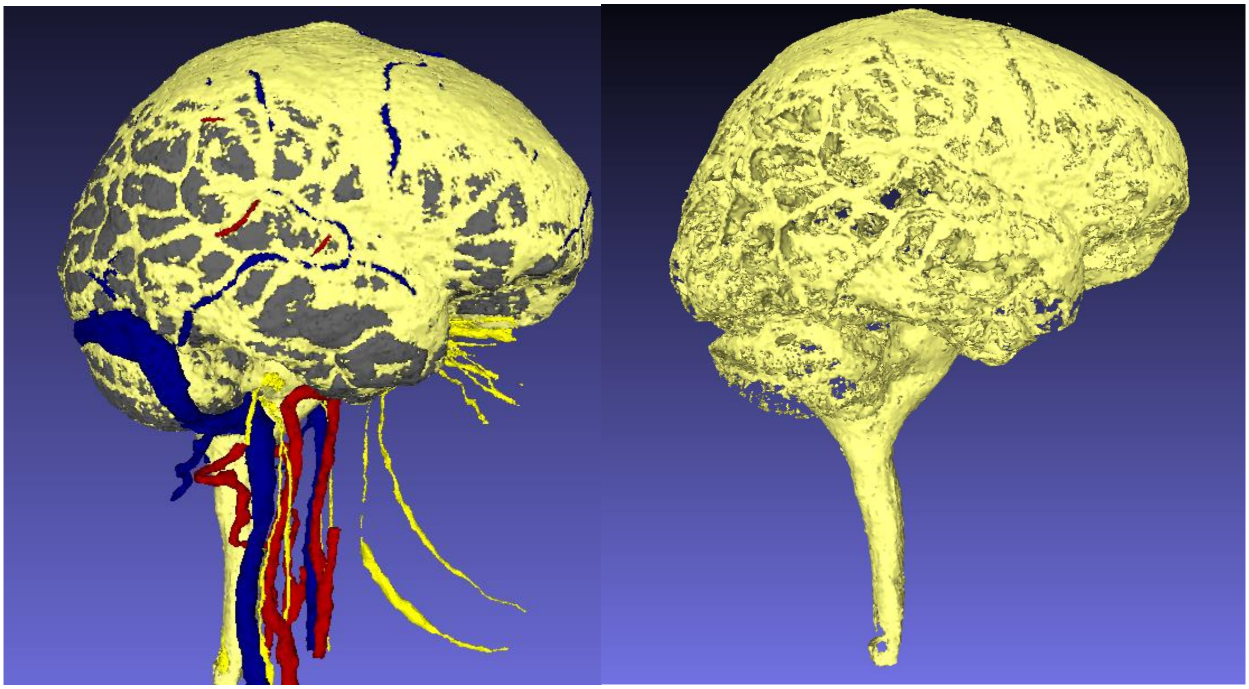


Fig. 22. Left: MIDA model's cerebrospinal fluid (pale yellow), gray matter (gray), veins (blue), arteries (red), and nerves (dark yellow). Right: The cerebrospinal fluid mesh presented in isolation from the other tissues; note the tight channels for the veins and arteries, the hollow areas for the gray matter, and the multitude of small, isolated facets.

The total number of facets in the MIDA model decreases from 11 M to 5.6 M after the coincident facet resolution step, and the operation requires approximately 81 seconds. Coincident facets need to be resolved only once during the model preprocessing phase; the time taken to resolve coincident facets does not affect the charge solution time once the model has been constructed.

Chapter VI will discuss an application example based on the MIDA model.

II.B.4. Example: a simple T-junction geometry

This section has been excerpted from Appendix B of [1].

II.B.4.a. Junction Geometry

We consider a T-junction of three distinct, non-nested conducting objects as shown in Fig. 23 below. We assume that Object 1 has conductivity σ_1 (where $\sigma_1 \neq 0$), Object 2 has conductivity σ_2 (where $\sigma_2 \neq 0$ and $\sigma_2 \neq \sigma_1$), and Object 3 has conductivity $\sigma_3 = 0$ (i.e. air). The junction line, marked by a gray circle near the top of the figure, extends out of the page. The cross section shown is perpendicular to the junction line, and the joint boundary between Object 1 and Object 2 is assumed to extend some distance in the direction of the junction line (out of the page). The primary electric field, marked in Fig. 23 by a red arrow, is parallel to the top of the T-junction (and thus perpendicular to the joint boundary between Object 1 and Object 2).

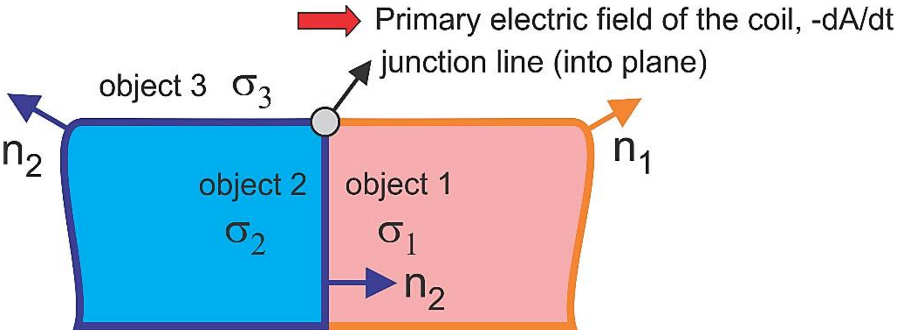


Fig. 23. Simple non-nested T-junction geometry with three objects. The joint interface between Object 1 and Object 2 is counted only for Object 2, with σ_2 being the inner conductivity and σ_1 being the outer conductivity (in the direction of the outer normal vector $\boldsymbol{n}_2$).

As previously explained, the joint interface between Object 1 and Object 2 in Fig. 23 must be counted only once. Here, we decide that it is only counted for Object 2, and thus Object 1's copies of the boundary facets will be discarded. In this case, σ_2 is assigned as the inner conductivity and σ_1 is assigned as the outer conductivity (in the direction of the normal vector $\boldsymbol{n}_2$) for this boundary as shown in Fig. 23.

II.B.4.b. Geometry Model with Secondary Field. Source Positioning

The geometry shown in Fig. 23 could be straightforwardly realized with two distinct hemispheres in air. However, a homogeneous precisely spherical geometry does not generate the secondary field for the most relevant co-axial magnetic dipoles (small coils). Therefore, we cannot directly compare the effect of the secondary electric field for the nested geometry with the additional effect of the junction for a similar yet non-nested geometry.

A more appropriate example is perhaps a composite cylinder consisting of two homogeneous cylinders joined together tip-to-tip as shown in Fig. 24a. A horizontal coil in the form of a single ring of current shown in the same figure does generate a secondary field even if the composite cylinder is homogeneous. The cylindrical geometry model and the coil model in Fig. 24a are characterized as follows (note that all dimensions have default units of meters, but they could be scaled as appropriate):

i. the two joined cylinders each have the radius of 1 m and the same height of 1 m;

ii. the conductivity of the rightmost cylinder in Fig. 24a (object 1) is 0.1 S/m, and the conductivity of the left cylinder (object 2) is 0.5 S/m. The ambient medium (object 3) is air with zero conductivity;

iii. the torus (single-ring coil) of current has the major radius of 0.25 m and the minor radius of 0.025 m; it is located at the height of 1.2 m above the center of the axis of the composite cylinder;

iv. the coil's dI/dt is chosen as $2\pi \times 3000$ A/s. For harmonic excitation used in ANSYS FEM software, which will be employed in what follows for comparison purpose, this would correspond to coil current amplitude of 1 A at 3 kHz.

II.B.4.c. BEM-FMM Result

The BEM-FMM solution for the geometry of Fig. 24a with 16 iterations runs in approximately 2 seconds on a 2.1 GHz multicore server; the final relative residual is 10^{-8}. Fig. 24b shows the resulting distribution of the surface charge density on the outer surface of the composite cylinder. A “wind rose” with four petals seen on top of the cylinder is the induced surface charge density that also exists for the homogeneous cylinder. It is responsible for the secondary (conservative) electric field of the homogeneous cylinder without the junction.

However, the two prominent narrow strips of charge with opposite polarity also seen on top of the cylinder are the effect of the junction. This effect appears to be quite strong. It is stronger than the conventional induced surface charge density.

Next, Fig. 24c shows the corresponding *total* electric field (magnitude) just inside the outer cylinder surface. A large gradient (or, strictly speaking, discontinuity) of the field is observed near the junction, in the direction perpendicular to the junction line.

Finally, Fig. 24d demonstrates the total electric field in a cross-section plane (marked white in Fig. 24a) which is located close to the cylinder top and at ¾ of the cylinder radius from the cylinder axis. A large gradient of the field is again observed near the junction, in the direction perpendicular to the junction line.

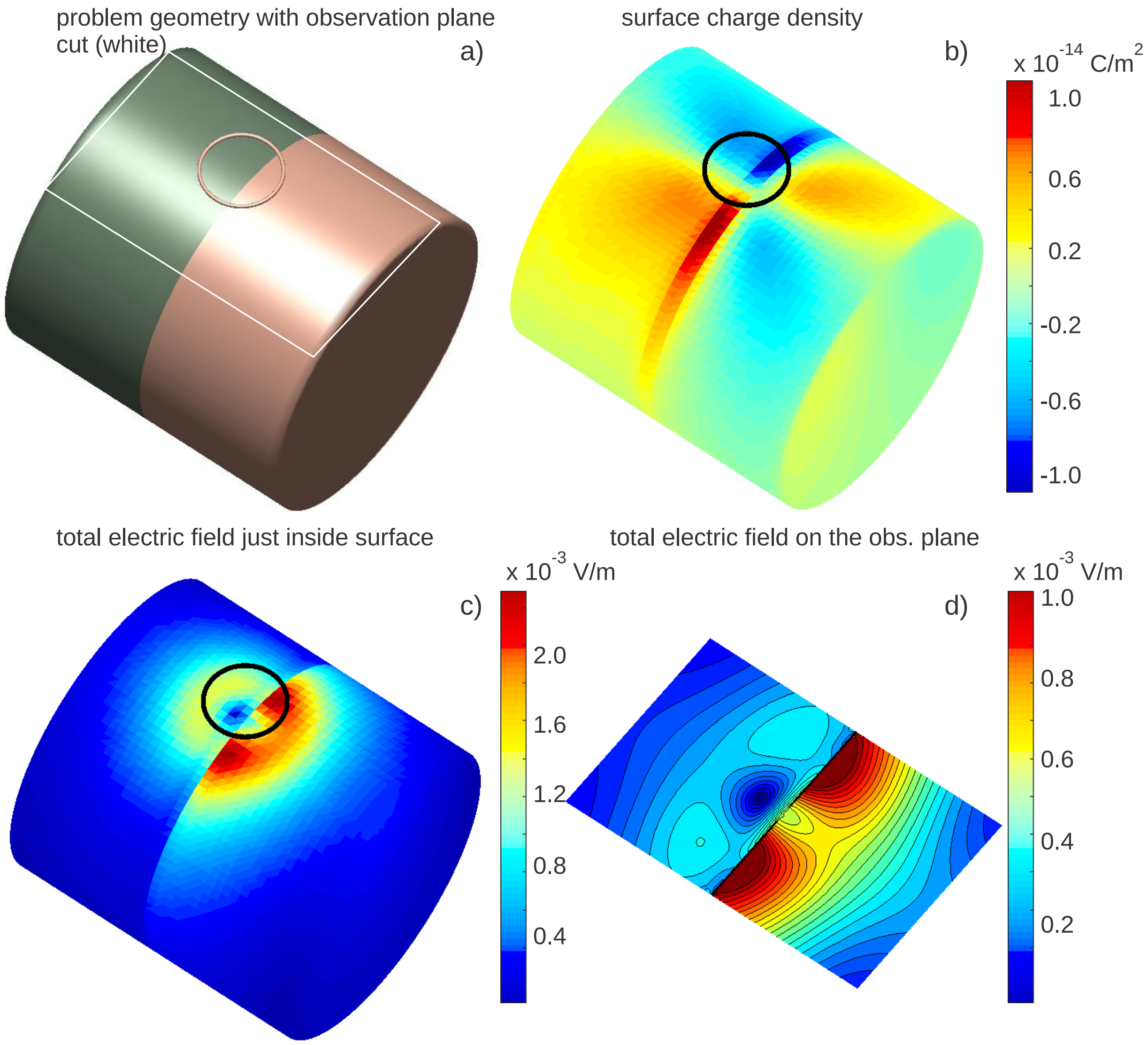
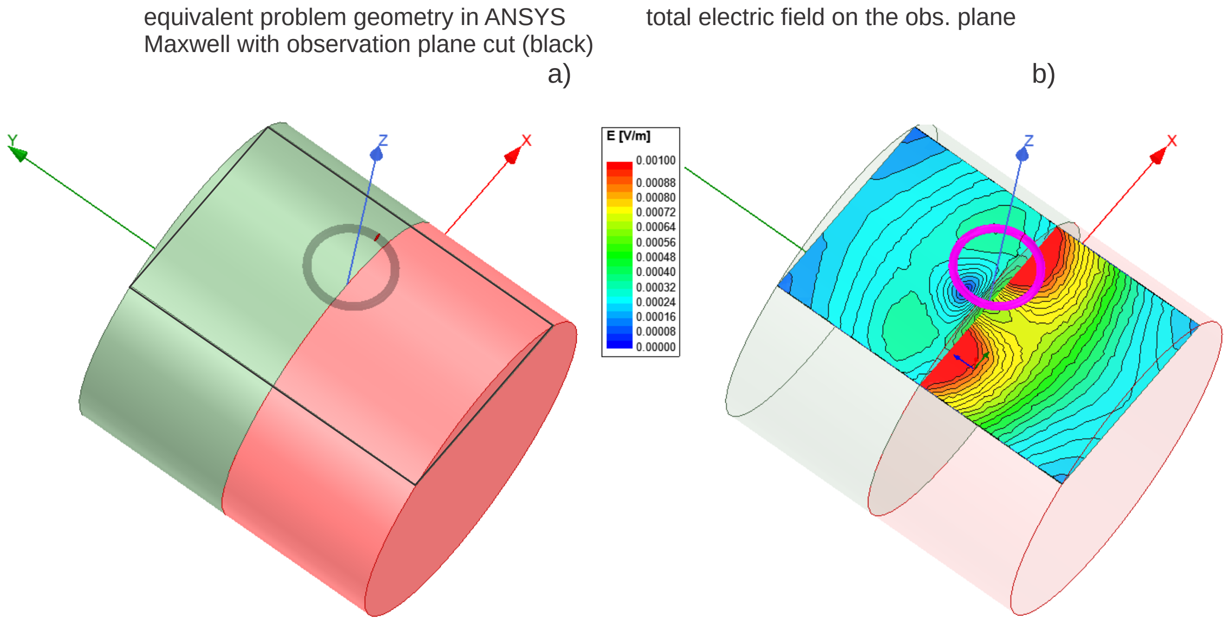


Fig. 24. a) – Geometry of the composite cylinder along with the single-ring coil and a cross-section plane marked white. b) – Resulting distribution of the surface charge density on the outer surface of the composite cylinder. c) – Total electric field (magnitude) distribution just inside the cylinder's outer surface. d) – Total electric field (magnitude) distribution in the observation plane marked white in a).

II.B.4.d. ANSYS Maxwell Result

An equivalent project was created in ANSYS® Maxwell Electronics Desktop 2019 R1 commercial FEM software (the eddy current solver) with adaptive mesh refinement. The FEM solution uses 5 adaptive passes and the final mesh with ca 0.6 M tetrahedra. The final energy error is 0.02%. The execution time is 1 hour on a 2.2 GHz multicore server.

Fig. 25 shows the corresponding problem geometry in ANSYS Maxwell and the total electric field distribution in the observation plane marked black in Fig. 25a. The observation plane is again located at ¾ of the cylinder radius from the cylinder axis. Note that, in contrast to Fig. 25 b and c, the ANSYS Maxwell eddy current modeling software does not output the surface charge distributions and/or the electric field distribution just inside or outside model surfaces.



equivalent problem geometry in ANSYS Maxwell with observation plane cut (black)

a)

total electric field on the obs. plane

b)

Fig. 25. a) – Equivalent problem geometry in ANSYS Maxwell ED 2019 R1 with the observation plane cut marked black. b) – Total electric field (magnitude) distribution in the observation plane marked black in a).

II.B.4.e. Comparison of Two Solutions

The two solutions for the electric field $\boldsymbol{E}$ in the same observation plane with $N = 40{,}000$ observation points (shown in Fig. B2d and Fig. B3b, respectively) have been compared with each other. The relative least-squares vector-field error has been calculated according to

$$\text{Error} = \sqrt{\sum_{n=1}^{N} \|\boldsymbol{E}_n^{BEM} - \boldsymbol{E}_n^{FEM}\|^2} \Big/ \sqrt{\sum_{n=1}^{N} \|\boldsymbol{E}_n^{FEM}\|^2} \qquad (21)$$

where $\|\boldsymbol{E}\|$ is the least-squares vector norm. The relative error in Eq. (B1) is equal to 3.7%, which confirms the accuracy of both solutions. The relative 2-norm (default of MATLAB) error between the two vector fields is 3.4%.

II.B.4.f. Conclusion

The surface junction in a non-nested tissue topology may cause strong localized surface charge accumulation and a (very) strong associated gradient of the electric field. The BEM-FMM solution adequately describes this effect.

II.C. Material Property Assignment to Volumetric Observation Points

II.C.1. Motivation

Several field quantities of interest depend on material properties of the observation points at which they are computed. The electric current density $\boldsymbol{J}$ at a given observation point, for instance, depends on both the electric field $\boldsymbol{E}$ at that point and the electrical conductivity σ at that point. In FEM and FDM systems, the material type and properties are already known at each volumetric element. In the surface-mesh-based BEM-FMM system, there are no volumetric elements, and material properties are assigned to the surface meshes. Material properties at volumetric observation points are thus not initially known. To compute material-dependent field quantities like current density and Lorentz force, material types for given observation points must be extrapolated based on nearby surface meshes.

II.C.2. Algorithm and Example

To assign a tissue type to a given observation point $\boldsymbol{r}$, the following method is used:

```
For each mesh index $n$:
    Find the facet on mesh $n$ whose centroid $\boldsymbol{r}'_n$ is nearest to $\boldsymbol{r}$; save this centroid.
    Assign inside/outside status to $\boldsymbol{r}'_n$ according to $sign\left(\boldsymbol{n}'_n \cdot (\boldsymbol{r}'_n - \boldsymbol{r})\right)$
        $\boldsymbol{n}'$ denotes the unit normal vector of the selected facet.
        If the expression is positive: the observation point is enclosed by mesh $n$.
        If the expression is negative: the observation point is not enclosed by mesh $n$.
Remove from consideration any meshes which do not enclose the observation point.
From the remaining centroids (one per mesh), select the mesh index $m$ that minimizes $|\boldsymbol{r}'_m - \boldsymbol{r}|$.
This mesh index $m$ is taken as the mesh whose material is present at the observation point.
```

The MATLAB code for this algorithm can be found in Appendix A, section AA.3.

Fig. 26 below shows an example of this algorithm applied to Connectome Subject 110411. Despite the complex geometry of the region shown, the algorithm correctly assigns tissue types to every location in a grid of 200x200 observation points. For these 40k observation points and this 7-compartment 860k facet model, the tissue types were assigned within 1.84 seconds.

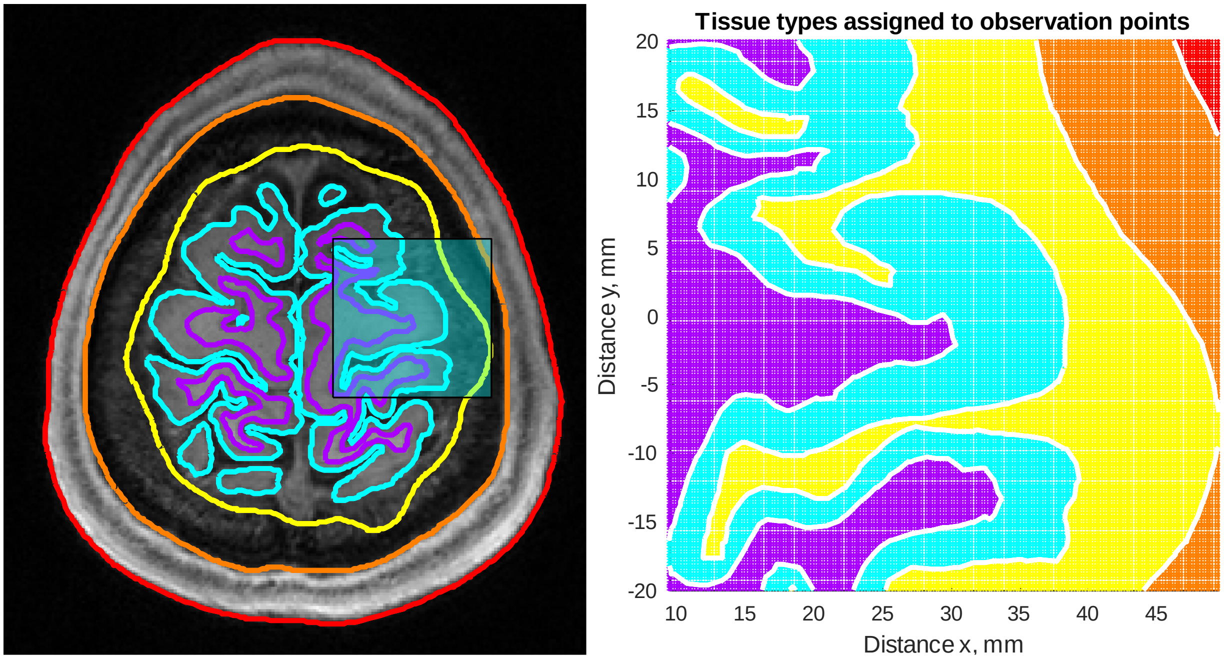


Fig. 26. Left: Transverse cross-section of Connectome Subject 110411 showing tissue meshes (red - skin; orange - skull; yellow - cerebrospinal fluid; cyan - gray matter; purple - white matter) against subject MRI data. Right: Tissue types assigned to a grid of 200x200 observation points in the cyan rectangle in the left figure, following the same color scheme. The white dots visible in the right figure are interstitial spaces between observation points.

III. TMS-Focused BEM-FMM Toolkit

This chapter has been excerpted from [1], published in April 2020. It describes specific ways in which the method was tailored to solve TMS problems and provides an example based on sixteen subjects of the Human Connectome Project.

III.A. TMS-Specific Implementation Details

III.A.1. Charge conservation law

In addition to excerpts from [1] proper, this section contains excerpts from Appendix C of that publication.

Before applying a TMS pulse, the human head or the entire human body is assumed to have the total charge Q equal to zero, i.e. be electrically neutral. Since the electric charge is a physical quantity carried by electrons or ions, and since there is no direct conduction (or charge transport) path between the head and a coil separated from the head by a gap, the total charge of the head must not change. Therefore, the total charge must be equal to zero at any time instant.

This observation is valid not only for TMS excitation but also for any isolated, conducting, or dielectric object in an external electromagnetic field. Examples pertinent to static, eddy-current, and full-wave problem statements are given in [73] and [74].

The charge conservation law is not explicitly included in Eq. 4; it can be enforced in the form

$$\int_S \rho(\boldsymbol{r}')d\boldsymbol{r}' = 0 \tag{22}$$

In Eq. 22, S is now the combination of *all* interfaces. Proper implementation of Eq. 22 implies a direct combination with Eq. 4 with a certain weighting; it provides a significantly better and unconstrained convergence rate of the iterative solution and prevents excess charge accumulation at sharp corners of the model. Section III.B.3 discusses the charge conservation law (Eq. 22) in detail and presents examples that demonstrate its impact on solution convergence and accuracy.

III.A.2. Primary TMS field

An arbitrary TMS coil is modeled in the form of a very large number of small straight elements of current $i_j(t)$ with orientation vector $\boldsymbol{s}_j$ and center coordinate vector $\boldsymbol{p}_j$. Those elements can be either uniformly (Litz wire) or non-uniformly (skin effect) distributed over every conductor's cross-section. The magnetic vector potential $\boldsymbol{A}^p$ of a current element with orientation $\boldsymbol{s}_j$ and position $\boldsymbol{p}_j$ at an observation point $\boldsymbol{c}_i$ is given by [74]

$$\boldsymbol{A}^p(\boldsymbol{c}_i, t) = \frac{\mu_0}{4\pi}\frac{i_j(t)\boldsymbol{s}_j}{|\boldsymbol{c}_i - \boldsymbol{p}_j|} \tag{23}$$

where μ_0 is magnetic permeability of vacuum and index p denotes the primary field. The corresponding solenoidal electric field is $\boldsymbol{E}^p = -\partial \boldsymbol{A}^p / \partial t$. Substitution of Eq. 23 yields

$$\boldsymbol{E}^p = \frac{\mu_0}{4\pi} \frac{\boldsymbol{s}_j}{\left|\boldsymbol{c}_i - \boldsymbol{p}_j\right|} \left\{ -\frac{di_j}{dt} \right\} \tag{24}$$

The multiplicative factor in curly brackets is a known characteristic of the coil pulse waveform. For every observation point, the primary electric field in Eq. 24 is computed via the FMM as a potential of a single layer repeated three times, i.e., separately for each component of the field. Examples of detailed TMS coil models are shown in Fig. 27; the coil geometry generator of the toolkit is described in Appendix B.

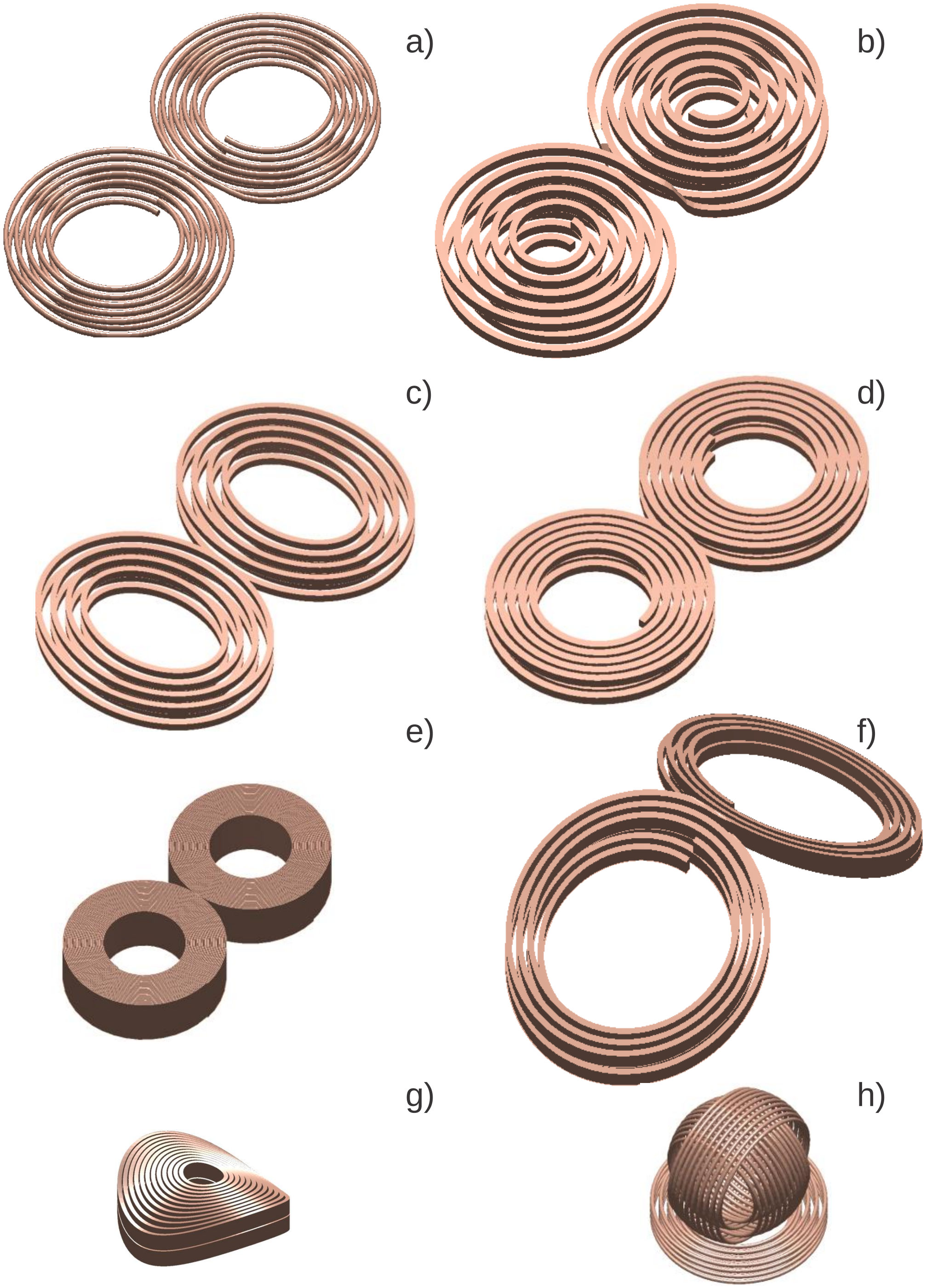


Fig. 27. Solid CAD models created using the MATLAB-based coil geometry generator. Figs. 27a-b are generic double figure-eight spiral coil models with different conductor cross-sections and bootstrapped interconnections. Figs. 27c-g are simplified commercial coils of MagVenture, Denmark. Fig. 27c is a MRi-B91 TMS-MRI coil with elliptical conductors of a rectangular cross-section used in the example; Fig. 27d is a MagPro C-B60 coil model; Fig. 27e is a compact Cool B-35 coil; Fig. 27f is a D-B80 coil; Fig. 27g is Cool-40 Rat small animal coil model. Fig. 27h is a three-axis multichannel TMS coil array radiator [75][76].

III.A.3. Excitation

The excitation is the term on the right-hand side of Eq. 4, which depends on $\boldsymbol{E}^p$. It is first evaluated at mesh nodes via the FMM, and then an average value for every facet center is obtained. The initial guess of an iterative solution is proportional to the excitation term. Based on analytical solutions for the sphere and ellipsoid, a scalar weighting parameter for the initial guess may be chosen in the range from 1 to 10.

III.A.4. Iterative solution

In the iterative matrix-free solution, Eqs. 4 and 22 are added together and solved simultaneously. The weighting parameter for the conservation law normalized by the total surface area is chosen as 0.5. The generalized minimum residual method (GMRES) was found to converge better than the bi-conjugate gradients method and its variations. Several sparse near-field preconditioners have been constructed, but so far none has provided a significantly better convergence speed. The relative residual of 10^{-10} is achieved in approximately 60 iterations for a typical head model discussed further. However, such a large number of iterations is not necessary as shown below.

III.A.5. Discretization error and surface charge averaging

From the viewpoint of electromagnetic field theory [77], any sharp edge in the surface mesh will lead to an infinite value of the surface charge density at that location, which indeed results in an infinite electric-field value when the mesh surrounding this edge is refined. Fortunately, this is an integrable singularity [77]. In order to compare models with different surface resolutions, we must therefore introduce averaging (integration) over a consistent and small surface area. For practical purposes, it is convenient to introduce equally weighted surface charge averaging (low pass filtering) over three or more immediate topological neighbors. After the solution is obtained, we substitute (for three topological neighbors)

$$\rho(t_m) \to \frac{1}{4}(\rho(t_m) + \rho(t_{m1}) + \rho(t_{m2}) + \rho(t_{m3})) \qquad (25)$$

where triangles t_{mn} share an edge with triangle t_m. Eq. 25 has been implemented in the code. For a very fine mesh, it could be applied twice to expand the relative averaging domain.

III.A.6. Postprocessing

The normal electric field just inside/outside an interface is computed directly from the known surface charge density following Eqs. 5. The total electric field just inside/outside an interface follows Eq. 2 with all neighbor potential integrals computed analytically. The total electric field anywhere in the volume follows Eq. 1 with all neighbor potential integrals again computed analytically.

III.A.7. Available subject head models

Sixteen Human Connectome Project (HCP) head models [78] with an initial isotropic voxel resolution of 0.7 mm are available with the software [72]. These MRI data have been converted to surface models using the SimNIBS v2.1 pipeline; each model includes seven brain compartments (skin, skull, CSF, GM, WM, ventricles, cerebellum). Each model has been checked against the original NIfTI images, and mesh manifoldness has been strictly enforced and confirmed using ANSYS HFSS mesh checker. The default average cortical surface mesh edge length is 1.5 mm, the cortical nodal density is 0.55 nodes per mm^2, and the total number of facets is 0.9 M.

Any other surface model may be used in *.stl or *.mat (MATLAB) format. A chief example is the MIDA model or its parts [71]. Additionally, the fifty CAD models included in the Population Head Model Repository or PHM [79],[80] may be used, which have been made available by the IT'IS Foundation, Switzerland via the web [81]. Note that some inconsistencies were observed when overlapping these PHM models with the original NIfTI images.

III.A.8. Model remeshing and surface mesh registration

The MATLAB package also makes use of CM2 SurfRemesh®, a mesh generation program from Computing Objects, France, that enables the user to create coarser and/or finer surface representations while minimizing the surface deviation error from the master mesh. A NIfTI viewer available in the core MATLAB package facilitates surface mesh registration in any plane by overlapping surface mesh cross-sections on NIfTI images. We should note that for most accurate registration the original MRI data header information should be used, and the provided viewer is predominantly to facilitate quick visualization in MATLAB.

III.A.9. Implementation and distribution

The complete BEM-FMM algorithm is implemented entirely in MATLAB® 2019b for both Windows (runs as is, but may require machine-specific compilation of the FMM library for further speed up) and Linux (may require machine-specific compilation of the FMM library). As mentioned above, the software includes the recent FMM library [66], is compatible with its latest modifications, and is bundled together with remeshing and registration modules. The complete package is available to interested researchers via a GitHub repository [54].

III.B. Results

III.B.1. Necessary number of iterations for the iterative solution

A critical point for the both the speed and performance of the method is the number of GMRES iterations to be used in the iterative solution. This number, which determines the overall algorithm's speed, was not quantified before.

In order to establish the representative estimate, six Connectome Project head models (subject numbers 110411, 117122, 120111, 122317, 122620, 124422), [78] and the corresponding surface models obtained via the SimNIBS v2.1 pipeline and described above were tested. The average cortical surface mesh edge length is 1.5 mm; the cortical nodal density is 0.55 nodes per mm^2; the total number of facets is 0.9 M.

Further, the remeshing program described above was applied. As a result, a coarser set with an average cortical edge length of 1.9 mm and an average cortical nodal density of 0.32 nodes per mm^2 was created; the total number of facets is 0.4 M. Likewise, a finer set with an average cortical edge length of 0.99 mm and a cortical nodal density of 1.2 nodes per mm^2 was created; the total number of facets is 1.8 M.

The models were augmented with the material conductivities identical to the default set of SimNIBS TMS software package: scalp average – 0.465 S/m, skull – 0.01 S/m, CSF – 1.654 S/m, GM – 0.275 S/m, cerebellum – 0.126 S/m, WM – 0.126 S/m, ventricles – 1.654 S/m. Note that the available literature data vary widely; the corresponding conductivity ranges have been reviewed in [82]. In particular, the IT'IS database of tissue properties [83] reports somewhat different values: skull – 0.0203 S/m, CSF – 2.0 S/m, GM – 0.106 S/m, cerebellum – 0.126 S/m, WM – 0.065 S/m, ventricles – 2.0 S at 3 kHz center frequency.

The widely used MRI compatible TMS coil MRi-B91 (MagVenture, Denmark) located above the motor hand area of the precentral gyrus (the hand knob area [84]), was employed in these tests. A detailed coil model has been constructed and approximated by 26 thousand elementary current segments in Eq. 24. The coil was driven by a time-varying current of $\frac{dI}{dt} = 9.4e7\ Amperes/sec$. The primary coil field was computed using Eq. 24.

In every case, the coil was positioned in order to follow three geometrical rules:

i. align the approximately identified hand knob area of the right precentral gyrus with the coil centerline;
ii. set the coil centerline approximately perpendicular to the skin surface, and position the coil 10±0.25 mm from the skin along this centerline;
iii. have the dominant field direction (the x-axis of SimNIBS coordinate system) approximately perpendicular to the gyral crown (and associated sulcal walls) of the precentral gyrus pattern at the target point.

These rules uniquely define the coil position and the rotation angles.

The most sensitive error parameter is the value of the absolute maximum field observed at the cortical interfaces and the exact position of this local maximum. Figs. 28 a,b,c show the error in the maximum value of the total field just inside the pial cortical surface or the GM shell (red) and just inside the inner cortical surface or the WM shell (blue) as a function of iteration number versus the most accurate solution with 100 iterations for the three different model resolutions described above. Figs. 28 d,e,f give the error in the position of this maximum field in millimeters as a function of iteration number versus the most precise solution for the same three cases. The vertical line in every plot corresponds to the 15th iteration. Results are only given for subject #110411 of the Connectome Project Database, but nearly identical results have been observed for the five other subjects considered. Based on these results we conclude that 14-15 iterations are sufficient to obtain a maximum-field error below 1% and a maximum-field position error below 0.25 mm.

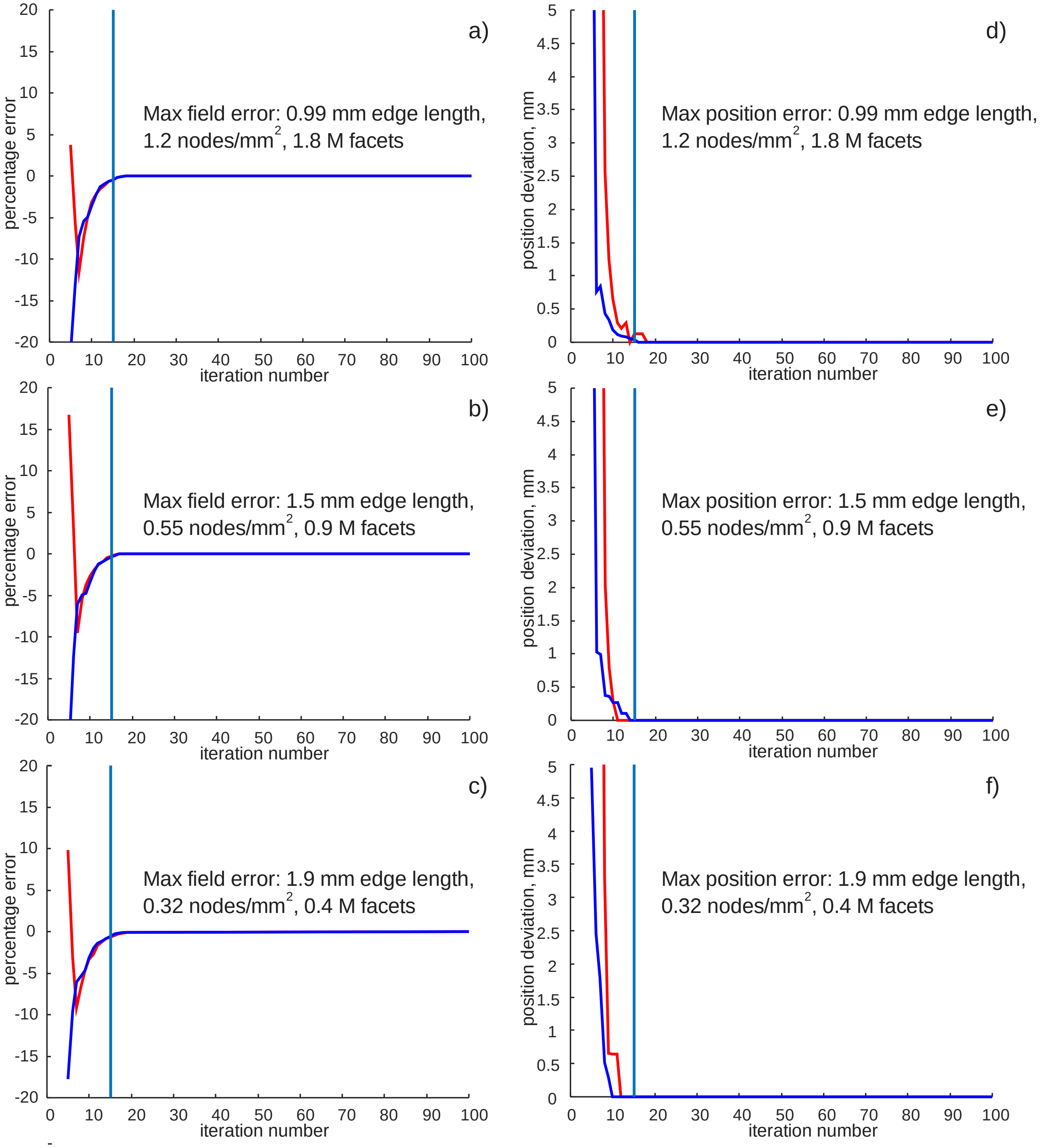


Fig. 28. a,b,c) – Error in the maximum value of the total field just inside the GM shell (red) and WM shell (blue) as a function of iteration number versus the most accurate solution with 100 iterations for three different model resolutions. d,e,f) – Error in the position of this maximum field in millimeters as a function of iteration number versus the most precise solution for three different model resolutions. The vertical line in each plot corresponds to the 15th iteration. Results are given for subject #110411 of the Connectome Project Database. Very similar results were observed for the five other subjects considered.

Table 6 gives the error in the maximum value of the total field just inside the inner cortical surface and the error in the position of this maximum field in millimeters versus the most accurate solution available (with 1.2 nodes/mm^2, 1.8 M facets, 100 iterations) for the two lower model resolutions. Results are given for all six subjects listed above. This surface-charge averaging area in Eq. 25 has been defined as 3 mm^2. One can see that both errors are relatively small, even for the coarser model with 0.4 M facets.

Table 6. Error in the maximum value of the total field just inside the inner cortical surface (the WM shell) and the error in the position of this maximum field in millimeters versus the more precise solution (with 1.2 nodes/mm^2, 1.8 M facets) for two lower model resolutions. Results are given for six subjects of the Connectome Project Database.

Connectome subject	**Max field error for 1.5 mm edge length (0.55 nodes/mm^2, 0.9 M facets)**	**Error in the position of the max field (0.55 nodes/mm^2, 0.9 M facets)**	**Max field error for 1.9 mm edge length (0.32 nodes/mm^2, 0.4 M facets)**	**Error in the position of the max field (0.32 nodes/mm^2, 0.4 M facets)**
110411	0.6%	0.2 mm	3.8%	0.9 mm
117122	1.2%	0.1 mm	1.2%	1.7 mm
120111	0.9%	0.2 mm	0.2%	1.1 mm
122317	0.3%	0.7 mm	2.3%	1.5 mm
122620	0.5%	1.1 mm	2.1%	0.9 mm
124422	2.2%	0.6 mm	3.5%	0.8 mm
Average	**0.9 %**	**0.5 mm**	**2.2%**	**1.2 mm**

Table 7. BEM-FMM run times for different model resolutions obtained with 14 iterations. These data were compiled using an Intel Xeon E5-2698 v4 CPU (2.10 GHz) server, 256 GB RAM, MATLAB 2019a.

Model resolution	**Preprocessing time (once per model)**	**Solution time for 14 iterations (cf. Fig. 1)**	**Post processing time for normal surface cortical fields and field discontinuities*)**
1.2 nodes/mm^2, 1.8 M facets	110 sec	74 sec	0 sec
0.55 nodes/mm^2, 0.9 M facets	70 sec	34 sec	0 sec
0.32 nodes/mm^2, 0.4 M facets	50 sec	18 sec	0 sec

) Post-processing time for the total/tangential interfacial fields should also approach zero once the last GMRES iterations are made available in the workspace.

III.B.2. Overall method speed

Based on results shown in Fig. 28 and quantified in Table 6, we estimate the speed of the method sufficient for an accurate solution pertinent to different model resolutions in Table 7. A significant speed improvement by a factor of approximately 3 as compared with previous results [49],[50] is achieved by using an improved FMM algorithm, by employing the proper number of iterations, by lowering the intrinsic FMM precision to an acceptable level without compromising the overall method accuracy, and by the explicit inclusion of the charge conservation law into the iterative solution.

III.B.3. Impact of the charge conservation law

In addition to excerpts from [1] proper, this section contains excerpts from Appendix C of that publication.

The global charge conservation law (Eq. 22 of III.A.1) is normalized as follows:

$$\int_S \rho(\boldsymbol{r}')d\boldsymbol{r}' / \int_S d\boldsymbol{r}' = 0 \tag{26}$$

where S is the combined area of all head compartments. Then, it is added to Eq. 4 of I.B.5. with the weight of ½ (found empirically – in fact, any weight value between 0.01 and 100 might also be acceptable). After that, modified Eq. 4 is solved iteratively using the generalized minimum residual method or GMRES. A MATLAB GMRES implementation (Drs. P. Quillen and Z. Hoffnung of MathWorks, Inc) is used.

III.B.3.a. Solution Convergence

Fig. 29 demonstrates the outcome of this correction on the solution convergence for three different models: the present example with three spheres (0.06 M facets) – Fig. 29a; the example of the main text for Connectome subject 110411 (0.86 M facets) – Fig. 29b; and the default Ernie head model of SimNIBS with 0.96 M facets [85] – Fig. 29c. Here, we compare two convergence curves: with (blue) and without (red) the conservation law.

In all three cases, the conservation law allows us to avoid convergence saturation at certain values of the relative residual and thus achieve a nearly uniform convergence rate for any residual values. In all three cases, the non-saturated convergence reduces the final relative residual by approximately two orders of magnitude or even slightly more.

In the first case in Fig. 29a, the plateau of saturated convergence is not very well visible. However, it becomes more apparent and increases in length when the model complexity increases (Figs. 29b and 29c). Also, the saturation zone in Fig. 29 moves toward smaller (and more important) values of the relative residual when model complexity increases. From the user's point of view, the plateau of saturated convergence creates a concern.

Note that in the exact spherical geometry and for a uniform primary field, the charge conservation is already satisfied very well. Introducing the conservation of charge has only a minor effect.

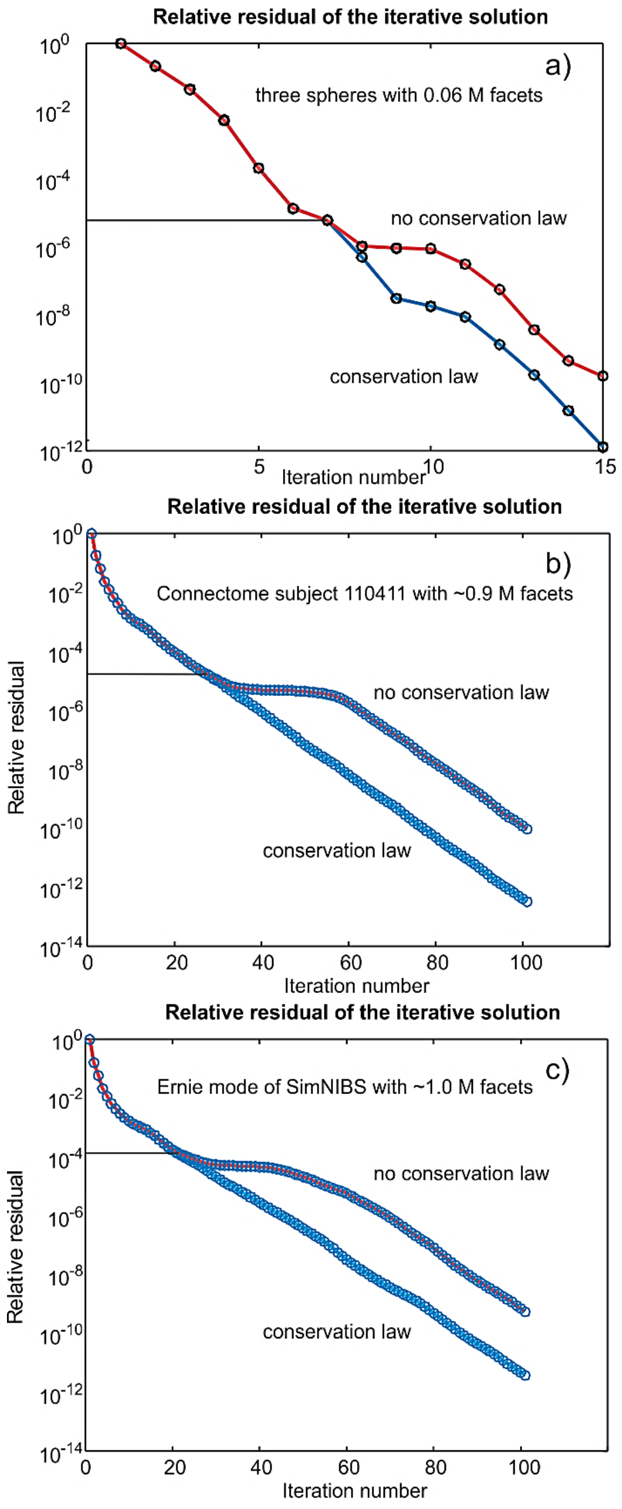


Fig. 29. GMRES convergence rates with (blue) and without (red) charge conservation with increasing model complexity. a) – The three-sphere example. b) – Connectome subject 110411. c) – Ernie model of SimNIBS. A horizontal line marks the iteration number where the two curves start to deviate.

III.B.3.b. Charge solution accuracy

As an example, we now consider the non-nested geometry of the composite cylinder previously studied in I.B.4. Fig. 30a shows the solution convergence for the example of I.B.4. with (blue) and without (red) charge conservation, while Fig. 30b displays the map of the solution error for the example of I.B.4 between the solutions with and without charge conservation, respectively. The solution error for the primary unknown – the surface charge density – is further quantified in Table 8. Despite small values of the relative residual in both the cases, the relative solution error (2-norm) reaches 3.1% while the maximum relative solution error versus the absolute mean charge density reaches 12.5%.

Table 8. Solution performance after 17 iterations (the first iteration is trivial) for the example of I.B.4. The solution error is given versus the solution with charge conservation law (Eq. 26).

Solution type	Relative residual	Relative error in global charge cons. law, %	Solution error for surface charge density (2-norm), %	Maximum relative solution error vs. absolute mean charge density, %
With charge conservation (Eq. 26)	5.9e-10	-3.2e-04	**0**	**0**
Without charge conservations	3.2e-07	-4.6e-02	**3.1%**	**12.5%**

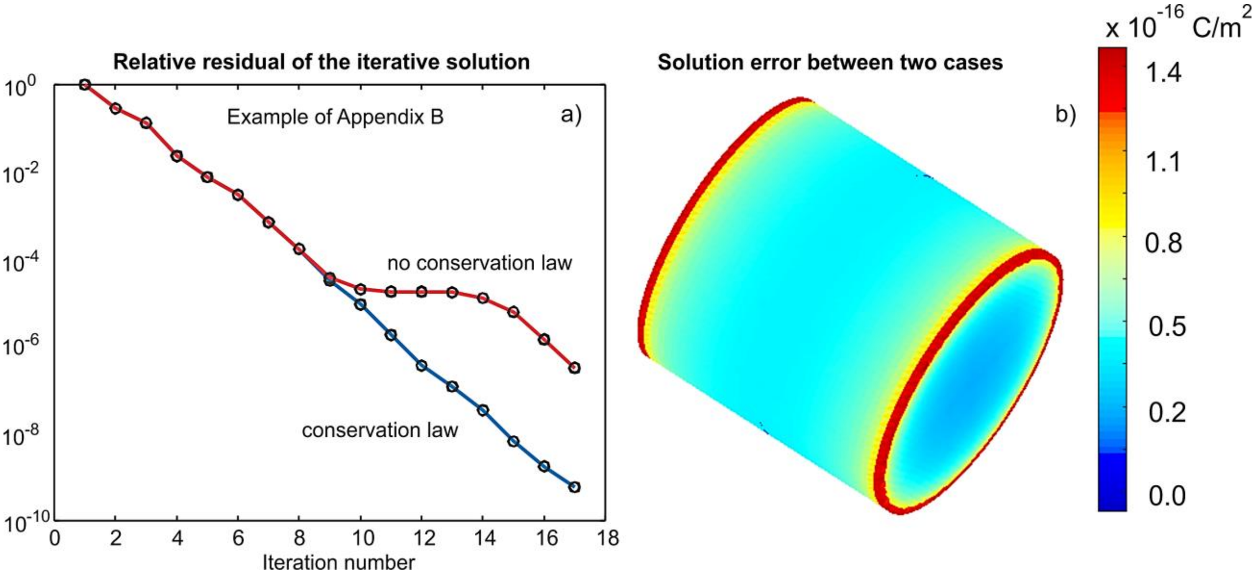


Fig. 30. a) – Solution convergence for the example of I.B.4 with (blue) and without (red) charge conservation. b) – plot of the solution error for the example of I.B.4 between the solutions with and without charge conservation, respectively.

III.B.4. Modeling example: computing total, tangential, and normal fields at the interfaces

The developed algorithm is applied to accurately compute normal, tangential, and total electric fields anywhere in the cerebral cortex for a specific subject and a specific coil orientation. In this example, particular attention is paid to modeling the field in the vicinity of the folded WM-GM interface (the inner cortical surface).

Electric fields in the brain were simulated using the MRi-B91 coil model, located above the motor hand area of the right hemisphere in six distinct subject models of the Connectome Project described above in section III.B.1. Only the finer resolution models with an average cortical edge length of 0.99 mm, a cortical nodal density of 1.2 nodes per mm^2, and a total number of facets of 1.8 M have been used for simulations. We employed straightforward geometrical coil positioning as described in section 3.1 above.

Fig. 31a shows the computed magnitude of the total surface electric field for subject #110411 just inside the pial cortical surface (the GM shell). This is the typical non-focal gray-matter field distribution observed in many relevant studies. The field distribution includes a number of sparse local maxima, one of which is located close to the coil centerline. The absolute total-field maximum for the plot is 123.6 V/m. In Fig. 31a, some red spots span small groups of triangles. We believe that this happens when the GM shell is situated very close to the CSF so that a “capacitor effect” occurs.

In general, the field just inside the GM shell corresponds to cortical layer I, which is likely of little interest for TMS activation as it is a molecular layer, and contains few scattered neurons [86].

A different situation might occur when we evaluate the total field magnitude just outside or inside the inner cortical surface (the WM shell), as shown in Fig. 31b. For this particular example, the absolute total-field maximum just inside the WM shell is somewhat higher, at 133.4 V/m. Interestingly, the field has become quite focal; the maximum field is concentrated in a well-defined domain marked by an arrow in Fig. 31b.

Remarkably, this focal domain resides in the area of the superior parietal lobule, just behind the postcentral sulcus and rather far from the targeted hand knob area of the precentral gyrus. The distance from the coil centerline intersection with the WM shell to the center of the depicted hot spot is 32 mm. The distance from the coil centerline intersection with the GM shell is even longer. This observation is consistent with the previously established fact that the apparent TMS motor map may extend due to remote hotspot activation [87].

Which field component generates this local maximum? To answer this question, Fig. 31c plots the absolute value of the normal field just inside the WM shell. In this instance, the focal area is even more pronounced with an absolute field maximum of 130.4 V/m. Comparing this number with the previous value of 133.4 V/m, we conclude that the normal field component is primarily responsible for this maximum.

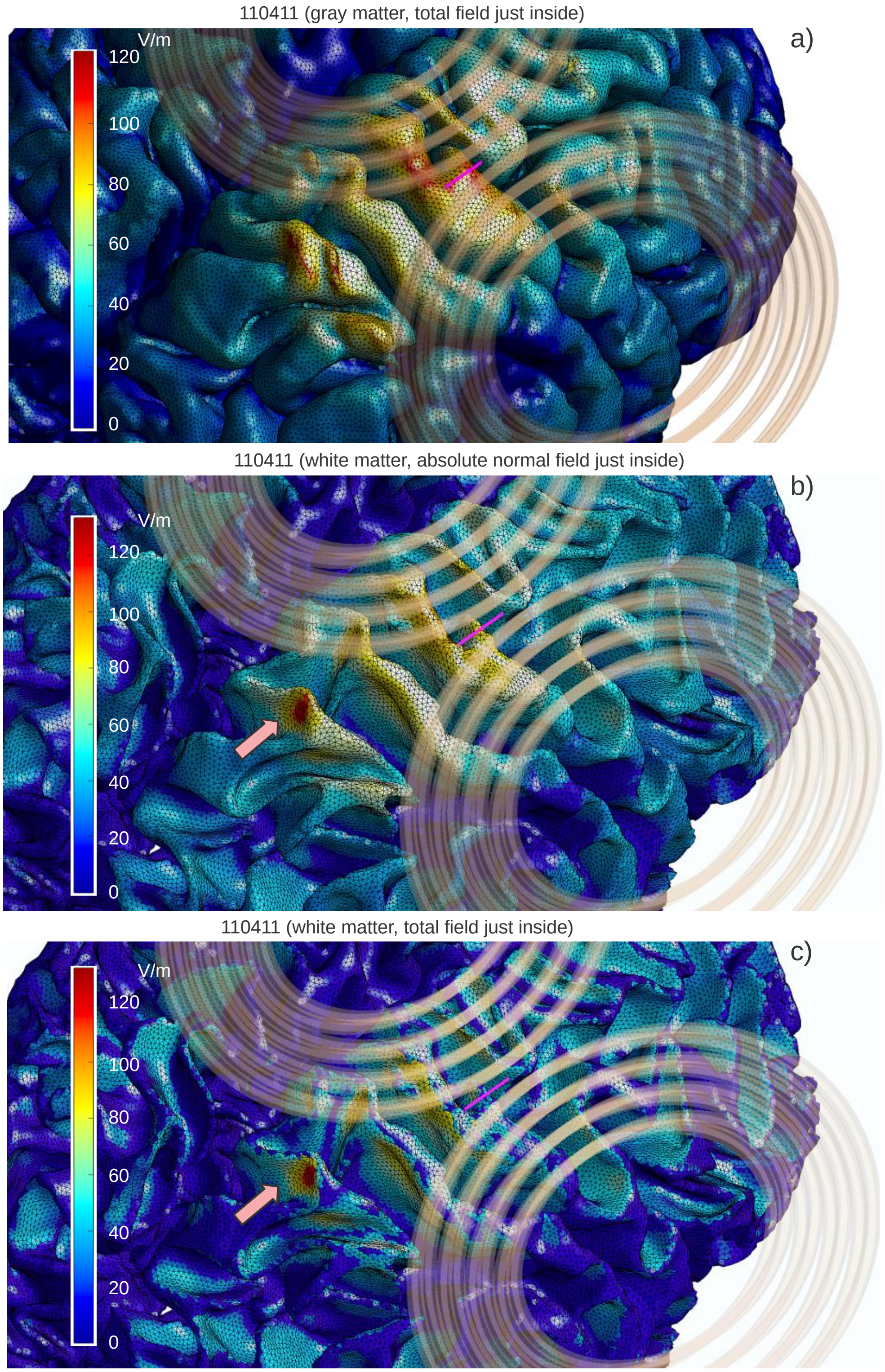

110411 (gray matter, total field just inside)
a)
V/m
120
100
80
60
40
20
0
110411 (white matter, absolute normal field just inside)
b)
V/m
120
100
80
60
40
20
0
110411 (white matter, total field just inside)
c)
V/m
120
100
80
60
40
20
0

Fig. 31. a) – Computed total surface electric field (magnitude) distribution just inside the pial surface (GM shell) for subject #110411. The absolute field maximum for the plot is 123.6 V/m. b) – The same result just inside the WM shell; the absolute field maximum there is even higher: 133.4 V/m. c) The normal-field magnitude just inside the WM shell; the field maximum is 130.4 V/m.

III.B.5. Modeling example: computing normal fields and their discontinuities at the WM-GM interface

The field normal to the inner cortical surface (the WM-GM interface) is the field parallel to the long, either straight or bent pyramidal axons of the fast-conducting pyramidal tract neurons passing through this interface. The normal field discontinuity (or, rather, a very rapid field variation) across the WM-GM interface creates a strong gradient of the component of the electric field along the axon, including the effect of the axonal bending [88][89]. It has been suggested that such a gradient may cause stimulation [88][89]. The stimulation of pyramidal axons of the fast-conducting pyramidal tract neurons results in D (direct or earliest) TMS wave generation [90]-[95].

The TMS activation mechanisms are complex. The most common TMS response observed at lower suprathreshold field intensities are not the D waves but multiple I (or indirect) waves, which originate from indirect, trans-synaptic activation of pyramidal neurons [91]-[94], [96]-[98].

Fig. 32 shows the WM-GM interfaces (the WM shells) for the six Connectome subjects considered. Small blue spheres are drawn at the center of every WM facet where the absolute normal field value just inside the surface is in the range of 80-100% of the maximum normal-field value for the same surface. Identical "hot spots" would also be observed for the normal component just outside the interface and for the normal field discontinuity across the interface, according to Eq. 5.

In every case, the TMS coil is targeting the motor hand area of the right hemisphere. The total WM areas with the normal-field values within 80-100% of the maximum normal-field value are 40, 55, 120, 150, 139, and 79 mm^2; the standard deviations of all high-field values from the maximum-field position are 5.5, 8.3, 8.4, 9.9, 15, and 6.0 mm for the six Connectome subjects 110411, 117122, 120111, 122317, 122620, and 124422 respectively.

One can see that, in the majority of cases in Fig. 32, the TMS response with regard to the normal inner cortical field becomes sparse and often significantly deviates from the coil centerline. The intersubject variations are also strong.

In this example, the most remarkable result has been observed for subject #110411 (Fig. 32a). The total WM area covering 80-100% of the maximum field in Fig. 32a is compact; its size is only 43 mm^2 (approximately 6.6 mm × 6.6 mm). According to Eq. 5, the same focal area is observed for both the normal field just outside the inner cortical surface and for the normal field discontinuity across the inner cortical surface. Note again that the maximum values of the field just inside and outside the WM shell are 134.2 V/m and 82.3 V/m, respectively, demonstrating significantly higher field values just inside the inner cortical surface and a large field discontinuity. This result directly follows from Eq. 5 when the corresponding conductivity values are substituted.

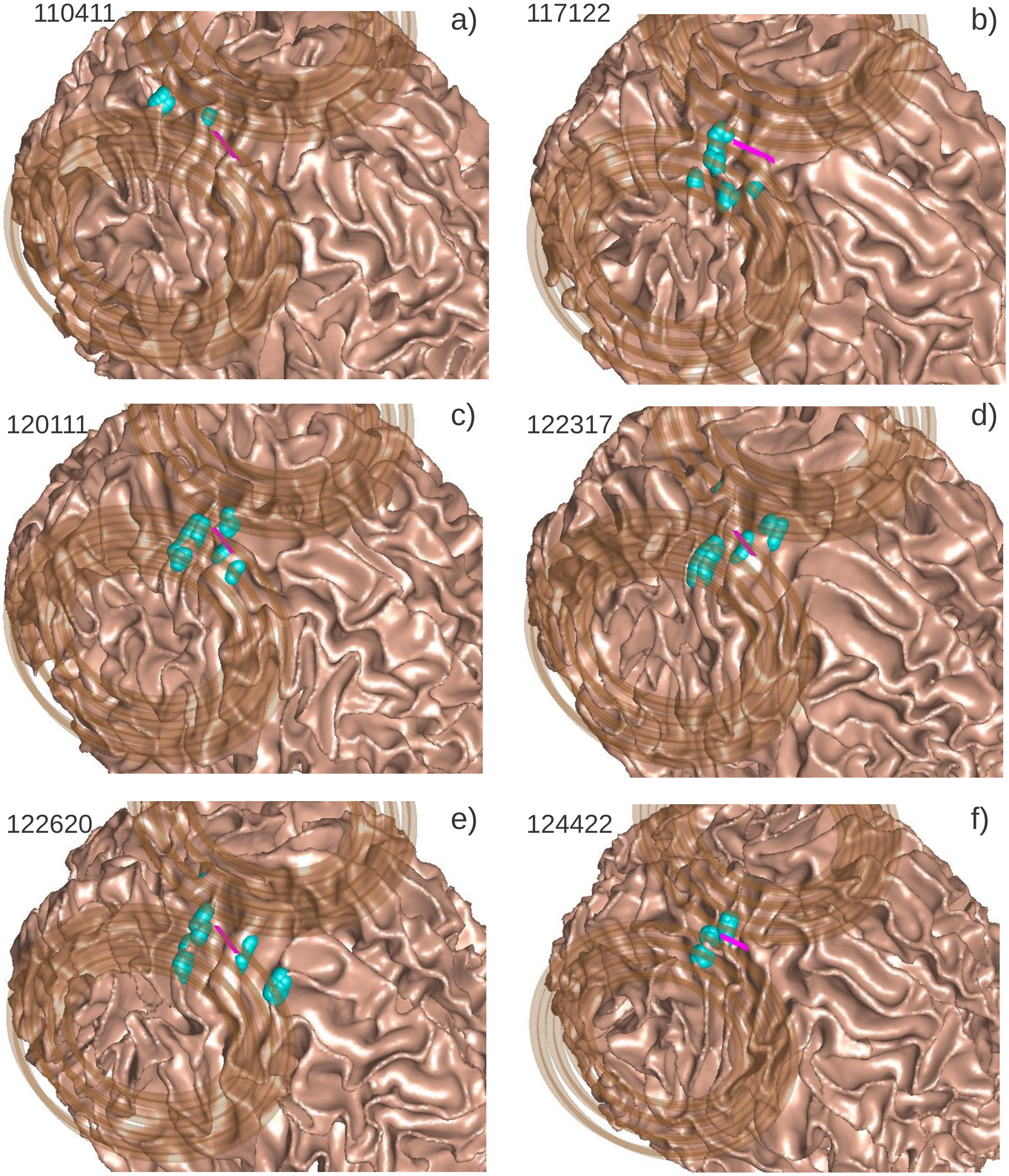


Fig. 32. Distribution of E-field “hot spots” for the normal-field component at the inner cortical surfaces using geometrical coil positioning. a)– f) – Inner cortical surfaces (WM-GM interfaces) for six Connectome subjects with small blue spheres drawn at the center of every WM facet where the absolute normal-field value just inside the surface is in the range of 80-100% of the maximum normal-field value for the same surface. In every case, the MagVenture MRiB91 coil is targeting the motor hand area of the right hemisphere via geometrical positioning. The total WM areas with the normal-field values within 80-100% of the maximum normal field are 43, 119, 120, 170, 197, and 82 mm^2; the standard deviation of all high-field values from the maximum-field position is 2.4, 9.0, 8.5, 12, 17, and 6.6 mm for Connectome subjects 110411, 117122, 120111, 122317, 122620, and 124422 respectively.

A separate study has been performed to test if the result for subject #110411 is stable with respect to perturbations in the coil position. In order to accomplish this, the algorithm was straightforwardly modified to run in a parametric loop with no graphical output. It was found that, if the variations in the coil rotation angles and in the three coil coordinates do not each exceed 3%, both the focal position and the focal field value remain stable.

III.B.6. Modeling example: computing high-resolution volumetric field distributions

To demonstrate the BEM-FMM numerical field resolution capability, we consider the region in the vicinity of the E-field maximum in Fig. 32a, where the electric field likely changes very rapidly. The corresponding results for three principal planes passing through the field maximum position in Fig. 32a are given in Figs. 33, 34, and 35. In Figs. 33a, 34a, and 35a, the surface mesh has been overlaid on top of the relevant NIfTI slice to demonstrate mesh registration with the original imaging data. The red dots again signify the centers of intersected WM facets where the absolute normal-field value is in the range of 80-100% of the maximum normal-field value. Field localization in these planes is very good.

The volumetric field distribution in Figs. 33b, 34b, and 35b was obtained in the areas labeled by white rectangles in Fig. 33a, 34a, and 35a. Each area is 20×20 mm^2 and is centered exactly at the location of the absolute field maximum. Each area contains 500×500 observation points, resulting in a field resolution of 40 µm. The volumetric field distribution is given for the magnitude of the *total* electric field. Specific colors designate interfaces: blue corresponds to the pial cortical surface, purple defines the inner cortical surface, and yellow defines the skull-CSF interface.

We see in Figs. 33b, 34b, and 35b that the local volumetric field inside the narrow WM gyrus changes very rapidly, at the submillimeter scales. The same is valid for the surrounding GM. The volumetric divergence of the electric field or its one-dimensional counterpart, the activating function dE/dz, easily reaches 1-10 V/mm^2 in both the WM gyral crown/lip and in the WM sulcal walls.

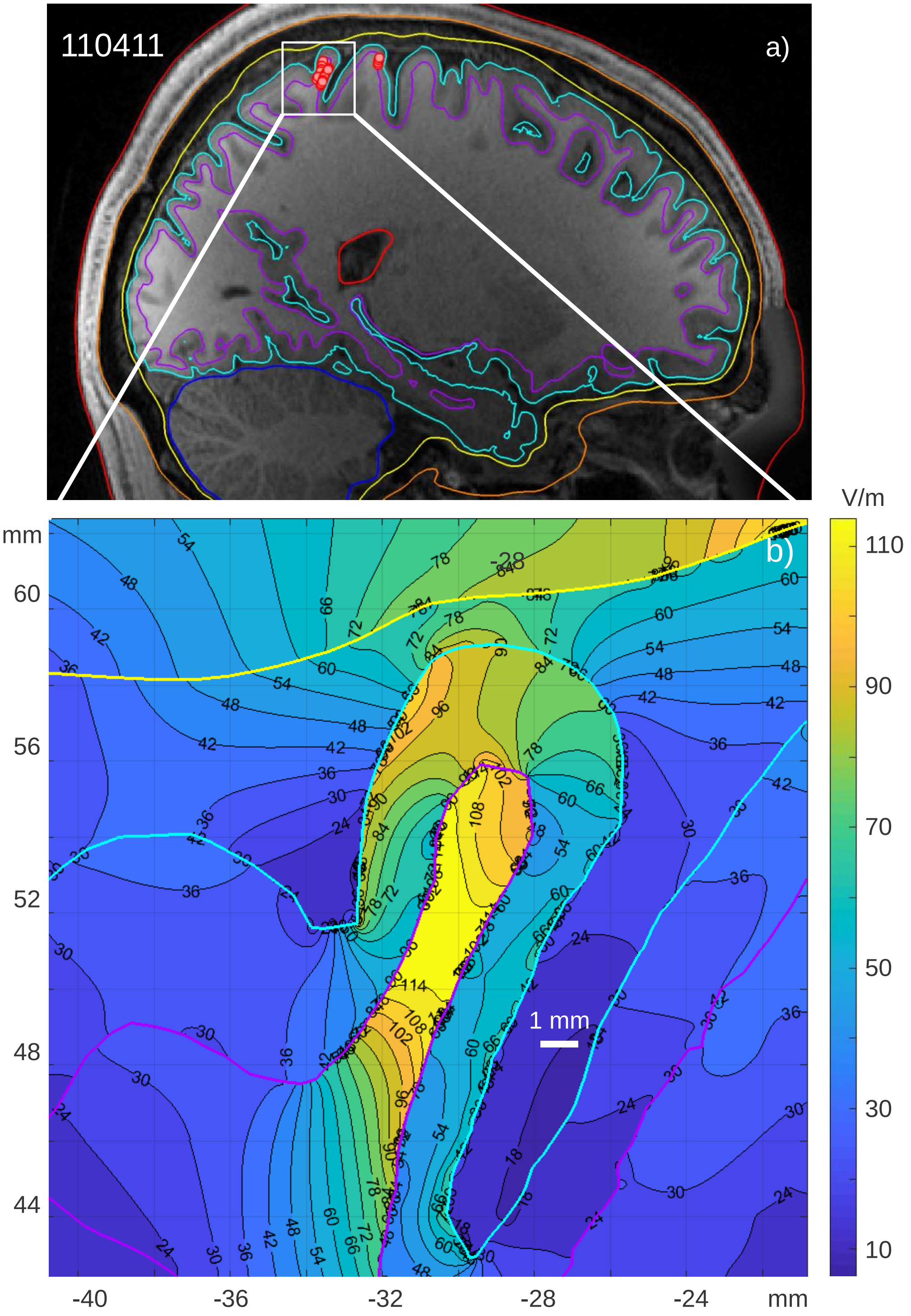


Fig. 33. a) – Sagittal plane passing through the location of the maximum field at the WM-GM interface and superimposed onto the corresponding NIfTI slices when using the MagVenture MRiB91 coil located above the hand knob area of the right precentral gyrus of subject #110411. The red dots indicate the centers of intersected WM facets where the absolute field value is in the range of 80-100% of the maximum field value. b) – volumetric total field (magnitude) distribution within the small white rectangle in Fig. 33a. Blue color – CSF-GM interface, purple color – WM-GM interface, yellow color – skull-CSF interface.

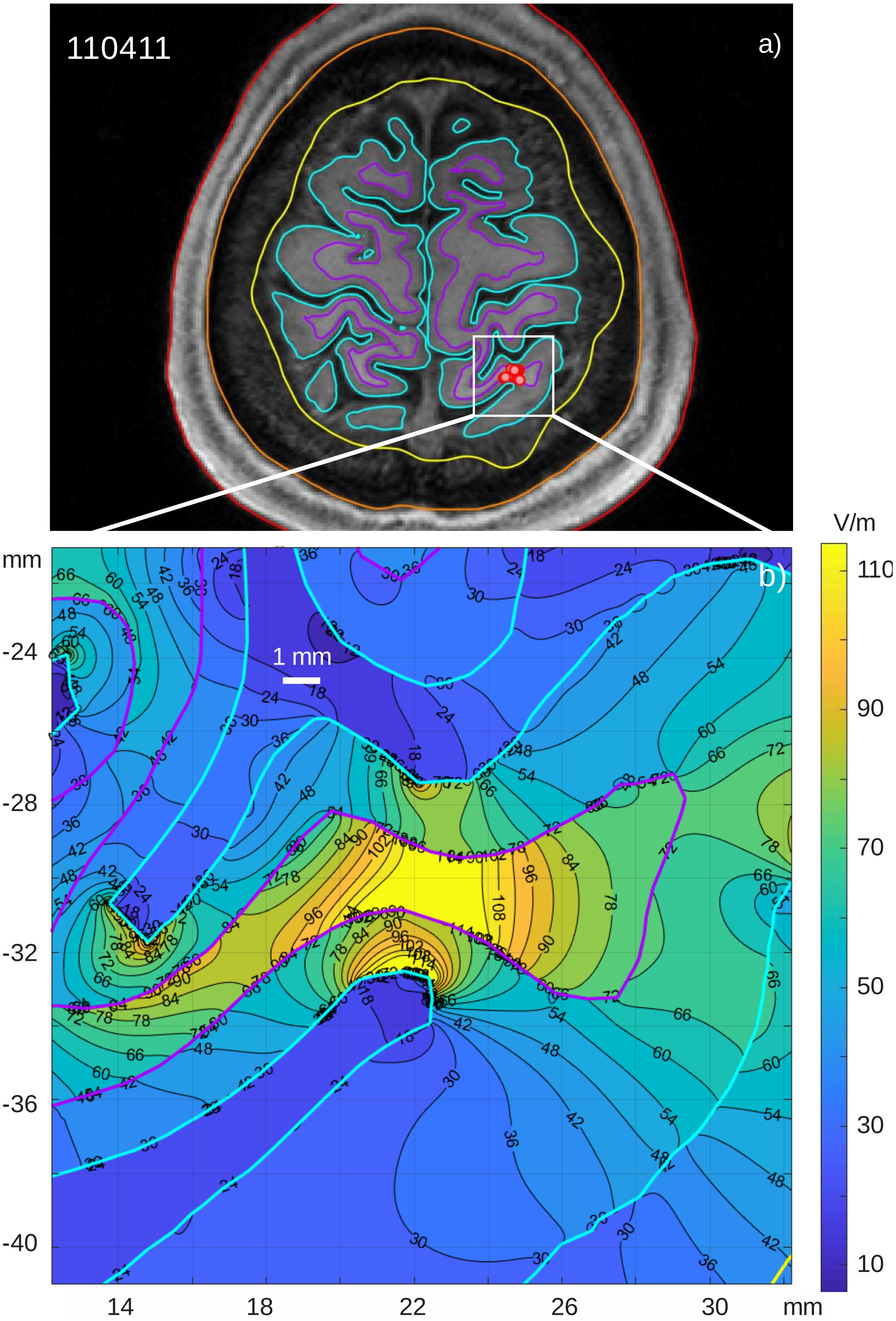


Fig. 34. a) – Transverse plane passing through the location of the maximum field at the WM-GM interface and superimposed onto the corresponding NIfTI slices for the MagVenture MRiB91 coil located above the hand knob area of the right precentral gyrus of subject #110411. The red dots depict the centers of intersected WM facets where the absolute field value is in the range of 80-100% of the maximum field value. b) – volumetric total field (magnitude) distribution within the small white rectangle in Fig. 34a. The same notations from Fig. 33 are used.

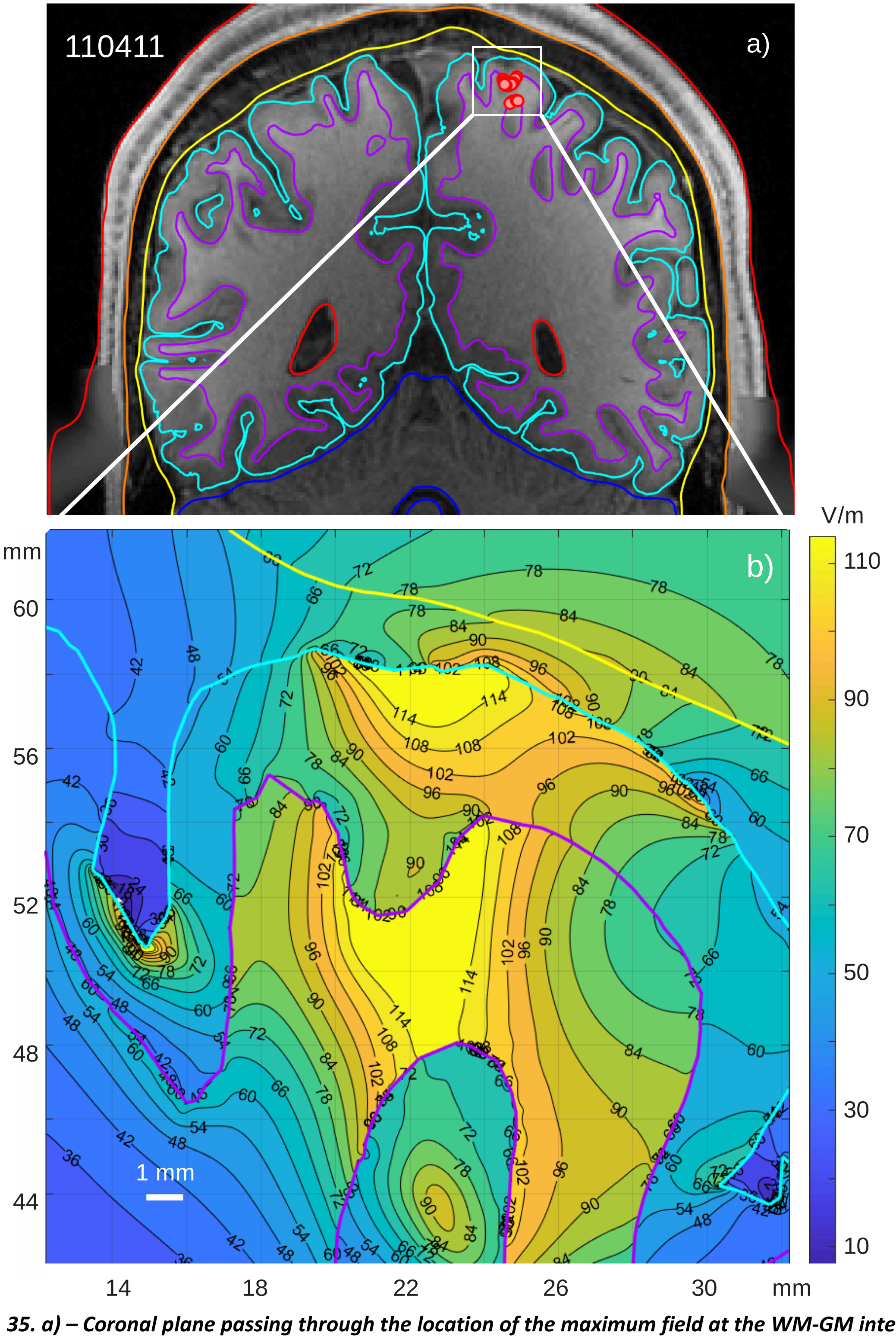


Fig. 35. a) – Coronal plane passing through the location of the maximum field at the WM-GM interface and superimposed onto the corresponding NIfTI slices for the MagVenture MRiB91 coil located above the hand knob area of the right precentral gyrus of subject #110411. The red dots show the centers of intersected WM facets where the absolute field value is in the range of 80-100% of the maximum field value. b) – volumetric total field (magnitude) distribution within the small white rectangle in Fig. 35a. The same notations from Fig. 33 are used.

III.C. Discussion

III.C.1. Comparison with analytical solutions and with other software packages/numerical methods

Three separate computational studies [49]-[51] have been performed to compare the method's performance in application to TMS problems. For the canonic multisphere geometry and an external magnetic-dipole excitation where the analytical solution is available, the BEM-FMM algorithm was tested against a fast open-source getDP solver running within the SimNIBS 2.1.1 environment [50]. It was observed that the BEM-FMM algorithm gives a smaller solution error for all mesh resolutions and runs significantly faster for high-resolution meshes when the number of triangular facets exceeds approximately 0.25 M [50]. The algorithm was further tested for 10 realistic head models of the Population Head Model Repository [79]-[81] excited by a realistic coil geometry. The algorithm's performance was compared against a high-end commercial FEM software package ANSYS Maxwell 3D with adaptive mesh refinement [49]. Excellent agreement was observed for electric field distribution across different intracranial compartments, and the BEM-FMM algorithm achieved a speed improvement of three orders of magnitude over the commercial FEM software [50].

A detailed and rigorous comparison study was recently performed independently by another group [51]. For MRI-derived head models, the method of the present study – the 0th order BEM-FMM – was determined to be the most accurate method that could be run with available computational resources. Other methods (from least to most accurate [51]): FDM or finite difference method, 1st order FEM, SPR (superconvergent patch recovery)-FEM, 2nd order FEM, 1st order BEM, and 3rd order FEM were benchmarked against the present method. It was concluded that, whereas at present the 1st order FEM is most commonly used, the 0th order BEM-FMM appears to be the judicious strategy for achieving negligible numerical error relative to modeling error, while maintaining tractable levels of computation.

III.C.2. Method speed and model size

The algorithm runs best on multicore workstations/servers and multicore PCs due to the inherent parallelization of MATLAB and the FMM package. The number of cores seems to be more important than the clock speed. According to Table 7, the numerical TMS solution for the head segmentation with 0.55 nodes/mm^2 and 0.9 M facets in total executes with the improved BEM-FMM in approximately 34 sec using a 2.1 GHz multicore server. This is threefold improvement compared to the initial formulation ([49]or [50]). The numerical TMS solution with 1.8 M facets in total from Table 7 executes in approximately 74 sec, i.e. scales nearly linearly with the number of facets.

A surface model with 70 M facets has been considered and computed with the present toolkit. For the same 2.1 GHz multicore server listed in Table 7, the corresponding execution time reaches approximately two hours.

However, for a laptop computer, the BEM-FMM algorithm is expected to run significantly slower than the FEM pipeline of SimNIBS 3.0 ([48]) when a comparable resolution model is used. The BEM-FMM algorithm is also, in the present version, unable to handle WM anisotropy.

Also, the BEM-FMM approach requires relatively large preprocessing times (on the order of one minute or longer) for an individualized head model in order to compute and store necessary neighbor electrostatic potential integrals on triangular facets. However, this preprocessing step is required only once for each head model.

III.C.3. Volumetric field resolution

In contrast to FEM, the BEM-FMM numerical resolution in the cortex is not limited by the volumetric tetrahedral mesh size; it may reach a micron scale if necessary. The key difference is that the solution is completely determined by the incident field and the induced charge distribution on the conductivity boundary surfaces, and the E-fields at arbitrary points of the 3D space can be subsequently evaluated. In the present study, we have demonstrated the ability to accurately compute TMS fields within the cortex at submillimeter scales with a field resolution of 40 µm as well as close to and across the cortical interfaces, in particular across the WM-GM interface.

The method accuracy is only limited by the surface segmentation itself and not by the volumetric mesh size. A meaningful solution is obtained at any distance from the interface, including a distance approaching zero. For example, for two interfaces separated by 2 mm, BEM-FMM is expected to generate more accurate results than the first-order FEM of SimNIBS at the distances of 0.5 mm or less from either interface unless many tetrahedra across the 2 mm gap or higher-order FEM basis functions are used. Indeed, the segmentation accuracy itself provides a limit on the overall modeling accuracy. However, detailed segmented models with isotropic resolution of 0.5 mm are already available [71].

Furthermore, it is seen in Figs. 33, 34, and 35 that the field in the cortex may change very rapidly and at submillimeter scales. Accurate high-resolution field modeling may therefore be important for subsequent multiscale modeling pertinent to evaluating the neuron activating function [99]-[101].

III.C.4. Interfacial field resolution

The electric field is discontinuous at the interfaces of brain compartments with different conductivities, due to surface charge accumulation and the abrupt jump of the normal field component across a single monopolar layer of charges. In the present study, we distinguish the normal field just inside the interface, the normal field just outside the interface, and the normal field discontinuity across the interface. The BEM-FMM approach (and any BEM method) accurately accounts for the interface field discontinuities. The ability of BEM-FMM to accurately simulate field changes across the GM/WM interface might allow for better estimations of the potential changes along axons passing this interface [88], [89].

III.C.5. Biophysical limitations

The present BEM-FMM algorithm rests on the assumption of isotropic conductivities inside the tissue compartments, which is not biologically accurate. Simulations suggest that taking WM anisotropy into account may change the TMS field by up to 20% in WM regions close to the brain surface. While GM anisotropy, as assessed with diffusion tensor imaging (DTI), is far less on a macroscopic scale, it might well matter when considering the microscopic scale of 40 μm presented in section 3.5. It is likely that the different cortical layers differ in their conductivity. It is also likely that the conductivity is anisotropic in some of the layers, in particular the lower layers. While the present BEM-FMM algorithm is numerically accurate at a microscopic scale, this should thus not be confused with "closer to the ground truth", as it is based on a number of simplifying assumptions. Further method development may overcome some of them.

III.D. Conclusions

In this chapter, we have described the improved BEM-FMM numerical algorithm for TMS modeling. Compared with previous results, the BEM-FMM algorithm has been improved in several novel ways. First, we have established fast, non-saturated solution convergence by incorporating an explicit global charge conservation law. Second, we utilized a simple analytical approach for obtaining electric fields (and electric field discontinuities) normal to the cortical surface (or any other interfaces) at no extra computational cost and without any postprocessing. Third, we established a minimum sufficient number of iterations for obtaining an accurate solution. Finally, we have incorporated a fully general treatment of boundary interface geometries, allowing non-nested surface models to be used.

The improvements have increased the method speed by a factor of approximately 3, while maintaining the same accuracy, and have provided fast non-saturated convergence to arbitrarily small values of the relative residual. The numerical TMS solution for the head segmentation with 0.55 nodes/mm^2 and 0.9 M facets in total now executes in approximately 34 sec using a 2.1 GHz multicore server.

The algorithm is based on the new general-purpose FMM kernel developed and supported by the group of L. Greengard [66]. The algorithm, coupled with tools that support surface remeshing and registration with corresponding NIfTI data, is implemented entirely in MATLAB and employs a few necessary toolboxes. It runs best on multicore machines due to the inherent parallelization of its platform. The number of cores seems to be more important than the clock speed. The complete computational code for this study, along with supporting documentation (Appendix B), is available online via a GitHub repository [54].

In the present chapter, we have demonstrated the ability to accurately compute TMS fields within the cortex at submillimeter scales as well as close to and across the cortical surfaces, in particular across the WM-GM interface. The method accuracy is only limited by the surface meshing itself and not by the nominal volumetric resolution of the MRI data. A meaningful solution is obtained at any distance from the conductivity boundary, including a distance approaching zero. It was found that it is possible to scan through electric fields normal to interfaces in real time and without postprocessing. The same functionality is true for the tangential and total interface fields once an intermediate integral in Eq. 2 is computed or acquired from the last step of the iterative solution.

The computational method developed may be useful for navigated TMS [87],[102],[103] and robotic TMS systems [104] operating with the guidance of available high-resolution MRI imaging and providing accurate and stable coil position and orientation.

The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health and Human Services.

IV. TMS Toolkit Application Example: Focality Estimates for Total Electric Field on Intracortical Surfaces

This chapter has been excerpted from [105], under review at the time of writing. In this application example, the TMS-focused BEM-FMM modeling toolkit is applied to search for a coil position and orientation that produces the most focal fields at certain sites in the brain.

IV.A. Introduction

Transcranial Magnetic Stimulation (TMS) activation zones are linked to the induced electric field in the cortex. The activation zone can be anywhere where the field exceeds the activation threshold, which does vary over space. Furthermore, different directions of the induced electric field (or electric current) within these domains can excite different intracortical circuits or the same circuits but at different sites [91] [106]-[108].

The domains of field maxima with respect to a certain field component – total, tangential, or normal to the cortical surfaces – are determined by the coil winding geometry, position and orientation in space as well as by the unique gyral pattern of the subject. Several such domains may simultaneously be generated, e.g., a primary domain in the originally targeted motor cortex or M1 and secondary "hot spots" in the somatosensory and/or premotor cortexes. Also, the size of the primary domain and the type of the dominant field component may vary.

All three electric-field components – total, tangential, and normal – separately, are considered in different TMS models of activation mechanisms for different neural populations [88],[89], [97],[101], [109]-[113]. Most important is likely the magnitude of the total TMS electric field along the gyral crown [97],[101],[109] although other studies have observed predominantly sulcal activations [88],[89],[111]-[113]. Detailed neuronal models have elucidated the different neuronal elements and their activation thresholds [112]. From the macroanatomical perspective, the activated region is considered as a comprise of the locations where the field intensity is highest [109].

A number of computational approaches have been developed to solve the TMS "forward problem": determine the corresponding intracranial E-field distributions given the coil position/orientation and the subject-specific anatomically realistic volume conductor geometry via a numerical solution of the quasistatic Maxwell equations using either the finite element method (cf. [48],[51], [114]) or the boundary element method (cf. [1],[49],[115]).

On the other hand, the TMS "inverse problem" or TMS-IP studied in this chapter aims to determine the coil position and orientation that will optimally focus the total TMS induced electric field around a user-specified cortical target point and minimize it otherwise.

In this study, the performance or degree of improvement of a TMS-IP solution will be quantified as compared to the well-known sulcus-aligned mapping of the $\mathrm{M1_{HAND}}$ (the hand knob [84]) area. The sulcus-aligned mapping is a common projection approach for the primary motor cortex or M1 with the 90° local sulcal angle [87],[108],[116], also called $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ [108]. This promising approach allows one to readily probe the within-hand motor somatotopy [107] in

$\mathrm{M1_{HAND}}$ with neuronavigated TMS that follows the sulcal shape and creates a tissue field/current perpendicular to the central sulcus at all mapping sites [108]. Furthermore, using sulcus-shape based mapping, novel evidence has been provided that fast sensorimotor integration in $\mathrm{M1_{HAND}}$ displays a center-surround organization, engaging center inhibition and surrounding facilitation [116].

In other words, we aim to answer the following questions:

(i) By how much would the TMS-IP solution help us to increase the focality of the $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ mapping; and

(ii) By how much would it help us to better differentiate between neighbor targets in $\mathrm{M1_{HAND}}$?

For a comprehensive and statistically accurate characterization of the TMS-IP performance, we analyze 16 subject-specific head models, each with six target points in the $\mathrm{M1_{HAND}}$ area as described in [108],[116], employ two different coil types (large MRiB91 and small CoolB35 coils, both of MagVenture), two different intracortical field observation surfaces, and two different segmentation models for the same subject (`mri2mesh` and `headreco`), both implemented in the SimNIBS segmentation pipeline [46].

As a focality metric or the cost function of the TMS-IP solution, we choose an average absolute deviation (AAD) of the field magnitude, $\|\boldsymbol{E}\|$, from the target point with respect to a certain field percentile – chosen as the 80th percentile in this study. The total electric field is evaluated either on the mid-surface between gray and white matter or on a surface shifted toward the white matter boundary, with the separation ratio of 4:1, or on a white matter interface itself. While the mid surface approximately corresponds to the cortical layer 2/3 (L2/3), the second observation surface approximately corresponds to the (bottom of) cortical layer 5 (L5, cf., for example, [101]).

The major stability considerations for the TMS-IP solution to be taken into account are

(i) Inverse-problem solution stability and deviation with respect to geometrical uncertainties in the relative coil-head position and with respect to all three coil coordinates and three orientation angles;

(ii) Inverse-problem solution stability and deviation caused by variations in tissue properties;

(iii) Inverse-problem solution stability and deviation with respect to the segmentation accuracy.

The numerical implementation of the TMS-IP solution uses the boundary element fast multipole method (BEM-FMM) [1],[49] as the forward solver and a simple gradient descent search method with a small step size in a full six-dimensional search space as the inverse solver. This search space includes three independent coil co-ordinates and three independent coil rotation angles as suggested in the recent TMS protocol [110]. The expected outcome is the best position and orientation of the coil for the given subject head model and for the given target point.

The gradient search method is tested and confirmed against the most reliable exhaustive approach applied previously in a comprehensive and detailed relevant study [109]. The method can be applied for studying multiple geometrical model variations and arbitrary field observation surfaces, arbitrary coil positions different from those strictly tangential to the scalp, and it does not require extra mesh conditioning.

Our main result shows that the TMS-IP solution may further improve the focality of the sulcus-aligned motor mapping approach for the total field in the gyral crown. This relative improvement is stable, consistent with an average value of 25%, and is weakly dependent on the coil type and the observation surface type. However, it is moderate. When the coil position/orientation and conductivity uncertainties are included into consideration, the focality improvement compared to the sulcus-aligned mapping is further reduced to 20% on average.

In addition, we demonstrate that the TMS-IP solution simultaneously improves the "somatotopic resolution" of TMS for the $\mathrm{M1_{HAND}}$ area of the precentral gyrus (the hand knob) which could in principle be "scanned" with a relatively high resolution when moving the focal hot spot for the likely most important magnitude of the TMS total field [97],[100],[109] along the gyral crown.

One significant problem that we encounter is related to the model segmentation uncertainty, which may considerably affect the TMS-IP solution at M1 and diminish its value. This is because different segmentation models lead to considerably different electric-field distributions [117].

IV.B. Materials and Methods

The major parameters of the present inverse-problem solution are initially summarized in Table 9 and then described in detail below in this section. 16×6×2×2×2=768 individual inverse problem solutions have been analyzed following the different parameter permutations from Table 9.

Table 9. Subjects, models, and methods of the TMS inverse-problem solution.

Type	Number	Short description
Subjects	**16**	Connectome [78] subjects 101309, 110411, 117122, 120111, 122317, 122620, 124422, 128632, 130013, 131722, 138534, 149337, 149539, 151627, 160123, 198451
Brain segmentation models (surfaces)	**2**	`mri2mesh` (based on FreeSurfer and FSL [118]) or `headreco` (based on SPM/CAT [119]), both from SimNIBS.
Coil models	**2**	Small Cool B35 coil (Fig. 1a) and a relatively large MRiB91 coil (Fig. 1b), both of MagVenture (Fig. 1).
Target domains	**1**(l+r)	$\mathrm{M1_{HAND}}$ of the left hemisphere, $\mathrm{M1_{HAND}}$ of the right hemisphere.
Target points per domain	**3+3**(l+r)	Three target points separated by approx.10 mm on gray matter interface within each $\mathrm{M1_{HAND}}$ domain (Fig. 2).

Initial guess	**1**	Sulcus-aligned mapping: 90^{o}_{flex}[108], [116] as an initial guess
Field observation domains	**2**	a. Mid-surface between gray/white matter (1:1, ~Layer 2/3); b. Surface shifted toward white matter (4:1, ~Layer 5)
Field type	**1**	Total field magnitude
Cost function (focality metric)	**1**	Average absolute deviation (AAD) of the field from target – Eq. (1) below
Somatotopy improvement metric	**1**	Average relative difference (ARD) in the field between neighboring target points – Eq. (2) below
Search method		Gradient descent in the $\mathfrak{R}^6$ search space (coil pos.+ orient.)
Forward solver	**1**	BEM-FMM **Error! Reference source not found.**,[1]
Uncertainty sources under study	**2**	Coil position/orientation uncertainties, conductivity and segmentation model uncertainties

IV.B.1. Subject MRI data

MRI T1/T2 data for sixteen Human Connectome Project (HCP) healthy subjects ([78], head only) with an initial isotropic voxel resolution of 0.7 mm have been selected [50]. The subjects' IDs are listed in Table 9 above. Those subjects were selected for their sufficiently good skull quality, which was observed visually after segmenting the models via the `mri2mesh` option of SimNIBS [50].

IV.B.2. Segmentation of MRI data

Two of the automated segmentation routines available in SimNIBS 3.1 software have been used: `mri2mesh` (based on FreeSurfer and FSL [118]) and `headreco` (based on SPM/CAT [119]), both with default options. By using the two segmentation routines, we aim to evaluate the effect of intra-segmentation variability on the computed TMS focality. It is known from the literature that segmentation results appear somewhat different in both cases [120].

These segmentation routines create surface triangular meshes for the six main brain compartments under study: skin, skull, cerebrospinal fluid (CSF), gray matter, white matter, and ventricles. The cerebellum was combined with the white matter in both cases. Every subject dataset was thus segmented twice, which resulted in 32 computational models in total. The typical model size is approximately 1 M (million) triangular facets for both segmentation pipelines.

Table 10 quantifies the resulting average distance d between CSF shells, gray matter shells, and white matter shells for both segmentation routines used in the study. This distance was determined as the mean of the shortest distances from every triangle centroid of one shell to all triangle centroids of the other shell, whose triangular mesh was previously refined with the ratio of 36:1 to achieve good accuracy. These results are only valid for the upper part of the cerebral cortex, strictly above the corpus callosum.

Table 10. Average distance d (in millimeters) between the CSF shells, the gray matter (GM) shells, and the white matter (WM) shells for both segmentation routines used in this study. Subject number (first row of the table) corresponds to the Connectome list from Table 9.

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_{CSF}	0.9	1.5	0.8	0.7	0.8	0.7	0.7	0.9	0.7	1.0	0.7	0.8	1.1	0.8	0.7	1.2
d_{GM}	0.4	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.3	0.4	0.3	0.4	0.3	0.4	0.3	0.4
d_{WM}	0.3	0.3	0.2	0.3	0.2	0.2	0.2	0.3	0.2	0.4	0.2	0.3	0.2	0.2	0.3	0.3

IV.B.3. Selection of TMS coils

A small Cool B35 TMS coil (Fig. 36, top) and a relatively large elliptical MRiB91 TMS coil (Fig. 36, bottom), both of MagVenture, Inc., were converted to CAD models according to the manufacturer's datasheets. By using two different coils with the size ratio of approximately two to one, we aim to evaluate the effect of the absolute coil size on the computed TMS focality. The coils are approximated by ~100,000-120,000 elementary current dipoles, and the coil fields are computed via the FMM [50].

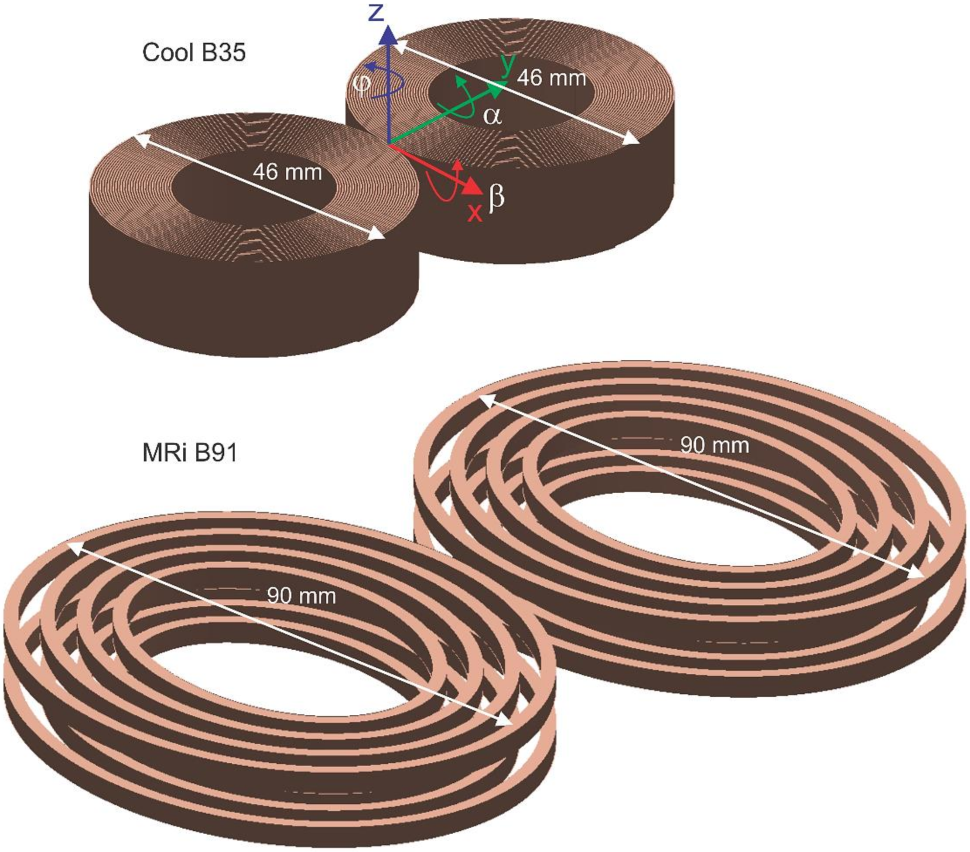


Fig. 36. Two coil models (Cool B35 and MRiB91) used in this study, presented to scale. The coil dimensions in the direction of the coil handle (induced electric field) differ by approximately a factor of 2. Principal rotation angles of the coil – pitch α, roll β, and yaw φ– are shown in the top figure.

IV.B.4. Selection of TMS target domains and target points

For every subject, six TMS target points, $\boldsymbol{T}$, within the $\mathrm{M1_{HAND}}$ areas on the crown of the precentral gyrus have been identified on the gray matter interface in global MRI co-ordinates – three points per every hemisphere. These points are illustrated in Fig. 37 for subject 120111 by blue spheres. The $\mathrm{M1_{HAND}}$ areas were defined with the help of a neuroanatomist. Exactly the same target points (16×6=96 in total) have been used for both segmentation types (`mri2mesh` and `headreco`). Following the relevant experimental setup in [108],[116], the average linear distance between the target points was maintained at 10 mm, with the standard deviation of less than 1.2 mm. The center of a target cloud approximately coincides with the center of the respective $\mathrm{M1_{HAND}}$ area found using its posterior convexity as central reference [84].

Simultaneously, Fig. 37 shows the directions of the coil axis (black lines) and the directions of the induced electric field (the coil handle, white lines) which will be used in the sulcus-aligned motor mapping – the initial guess of an optimization solution.

IV.B.5. Selection of initial coil positions: sulcus-aligned coil mapping

The initial coil position is based on a direct projection onto the cortical surface with the field direction being perpendicular to the nearest sulcal wall. It is constructed following three steps:

iv. make the coil centerline pass through a given target point $\boldsymbol{T}$ defined in global MRI coordinates (on the gray matter interface);

v. make the coil centerline perpendicular to the skin surface at the skin-centerline intersection, and position the coil bottom at a distance of 10±0.25 mm from the skin surface along this centerline;

vi. make the dominant coil field direction (coil handle) approximately perpendicular to the sulcal wall (of the central sulcus for $\mathrm{M1_{HAND}}$) nearest to the target point, which is the sulcus-aligned approach [108],[116],[87].

The examples of the initial coil positioning for subject 120111 are shown in Fig. 37.

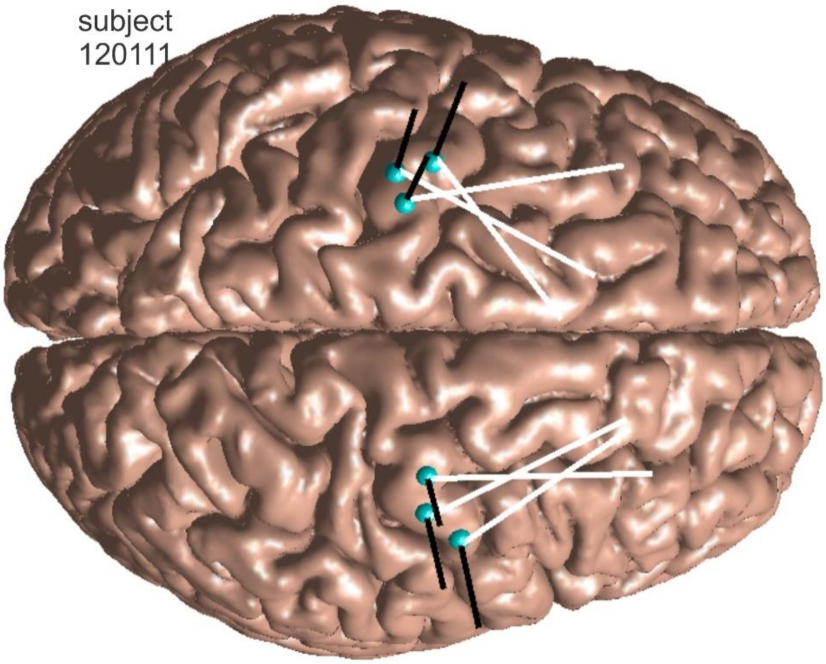


Fig. 37. Six TMS target points T within left and right $M1_{HAND}$ areas of subject 120111 (superimposed onto `mri2mesh` gray matter segmentation) as indicated by blues spheres. Directions of the coil axis (black lines) and the directions of the induced electric field (coil handle, white lines) are shown used in the sulcus-aligned motor mapping.

IV.B.6. Selection of observation domains

To be consistent with the previous relevant studies [109],[82] and with the biophysical activation mechanisms, the first E-field observation domain was chosen as a mid-surface between gray and white matter. The total electric field (field magnitude, $\|\boldsymbol{E}\|$) was sampled on this surface. Another observation surface was also chosen, shifted closer toward the white matter boundary, with the separation ratio of 4:1. While the mid surface approximately corresponds to the cortical layer 2/3 (L2/3), the second observation surface approximately corresponds to the bottom of cortical layer 5 (L5, cf., for example, [101]).

The two observation surfaces were created by finding the shortest vector distances from every triangle centroid of the gray matter shell to all triangle centroids of the white matter shell, whose triangular mesh was previously refined with the ratio of 36:1 to achieve sufficient accuracy. Then, the nodes of the observation surfaces were defined as the centroids of gray matter facets moved along the established vector distances, with the appropriate relative separation from both boundaries (1:1 or 4:1).

IV.B.7. Cost function (focality metric) of the inverse problem

Consider any target point $\boldsymbol{T}$ on the gray matter interface of a subject's head model shown, for example, in Fig. 37. The goal of the present TMS-IP is to vary the TMS coil's position and orientation in order to maximize the field on the observation surface close to the target point $\boldsymbol{T}$

and minimize it everywhere else. In other words, the best coil focality is sought in the vicinity of the target point. Other formulations of the inverse problem are indeed possible.

The search space is a six-dimensional space (denoted by $\mathfrak{R}^6$) that includes the three Cartesian coordinates of the coil center and the three coil rotation angles (pitch-roll-yaw) shown in Fig. 36. The only physical constraint imposed on the search space requires that the *nearest* distance from any part of the coil metal conductor (without the plastic case) to the skin of the subject under study is no less than 10 mm. This constraint is enforced by monitoring the minimum distance between the scalp and the coil windings, for every coil position and orientation.

The cost function to be minimized is defined in this study as the mean or average absolute deviation (AAD) of the field maxima from the target point $\boldsymbol{T}$ in 3D, that is

$$AAD = mean(\|\boldsymbol{T} - \boldsymbol{T}_i\|) \ \text{[mm]}, \ \ i = 1, \dots, N \tag{27}$$

with $\|\cdot\|$ being the Euclidian norm, and $\boldsymbol{T}_i$ being N nodes of the observation surface where the absolute field value $\|\boldsymbol{E}\|$ of interest is in the range of 80-100% of the maximum field value. The measure of Eq. 27 does not square the distance from the target, so it is less affected by a few extreme observations than the variance and standard deviation. The threshold value of the 80th percentile used in this study is a soft estimate; it may be altered if necessary.

IV.B.8. Target differentiation or "somatotopy" metric

In addition to the absolute focality, it is useful to know how different the field would become at the immediate neighbor target points after the AAD has been minimized for one of them. Note that in the present case and in the background experimental studies [108],[109],[116] the neighboring target points are separated by 10 mm (7-10 mm in [109]). This "somatotopy" metric may be defined as an average ratio (ARD) of the electric-field values $\boldsymbol{E_T}$ and $\boldsymbol{E_{T_n}}$ between the selected target point $\boldsymbol{T}$ and its immediate neighboring target points $\boldsymbol{T}_n$, respectively. One has

$$ARD = 1 - \ mean\left(\frac{\|\boldsymbol{E_{T_n}}\|}{\|\boldsymbol{E_T}\|}\right) \ [a.u.], \ \ n = 1{,}2 \tag{28}$$

In the present case, the number of immediate neighbors, n, is equal to two. Eq. (28) predicts a zero ARD when the fields at the neighbor targets are equal to the field at the selected target. On the other hand, it predicts the ARD of one (or 100%) when the fields at the neighbor targets are vanishingly small compared to the field at the target in question (maximum differentiation).

IV.B.9. Search method; forward solver

A simple (perhaps the simplest) gradient descent search method with a sequential variable-by-variable update and a relatively small step size was tested. For eight test cases considered (target points #1 and #2 were tested for two subjects, for two coil types, and two observation surfaces), the gradient descent search method generated nearly the same results as the straight exhaustive search method. In the one situation where this was not the case, the exhaustive-search solution appeared *unstable* and was forced to revert to the previous stable solution.

Therefore, the gradient descent method in the six-dimensional space of three coil coordinates and three rotation angles shown in Fig. 36 was chosen in this study, along with a relatively small and fixed step size. The constraint on the minimum coil-scalp distance of 10 mm was additionally imposed. The method is based on variations

$$x = x_0 \pm \Delta x, y = y_0 \pm \Delta y, z = z_0 \pm \Delta z, \alpha = \alpha_0 \pm \Delta\alpha, \beta = \beta_0 \pm \Delta\,\beta\,, \varphi = \varphi_0 \pm \Delta\varphi \qquad (29)$$

executed and processed sequentially. First, the coil azimuthal angle $\varphi = \varphi_0 \pm \Delta\varphi$ in Fig. 36 is varied about its initial state φ_0, and the angle providing the minimum of the cost function is selected. Next, angles $\alpha = \alpha_0 \pm \Delta\alpha$ and $\beta = \beta_0 \pm \Delta\,\beta$, which characterize the direction of the coil axis, are varied similarly. Finally, we vary the three spatial coordinates of the coil: $x = x_0 \pm \Delta x, y = y_0 \pm \Delta y, z = z_0 \pm \Delta z$ sequentially and then update the cost function (and coil position) at every individual iteration step.

The iterative process repeats until the solution saturates (i.e. no longer changes or converges). Given sufficiently small step values, no step refinement was required. We found that the following values: $\Delta x = \Delta y = \Delta z = 2$ mm and $\Delta\varphi = \Delta\alpha = \Delta\beta = 0.1\,rad = 5.7°$ provide convergence in 4-10 iterations.

For the forward-problem solution, we use the boundary element fast multipole method formulated in terms of induced surface charge density $\rho(\boldsymbol{r})$ residing at the conductivity interfaces or BEM-FMM [1],[49],[54]. The method possesses high numerical accuracy, which was shown to exceed that of the comparable finite element method of the first order [51]. Most recent efforts made it possible to obtain the forward-problem solution in approximately 12-14 sec for a head model with 0.9 million facets and with 7 brain compartments using a 2.8 GHz server. The RAM requirements (6-12 Gbytes) of the forward solver are moderate; the number of cores is more critical: 16 or more cores are preferred. Without parallelization of the search algorithm, a single-target optimization run (the TMS-IP solution) requires approximately 3-5 min.

IV.B.10. Testing numerical accuracy of the inverse-problem solution

While the BEM-FMM solver itself was tested previously [1],[49],[51], the inverse-problem solution was not. To do so, we have rerun

(i) 96 inverse problems with a different number of iterative passes (12 instead of 6);

(ii) 8 inverse problems (target points #1 and #2 for two subjects, for two coil types, and two observation surfaces) with the straight exhaustive search method;

(iii) 8 inverse problems with a different mesh resolution (using a 4× barycentric mesh refinement).

In every case, the differences in the mean AAD values reported below in Table 11 did not exceed 1-5%, which was considered satisfactory. In one case (ii), a better but unstable solution was found.

IV.B.11. Solution stability vs geometrical uncertainties in coil position; stability correction

The stability of the inverse problem solution against uncertainties in coil position and orientation, as well as model segmentation defects is critical. The geometrical stability has been checked for every single TMS-IP solution by introducing a relative average de-focalization, DF, in the following way. Assume that in Eq. 29, index 0 now denotes the final solution with the cost function AAD_0. Its variations corresponding to $\Delta x = \Delta y = \Delta z = 1.5$ mm and $\Delta\varphi = \Delta\alpha = \Delta\beta = 0.1\, rad = 5.7°$ are tested. These dimensions approximately correspond to the accuracy of a modern robotized TMS navigator or cobot [121] (controlled 2 mm linear and 4 deg angular coil placement accuracy). These solution variations give us 12 more AAD values. The relative average defocalization DF_{coil} (always greater or equal than one) is found in the form:

$$DF_{coil} = \underset{n=0:12}{mean}(AAD_n)/AAD_0 \geq 1 \tag{30}$$

When DF exceeded 1.25 at the last iteration, the inverse-problem solution was classified as unstable. This has been found in approximately only 1-2% of all considered cases. The iterative solution was then reverted to the second-to-last iteration and the stability check was performed again, etc., until the inequality DF<1.25 is met. We were able to accomplish this task in all cases.

IV.B.12. TMS-IP solution result; measure of success

After the TMS-IP solution is obtained, the cost function for the initial coil position, $AAD_{initial}$, is compared with the final cost function, AAD_{final}. The dimensionless relative difference of these (greater than one) or the absolute difference (in mm) is the field focality improvement. If the final ratio is substantially greater than one, the usefulness of the TMS-IP solution will be proven.

This relative difference should further be adjusted to include the effect of defocalization, DF, from Eq. 30 or elsewhere to obtain a more realistic estimate. The corrected relative focality improvement to within the main order of magnitude can then be formulated in the form

$$Focality\ Improvement = \frac{AAD_{initial} - DF_1 \times DF_2 \ldots \times AAD_{final}}{AAD_{initial}} \tag{31}$$

where $DF_{1,2}$, ... are the individual focality corrections (comprising the average defocalizations of AAD_{final}) due to the different uncertainty mechanisms. In this study, the focality corrections due to position/orientation inaccuracy and inaccuracy in tissue conductivities were explicitly included.

IV.C. Major Results

IV.C.1. Typical optimization results for the mid-surface

Fig. 38 shows a typical optimization result b) and d) as compared to the original sulcus-aligned mapping a) and c) for subject 120111 and for the large MRiB91 coil. The total field at the mid-surface is projected on the gray matter interface and then plotted, both in binary and continuous form. The AAD value (which may perhaps be treated as an "effective radius" of the suprathreshold area) moderately decreases from 9.4 mm to 6.7 mm. This is at the expense of increasing the subthreshold field spread, mostly in the premotor area. At the same time, the initially observed second suprathreshold maximum in the somatosensory area presumably becomes subthreshold.

Fig. 39 shows a typical optimization result b) and d) as compared to the original sulcus-aligned mapping a) and c) for the same subject 120111 but for the small CoolB35 coil and a different target point. The total field at the mid-surface is projected on the gray matter interface and then plotted, both in binary and continuous form. The AAD value (the "effective radius" of the suprathreshold area) now decreases more significantly as compared to the previous case, from 10.2 mm to 5.2 m. This is again at the expense of some increasing the subthreshold field spread, in the premotor area and in the $\mathrm{M1}_{\mathrm{HAND}}$ itself. At the same time, the initially observed second suprathreshold maximum in the somatosensory area again becomes subthreshold.

IV.C.2. Summary of focality improvement for the mid-surface

Table 11 summarizes the average absolute deviation, AAD, at the mid-surface before and after the inverse-problem solution, along with the defocalization, DF_{coil}, estimates.

Table 11. Focality and its improvement, and the defocalization due to coil position/orientation uncertainty for the mid-surface. Every number is an averaged value for 96 targets points (6 points per subject for 16 subjects). The standard-deviation (STD) values are given using a small font.

Segmentation type	mri2mesh		headreco	
Coil type	MRiB91	CoolB35	MRiB91	CoolB35
AAD for sulcus-aligned mapping, mm	**9.99** 2.8	**7.94** 2.4	**8.81** 2.5	**7.12** 2.1
AAD optimized via TMs-IP, mm	**7.67** 2.2	**5.86** 1.5	**6.27** 2.3	**5.25** 1.7
Defocalization, DF_{coil}, a.u.	**1.02**	**1.03**	**1.02**	**1.02**

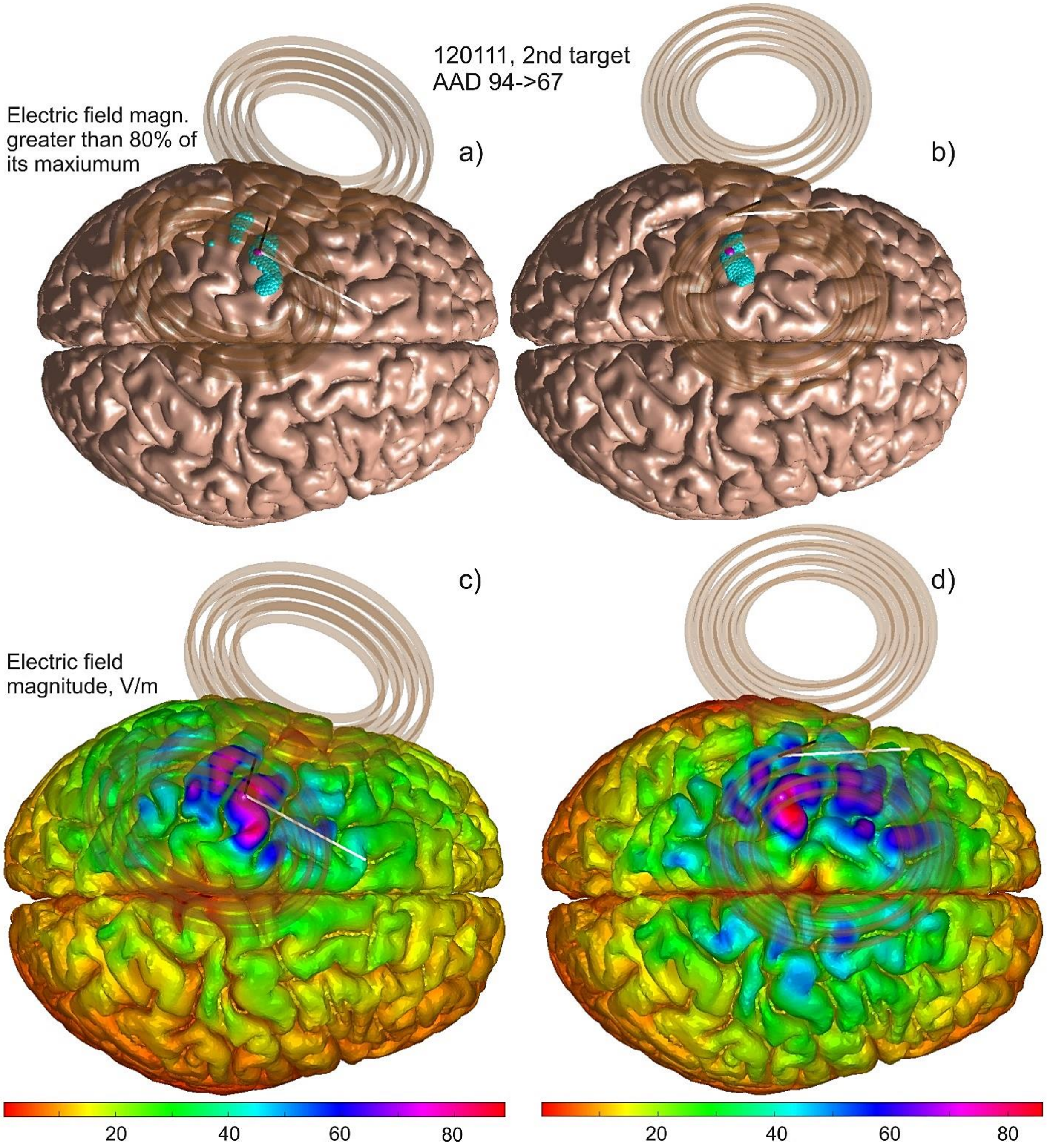


Fig. 38. MRiB91 (large coil) of MagVenture targets the left motor hand area of Connectome subject 120111; TMS pulse strength is 9.4e7 A/s. Target point T_2 ($M1_{HAND}$) on the gray matter surface is shown by a small magenta sphere. a) and c) – 80th percentile and continuous electric field magnitude, respectively, for the sulcus-aligned mapping; b) and d) – the same result after the AAD optimization. For plotting, the total electric field at the mid-surface is projected on the gray matter surface and then plotted.

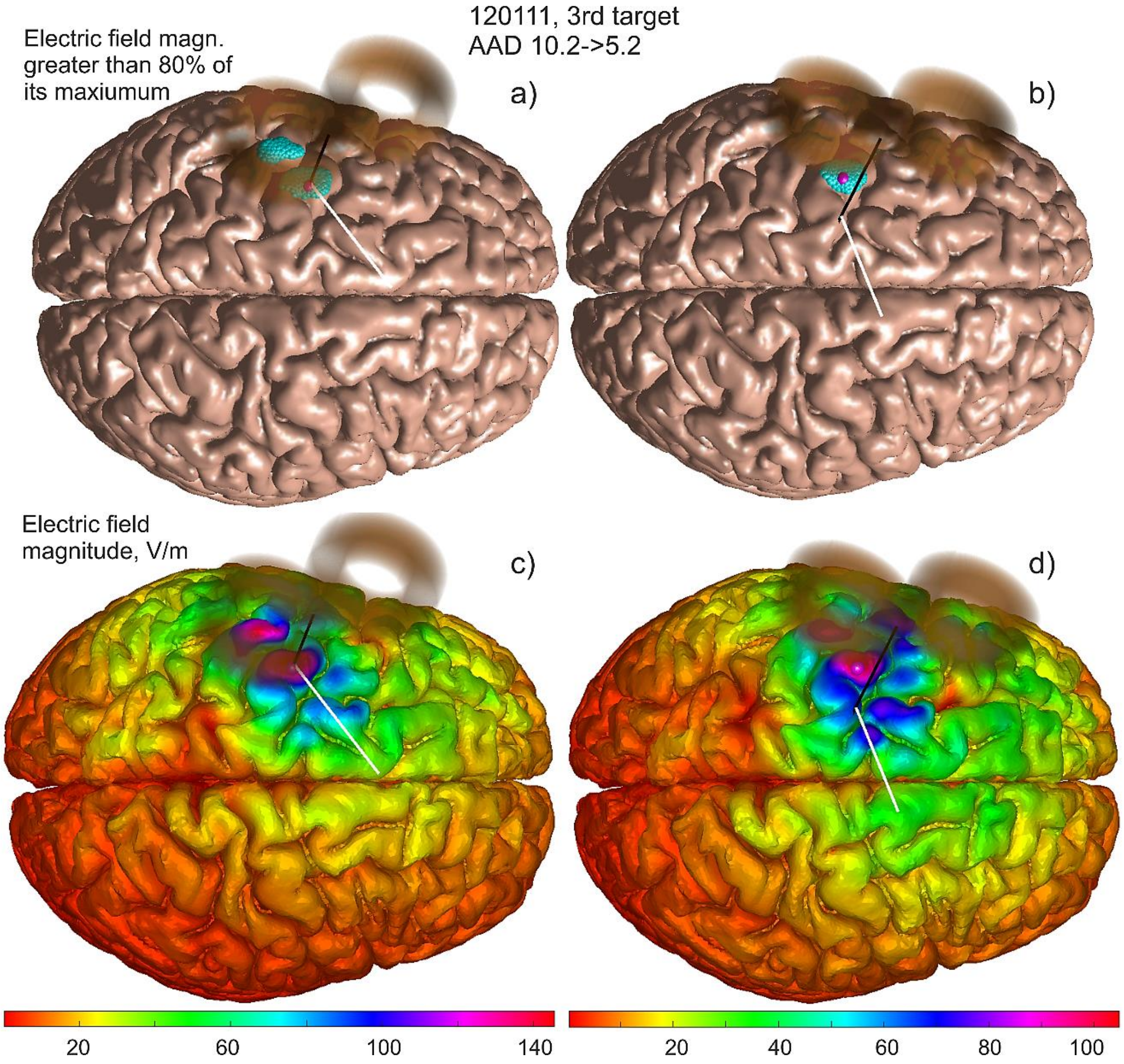

120111, 3rd target
AAD 10.2->5.2
Electric field magn. greater than 80% of its maxiumum
a)
b)
Electric field magnitude, V/m
c)
d)
20
60
100
140
20
40
60
80
100

Fig. 39. CoolB35 (small coil) of MagVenture targets the left motor hand area of Connectome subject 120111; TMS pulse strength is 9.4e7 A/s. Target point T_3 ($M1_{HAND}$) on the gray matter surface is shown by a small magenta sphere. a) and c) – 80th Percentile and continuous electric field magnitude, respectively, for the sulcus-aligned mapping; b) and d) – the same result after the AAD optimization. For plotting, the total electric field at the mid-surface is projected on the gray matter interface and then plotted in the binary and continuous form.

Every number in Table 11 is an averaged value for 96 targets points. The corresponding standard-deviation values are given using a small font. The corresponding target-by target individual raw data are given in Fig. 40.

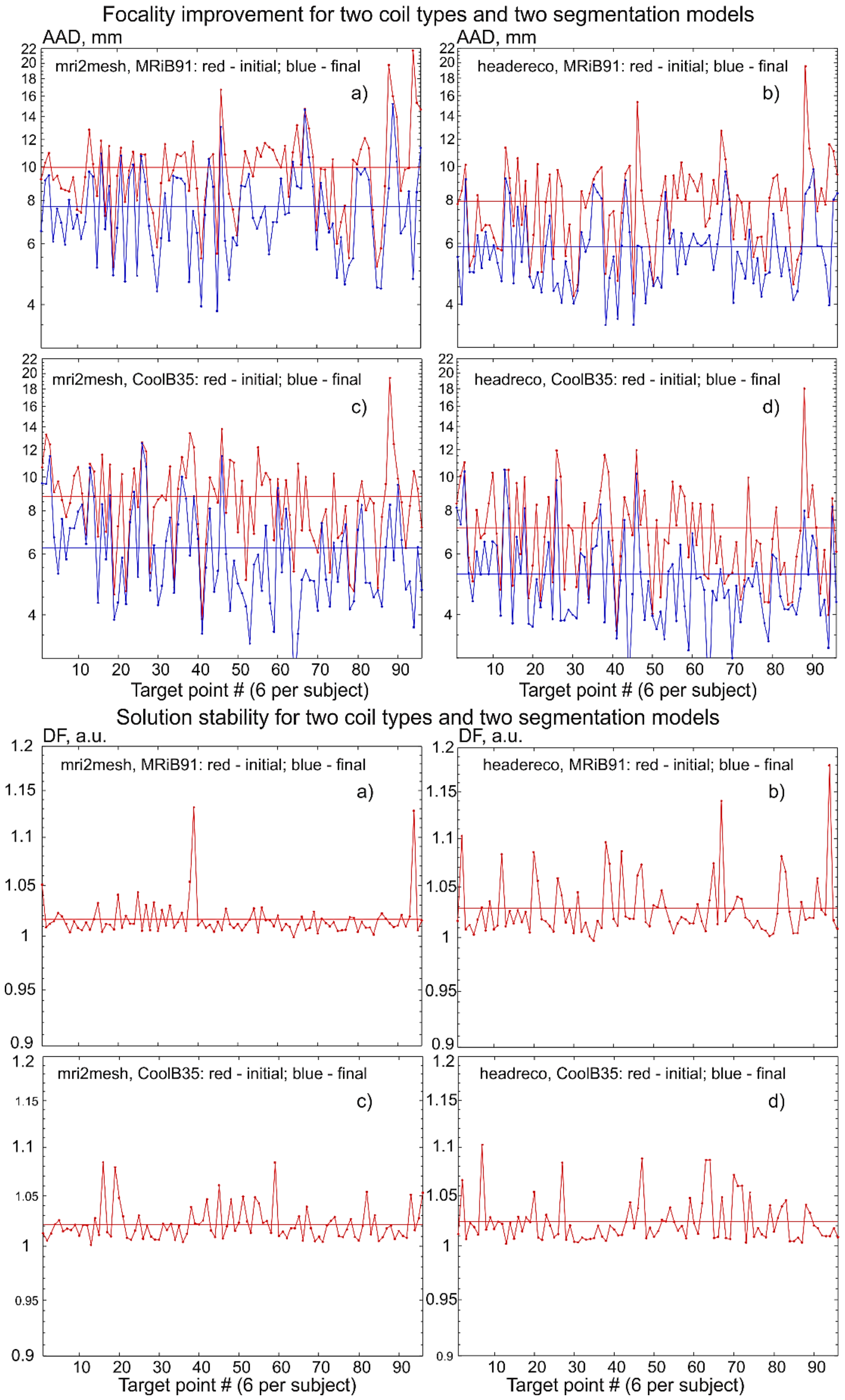


Fig. 40. Target-by-target AAD and stability results for the mid-surface. The target points are numbered sequentially. Average data reported in Table 11 are shown by straight lines.

IV.C.3. Degree of "somatotopic" resolution improvement for the mid-surface

Table 12 summarizes the average relative difference or the somatotopy metric, ARD, at the mid-surface before and after the inverse-problem solution, respectively. The corresponding target-by target individual raw data are given in Fig. 41. We emphasize that inverse-problem solution is not originally meant to improve the somatotopy; its original cost function is AAD.

Table 12. Somatotopy percentage and its improvement for the mid-surface. Every number is an averaged value for 96 targets points (6 points per subject for 16 subjects). STD values are given using a small font. Very similar results have been obtained for the 4:1 observation surface.

Segmentation type	`mri2mesh`		`headreco`	
Coil type	MRiB91	CoolB35	MRiB91	CoolB35
ARD for sulcus-aligned mapping, %	**14.2** 14.4	**21.4** 15.5	**13.5** 20.0	**21.3** 21.1
ARD for inverse-problem solution (TM-IP), %	**19.5** 13.7	**27.3** 14.2	**19.7** 20.4	**27.3** 22.0

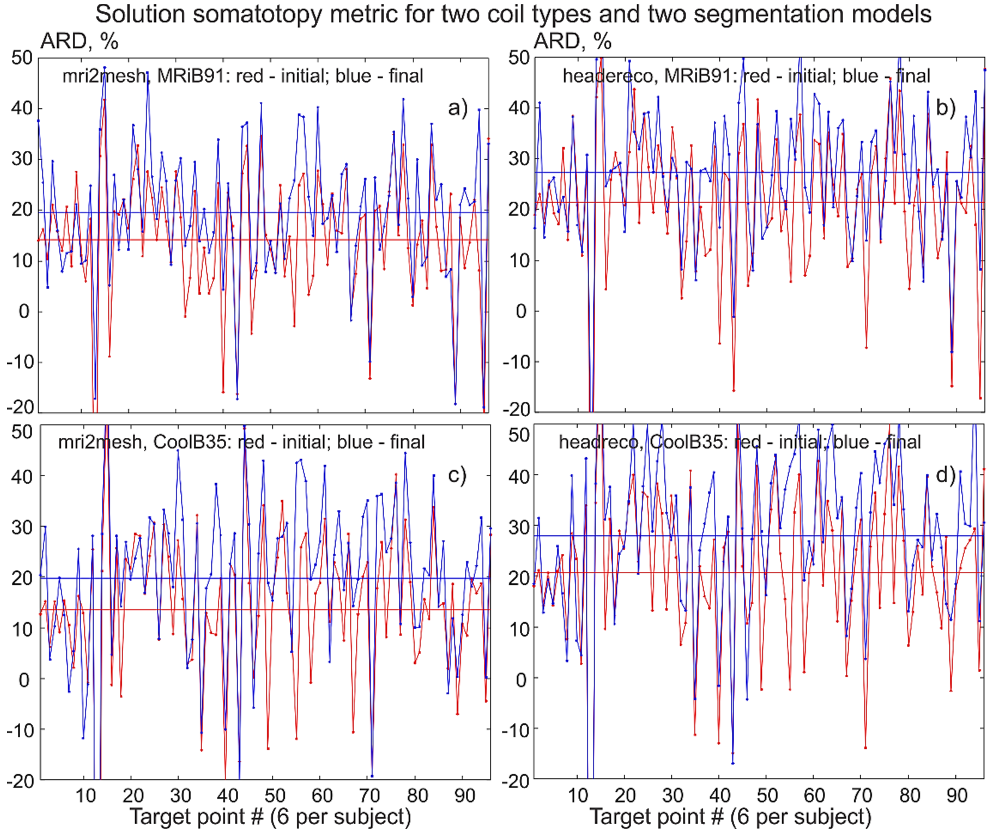


Fig. 41. Target-by-target ARD percentage for the mid-surface. The target points are numbered sequentially. Average data reported in Table 12 are shown by straight lines.

IV.C.4. Summary of field loss for the mid-surface

The inverse-problem solution is characterized by an overall field intensity decrease or field loss at the target point. Table 13 summarizes the average ratio of the maximum total field magnitudes at the mid-surface before and after the inverse-problem solution, respectively.

$$Field\ Loss = 100\% \times \left(mean\left(\frac{max(\|\boldsymbol{E}_{\text{sulcus aligned}}\|)}{max(\|\boldsymbol{E}_{\text{inverse problem}}\|)} \right) - 1 \right) \tag{32}$$

The corresponding target-by target data are given in Fig. 42.

Table 13. Field loss for the mid-surface. Every number is an averaged value for 96 target points (6 points per subject for 16 subjects). The standard-deviation values are given using a small font. Very similar results have been obtained for the 4:1 observation surface.

Segmentation type	mri2mes		headreco	
Coil type	MRiB91	CoolB35	MRiB91	CoolB35
Field loss, %	**5.4**/6.9	**6.2**/8.7	**5.2**/8.5	**6.6**/9.4

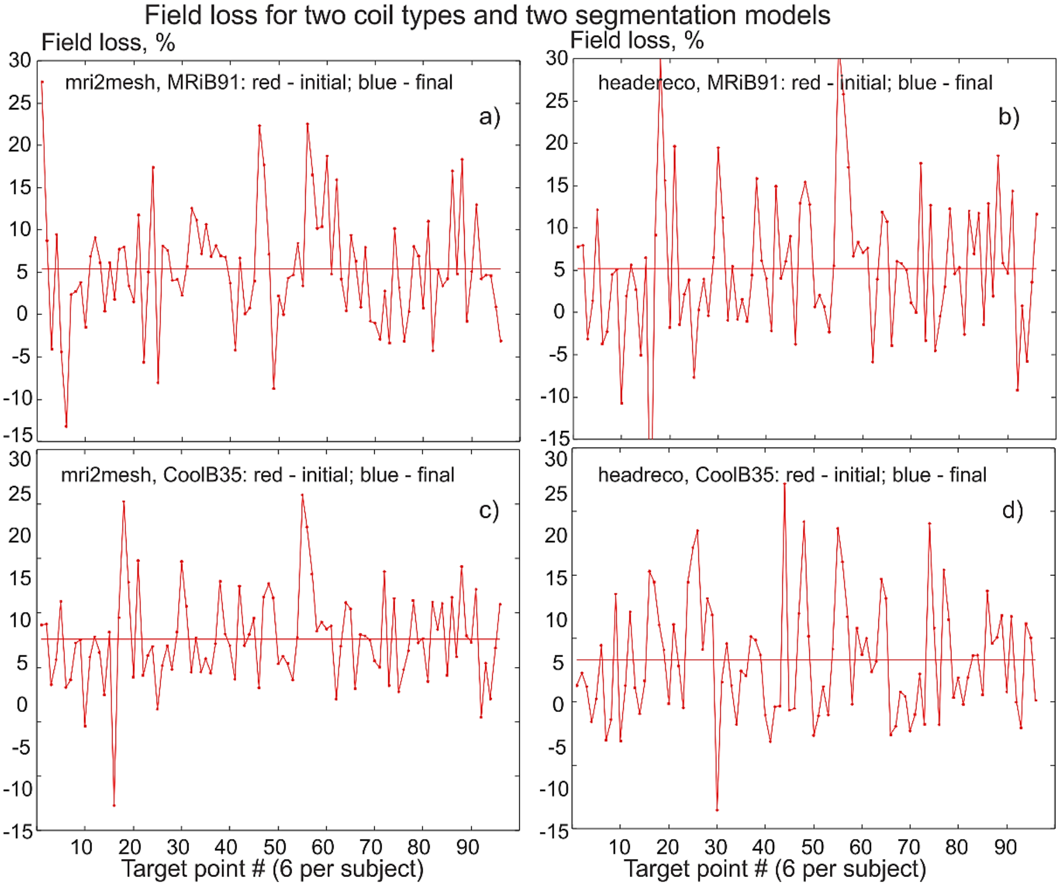


Fig. 42. Target-by-target field loss percentage for the mid-surface. The target points are numbered sequentially. Average data are shown by straight lines.

IV.C.5. Effect of observation surface position: results for the 4:1 intracortical observation surface between gray and white matter

The observation surface now approximately corresponds to the bottom of L5 with thick-tufted pyramidal cells with an early bifurcating apical tuft [101]. Only the results for `headreco` segmentation are shown here. Table 14 summarizes the average absolute deviation, AAD, at the 4:1 surface before and after the inverse-problem solution and defocalization, DF_{coil}, estimates. Every number is an averaged value for 96 targets points. The relevant target-by target data are given in Fig. 43.

Table 14. Focality improvement and uncertainty driven defocalization at the 4:1 observation surface. Every number is an average for 96 targets points. STD values are given using a small font.

Coil type	MRiB91	CoolB35
AAD for sulcus-aligned mapping, mm	**8.91**/2.8	**7.22**/2.4
AAD optimized via TMs-IP, mm	**6.37**/2.2	**5.31**/1.5
Defocalization, DF_{coil}, a.u.	**1.02**	**1.03**

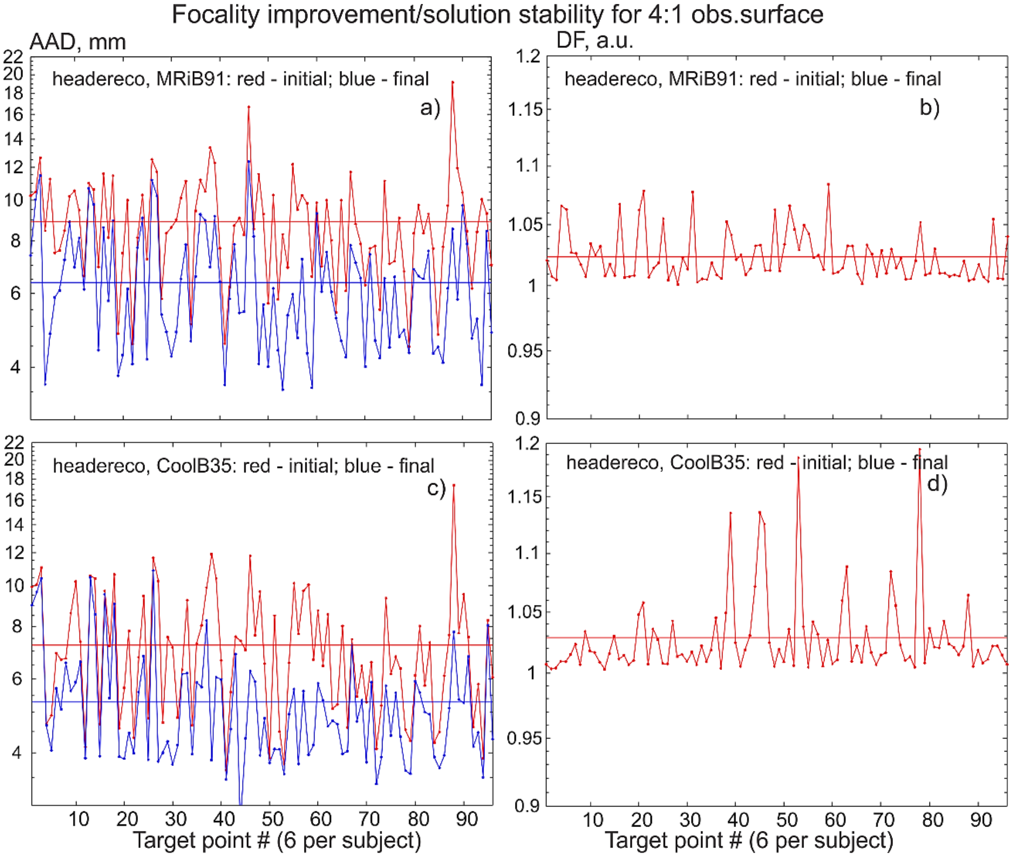


Fig. 43. Target-by-target AAD and stability results for the 4:1 surface. The target points are numbered sequentially. Average data reported in Table 14 are shown by straight lines.

IV.C.6. Differences in optimized coil positions/orientations for 1:1 and 4:1 intracortical observation surfaces

Table 15 summarizes deviations between the two sets of the TMS-IP results for the final coil position/orientation: one for the 1:1 mid-surface and another for the 4:1 surface. Every number is an averaged value for 96 targets points. The standard-deviation values are given using a small font. Only the results for `headreco` segmentation are shown here.

Table 15. Deviations between the two sets of the TMS-IP results for the final coil position/orientation: the 1:1 mid-surface optimization vs. the 4:1 surface optimization. Every number is an average for 96 targets points. Standard-deviation values are given using a small font.

Coil position/angle deviation	MRiB91	CoolB35
Average medial-lateral deviation (*x*), mm	**0.0/**2.4	**-0.1/**2.4
Average posterior-anterior deviation (*y*), mm	**0.0/**3.5	**-0.1/**2.7
Average superior-inferior deviation (*y*), mm	**0.3/**2.7	**0.0/**2.2
Deviation in coil handle (*E*-field) angle, deg	**0.4/**9.8	**0.4/**10.3

IV.D. Discussion

In the present chapter, we perform, arguably for the first time, a rather comprehensive investigation aimed to focus the TMS induced electric field close to a specified target point defined on the gray matter interface in the $\mathrm{M1_{HAND}}$ area while otherwise minimizing it. The goal is to numerically evaluate the usefulness and degree of improvement of the TMS-IP (inverse problem) solution relative to the well-known sulcus-aligned mapping. The TMS-IP is an extremely interesting and complicated problem, with many important questions yet to be answered.

Our major finding is that the TMS-IP provides a moderate yet stable and consistent average improvement, which implies reducing the size of the total electric field "hot spot" with the respect to the 80th percentile and its deviation from the target. This hot spot is determined via the average absolute deviation (AAD) of the field maxima from the target point $\boldsymbol{T}$ in 3D given by Eq. 27 and evaluated over an observation surface. The observation surface is either the 1:1 mid-surface between the gray and white matter or its 4:1 counterpart located significantly closer to the white matter interface, or the white matter interface itself.

Below, we quantify this result with respect to different geometrical parameter configurations and other conditions while paying special attention to coil angles, attempt to link our modeling results to the recent experimental data [109],[108], and finally discuss the limitations of the study.

IV.D.1. TMS-IP solution moderately improves AAD–based TMS focality by approximately 25%

Based in particular on Tables 11 and 14, we observe that the TMS-IP solution provides only a moderate improvement of the TMS focality for $\mathrm{M1_{HAND}}$ at both the mid-surface (~L2/3) and the 4:1 (~L5) surface as compared to the sulcus-aligned motor mapping ($\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ [108],[116]). This improvement – the percentage of the AAD reduction as compared to the original AAD of the sulcus-aligned mapping – is slightly greater than 25% on average.

Table 16 summarizes the percentage of the AAD reduction as compared to the original AAD of the sulcus-aligned mapping, along with its standard deviation. These results are consistent. There are some secondary minor variations due different coil types, segmentation types, and observation surface types.

Table 16. Percentage of the AAD reduction as compared to the original AAD of the sulcus-aligned mapping. Every number is an averaged value for 96 targets points (6 points per subject for 16 subjects). The standard-deviation values are given using a small font.

Segmentation type	mri2mes		headreco	
Coil type	MRiB91	CoolB35	MRiB91	CoolB35
AAD reduction, L2/3 mid-surface, %	**23** 14	**26** 15	**29** 17	**26** 16
AAD reduction, 4:1 L5 surface, %	**24** 14	**27** 16	**29** 17	**26** 15

IV.D.2. TMS-IP solution simultaneously and consistently improves TMS somatotopic resolution

It is observed that the AAD optimization simultaneously and consistently improves the ARD given by Eq. 28 – i.e., enables a potentially better $\mathrm{M1_{HAND}}$ somatotopy-based scanning. Although this improvement is quite modest in terms of the absolute ARD numbers as reported in Table 12, the relative ARD increase might be higher.

IV.D.3. TMS-IP solution is stable with regard to small variations of coil position and orientation

Results of Tables 11 and 14 predict exemplary solution stability at both the 1:1 mid-surface and at the 4:1 observation surface: the average de-focalization (increase in the AAD size) does not exceed 2-3% when coil is moved/rotated by ± 1.5 mm and by ± 0.1 rad about the optimized position. These dimensions approximately correspond to the accuracy of a modern robotized TMS navigator or cobot [121] (controlled 2 mm linear and 4 deg angular coil placement accuracy).

IV.D.4. TMS-IP solution does not change the electric-field direction (coil handle direction) significantly and leads to a moderate field loss

It is well known that the direction of the induced electric field has a large influence on TMS performance [91],[108]. Furthermore, TMS applied to $\mathrm{M1}_{\mathrm{HAND}}$ excites corticospinal neurons most optimally, if the TMS pulse induces an electrical current that flows perpendicular to the central sulcus in a posterior to anterior direction [91]. Table 17 shows the deviation of the coil handle angle and its STD between the initial the sulcus-aligned motor mapping ($\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ [116]) and the final position, respectively. The mean deviation is rather modest while the maximum deviation never exceeds 45°. All tested deviations of the coil handle angle average to approximately 17°.

Table 17. Deviation of the coil handle angle and its STD between the initial the sulcus-aligned motor mapping ($\mathbf{CURVED} - \mathbf{90}^{\mathbf{o}}_{\boldsymbol{flex}}$ [116]) and the final position, respectively. Every number is an averaged value for 96 targets points (6 points per subject for 16 subjects). The standard-deviation values are given using a small font.

Segmentation type	mri2mesh		headreco	
Coil type	MRiB91	CoolB35	MRiB91	CoolB35
Mean angle deviation, deg	**16** 11	**17** 12	**18** 12	**17** 11

A moderate field loss of approximately 6% on average (Table 13) is simultaneously observed for all tested configurations after the application of the TMS-IP solution as compared to the sulcus-aligned mapping.

IV.D.5. TMS-IP solution slightly lifts the coil up to provide space for a small tilt, moves it slightly anterior, and does change the medial-to-lateral position on average. Strictly tangential TMS-IP solution might yield comparable results

The following numbers characterize the average coil movement with respect to the initial sulcus-aligned projection approach given that the field observation surface is the mid-surface. The mean value of the relative vertical coil movement is +2 mm with the standard deviation of 3 mm. This means that the coil is slightly lifted up to provide some space for a more adequate tilt. For all considered cases, the coil is moved slightly *anterior* on average, with the mean value of +1.5 mm and with the standard deviation of 4 mm. The medial-to-lateral movement does not have such an offset; it is characterized by an average value close to 0 mm and by the standard deviation of 3 mm. The total coil movement is 6 mm on average with a 4 mm standard deviation for both intracortical observation surfaces.

A practically interesting question is: what happens if we reduce the search space to three parameters (i.e., restrict ourselves to the only tangential coil movements and rotations)? The relevant simulations have been performed for the `headreco` segmentation and for the mid-surface as the field observation surface. In this case, the improvement reported in Table 16 above is reduced from 29% to 20% for the MRiB91 coil and from 26% to 20% for the CoolB35 coil, respectively. In other words, the tangential TMS-IP might be a simpler viable alternative to the more general solution.

IV.D.6. Focality improvement is reduced from 25% to approximately 20% due to coil position uncertainty and tissue properties uncertainty

The focality improvement discussed in Section IV.D.1 further deteriorates due to uncertainties in the coil position and uncertainties in the tissue properties. According to Tables 11 and 14, the average defocalization factor, DF_{coil}, in the studied range of position variations, is limited by the value $DF_{coil} = 1.03$. As to the conductivity uncertainties, we will employ here the results of previous very comprehensive relevant studies reported in [82] and [109], respectively. There and for TMS, the mean of the electric field on the gyral crown is characterized by a relatively small error of generally less than 5% for a wide range of the electrical conductivities assigned to the different tissue types [82]. We therefore assume the corresponding defocalization factor in the form: $DF_{cond} = 1.05$. According to Eq. 31, the corrected focality improvement to within the main order of magnitude could then be estimated as

$$Focality\ Improvement = \frac{AAD_{initial} - DF_{coil} \times DF_{cond} \times AAD_{final}}{AAD_{initial}} \qquad (33)$$

Plugging in the number computed previously gives us an average value of the corrected focality improvement of slightly higher than 20%, for all cases considered in this study.

IV.D.7. Similar overall focality improvement of the total field is observed for the white matter surface

As shown in Tables 11, 14, and 15, the TMS-IP solution is weakly affected by the type of the intracortical field observation surface: either the 1:1 surface (the mid-surface between the ray and white matter) or the 4:1 surface.

An interesting question is: what will happen to the TM-IP solution when the field observation surface is the white matter interface itself? In this case, the normal field is discontinuous, so we need to distinguish between the two states: either just outside or just inside.

In short, the results for both sides of the conductivity interface look similar but not identical when the total field is concerned. The average TMS-IP based AAD improvement might be somewhat higher. This is because the fields close to the white matter interface are less homogeneous due to the localized surface charges at this interface. The stability against coil position/orientation error is approximately the same (only slightly lower), but the total field loss increases.

As an example, Fig. 44 shows a TMS-IP optimization example (the volumetric total electric field is shown) for one target point for Connectome subject 101309 when the AAD just inside the white matter interface is maximized in the vicinity of the target point. The mri2mesh segmentation has been used in this case.

When the normal field (quite significant at the white matter interface) is concerned, the results become very different. The average TMS-IP based AAD improvement becomes much higher, and the absolute AAD values become lower. This is because the fields close to the white matter interface are less homogeneous due to the localized surface charges at this interface. Simultaneously, the stability against coil position/orientation error decreases and the field loss increases.

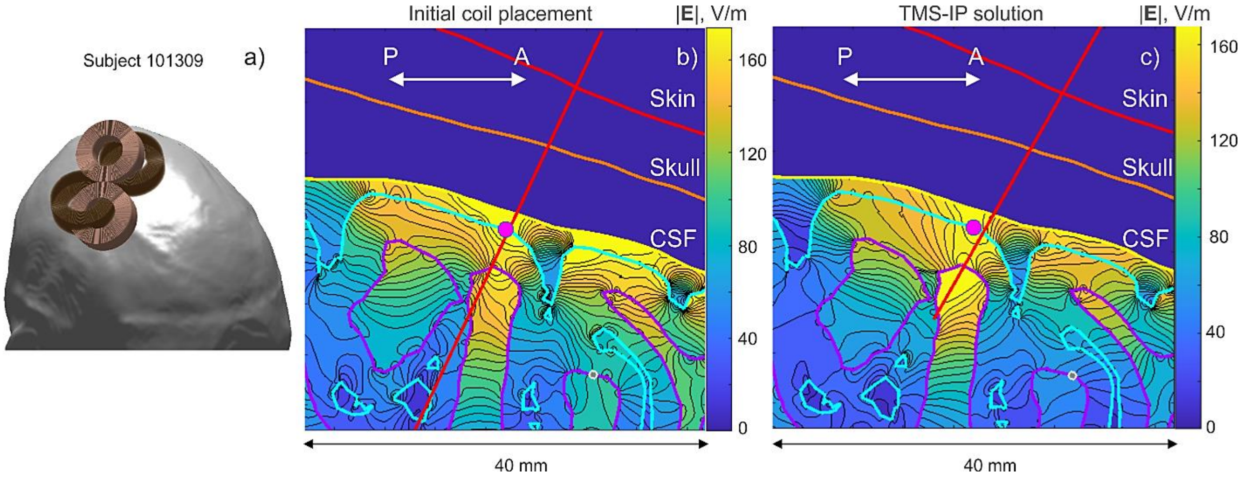


Fig. 44. Volumetric cortical electric fields (total field magnitude, sagittal plane) before and after the TMS-IP solution when the field observation surface is exactly the white matter interface (just inside) and using the Cool B35 coil. TMS pulse strength is 9.4e7 A/s. a) – Initial (dark copper) and final (light copper) coil positions. b), c) – Fields before and after the TMS-IP solution. The target point on the gray matter interface is shown by a magenta circle; the coil axis is marked red. Color scale expand from 0 to the 75th percentile of the maximum total field within the CSF shell. The fields within the scalp and the skull are not plotted for clarity. Numerical field resolution is 100 µm.

IV.D.8. TMS-IP solution strongly depends on the segmentation accuracy

To demonstrate this fact, we consider two extreme cases. In the first one, we perform optimization with the headreco segmentation but assume that the ground truth is the mri2mesh segmentation. We therefore compute the final AAD for the mri2mesh model but based on the TMS-IP solution for headreco. The results for the two coil types are shown in Fig. 45a,c. In the second case, we perform optimization with the mri2mesh segmentation but assume that the ground truth is the headreco segmentation. We therefore compute the final AAD for the headreco model but based on the TMS-IP solution for mri2mesh. The results for the two coil types are shown in Fig. 45b,d.

In Fig. 45, the results are somewhat different for the initial AAD (by 1.5-3%) since the normal vectors to the skin surface are slightly different for both segmentations. However, the results are quite significantly different for the final AAD: a very minor AAD improvement is observed in Fig. 44 in general. This is in stark contrast with Fig. 40 and Fig. 43, respectively.

To explain this fact, we overlapped both segmentations onto the T1 data for several subjects and observed the differences. As an example, Fig. 46 shows both segmentation types for Connectome subject 120111 targeting the motor hand area, $\mathrm{M1_{HAND}}$, of the left hemisphere and superimposed onto the original T1 images.

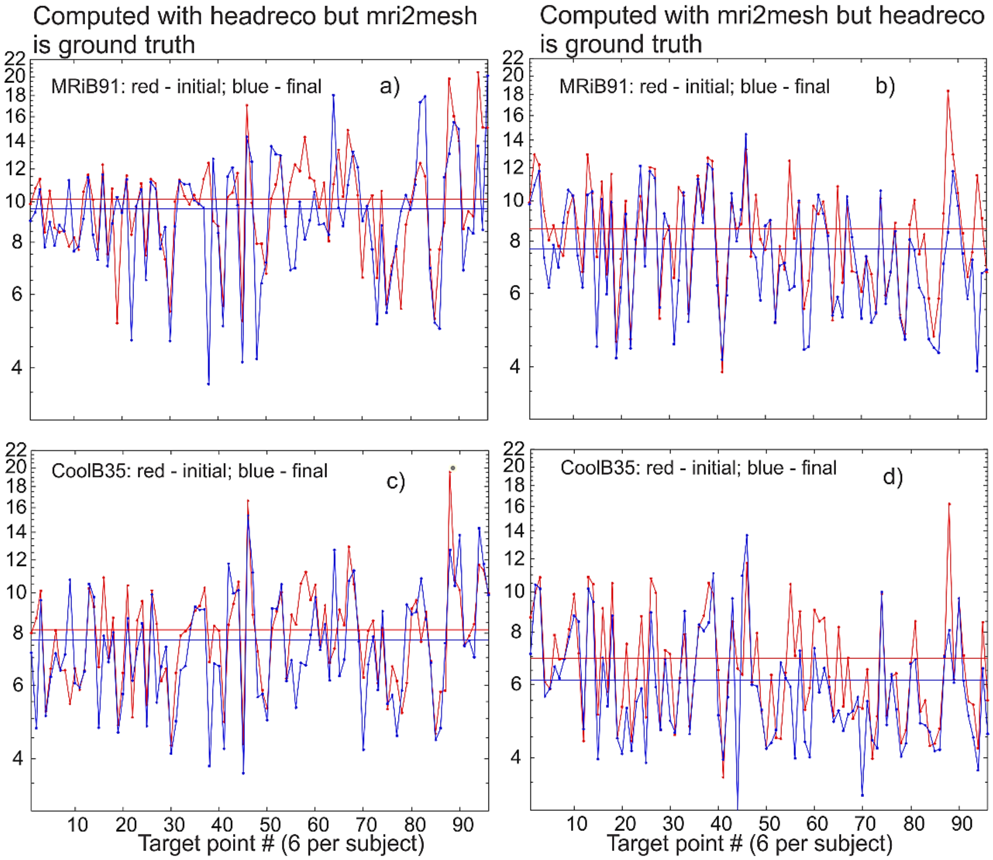


Fig. 45. An extreme case illustrating the effect of segmentation uncertainties on TMS-IP solutions. There is almost no improvement of AAD when `mri2mesh` is the ground truth but optimization is done for `headreco` and a minor improvement when `headreco` is the ground truth but optimization is done for `mri2mesh`. Average data are shown by straight lines.

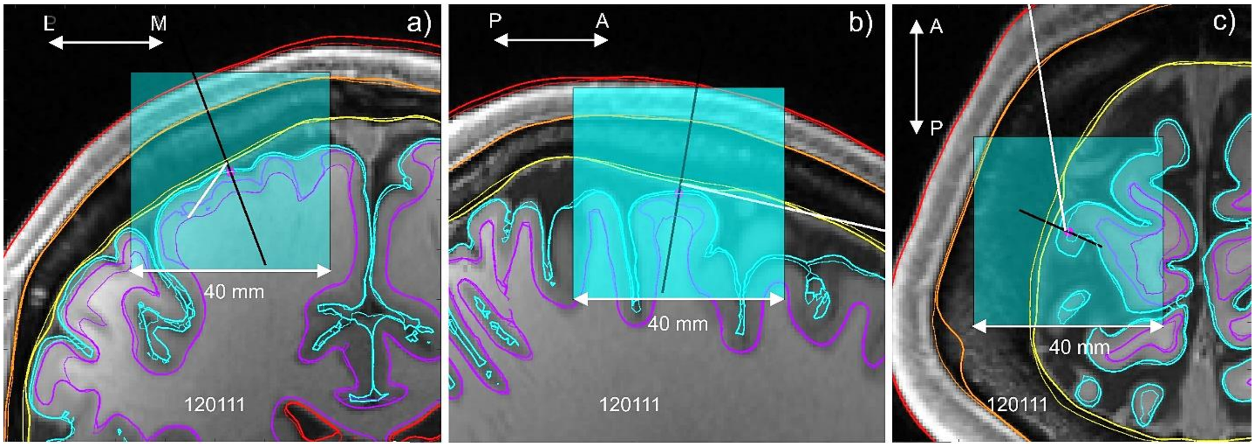


Fig. 46. Segmentation data for **`mri2mesh`** ***(thick curves) and*** **`headreco`** ***(thin curves) superimposed onto T1 data for Connectome subject 120111 in three planes when targeting the motor hand area, $M1_{HAND}$. The target point is shown by a magenta circle.***

In Fig. 46, segmentation data for `mri2mesh` are shown by thick curves; segmentation data for `headreco` are shown by thin curves. While the average deviation between the two models is still modest (cf., e.g., Table 10), the white matter interfaces in Fig. 46 deviate quite significantly, specifically in the $\mathrm{M1_{HAND}}$ and also in the entire motor cortex area. This difference, also known from a number of previous studies [109], [117],[122],[123], appears to have a large influence on the TMS-IP performance as observed in Fig. 45.

IV.D.9. Link to recent experimental data [108],[109]

In Ref. [109], the hotspots (the anticipated target points) of a high congruence factor between the applied electric field and the motor evoked potential (MEP) have been initially determined for three subjects while recording MEPs over the first *dorsal interosseous* muscle belly and one at the proximal interphalangeal joint. A CB60 coil of MagVenture was used in that experiment, which is similar in size to the first coil of the present study (MRiB91).

Next, using an exhaustive search with ~5,000 search points, the coil positions and orientations were found that maximize the electric-field magnitude in the hotspots or the target points in $\mathrm{M1_{HAND}}$. The subsequently measured (over 16 experiments with the similar scan spacing of 7-10 mm) motor thresholds were always lowest for the predicted optimal positions and orientations. This study specifically mentions that these optimal coil orientations were similar to the commonly used 45° angle towards the *fissura longitudinalis* (cf. [109], Fig. 46) or $\mathrm{CURVED} - 45^{o}_{fix}$ in terms of Ref. [108].

While we were unable to perform our own relevant experiments, we could establish quantitative correlations with regard to the optimal angular coil positioning. The sulcus-aligned mapping (the initial guess shown for one head model in Fig. 37) gave us the (absolute) angle towards the *fissura longitudinalis* of 29° on average for 16 subjects and 96 target points (cf. Fig. 37). The TMS-IP solution increased this average value to 36° (`headreco` segmentation) or to 35° (`mri2mesh` segmentation) for both coil types, with a 19° standard deviation. This means a closer agreement with the observation stated above for the minimum motor threshold given the limited angular experimental resolution.

On the other hand, Ref. [108] reports a similar experimental somatotopic resolution for the $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ and $\mathrm{CURVED} - 45^{\mathrm{o}}_{fix}$ approaches, respectively, when *abductor digiti minimi* and *first dorsal interosseus* muscles are concerned. Still, the cross-correlation was the lowest for the $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ approach [108]. Our average angular TMS-IP data are almost in between the two approaches, but with a small yet visible bias toward the $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ approach too.

One must emphasize that only the average angular directions for the coil handle are reported here. Certain TMS-IP solutions may significantly deviate from them as shown in Figs. 38 and 39, respectively.

IV.D.10. Study limitations

The $\mathrm{M1}_{\mathrm{HAND}}$ area has been the main focus of his study. Along with this, we have also collected and processed some less extensive results related to the dorsolateral left prefrontal cortex (DLPFC), which is pathophysiologically linked to depression [17],[124],[125],[126]. A single observation point was chosen at the middle frontal gyrus of DLPFC and TMS-IP has been solved for all 16 subjects. Generally, these results look similar to the results reported in this study, but they will need additional verification and extensions to multiple observation points.

With respect to the numerical modeling, the major drawback of the BEM-FMM approach used in this study is an inability to straightforwardly include into consideration the white matter anisotropy, which may be quite significant in the subcortical white matter [123] and which may have a substantial effect of the E-field distribution [82]. On the other hand, this approach is free of volumetric field averaging throughout the gray matter volume. The method requires a relatively small amount of RAM (6-12 Gbytes for the present head models), but it runs best on multicore machines due to inherent FMM parallelization.

Powerful mathematical tools have recently been developed and implemented [127],[128] for fast computations of the TMS-IP solution via the auxiliary dipole method (ADM) or the magnetic stimulation profile approach, for determining the optimum coil position and orientation. The goal of the present study is not to compete with these tools, but is rather to evaluate the usefulness and degree of improvement of the TMS-IP solution itself. We attempted to answer the question by how much the TMS-IP solution can improve the focality of one common projection approach – the $\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ mapping – and by how much it can help us to differentiate between the neighbor targets. For this purpose, we employ a much slower but more general traditional approach. This approach might be more flexible for studying multiple geometrical model variations and arbitrary field observation surfaces, arbitrary coil positions (slightly) different from those strictly tangential to the scalp, and it does not require extra mesh conditioning.

From the practical point of view, the solution of a particular TMS-IP will likely be best accomplished by using specialized highly efficient algorithms such as [127],[128] instead of the straightforward yet slow approach of this study.

IV.E. Conclusions

The ideal TMS inverse-problem solution studied in this chapter predicts a stable focality improvement of approximately 25% on average in the $\mathrm{M1_{HAND}}$ area for all considered subjects, for two coil distinct coil types, and for both intracortical observation surfaces under study. The solution is using the average absolute deviation, AAD, from target given by Eq. 27 as the cost function. The predicted improvement is relative to the projection-based perpendicular-to-sulcus initial coil placement or the sulcus-aligned mapping ($\mathrm{CURVED} - 90^{\mathrm{o}}_{flex}$ [108],[116]). The inverse-problem solution simultaneously and consistently improves the somatotopic TMS resolution, although this improvement is also moderate. Last but not least, the TMS-IP solution does not significantly change the electric-field direction (coil handle direction) and leads to a moderate field loss of approximately 6% on average.

When the coil position/orientation and conductivity uncertainties are included into consideration, the improvement deteriorates. Given the applied level of uncertainties, the focality improvement is reduced to approximately 20% for the total electric field with its maximum located mostly close to the gyral crown. A similar final estimate is obtained for white matter as an observation surface. The present results will change when the levels of uncertainties change. The results could be expanded by considering more representative cases and by performing a more rigorous uncertainty power analysis [129].

The TMS-IP solution strongly depends on the segmentation accuracy. For the extreme case when the solution for one realistic segmentation model (`mri2mesh` or `headreco`) is used in another model and vice versa, the TMS-IP solution generates minor, if any, improvement.
The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health and Human Services.

V. TES-Focused BEM-FMM Toolkit

This chapter has been mainly excerpted from [2] (under review at the time of writing).

V.A. Introduction

The finite element method (FEM) is widely and successfully used to solve the corresponding bioelectric field problem for different electrode configurations using anatomical isotropic and anisotropic head models. The leading open-source TES FEM software is SimNIBS [46] along with ROAST [130], but commercial packages such as Ansys Maxwell and COMSOL are also frequently used [131]. In the past, DBS problems have been subject to extensive FEM investigations [41],[42],[132]-[134].

In this chapter, we formulate, validate, and demonstrate possible applications of an alternative numerical approach to model electric brain stimulation. This is the boundary element fast multipole method or BEM-FMM, which is formulated in terms of the induced surface charge density. The approach possesses numerically unconstrained field resolution close to fine anatomical structures or miniature electrodes. This resolution is not limited by the size of the underlying tetrahedral mesh. Therefore, it might potentially be capable of modeling multiscale problems with different mesh resolutions at macroscopic (major brain compartments) and microscopic (submillimeter and finer) scales, respectively. In the past, the BEM-FMM approach was formulated for transcranial magnetic stimulation (TMS) [1],[49],[50] and electroencephalography/magnetoencephalography (EEG/MEG) recording problems [135]. Models of voltage and current electrodes require modifications and extensions of the algorithm.

The chapter is organized as follows. Section V.B. describes the implementations of voltage, shunt, current, and floating electrodes, either surface or embedded. Section V.C. presents several validation examples. Section V.D. discusses two application examples pertinent to DBS and ICMS, respectively. Section V.E. discusses the results, and Section V.F. concludes the study.

V.B. TES-Specific Implementation Details

Fig. 47 schematically illustrates the electrode configurations under study. They may include (i) imprinted TES surface electrodes, (ii) embedded ICMS electrode arrays of different configurations (planar or laminar), and (iii) embedded DBS electrodes.

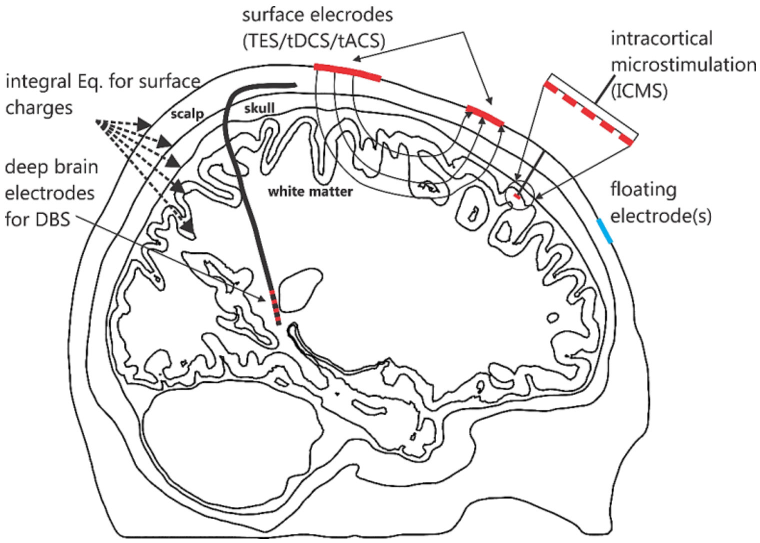


Fig. 47. Different electrode types modeled by BEM-FMM: (i) imprinted TES surface electrodes, (ii) embedded ICMS electrode arrays of (planar or laminar), and (iii) embedded DBS electrodes.

As compared to a simpler excitation by a prescribed external volumetric field (e.g., a primary coil field of TMS [1],[49],[50] or a primary field of internal EEG/MEG dipole sources[135]), the integral equation of the BEM-FMM algorithm has to be modified for voltage, shunt, current, and/or floating electrodes with reference (Fig. 47). The voltage electrodes may refer to TES surface electrodes, ICMS embedded electrode arrays, and DBS embedded electrodes. These electrodes may be normalized to generate a constant total current. In that case, they are often called current electrodes (or shunt electrodes [136][136]). True current electrodes (with a strictly uniform current distribution over the entire electrode surface and a non-uniform voltage distribution) may approximate contact sponges [137] and are useful as a tool for reciprocity analyses [138],[139].

V.B.1. Integral equation for tissue conductivity interfaces

Induced electric charges with a surface charge density $\rho(\boldsymbol{r})$ will reside on every macroscopic or microscopic tissue conductivity interface S including the electrode surfaces S_e. These induced surface charges will alter the field created by the electrodes in vacuum or in a background homogeneous medium to fulfill the law of current conservation across the boundaries. The electric field generated by all surface charges anywhere in space except the charged interfaces themselves is governed by Coulomb's law,

$$\boldsymbol{E}(\boldsymbol{r}) = \frac{1}{4\pi\varepsilon_0}\int_S \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\rho(\boldsymbol{r}')d\boldsymbol{r}', \quad \boldsymbol{r} \notin S \tag{1}$$

where ε_0 is dielectric permittivity of vacuum. The electric field in Eq. 1 is discontinuous at the interfaces. When approaching a charged interface S with a certain normal vector $\boldsymbol{n}$ and assigning index *in* to the medium from which $\boldsymbol{n}$ is pointing and index *out* to the medium toward which $\boldsymbol{n}$ is pointing, the electric field close to the boundary is given by two limiting values [140]

$$\boldsymbol{E}_{in/out}(\boldsymbol{r}) = \frac{1}{4\pi\varepsilon_0}\int_S \frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\rho(\boldsymbol{r}')d\boldsymbol{r}' \mp \boldsymbol{n}(\boldsymbol{r})\frac{\rho(\boldsymbol{r})}{2\varepsilon_0}, \qquad \boldsymbol{r} \in S \tag{2}$$

where $\boldsymbol{E}_{in/out}(\boldsymbol{r}) \equiv \lim_{\Delta\to 0}\boldsymbol{E}\big(\boldsymbol{r} \mp \Delta\boldsymbol{n}(\boldsymbol{r})\big)$ and $\boldsymbol{r} \mp \Delta\boldsymbol{n}(\boldsymbol{r})$ is not on the surface. The second term on the right-hand side of Eq. 2 is a continuous contribution of all other surface charges while the last term is a discontinuous contribution of a local planar sheet of charge located exactly at $\boldsymbol{r}$ resulting in a jump of the normal electric field by $\frac{\rho(\boldsymbol{r})}{\varepsilon_0}$. Another "discrete" numerical interpretation is that the integral on the right-hand side of Eq. 2 is the continuous contribution of surface charges of all facets except the facet located exactly at $\boldsymbol{r}$. The last term on its right-hand side is a discontinuous contribution of the facet located exactly at $\boldsymbol{r}$. This facet is locally a planar sheet of charge when $\boldsymbol{r}' \to \boldsymbol{r}$.

An integral equation for $\rho(\boldsymbol{r})$, which is the Fredholm equation of the second kind, is obtained after substitution of Eq. 2 into the quasistatic boundary condition which enforces the continuity of the normal component of the electric current across the interface. That is,

$$\sigma_{in}\boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{in}(\boldsymbol{r}) = \sigma_{out}\boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{out}(\boldsymbol{r}),\ \boldsymbol{r} \in S\backslash S_e \tag{3b}$$

Here, $\sigma_{in}, \sigma_{out}$ are the conductivities just inside and outside with respect to the direction of the normal vector. After combining the similar terms, the result has the form of the homogeneous Fredholm equation of the second kind,

$$\frac{\rho(\boldsymbol{r})}{2} - K(\boldsymbol{r})\boldsymbol{n}(\boldsymbol{r}) \cdot \int_S \frac{1}{4\pi}\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\rho(\boldsymbol{r}')d\boldsymbol{r}' = 0, \qquad \boldsymbol{r} \in S\backslash S_e \tag{4b}$$

where K is the electric conductivity contrast $K = \frac{\sigma_{in}-\sigma_{out}}{\sigma_{in}+\sigma_{out}}$. Here, the integration surface S contains *all* charged interfaces *including* those of the electrodes. For a harmonic excitation with angular frequency ω in a medium with permittivity ε, the contrast value in Eq. 4b becomes complex in the form $\sigma \to \sigma + j\omega\varepsilon$. It is noted that homogeneous Eq. 4b is applicable to the tissue interfaces but not to the electrode surfaces, which have to be treated separately.

V.B.2. Integral equation for voltage electrodes (or shunt electrodes normalized by constant current)

For a voltage electrode, we assume that a certain voltage V is impressed at a certain electrode surface S_e. The voltage may be different for each electrode. The electrode surfaces may be considered as an imprinted part of other closed surfaces, for example a part of the insulated lead surface for DBS electrodes. At the electrode surface(s), the electric potential generated by all surface charges must be equal to the prescribed voltage(s). The Fredholm integral equation of the first kind is therefore imposed instead of Eq. 4b. That is,

$$\frac{1}{4\pi\varepsilon_0}\int_S \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}' = V, \qquad \boldsymbol{r} \in S_e \tag{34}$$

V.B.3. Integral equation for current electrodes with uniform inflowing current density

True current electrodes inject a constant current density $j_{\boldsymbol{e}}$ into the volume conductor. Numerically, it is convenient to point the normal vector $\boldsymbol{n}(\boldsymbol{r})$ from a contact tissue (*in*) toward the electrode material (*out*). The current continuity across the electrode surface requires that

$$j_{\boldsymbol{e}} = -\sigma_{in}\boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{in}, \qquad \boldsymbol{r} \in S_e \tag{35}$$

given that a positive current flows from the electrode. Here, σ_{in} is the conductivity of the tissue in contact and $\boldsymbol{E}_{in}$ is the total electric field close to the contact. Substituting Eq. 35 into Eq. 2, one obtains

$$\frac{\rho(\boldsymbol{r})}{2} - \boldsymbol{n}(\boldsymbol{r}) \cdot \int_S \frac{1}{4\pi}\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\rho(\boldsymbol{r}')d\boldsymbol{r}' = \frac{\varepsilon_0 j_{\boldsymbol{e}}}{\sigma_{in}} \quad \boldsymbol{r} \in S_e \tag{36}$$

Eq. (36) has the same form as Eq. (4b) with $K = 1$ and, additionally, with the extra source term describing current injection.

V.B.4. Integral equation for floating metal electrodes

For floating metal electrodes (constant but unknown voltage over the electrode surface), Eq. (4b) can still be used, but with $K = -1$ which physically corresponds to a very high metal (copper, platinum iridium, etc.) conductivity as compared to the conductivity of the surrounding tissues.

V.B.5. Current conservation law (Kirchhoff's current law) on electrodes

The current conservation law is essential for voltage electrodes. It integrates the resulting normal current density, $\sigma_{in}\boldsymbol{n}(\boldsymbol{r}) \cdot \boldsymbol{E}_{in}$, over all N_e electrode surfaces and requires the net result to be zero, i.e.

$$\sum_{n=1}^{N_e}\int_{S_e} \sigma_{in}\left\{\frac{\rho(\boldsymbol{r})}{2} - \boldsymbol{n}(\boldsymbol{r}) \cdot \int_S \frac{1}{4\pi}\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\rho(\boldsymbol{r}')d\boldsymbol{r}'\right\}d\boldsymbol{r} = 0 \tag{37}$$

Scalar Eq. 38 will be enforced in a "weak" form, i.e. by additively combining it with Eqs. 4b, 34, and/or 36.

V.B.6. Discretization and preconditioning

The surface charge density is expanded into pulse bases (zeroth-order basis functions) on triangular facets t_m with area A_m. The standard Galerkin method is then applied. When only voltage (and/or shunt/floating) electrodes are used, Eq. 4b and Eq. 34 are solved simultaneously, which gives us a system of $M + N$ linear equations for unknown expansion coefficients c_m. Here, M is the number of electrode facets and N is the number of all other facets. In a matrix formulation, one introduces the corresponding matrices $\mathbb{R}$ and $\mathbb{Z}$, and produces the system of equations in the following form:

$$\begin{array}{c} \begin{bmatrix} \mathbb{R}_{mn} = \iint_{A_m A_n} \dfrac{1}{4\pi\varepsilon_0}\dfrac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\, d\boldsymbol{r}' d\boldsymbol{r} \\ \\ m = 1{:}M, n = 1{:}M+N \end{bmatrix} \\ \\ \begin{bmatrix} \mathbb{Z}_{mn} = \dfrac{1}{2}\delta_{mn} - \dfrac{K}{A_m}\boldsymbol{n}_m \cdot \iint_{A_m A_n} \dfrac{1}{4\pi}\dfrac{(\boldsymbol{r}-\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|^3}\, d\boldsymbol{r}' d\boldsymbol{r} \\ \\ m = M+1{:}M+N, n = 1{:}M+N \end{bmatrix} \end{array} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ c_{M+N} \end{bmatrix} = \begin{array}{c} \begin{bmatrix} \int_{A_m} V d\boldsymbol{r} \end{bmatrix} \\ m = 1{:}M \\ \\ \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \end{array} \tag{38}$$

Coupled Eqs. 38 are solved iteratively. The iterative solution does not require the explicit matrix formulation; only the vector-matrix product in 38 needs to be formed. A proper preconditioner must be used for the first M electrode equations. Otherwise, an iterative solution will not converge. The simple diagonal preconditioner $\mathcal{M}$ is as follows:

$$\mathcal{M}_m = \left[\frac{1}{4\pi\varepsilon_0}\iint_{A_m A_m} \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\, d\boldsymbol{r}' d\boldsymbol{r}\right], \quad m = 1{:}M \tag{39a}$$

The full left electrode preconditioner, $\mathcal{M}$, for the first M electrode equations, reads:

$$\mathcal{M}_{mn} = \left[\frac{1}{4\pi\varepsilon_0}\iint_{A_m A_n} \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}\, d\boldsymbol{r}' d\boldsymbol{r}\right], \quad m, n = 1{:}M \tag{39b}$$

The preconditioner dramatically improves performance. Facets of different electrodes are put up front of the combined mesh, and the matrix on the right-hand side of Eq. 39b is computed for all electrodes at once. Matrix $\mathcal{M}$ is easily and explicitly invertible; it solves the Fredholm integral equation(s) of the first kind (Eq. 38) exactly if only the electrodes were present.

The double potential integrals present in Eqs. 38 and 39 require care. Uniformly charged facets, which are spatially close to one another, cannot be accurately approximated by point charges located at their centers and cannot therefore be treated with the FMM. The corresponding facet-to-facet nearfield potential integrals in Eqs. 38 and 39 are instead directly calculated and stored in a sparse near-field BEM matrix using analytical integration for the inner integral and a Gaussian quadrature of 10th degree of accuracy for the outer integrals at the preprocessing stage. The number of geometrical (based on Euclidian distance) neighbors in Eqs. 38 may vary, but a relatively reasonable number (4-16 for TES/DBS/ICMS problems, specified below in every case) was found to be adequate. It must be noted that the spatially close facets may belong to different tissues or implanted devices when tightly-coupled structures (e.g., brain membranes) are considered.

V.B.8. Fast multipole method. Iterative solution

The general-purpose fast multipole method (FMM) [64] and its most recent software implementation [66] are applied to compute the matrix-vector product in Eqs. 38. Applying the FMM is equivalent using the center-point approximation at face centers $\boldsymbol{r}_m$, yielding in Eqs. 38

$$\int_{A_m}\int_{A_n}\frac{\boldsymbol{r}-\boldsymbol{r}'}{|\boldsymbol{r}-\boldsymbol{r}'|^3}d\boldsymbol{r}d\boldsymbol{r}' \approx A_mA_n\frac{\boldsymbol{r}_m-\boldsymbol{r}_n}{|\boldsymbol{r}_m-\boldsymbol{r}_n|^3}, \int_{A_m}\int_{A_n}\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}d\boldsymbol{r}' \approx A_mA_n\frac{1}{|\boldsymbol{r}_m-\boldsymbol{r}_n|} \tag{40}$$

In terms of an iterative solution, the problem is thus equivalent to finding the electric field and the potential at target points $\boldsymbol{r}_m$ generated by the point charges located at source points $\boldsymbol{r}_n$. The accuracy of the corresponding FMM (the number of levels) is unlimited; it is conventionally estimated for arbitrary volumetric charge distributions. However, for surface-based charge distributions, a better relative accuracy is observed than for the arbitrary volumetric distributions. Therefore, the relative FMM precision levels of 10^{-2} to 10^{-3} were found to be adequate when compared to the most accurate numerical solution.

In the iterative matrix-free solution, Eqs. 38 are solved simultaneously and the FMM is applied at every iteration step for the matrix-vector computation. Then, the inaccurate near-field contributions due to Eq. 40 are subtracted, and accurate double potential integrals are used instead as explained in the previous section.

The generalized minimum residual method (GMRES) [141] and its flexible version [142] used in this study were found to converge better than the bi-conjugate gradients method and its variations. Since the Fredholm equation of the second kind is essentially solved, with a strong diagonal dominance, the relative residual error approaches the relative solution error.

V.C. Results – Validation Examples

In the following, we investigated the performance and accuracy of the BEM-FMM for different validation examples and compared the solutions against analytical and numerical reference solutions. The validation examples are available online with appropriate documentation [55]. All codes are written in MATLAB and do not require extra dependencies or downloads.

V.C.1. Comparison with analytical solution

The problem geometry is shown in Fig. 48a. It includes a sandwiched brick with sixteen 1 mm thick layers having interleaving static conductivities of 0.5 S/m and 0.2 S/m, respectively. The brick has the dimensions of 30×30×16 mm. The media are isotropic, with the same values of the dielectric constant. Two electrodes (± 10 V, chosen as infinitely thin plates for the BEM-FMM solution) cover the top and the bottom of the brick. The goal is to obtain the BEM-FMM numerical solution and compare it with the analytical reference solution. The analytical solution *inside* (but not *outside*) the brick implies one-dimensional *z*-directed current density and electric field distributions, and a linear electric potential distribution which satisfy all the necessary boundary conditions and Laplace equation. The electric current density is constant inside the volume, while the electric field is piecewise constant. Using the corresponding lumped-circuit model, one has

$$I_{total} = 0.3214\text{ A}\,, J_z = 357.143\frac{A}{m^2}, E_{z1} = \ 714.286\frac{V}{m}, E_{z2} = \ 1785.714\frac{V}{m} \qquad (41)$$

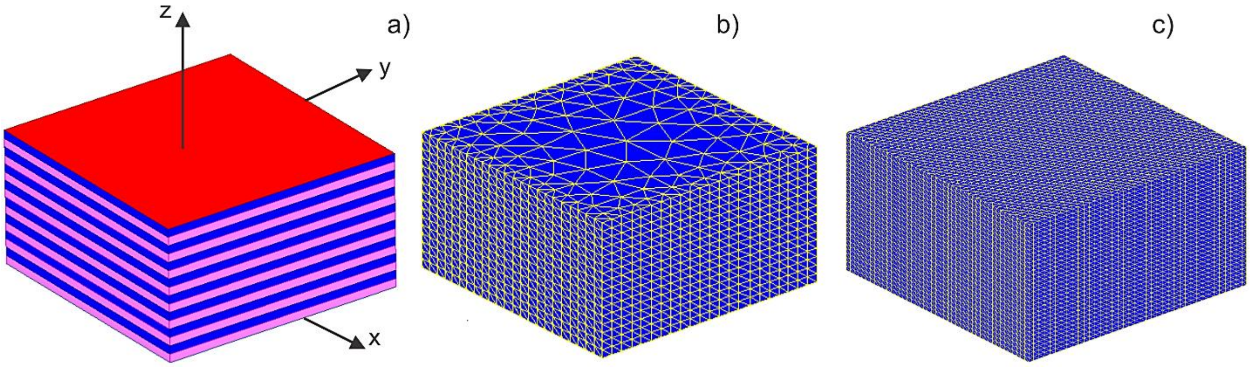


Fig. 48. a) – Problem geometry under study. The top electrode (+10 V) is marked in red. Coarse b) and fine c) surface meshes used for the solution are shown. The inner facets separating every conducting layer are not shown.

The observation grid is composed of the centers of cubic voxels inside the grid. Two voxel sizes are considered: 0.5×0.5×0.5 mm (0.12 M voxels in total) and 0.2 mm (1.8 M voxels in total), respectively. The error in the vector electric field, $\boldsymbol{E}_t(\boldsymbol{r})$**,** is computed as a relative 2-norm error as follows:

$$Error_t = \frac{\|\boldsymbol{E}_t(\boldsymbol{r}) - \boldsymbol{E'}_t(\boldsymbol{r})\|}{\|\boldsymbol{E}_t(\boldsymbol{r})\|} \qquad (42)$$

where the BEM-FMM solution is marked by apostrophe, and the analytical solution is unmarked.

The BEM-FMM solution uses 20 adaptive passes, precise analytical expressions for sixteen neighbor surface integrals, relative FMM precision of 10^{-2}, and the full left preconditioner (Eq. 39b) for the electrode surfaces. The BEM-FMM solution runs in approximately 2-7.5 sec on a 2.1 GHz Windows server (MATLAB 2020b) depending on the surface mesh size. Two surface meshes are considered: an extruded coarse transversally unstructured mesh with 7,693 facets and an extruded fully structured mesh with 69,593 facets, respectively, as shown in Fig. 48b,c. Table 18 reports the 2-norm relative error values (computed from the electric field observed at the center of each voxel) and the BEM-FMM iterative solution run times for both meshes. The error is somewhat larger in the second case due to smaller distances from the electrodes.

Table 18. 2-Norm relative error values computed at the center of each voxel for two different BEM-FMM meshes. Voxels are used for error calculations only.

Voxel size	$Error_t$ for the BEM-FMM mesh with 7,693 facets	$Error_t$ for the BEM-FMM mesh with 69,593 facets	Solution time for the mesh with 7,693 facets	Solution time for the mesh with 69,593 facets
0.5 mm (0.1152 M voxels)	**3.96%**	**0.83%**	**2.4 sec**	**7.5 sec**
0.2 mm (1.8 M voxels)	**4.34%**	**1.44%**	**2.4 sec**	**7.5 sec**

Surface field recovery times just inside/outside any surface are less than 0.5 sec. Volumetric field recovery times at the voxel centers are significantly longer. For the BEM-FMM mesh with 69,593 facets, they are 42 sec (0.5 mm voxel size) and 392 sec (0.2 mm voxel size), respectively. Sub-percent vector field accuracy can be obtained for both voxel grids with the BEM-FMM meshes of ~100,000 triangular facets in 7.5-10 sec on a 2.1 GHz machine. Fine tuning (e.g., reducing the number of iterations) substantially reduces the execution times in Table 18.

V.C.2. Comparison to FEM for simple geometries

The geometry shown in Fig. 49a is tested first. A brick with the side of 1 m has two *square* surface electrodes at ± 1 V on its top with the size of 0.2×0.2 m each, and separated by 0.3 m. The isotropic conductivity is set as $\sigma = 0.1\ \mathrm{S/m}$. The BEM-FMM solution is compared to an FEM solution obtained by Ansys Maxwell (Ansys® Electronics Desktop 2020 R2), which is using 35 adaptive mesh refinement passes and a final mesh with ca 1.2 M tetrahedra. The BEM-FMM solution is using a structured surface triangular mesh created with three brick subdivision schemes detailed in Table 19. The GMRES-based BEM-FMM with sixteen neighbor integrals was run for 15 iterations, and the relative residual values are in the range of 10^{-4} to 10^{-5}.

The observation domain includes centers of all cubic voxels of a structured mesh within a smaller brick having the side of 0.9 m as shown in Fig. 49a. This was done in order to (i) support a more realistic scenario where smaller fields at a certain distance from the electrodes (e.g., in the in gray and white matter) are most important and; (ii) eliminate larger interpolation errors very close to the electrodes which could mask the rest of the solution. A 2-norm relative error percentage in the vector electric field from Eq. 40 within the observation domain is then compared for the two numerical solutions. The corresponding results are given in Table 19. In the last column of Table 19, the error for all voxel centers within the *entire* brick volume is given.

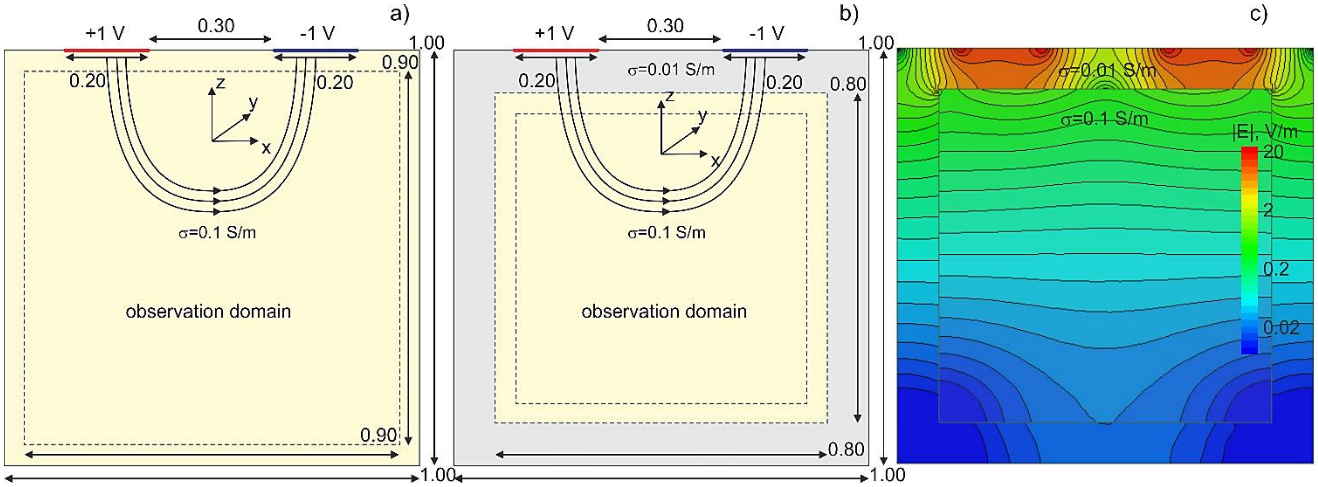


Fig. 49. Electrode assembly for surface electrode testing and comparison against FEM solutions. a) – Homogeneous isotropic brick model, b) – heterogeneous isotropic brick model, and c) – electric field distribution of the heterogeneous brick model.

Table 19. 2-Norm relative error in the vector electric field within the 0.9 m observation volume in Fig. 49a and within the entire volume, respectively. Voxels are used for error calculations only.

Mesh res./ base brick voxel subdivisions	Surface mesh size, facets	Solution time	$Error_t$ for the 0.9 m subdomain	$Error_t$ for the entire domain
20×20×20	**4,800**	**1.3 sec**	**3.2%**	**4.2%**
40×40×40	**19,200**	**2.0 sec**	**1.8%**	**2.8%**
80×80×80	**76,800**	**4.0 sec**	**0.9%**	**2.1%**

The geometry of Fig. 49b is tested next. The brick of the previous example is made composite. It consists of an outer shell to emulate an insulating domain such as skull with a thickness of 0.1 m. It has a low isotropic conductivity of 0.01 S/m. The inner brick, representing the brain, has a much higher conductivity of 0.1 S/m. This geometry is more challenging since the outer shell blocks the field as shown in Fig. 49c and a larger relative error may be observed within the inner volume. The observation domain included all voxel centers within a smaller brick with a side-length of 0.7 m. The corresponding results are given in Table 20. In the last column, the error for all voxel centers within the inner volume including those close to its boundary is given.

Table 20. 2-Norm relative error in the vector electric field within the 0.7 m observation volume in Fig. 49b and within the entire inner-brick volume, respectively. Voxels are used for error calculations only. The GMRES-based BEM-FMM iterative solution uses 25 iterations.

Mesh res./ base brick voxel subdivisions	Surface mesh size, facets	Solution time	$Error_t$ for the 0.7 m subdomain	$Error_t$ for the entire inner brick domain
20×20×20	**9,600**	**1.8 sec**	**4.7%**	**4.7%**
40×40×40	**38,400**	**3.5 sec**	**2.7%**	**2.8%**
80×80×80	**153,600**	**10.0 sec**	**2.1%**	**2.4%**

V.C.3. Comparison to FEM considering a realistic head model

A validation example pertinent to TES is adopted from [50] and subsequently revised. Two electrodes with a fronto–medial placement and with diameters of 30 mm each are applied onto the skin surface of subject #110411 of the Connectome database [78] as shown in Fig. 50a. Surface meshes are obtained with the SimNIBS 2.1 pipeline [46] using mri2mesh; tissue properties are adopted from [83]. The composite model has 0.86 M facets. The model of current electrodes (with a uniform current distribution and total current of 1 mA) from Eq. 36 is used. A numerical solution is again compared to FEM using Ansys® Electronics Desktop 2020 R2 but with voltage electrodes generating the same current. Fig. 50b shows the resulting electric field distribution (field magnitude) in the intracranial volume using BEM-FMM with eight neighbor integrals. Fig. 50c displays an arc connecting the electrode centers. Fig. 50d compares the electric field magnitude for both methods computed along the arc connecting the electrode centers. The FEM curve is shown in red, while the BEM-FMM curve is blue. Table 21 compares both simulations. For BEM-FMM, the required RAM did not exceed 4 Gbytes for the iterative solution while the FEM solution required 8 Gbytes for three adaptive passes and nearly 30 Gbytes of RAM for 6 adaptive passes.

Table 21. Comparative performance of the two modeling methods. The FEM run times are for Intel Xeon E5-2698 v4 CPU (2.20 GHz) server with 256 Gbytes of RAM. The BEM-FMM run times are given for Intel Xeon E5-2683 v4 CPU (2.1 GHz) based server. The relative error in the field magnitude is given for intracranial tissues (CSF, gray/white matter) only.

Ansys FEM	Ansys sol. time (8 cores)	BEM-FMM sol. Time	Intracranial rel. field magn. error for an arc in Fig. 4c	Intracranial rel. field magn. error in the entire coronal plane

4.8 M tets, 3 adapt. passes	**2 h 29 m**	**~60 sec (rel. residual of 10^{-3})**	**3.1%**	**6.31%**
6.9 M tets, 4 adapt. passes	**6 h 23 m**		**3.0%**	**6.15%**
9.0 M tets, 5 adapt. passes	**8 h 43 m**		**2.9%**	**6.06%**

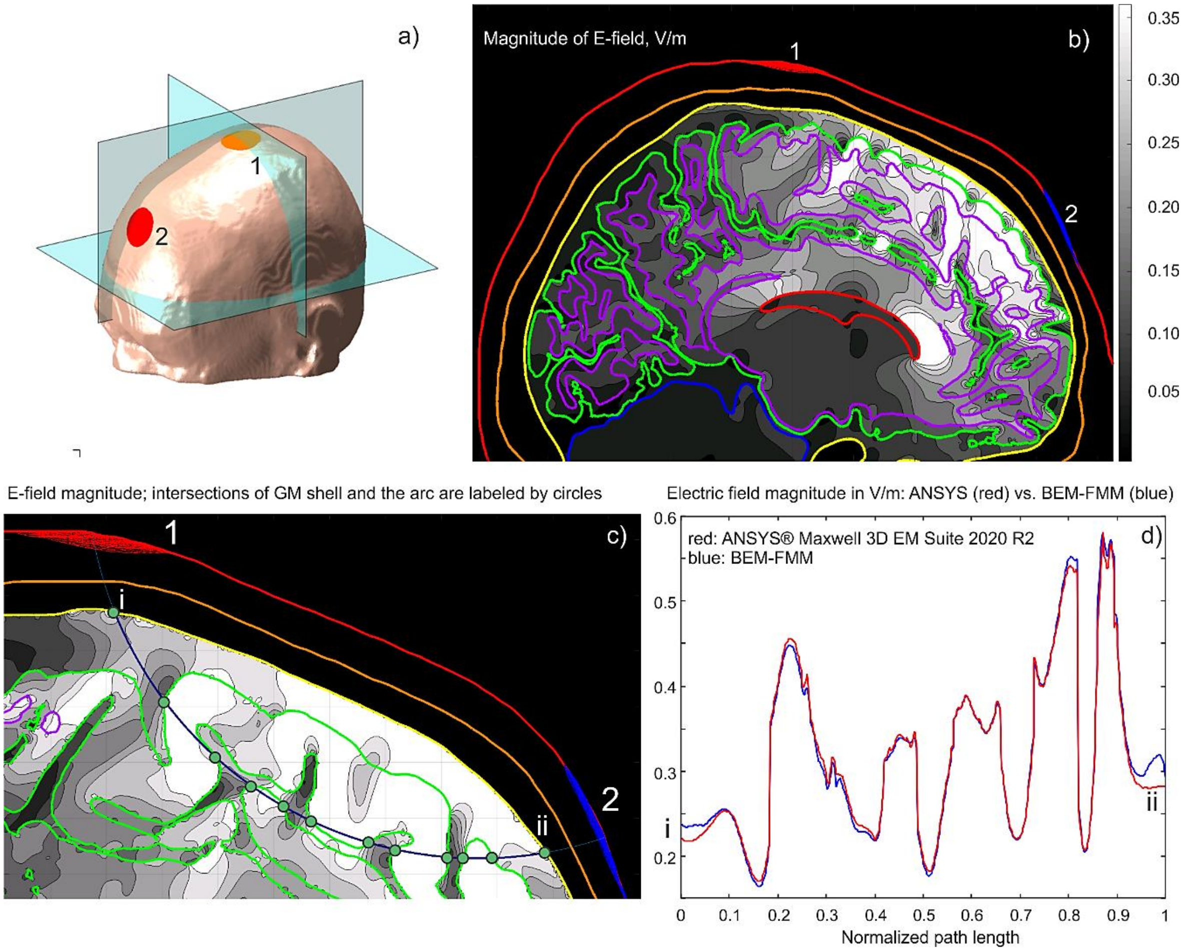


Fig. 50. a) – Conformal electrodes with a fronto–medial placement for Connectome subject #110411; b) – Electric field magnitude in the intracranial volume in the sagittal plane; c) – Electric field magnitude in a plane containing an arc connecting electrode centers. Tissue intersections (CSF and GM) are shown by green dots. d) – Comparison of electric field magnitude along the arc connecting electrode centers with Ansys FEM software.

V.D. Results – Application Examples

V.D.1. DBS electrodes in a multicompartment model: Field and potential distributions

A multi-compartment head-body model with a fully implanted DBS device and DBS electrodes (Fig. 51a) was constructed based on a realistic device configuration extracted from CT images of a patient at Northwestern University, Chicago IL (Figs. 51b,c).

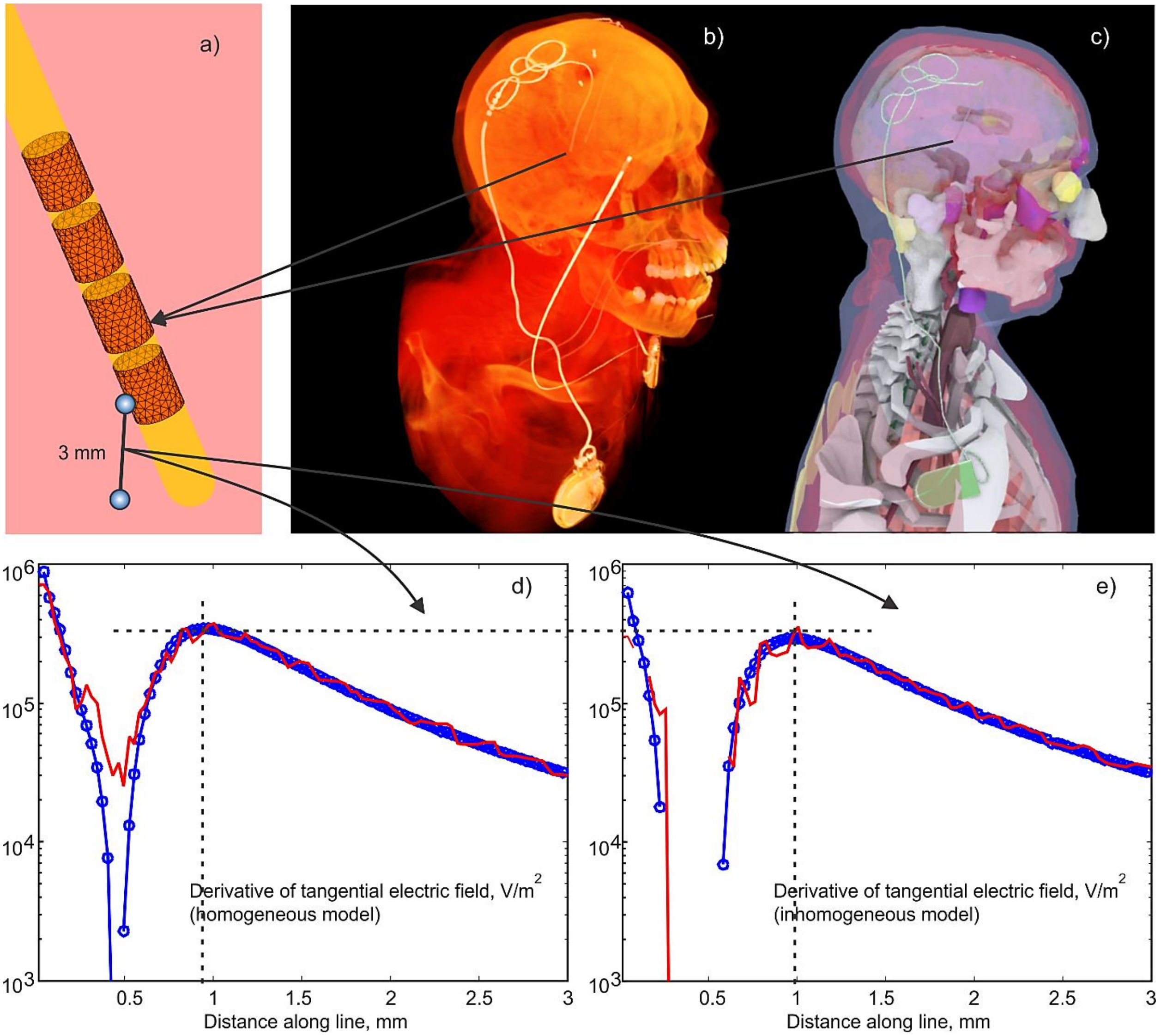


Fig. 51. a) – BEM-FMM Electrode mesh assembly around insulating urethane lead. Active electrode is assigned –1 V vs IPG (held at zero voltage), and no specific voltage is assigned to other electrodes (floating condition); b) – CT of insulating lead/IPG with the electrodes; the second electrode relates to another device. c,d) – IPG Device with the lead and electrodes embedded into the CAD model; d,e) – magnitude of the derivative of the tangential electric field (the second derivative of the electric potential) along a small fraction of the polyline with the length of 3 mm shown in Fig. 51a which starts in the middle of the active electrode held at –1 V for homogeneous (d) and inhomogeneous (e) head/torso model, respectively. Ansys Maxwell FEM solution is marked red; BEM-FMM solution is marked blue.

A head model was further combined with a torso of another subject having multiple tissue objects to additionally model the fields close to the IPG when necessary. The device trajectory was manually segmented, and a 3D model of the implant was constructed and registered to a heterogeneous human body model consisting of 77 individual head/upper torso compartments. The DBS electrode is a Medtronic 3389 electrode (Medtronic$^{®}$ Inc, Minneapolis, MN, USA) implanted into the thalamus volume.

The resulting surface-based CAD model was imported into FEM software Ansys$^{®}$ Maxwell 3D Electromagnetics Suite 2019 R1 and assigned the corresponding conductivity values adopted from [83]. The identical model was used in the BEM-FMM algorithm.

Fig. 51a shows the BEM-FMM electrode mesh assembly around the insulating urethane lead (assigned zero conductivity). Every electrode is assigned an independent voltage vs IPG (which is held at zero volts) following Section V.B.2, or no specific voltage is assigned to some electrodes (the floating condition, Section V.B.4).

Fig. 51b shows an original CT image of the IPG device with embedded lead and electrode assembly. Fig. 51c shows the resulting multi-tissue human model with the registered lead and electrodes. Figs. 51d,e show the activation function, i.e. the magnitude of the derivative of the tangential electric field (second derivative of the electric potential [99],[143]-[148]) along a small fraction of a polyline with the length of 3 mm shown in Fig. 51a. The polyline itself starts in the middle of the active electrode (#1) held at –1 V for both homogeneous (d) and inhomogeneous (e) head/torso models, respectively, and terminates at the IPG. All other electrodes are floating. The Ansys Maxwell FEM solution is marked in red and the BEM-FMM solution is marked in blue in both plots.

The ANSYS FEM solution with 7 adaptive passes executes in approximately 1 h 36 min using a PowerEdge R815 4 AMD Opteron 6378/32 cores 2.4 GHz Windows S 2016 server using the high-performance parallel computing (HPC) option of ANSYS. The BEM-FMM solution with 50 iterations, sixteen neighbor integrals, and a relative residual of 10^{-6} executes in approximately 2.2 min using an Intel Xeon E5-2683 v4 CPU/48 cores 2.1 GHz Windows S 2016 server and MATLAB$^{®}$ 2020b platform. This relatively large execution time (the model has only 0.4 M facets) is due to (i) a larger number of iterations needed and; (ii) a larger number of neighbor analytical integrals (128) needed in the solution. Such a number is found to be important because the model has facets of very different sizes.

For the polyline connecting the center of the active electrode and the IPG, the least-squares error for the electric potential between the two solutions is 2.3%. The least-squares error in the tangential component of the electric field along the polyline is 2.0% despite the fact that both solutions change abruptly when passing through the tissue interfaces. Overall, both solutions for the electric potentials and the electric fields are hardly distinguishable visually. Similar results have been obtained for the homogeneous model.

Close to the electrode surface in Fig. 51a, the tangential electric fields along the polyline are similar to each other for both numerical solutions. However, the tangential-field derivatives, which are proportional to the activating function of an axonal fiber [99],[143]-[148], are not. Figs. 51d,e show the magnitude of the derivative of the tangential electric field (the second derivative of the electric potential) computed along a small fraction of the polyline (Fig. 51a). The ANSYS FEM solution is marked red while the BEM-FMM solution is marked blue in both plots. The BEM-FMM solution is much smoother compared to the FEM solution.

V.D.2. Epidural planar ICMS array

A planar rectangular epidural ICMS array of size 5×5×0.5 mm with 100 circular electrodes was embedded into the head model of a 27 year old healthy female above the $\mathrm{M1_{HAND}}$ area of the left hemisphere. The surface meshes were generated with T1 and T2 images (Siemens Skyra fit 3 T, Max Planck Inst., Leipzig) using the "headreco" pipeline from SimNIBS 3.1. The default SimNIBS conductivity values have been used. Additionally, a dura (combined with arachnoid) mater shell was constructed with a thickness of 1.21 mm and a conductivity of 0.1 S/m by expanding the CSF shell and reducing the overall skull thickness accordingly. Fig. 52 shows the topology of the surface mesh superimposed onto T1 NifTI data of the subject in the sagittal plane along with the synthetic dura shell. Fig. 53a shows the array electrode assembly and dimensions.

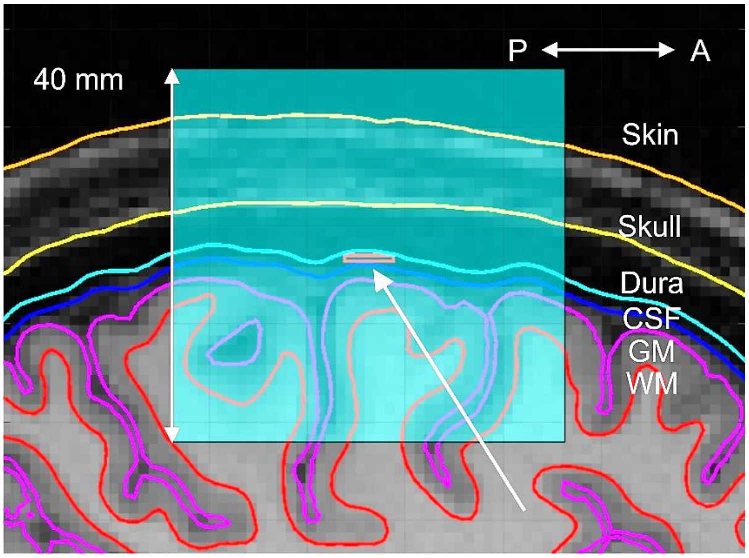


Fig. 52. Topology of the surface mesh (sagittal plane) along with the synthetic dura shell superimposed onto T1 NifTI data of the subject and the cross-section of the embedded array. The array position is indicated by a white arrow. The observation plane(s) (40×40 mm, also used in Fig. 54 below) is indicated by the blue transparent rectangle.

The array case in Fig. 53a is made insulating. The electrodes are driven as follows. Electrode #91 is assigned +1 V, electrode #10 is assigned – 1 V, and all other electrodes are assigned zero volts. After appropriate mesh refinement and adding the dura mater, the total number of faces in the model approached 2.55 M. The solution time of the BEM-FMM with four neighbor integrals was approximately 200 secs for 20 iterations and the relative residual of 10^{-3} on a 2.1 GHz Windows server (MATLAB 2020b); the required RAM did not exceed 11 Gbytes.

Fig. 53b shows the magnitude of the resulting total electric field at a mid-layer between the gray and white matter interfaces. The array is targeting the $\mathrm{M1}_{\mathrm{HAND}}$ area (Ω-shaped in the figure) of the left hemisphere. Simulation results reveal that, for the given array position and assembly, reasonable values of the electric field at the mid-layer (approximately corresponding to cortical layers 2,3) will be generated. These values approach 1 V/m, despite the small array size. Next, the array fields were computed just outside the white matter shell (approximately corresponding to cortical layer 5). The field values there approach 0.6 V/m, and the field becomes more disperse in the median-lateral direction. The distance between the array and the white matter shell is 8 mm. The same array could be used for iEEG recordings.

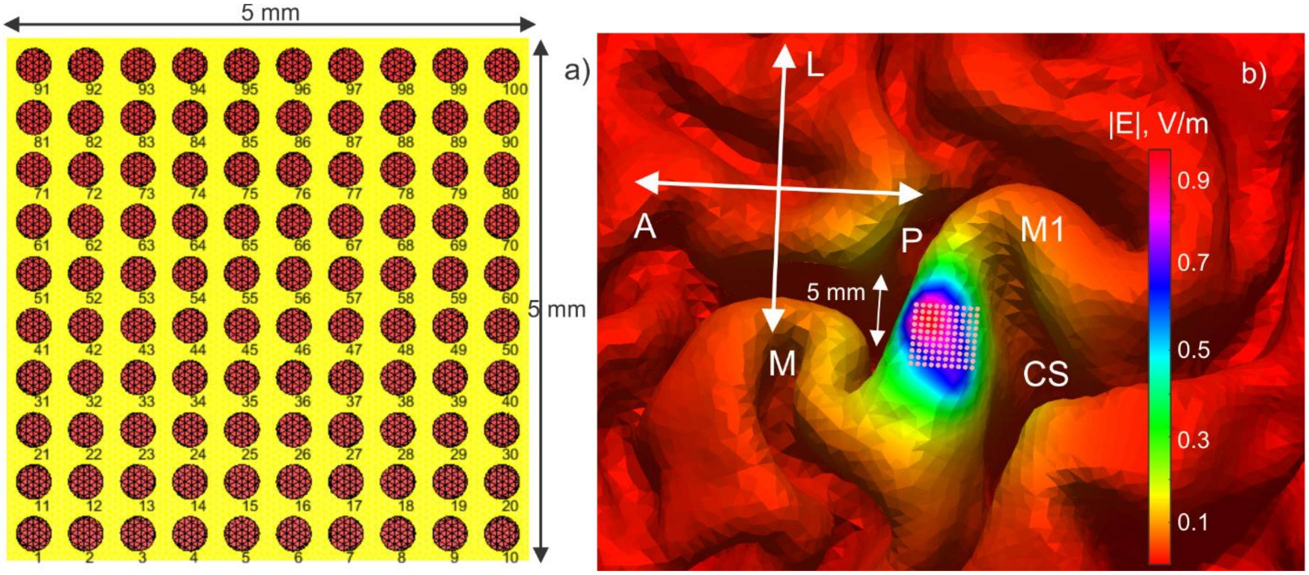


Fig. 53. a) – Planar ICMS array assembly. The insulating case is yellow. The array thickness is 0.5 mm. b) – Magnitude of the electric field at the mid-layer between the gray and white matter shells. The array is targeting the $M1_{HAND}$ area of the left hemisphere.

To examine the array field distribution into the depth, Figs. 54a and 54b demonstrate spatial distributions of the magnitude of the electric field in the coronal and sagittal planes, respectively, centered at the array center. The observation domain is 40×40 mm in every case. One may note a blocking effect of the cerebrospinal fluid, which is clearly seen in Fig. 54b.

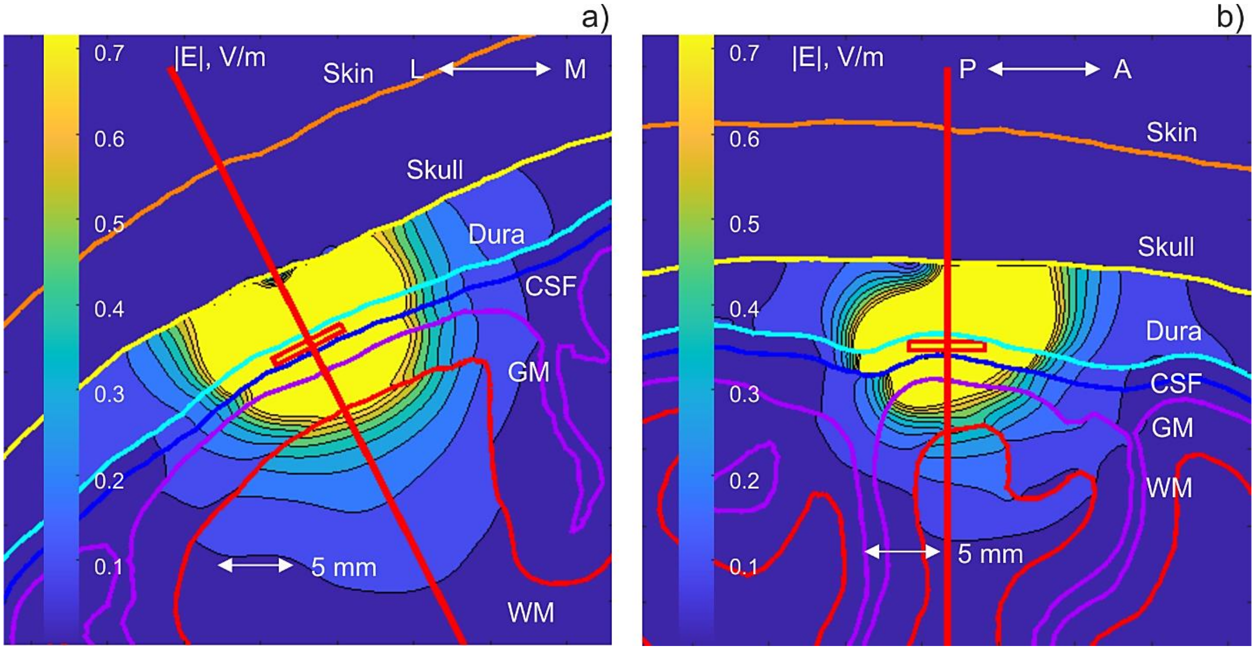


Fig. 54. a), b) – Magnitude of the total electric field in the coronal and sagittal planes, respectively. Both planes are centered at the array center, whose cross-section is shown in red.

V.E. Discussion

We formulated the BEM-FMM solution for electric stimulation by implementing the current and charge conservation laws in a weak form, i.e. by adding the corresponding scalar functional to be minimized to the right-hand side of the integral equation.

We further provide necessary modifications of the governing charge-based BEM equations to model voltage, shunt, current, and floating electrodes. After obtaining the solution for the surface charge density [1], it is possible to precisely obtain electric fields normal to the cortical surfaces (or any other interfaces) in real time and without postprocessing.

The present validation and application examples demonstrate that the BEM-FMM approach may be a viable tool for modeling electric brain stimulation with macro- and microelectrodes. While the concepts of the voltage, shunt, and floating electrodes are common, the model of the true current electrodes is rarely used and not implemented in any software packages known to the authors. On the other hand, it may be useful for the surface electrodes with sponges as well for modeling neurophysiological recordings via the reciprocity principle [138],[139]. The quasistatic reciprocity theorem allows us to interrelate the original field of intracortical sources and an auxiliary field obtained by injecting external electric current at the same EEG electrodes.

The BEM-FMM execution times are ~ 3 sec per iteration for the realistic head model with 0.86 M facets and ~10 sec per iteration for the model with 2.55 M facets using a 2.1 GHz server. Thus, the simulation time scales nearly linearly with increasing mesh size for the surface-based model, which is an advantage compared to FEM [64],[66]. In consequence, we observed that the memory requirements of the BEM-FMM may be twofold or even sixfold less than for the FEM (Section V.C.3; see also Section V.D.2) when the same problem is solved. It should be noted that the differences between BEM-FMM and FEM are not solely originating from inaccuracies of the BEM-FMM. They also result from the approximate nature of the FEM model due to coarse discretizations as shown by Gomez et al. 2020 [51] in the framework of TMS. In this case, the FEM solution should not be interpreted as a reference solution. Rather, it is a comparative solution.

The BEM-FMM computational speeds reported in this study are slower than those for TMS modeling [1] due to inclusion of a larger number of neighbor potential integrals and due to the fact that a typical electrode-based solution requires around 20-25 iterations to resolve the field singularities at the electrodes reasonably well. It is well known that both the surface charge density and the electric field are singular at the electrode rims [77].

For the DBS application example in Section V.D.2, one observes that the FEM solution on a submillimeter scale shows numerical noise for both the homogeneous and the inhomogeneous models. In this respect, BEM-FMM offers the possibility to calculate much more accurate electric field profiles on a small scale without the necessity of volumetric mesh refinement, which plays a crucial role especially when these models are coupled with neuron models. A numerically unconstrained spatial resolution was further observed for the ICMS modeling solution in Section V.D.2.

The present approach is not without its limitations. While the memory requirements of the BEM-FMM are more advantageous, the FMM algorithm used in this study requires heavy parallelization. All results reported above have been obtained on two servers with 40 and 48 cores, respectively. Our experience indicates that the number of cores is much more critical than the amount of RAM. The low RAM requirement and the good scalability in terms of CPU cores opens the possibility of an efficient GPU implementation which may reduce the computing time significantly. Additionally, the BEM-FMM cannot yet account for anisotropic tissue conductivities as opposed to the FEM.

V.F. Conclusions

In this chapter, the boundary element fast multipole method (BEM-FMM) was formulated as an alternative to the finite element method (FEM) for modeling electrode-induced fields in the brain. The suggested method implementation is based on two properly coupled integral equations: one for tissue interfaces and another for the electrodes. It supports voltage, shunt, floating, and current electrodes.

Comparison with the analytical solution and with the commercial FEM software demonstrated high accuracy of the BEM-FMM algorithm for various models, both in the vicinity of the electrodes as well as in the entire tissue volume. The method can be easily combined with existing head modeling pipelines such as headreco or mri2mesh by using the segmented and meshed surfaces. The simulation time is similar to fast TES FEM solvers such as SimNIBS and exceeds the speed of the commercial solvers (Ansys Maxwell and COMSOL) by approximately two orders of magnitude.

Two application examples demonstrated the advantage of the BEM-FMM, namely its potential applicability to model multiscale problems with submillimeter (and finer if necessary) resolution.

VI. TES Toolkit Application Example: Multilayer Skull Effect on TES Simulations

This chapter was excerpted from [153], a submission to the 10th International IEEE EMBS Conference on Neural Engineering

VI.A. Introduction

Transcranial magnetic stimulation (TMS) and transcranial electrical stimulation (TES) are noninvasive neurostimulation methods in which electrical currents are induced via an electromagnetic coil (TMS) or injected via electrodes placed on the scalp (TES). Due to the complicated geometry of the brain itself, coupled with insulating properties of the skull and conducting properties of the cerebrospinal fluid (CSF), electric field behavior within the brain is unintuitive. Computational electromagnetic modeling methods must be applied to precisely predict the electric field induced by either of these stimulation methods; common methods include the finite element method (FEM) and finite difference method (FDM). The human head models employed for this task often lack data on two major tissue types that may affect the total electric field in the cortical volume: the cancellous bone of the skull and the dura mater [50][149]. The high-definition MIDA head model [71] was employed for computational modeling of these tissues' effects on the electric fields induced by both TMS and TES using the boundary element fast multipole method (BEM-FMM).

VI.B. Background

VI.B.1. MIDA Model

The surface-based MIDA model developed by Iacono et al. [71] comprises 116 triangular surface meshes with a total of 11 M facets, making it one of the most detailed head models currently available [151]. For example, the superior region of the skull is composed of three distinct surfaces: two for the exterior and interior layers of cortical bone, respectively, and one for the cancellous bone (the skull diploë) between those layers of cortical bone. The model also contains explicit segmentations of adipose tissue between the skin and skull; the dura mater; major nerves, veins, and arteries; as well as gray matter (GM), white matter (WM), cerebrospinal fluid (CSF), and other large tissues. The detailed anatomical representation of the head renders the MIDA model an ideal candidate for exploration of the effects of simplifying assumptions on solutions to simulations of noninvasive transcranial stimulation methods.

VI.B.2. Boundary Element Fast Multipole Method

The Boundary Element Fast Multipole Method (BEM-FMM), described in [49] and available at [54][55][56], is a computational alternative to the Finite Element Method (FEM) or Finite Difference Method (FDM). Based on charges that accumulate at interfaces between tissues of different electrical conductivity, the BEM-FMM operates directly on surface meshes rather than operating on volumetric meshes generated from those surface meshes. For extremely detailed meshes like the MIDA model, the BEM-FMM is likely to require two-fold to six-fold less memory and execute just as fast (or faster) while maintaining equivalent accuracy [50],[51]. Additionally, observation points for various field quantities may be placed arbitrarily close to surfaces since the field resolution is unconstrained by a volumetric FEM/FDM grid. To the authors' knowledge, this is the first application of the BEM-FMM to modeling of TES.

VI.C. Materials and Methods

VI.C.1. Numerical verification of BEM-FMM results via Sim4Life results for a TMS test case

The first task was to establish accuracy of the BEM-FMM algorithm against a commercial FEM magneto quasi-static solver (Sim4Life [152]). The test setup used a single-loop coil with a radius of 5 mm, cross-sectional radius of 1 mm, and 16 elementary current filaments. The coil was placed 10 mm above the skin surface as shown in Fig. 55. It was driven with current amplitude I_0 = 5 kA and dI/dt = 94 A/µs, representing a sinusoidal current of frequency 3 kHz at the moment of maximum dI/dt. For this test and all subsequent simulations, conductivities were assigned to the MIDA model's tissues according to the IT'IS Foundation's Database of Tissue Properties [83].

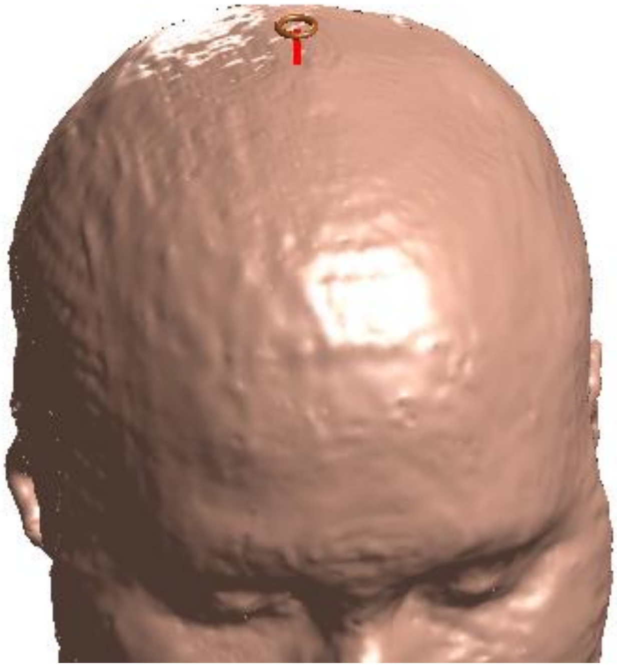


Fig. 55. Test setup for the comparison of results between the BEM-FMM and Sim4Life

The test setup was simulated using both Sim4Life and BEM-FMM. The induced electric field was calculated throughout the head volume on a grid with 0.5 mm resolution. The absolute error at each observation point was calculated as

$$\Delta E = \left\| \vec{E}_B - \vec{E}_S \right\|, \qquad (43)$$

where $\vec{E}_B$ is the field calculated using BEM-FMM, $\vec{E}_S$ is the field calculated using Sim4Life, and $\|*\|$ denotes the Euclidean vector norm.

VI.C.2. Effect of skull simplification on TMS simulation

The next task was to evaluate the error introduced by neglecting the effects of certain large tissues on the TMS E-field. To eliminate the contribution of the dura mater to the TMS E-field, it was assigned a conductivity equal to that of cerebrospinal fluid. Similarly, the skull diploë's conductivity was replaced by that of cortical bone, approximating the case (common to other head models) where the entire skull is treated as homogeneous cortical bone.

The test setup is shown in Fig. 56 below. The motor hand area was targeted for stimulation by a MagVenture MRi-B91 figure-eight coil model, which was positioned according to several simple geometric rules: (i) The coil's centerline is approximately perpendicular to the surface of the gyral crown at the targeted region; (ii) The dominant incident E-field direction is approximately perpendicular to the sulcal walls of the targeted region; and (iii) The coil's centerline intersects the skin surface approximately 9 mm from the coil's origin.

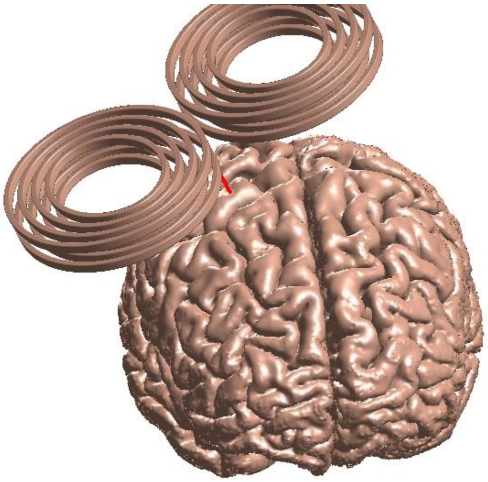


Fig. 56. Test setup for evaluation of the impact of the diploë and dura mater on TMS simulations. The coil used for this test was an MRi-B91 coil placed roughly above the motor hand area.

Two simulations were run with this test setup: the base case with all tissues present, and the test case where both the skull diploë and dura mater were neglected. In each case, the total electric field magnitude was evaluated on a grid with 0.5 mm resolution in the sagittal plane where the coil's centerline intersected the GM surface. Error between the test case and the base case was computed as described in Section VI.C.1.

VI.C.3. Effect of skull simplification on TES simulation

The final task was to evaluate the error introduced by neglecting the effects of the dura mater and the skull diploë on the TES-induced E-field. Sixty-four electrodes were placed on the skin surface in an approximate 10-10 configuration as shown in Fig. 57. Each electrode's E-field was computed independently at the gray matter and white matter surfaces by assigning the currently-active electrode a potential of 1 V and assigning all inactive electrodes potentials of 0 V. The set of 64 E-field solutions was computed for two cases. In the base case, the skull diploë and dura were assigned their own respective properties. In the test case, the skull diploë and dura mater were subsumed into the cortical bone of the skull. The white matter and gray matter were used as observation surfaces because a single sagittal plane (as applied in the prior sections) would not capture all effects of all electrodes.

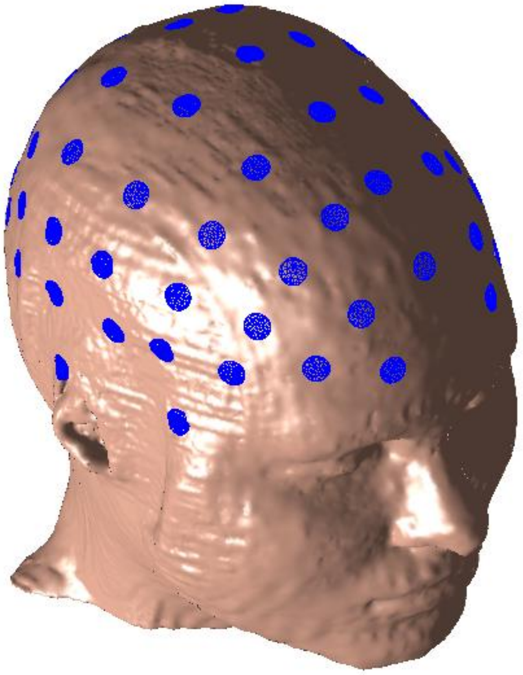


Fig. 57. Test setup for the TES dura/diploë sensitivity test. 5-mm-radius electrodes were imprinted on the MIDA model's epidermis mesh in an approximate 10-10 configuration.

For each electrode, an attempt was made to find a coefficient α that scales the vector of test-case E-field magnitudes $\boldsymbol{E}_{test}$ at the WM or GM surfaces to match the vector of base-case E-field magnitudes $\boldsymbol{E}_{base}$. This scalar was chosen to minimize the error expression,

$$\Delta E = \|\alpha \boldsymbol{E}_{test} - \boldsymbol{E}_{base}\| / \|\boldsymbol{E}_{base}\|. \tag{44}$$

VI.D. Results

VI.D.1. BEM-FMM validation via Sim4Life

Fig. 58 below shows the magnitude of the electric field in the sagittal cross section coplanar with the single-loop coil's centerline as evaluated by BEM-FMM and Sim4Life, followed by the magnitude of the difference between the two fields.

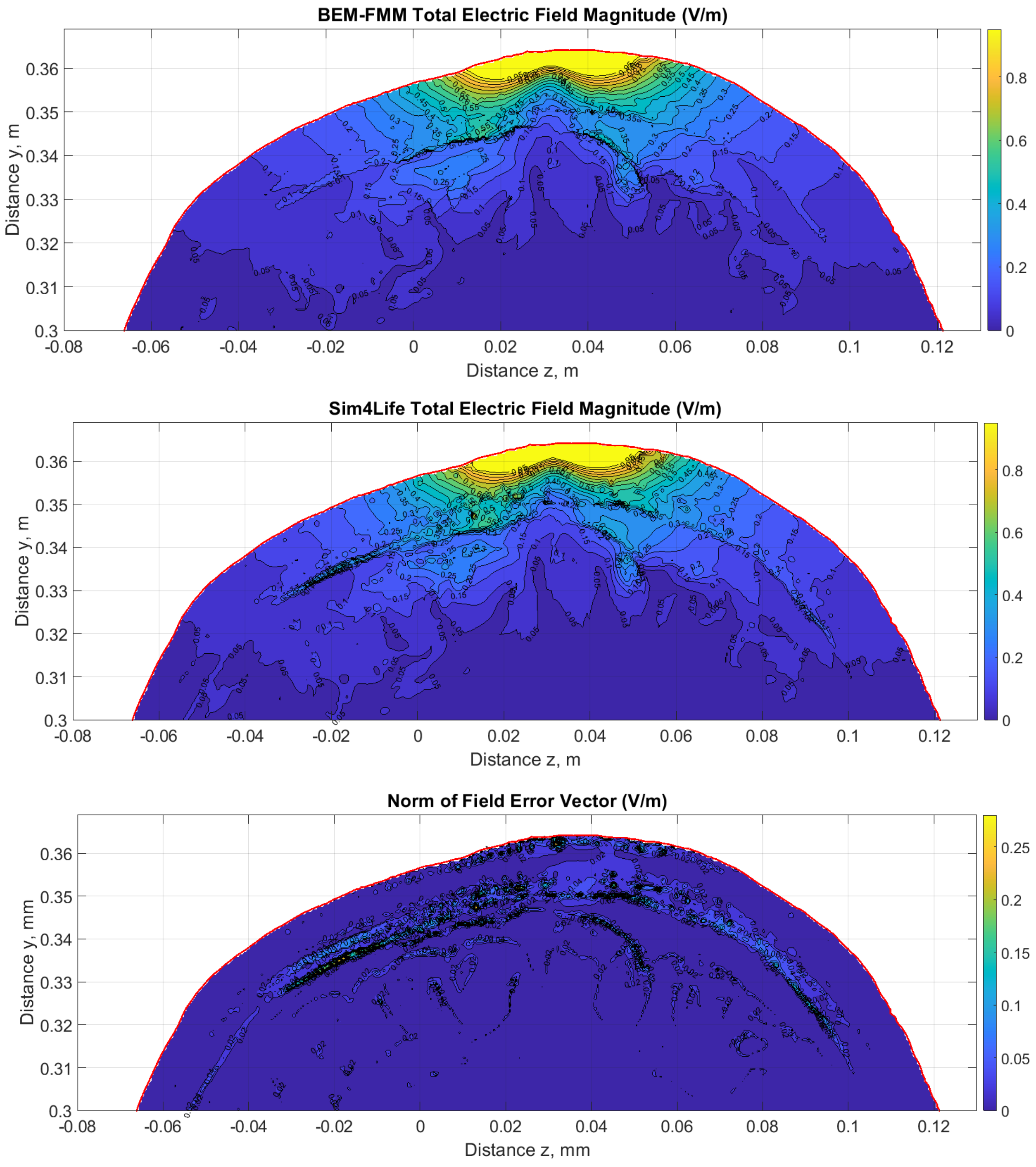


Fig. 58. E-field predicted by BEM-FMM (top), E-field predicted by Sim4Life (center), and magnitude of absolute error vector between the two cases (bottom). E-field units are V/m.

VI.D.2. Effect of skull simplification on TMS simulations

Fig. 59 below shows results in the sagittal cross-section where the coil's centerline intersects the gray matter surface for the full and simplified models. The magnitude of the total

E-field is shown for the full model and the simplified model, along with the magnitude of the absolute difference in the E-field between the two cases.

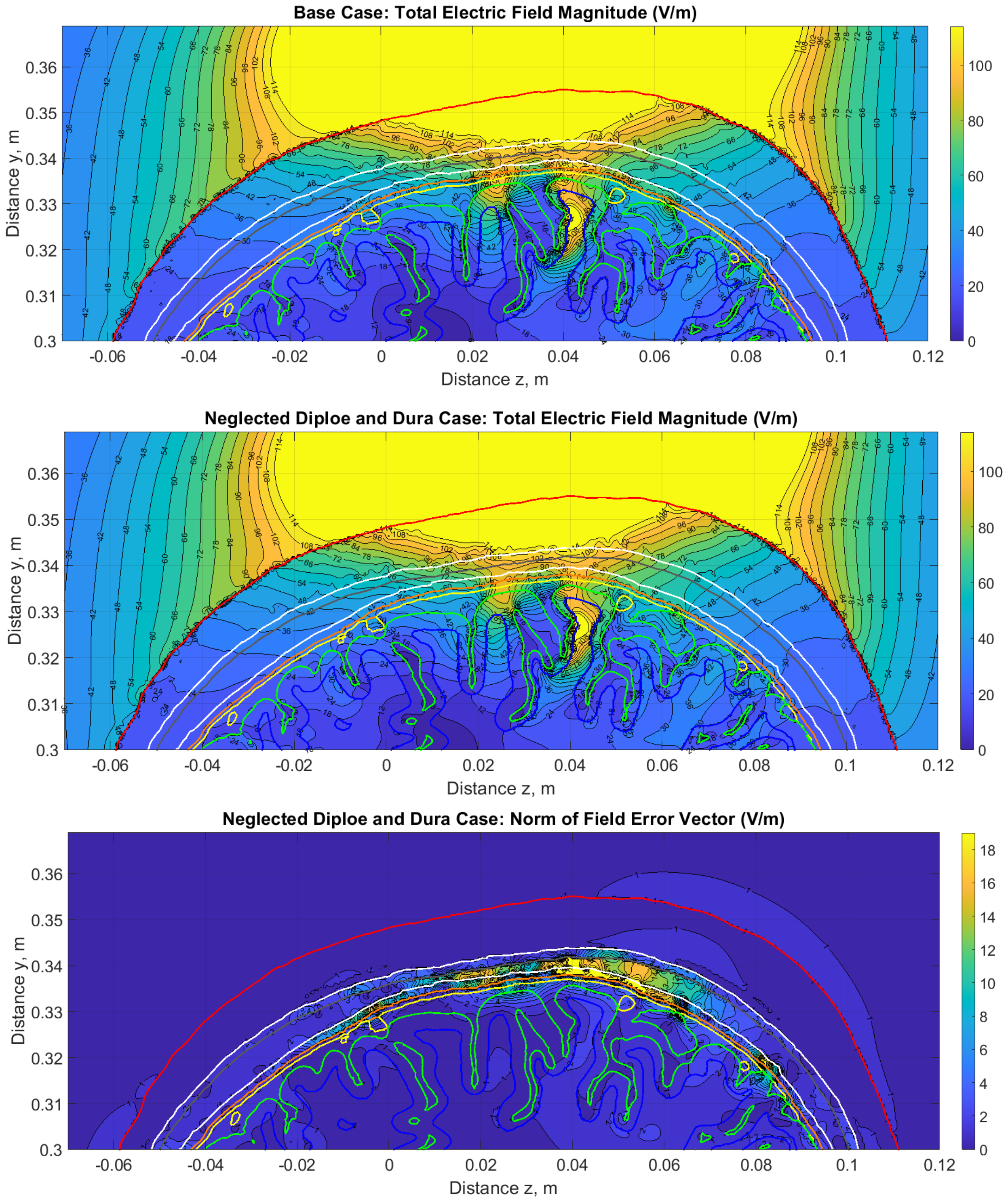


Fig. 59. TMS E-field magnitude evaluated for the base case (top), E-field magnitude evaluated for the simplified case (center), and magnitude of absolute difference in the E-fields for those cases (bottom). E-field units are V/m.

VI.D.3. Effect of skull simplification on TES simulations

The electrode-specific coefficients for the white matter surface ranged from 0.5 to 0.9 with a clear and reasonable dependence on electrode positions. The electrode-specific coefficients for the gray matter surface ranged from 0.5 to 0.8 and exhibited a visually similar dependence on electrode position to the white matter case. Aggregate error at the white matter surface between the scaled E-field predicted by the simplified model and the E-field predicted by the full model was between 6% and 13%, with a single outlier at 19%. The aggregate error at the gray matter surface ranged from 21% to 34% with no apparent outliers.

Fig. 60 below shows mappings of the electrode-specific coefficients α onto the skin surface, as well as the least-squares error that results when those coefficients are used to scale the E-field from the simplified model to match the E-field from the full model. The coefficients measure the change in absolute overall field strength, while the least-squares errors measure the change in spatial distribution of the relative electric field strength.

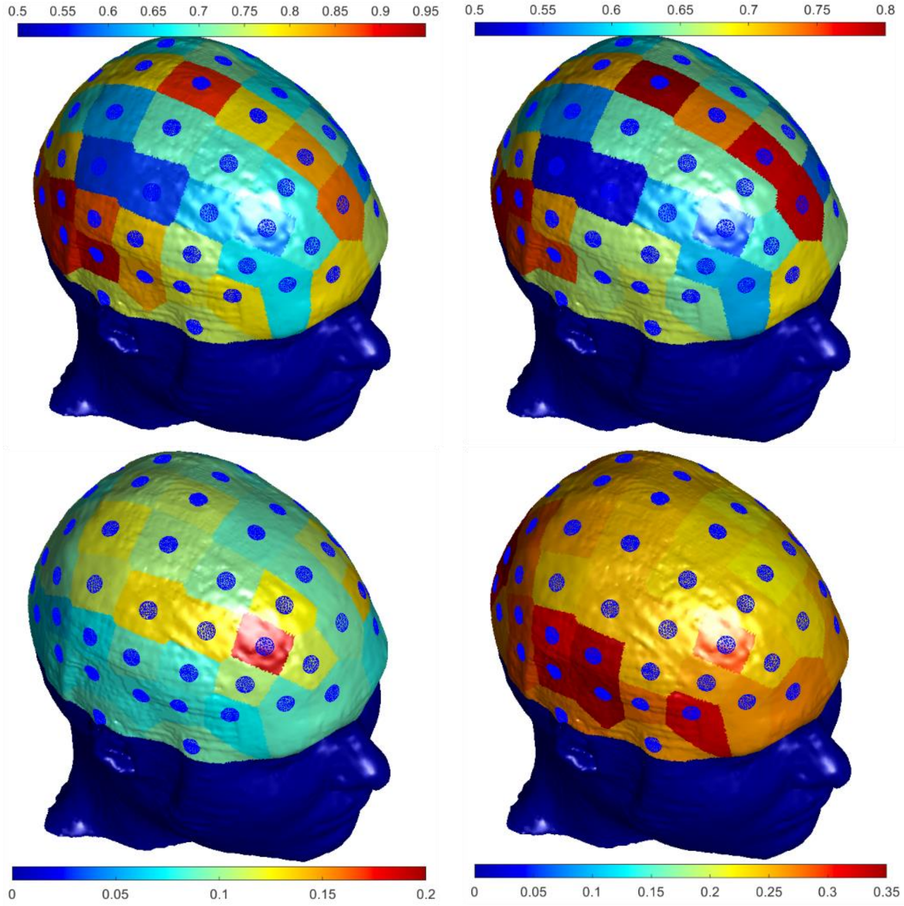


Fig. 60. Top left: Corrective coefficients to scale the TES E-field from the simplified model to match the E-field from the full model at the WM surface, mapped to the regions of the skin surface dominated by their respective electrodes. The maximum coefficient is 0.92 (dark red), and the minimum is 0.5 (dark blue). The lower half of the head, free of electrodes, is assigned a coefficient of 0.5. Top right: Corrective coefficients for the gray matter surface, ranging from 0.5 to 0.8. Bottom left: Aggregate error post-scaling for the E-field at the WM surface, ranging from 6 % (pale blue) to 19% (dark red). Bottom right: Aggregate error post-scaling for the E-field at the GM surface, ranging from 21% to 34%.

VI.E. Discussion

The error between the BEM-FMM simulation and the Sim4Life simulation appears to be fully restricted to regions very close to tissue boundaries. This behavior is expected, since nearby E-fields vary quickly due to charge accumulation at those boundaries. For a first-order FEM approach, the electric field is constant within a single element, so further analysis is necessary to obtain higher-order derivatives (e.g., evaluating potential values on a regular grid and calculating the derivatives using finite differences) [154]. Thus, FEM/FDM typically fail in these cases due to limited spatial resolution, while the quick field variation is accurately captured by the BEM-FMM. Within volumes sufficiently distant from boundaries, the two methods agree well, so we are confident in predictions based on BEM-FMM results.

For the TMS skull simplification test, the error in the E-field is almost fully constrained to the skull. Absolute differences in the field within the WM do not exceed 3 V/m (less than 3% relative error), and absolute differences in the field within the GM do not exceed 5 V/m (less than 5% relative error) except in locations close to (or touching) the outer CSF boundary. Since the maximum field magnitude in the joint GM/WM volume exceeds 100 V/m, these errors are sufficiently small to claim that the dura mater and skull diploë do not heavily affect the electric field induced within the brain by TMS. This is a reasonable result because the TMS E-field is induced by a time-varying magnetic field, which is not affected by intervening tissues between the coil and brain.

TES E-fields, on the other hand, appear to be very strongly affected by properties of the skull diploë and the dura mater. The fields measured at the WM surface for the base case are typically between 50% and 90% as strong as the fields measured for the test case, and the fields measured at the GM surface for the base case are typically between 50% and 80%. Based on the post-scaling field error at the WM surface, the field generated by the simplified model at that surface may be linearly scalable to the full-model field at that surface. Based on the post-scaling field error at the GM surface, however, the field computed at the GM surface by the simplified case is likely not linearly scalable to the field computed at the GM surface by the simplified model. The large change in the E-field observed when neglecting the diploë and dura is reasonable because these tissues' properties should directly affect the path of current from the active electrode to the brain and the return electrodes.

VI.F. Conclusion

We have demonstrated that the Boundary Element Fast Multipole Method obtains good agreement with Sim4Life for the MIDA model, that transcranial magnetic stimulation simulations likely are not strongly affected by neglecting the cancellous bone of the skull or the dura mater, and that transcranial electric stimulation simulations likely are strongly affected by both materials. We have further investigated a simple method of compensation for neglect of these tissues, namely applying electrode-specific coefficients to the E-fields at the gray matter surface and the white matter surface separately. While this method may be applicable to the E-field at the white matter surface, it is less applicable (or not applicable) to the fields at the gray matter surface.

To further explore and validate these results, the experiment should be undertaken with a larger set of highly detailed human head models. It would also be worthwhile to explore the interactions that cause the post-scaling error to be relatively high for the gray matter surface and relatively low for the white matter surface. Further attention should be given to simplifications other than those presented here – for example, the treatment of the skin and fat as the same uniform tissue, or an investigation of the results when the dura is subsumed into CSF instead of cortical bone. Finally, investigation of these results' frequency-dependence is also recommended.

References

[1] Makarov SN, Wartman WA, Daneshzand M, Fujimoto K, Raij T, Nummenmaa A. A software toolkit for TMS electric-field modeling with boundary element fast multipole method: An efficient MATLAB implementation. *J Neural Eng*. 2020 Aug 4;17(4):046023. doi: 10.1088/1741-2552/ab85b3.

[2] Makarov SN, Golestanirad L., Wartman WA, Nguyen BT, Noetcher GM, Ahveninen JP, Fujimoto K, Raij T, Weise K, Nummenmaa A. Boundary Element Fast Multipole Method for Modeling Electrical Brain Stimulation with Voltage and Current Electrodes. *J Neural Eng.*, under review

[3] National Institutes of Mental Health (NIMH). *Major Depression*. Feb.2019. Online: https://www.nimh.nih.gov/health/statistics/major-depression.shtml.

[4] Stern AP. *Transcranial magnetic stimulation (TMS): Hope for stubborn depression*. Harvard Health Publishing. Feb. 23 2018. Online: https://www.health.harvard.edu/blog/transcranial-magnetic-stimulation-for-depression-2018022313335

[5] 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.

[6] COVID's Mental-health toll: how scientists are tracking a surge in depression. *Nature Briefing*. 03 Feb. 2021. doi: https://doi.org/10.1038/d41586-021-00175-z. Online: https://www.nature.com/articles/d41586-021-00175-z

[7] FDA 510(k)/De Novo Clearance and Premarket Approval Applications: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/search/default.cfm, accessed Nov. 2020.

[8] Jack CR Jr, Thompson RM, Butts RK, Sharbrough FW, Kelly PJ, Hanson DP, Riederer SJ, Ehman RL, Hangiandreou NJ, Cascino GD. Sensory motor cortex: correlation of presurgical mapping with functional MR imaging and invasive cortical mapping. *Radiology*. 1994 Jan;190(1):85-92. doi: 10.1148/radiology.190.1.8259434.

[9] Grande KM, Ihnen SKZ, Arya R. Electrical Stimulation Mapping of Brain Function: A Comparison of Subdural Electrodes and Stereo-EEG. *Front Hum Neurosci*. 2020 Dec 7;14:611291. doi: 10.3389/fnhum.2020.611291.

[10] Collavini S, Fernández Corazza M, Oddo S, Princich JP, Kochen S, Muravchik CH. Improvements on spatial coverage and focality of deep brain stimulation in pre-surgical epilepsy mapping. *J Neural Eng*. 2021 Feb 12. doi: 10.1088/1741-2552/abe5b9.

[11] Crone NE, Miglioretti DL, Gordon B, Lesser RP. Functional mapping of human sensorimotor cortex with electrocorticographic spectral analysis. II. Event-related synchronization in the gamma band. *Brain*. 1998 Dec;121 (Pt 12):2301-15. doi:

10.1093/brain/121.12.2301. PMID: 9874481.

[12] Ramsey NF, van de Heuvel MP, Kho KH, Leijten FS. Towards human BCI applications based on cognitive brain systems: an investigation of neural signals recorded from the dorsolateral prefrontal cortex. *IEEE Trans Neural Syst Rehabil Eng.* 2006 Jun;14(2):214-7. doi: 10.1109/TNSRE.2006.875582.

[13] Schalk G, Miller KJ, Anderson NR, Wilson JA, Smyth MD, Ojemann JG, Moran DW, Wolpaw JR, Leuthardt EC. Two-dimensional movement control using electrocorticographic signals in humans. *J Neural Eng*. 2008 Mar;5(1):75-84. doi: 10.1088/1741-2560/5/1/008.

[14] Miller KJ, Schalk G, Fetz EE, den Nijs M, Ojemann JG, Rao RP. Cortical activity during motor execution, motor imagery, and imagery-based online feedback. *Proc Natl Acad Sci U S A*. 2010 Mar 2;107(9):4430-5. doi: 10.1073/pnas.0913697107. Erratum in: Proc Natl Acad Sci U S A. 2010 Apr 13;107(15):7113.

[15] Yanagisawa T, Hirata M, Saitoh Y, Kishima H, Matsushita K, Goto T, Fukuma R, Yokoi H, Kamitani Y, Yoshimine T. Electrocorticographic control of a prosthetic arm in paralyzed patients. *Ann Neurol*. 2012 Mar;71(3):353-61. doi: 10.1002/ana.22613.

[16] Vansteensel MJ, Pels EGM, Bleichner MG, Branco MP, Denison T, Freudenburg ZV, Gosselaar P, Leinders S, Ottens TH, Van Den Boom MA, Van Rijen PC, Aarnoutse EJ, Ramsey NF. Fully Implanted Brain-Computer Interface in a Locked-In Patient with ALS. *N Engl J Med*. 2016 Nov 24;375(21):2060-2066. doi: 10.1056/NEJMoa1608085.

[17] Kobayashi M, Pascual-Leone A. Transcranial magnetic stimulation in neurology. *Lancet Neurol*. 2003 Mar;2(3):145-56. PMID: 12849236.

[18] Rossi S, Hallett M, Rossini PM, Pascual-Leone A; Safety of TMS Consensus Group. Safety, ethical considerations, and application guidelines for the use of transcranial magnetic stimulation in clinical practice and research. *Clin Neurophysiol*. 2009 Dec;120(12):2008-2039. doi: 10.1016/j.clinph.2009.08.016.

[19] McMullen D. *NIMH Non-Invasive Brain Stimulation E-Field Modeling Workshop*. November 11, 2017. Online: https://www.nimh.nih.gov/news/events/2017/brainstim/nimh-non-invasive-brain-stimulation-e-field-modeling-workshop.shtml.

[20] Bonmassar G, Lee SW, Freeman DK, Polasek M, Fried SI, Gale JT. Microscopic magnetic stimulation of neural tissue. *Nat Commun*. 2012 Jun 26;3:921. doi: 10.1038/ncomms1914.

[21] Lee SW, Fallegger F, Casse BDF, Fried SI. Implantable microcoils for intracortical magnetic stimulation. *Sci. Adv*. (2016) Dec 9;2(12):e1600889. doi: 10.1126/sciadv.1600889.

[22] Lee SW, Fried SI. Enhanced control of cortical pyramidal neurons with micromagnetic stimulation. *IEEE Trans. Neural. Syst. Rehabil. Eng*. (2016) Nov 22. doi: 10.1109/TNSRE.2016.2631446.

[23] Liu A, Vöröslakos M, Kronberg G, Henin S, Krause MR, Huang Y, Opitz A, Mehta A,

Pack CC, Krekelberg B, Berényi A, Parra LC, Melloni L, Devinsky O, Buzsáki G. Immediate neurophysiological effects of transcranial electrical stimulation. *Nat Commun*. 2018 Nov 30;9(1):5092. doi: 10.1038/s41467-018-07233-7. PMID: 30504921; PMCID: PMC6269428.

[24] Mutz J, Vipulananthan V, Carter B, Hurlemann R, Fu CHY, Young AH. Comparative efficacy and acceptability of non-surgical brain stimulation for the acute treatment of major depressive episodes in adults: systematic review and network meta-analysis. *BMJ*. 2019 Mar 27;364:l1079. doi: 10.1136/bmj.l1079.

[25] Karabanov AN, Saturnino GB, Thielscher A, Siebner HR. Can Transcranial Electrical Stimulation Localize Brain Function? *Front Psychol*. 2019; 10: 213. doi: 10.3389/fpsyg.2019.00213.

[26] Kasschau M, Reisner J, Sherman K, Bikson M, Datta A, Charvet LE. Transcranial Direct Current Stimulation Is Feasible for Remotely Supervised Home Delivery in Multiple Sclerosis. *Neuromodulation*. 2016 Dec;19(8):824-831. doi: 10.1111/ner.12430.

[27] Lesser RP, Arroyo S, Crone N, Gordon B. Motor and sensory mapping of the frontal and occipital lobes. Epilepsia. 1998;39 Suppl 4:S69-80. doi: 10.1111/j.1528-1157.1998.tb05127.x. Erratum in: Epilepsia 1998 Aug;39(8):912. PMID: 9637595.

[28] Tarapore PE, Tate MC, Findlay AM, Honma SM, Mizuiri D, Berger MS, Nagarajan SS. Preoperative multimodal motor mapping: a comparison of magnetoencephalography imaging, navigated transcranial magnetic stimulation, and direct cortical stimulation. J Neurosurg. 2012 Aug;117(2):354-62. doi: 10.3171/2012.5.JNS112124. Epub 2012 Jun 15. PMID: 22702484; PMCID: PMC4060619.

[29] Berger MS, Ojemann GA. Intraoperative brain mapping techniques in neuro-oncology. Stereotact Funct Neurosurg. 1992;58(1-4):153-61. doi: 10.1159/000098989. PMID: 1439333.

[30] Ebeling U, Reulen HJ. Space-occupying lesions of the sensori-motor region. Adv Tech Stand Neurosurg. 1995;22:137-81. doi: 10.1007/978-3-7091-6898-1_3. PMID: 7495418.

[31] Fernandes RA, Diniz B, Ribeiro R, Humayun M. Artificial vision through neuronal stimulation. Neurosci Lett. 2012 Jun 25;519(2):122-8. doi: 10.1016/j.neulet.2012.01.063. Epub 2012 Feb 3. PMID: 22342306.

[32] Billock VA, Tsou BH. Elementary visual hallucinations and their relationships to neural pattern-forming mechanisms. Psychol Bull. 2012 Jul;138(4):744-74. doi: 10.1037/a0027580. Epub 2012 Mar 26. PMID: 22448914.

[33] Flesher SN, Collinger JL, Foldes ST, Weiss JM, Downey JE, Tyler-Kabara EC, Bensmaia SJ, Schwartz AB, Boninger ML, Gaunt RA, Intracortical microstimulation of human somatosensory cortex. *Sci Transl Med*. 2016 Oct 19;8(361):361ra141. doi: 10.1126/scitranslmed.aaf8083.

[34] Voigt MB, Hubka P, Kral A. Intracortical microstimulation differentially activates cortical layers based on stimulation depth. *Brain Stimul*. 2017 May-Jun;10(3):684-694. doi: 10.1016/j.brs.2017.02.009.

[35] Richardson AG, Ghenbot Y, Liu X, Hao H, Rinehart C, DeLuccia S, Torres Maldonado

S, Boyek G, Zhang M, Aflatouni F, Van der Spiegel J, Lucas TH. Learning active sensing strategies using a sensory brain-machine interface. *Proc Natl Acad Sci U S A*. 2019 Aug 13. pii: 201909953. doi: 10.1073/pnas.1909953116.

[36] Parvizi J, Kastner S. Promises and limitations of human intracranial electroencephalography. Nat Neurosci.2018;21(4):474-483. doi: 10.1038/s41593-018-0108-2.

[37] Buzsáki G, Anastassiou C, Koch C. The Origin of Extracellular Fields and currents--EEG, ECoG, LFP and Spikes. *Nat Rev Neurosci*. 2012 May 18;13(6):407-20. doi: 10.1038/nrn3241.

[38] Mayberg HS, Lozano AM, Voon V, McNeely HE, Seminowicz D, Hamani C, Schwalb JM, Kennedy SH. Deep brain stimulation for treatment-resistant depression. *Neuron*. 2005 Mar 3;45(5):651-60. doi: 10.1016/j.neuron.2005.02.014.

[39] Dandekar MP, Fenoy AJ, Carvalho AF, Soares JC, Quevedo J. Deep brain stimulation for treatment-resistant depression: an integrative review of preclinical and clinical findings and translational implications. Mol Psychiatry. 2018 May;23(5):1094-1112. doi: 10.1038/mp.2018.2.

[40] Schuepbach WM, Rau J, Knudsen K, Volkmann J, Krack P, Timmermann L, Hälbig TD, Hesekamp H, Navarro SM, Meier N, Falk D, Mehdorn M, Paschen S, Maarouf M, Barbe MT, Fink GR, Kupsch A, Gruber D, Schneider GH, Seigneuret E, Kistner A, Chaynes P, Ory-Magne F, Brefel Courbon C, Vesper J, Schnitzler A, Wojtecki L, Houeto JL, Bataille B, Maltête D, Damier P, Raoul S, Sixel-Doering F, Hellwig D, Gharabaghi A, Krüger R, Pinsker MO, Amtage F, Régis JM, Witjas T, Thobois S, Mertens P, Kloss M, Hartmann A, Oertel WH, Post B, Speelman H, Agid Y, Schade-Brittinger C, Deuschl G; EARLYSTIM Study Group. Neurostimulation for Parkinson's disease with early motor complications. *N Engl J Med*. 2013 Feb 14;368(7):610-22. doi: 10.1056/NEJMoa1205158.

[41] Aström M, Zrinzo LU, Tisch S, Tripoliti E, Hariz MI, Wårdell K. Method for patient-specific finite element modeling and simulation of deep brain stimulation. *Med Biol Eng Comput*. 2009 Jan;47(1):21-8. doi: 10.1007/s11517-008-0411-2.

[42] Malaga KA, Costello JT, Chou KL, Patil PG. Atlas-independent, N-of-1 tissue activation modeling to map optimal regions of subthalamic deep brain stimulation for Parkinson disease. *Neuroimage Clin*. 2021;29:102518. doi: 10.1016/j.nicl.2020.102518.

[43] 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.

[44] 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.

[45] 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.

[46] Saturnino GB, Puonti O, Nielsen JD, Antonenko D, Madsen KH, Thielscher A. SimNIBS 2.1: A Comprehensive Pipeline for Individualized Electric Field Modelling for Transcranial Brain Stimulation. In: Makarov S, Noetscher G, Horner M. Eds. *Brain and Human Body Modeling*. Springer Nature. NY 2019a. ISBN 9783030212926.

[47] Saturnino GB, Madsen KH, Thielscher A. Efficient Electric Field Simulations for Transcranial Brain Stimulation. *bioRxiv* 541409; 2019b. doi: https://doi.org/10.1101/541409.

[48] Saturnino GB, Madsen KH, Thielscher A. Electric field simulations for transcranial brain stimulation using FEM: an efficient implementation and error analysis. *J Neural Eng*. 2019c Nov 6;16(6):066032. doi: 10.1088/1741-2552/ab41ba.

[49] 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;65(12):2675-2683. doi: 10.1109/TBME.2018.2813261.

[50] Htet AT, Saturnino GB, Burnham EH, Noetscher G, Nummenmaa A, Makarov SN. Comparative performance of the finite element method and the boundary element fast multipole method for problems mimicking transcranial magnetic stimulation (TMS*). J Neural Eng*. 2019a:16:1-13. doi: https://dx.doi.org/10.1088/1741-2552/aafbb9.

[51] Gomez LJ, Dannhauer M, Koponen LM, Peterchev AV. Conditions for numerically accurate TMS electric field simulation. *Brain Stimul*. 2019 Oct 3. pii: S1935-861X(19)30378-X. doi: 10.1016/j.brs.2019.09.015.

[52] Li G, Nie J, Wu G, Wang Y, Shen D, Alzheimer's Disease Neuroimaging Initiative. Consistent reconstruction of cortical surfaces from longitudinal brain MR images. *Neuroimage*. 2012 Feb 15;59(4):3805-20. doi: 10.1016/j.neuroimage.2011.11.012.

[53] Makarov SN, Navarro de Lara L, Noetscher GM, Nummenmaa A. Modeling Primary Fields of TMS Coils with the Fast Multipole Method. biorxiv. 2019. doi: https://doi.org/10.1101/514919

[54] Makarov SN, Nummenmaa A, Wartman WA. “Software Toolkit for Transcranial Magnetic Stimulation Modeling using Boundary Element Fast Multipole Method – Connectome Subject 110411”. *GitHub Repository*. Online: https://tmscorelab.github.io/TMS-Modeling-Website/

[55] Makarov SN, Nummenmaa A, Wartman WA. “Software Toolkit for TES Modeling Using Boundary Element Fast Multipole Method”. *GitHub Repository.* Online: https://tmscorelab.github.io/TES-Modeling-Website/

[56] Makarov SN, Nummenmaa A, Wartman WA. “Software Toolkit for EEG and MEG Modeling Using Boundary Element Fast Multipole Method”. *GitHub Repository.* Online: https://tmscorelab.github.io/EEG_MEG-Modeling-Website/

[57] Song J. Lu CC, Chew WC. Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects. *IEEE Trans. Antennas and Propagation*. 1997

45(10):1488-1493. doi: r S 0018-926X(97)07215-3.

[58] Chew WC, Jin JM, Michielssen E, Song J, Eds., *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House,Norwood, MA, 2001.

[59] Chen Z, Waubke H, Kreuzer W. A formulation of the fast multipole boundary element method (FMBEM) for acoustic radiation and scattering from three-dimensional structures, *J. Comput. Acoust*. 16 (2008) 303-320. doi: https://doi.org/10.1142/S0218396X08003725

[60] Burgschweiger R, Ochmann M, Schäfer I, Nolte B. Performance-Optimierung und Grenzen eines Multi-Level Fast Multipole Algorithmus für akustische Berechnungen. *38. Jahrestagung für Akustik* (DAGA 2012), März 2012, Darmstadt, Deutschland.

[61] Burgschweiger R, Schäfer I, Ochmann M, Nolte, B. Optimization and Limitations of a Multi-Level Adaptive-Order Fast Multipole Algorithm for Acoustical Calculations. *Acoustics* 2012, Mai 2012, Hong Kong.

[62] Burgschweiger R, Schäfer I, Ochmann M, und Nolte B. The Combination of a Multi-Level Fast Multipole Algorithm with a Source-Clustering Method for higher expansion orders. *39. Jahrestagung für Akustik* (AIA/DAGA 2013), März 2013, Meran, Italien.

[63] Wu H, Liu Y, Jiang W. A fast multipole boundary element method for 3D multi-domain acoustic scattering problems based on the Burton–Miller formulation. *Engineering Analysis with Boundary Elements*. 36(2012)779–788. doi: https://doi.org/10.1016/j.enganabound.2011.11.018.

[64] Greengard L, Rokhlin V. A fast algorithm for particle simulations. *J. Comput. Phys.* 1987;73(2):325-348. doi: 10.1016/0021-9991(87)90140-9.

[65] *Fast Multipole Method (FMM) library in R^3*. V. 1.2.1. Nov. 9th 2017. Online: https://github.com/zgimbutas/fmmlib3d.

[66] Gimbutas Z, Greengard L, Magland J, Rachh M, Rokhlin V. *fmm3D Documentation.* Release 0.1.0. 2019. Online: https://github.com/flatironinstitute/FMM3D

[67] Barnard ACL, Duck IM and Lynn MS. The application of electromagnetic theory to electrocardiology: I. Derivation of the integral equations. Biophys J. 1967 Sep;7(5):443-62. DOI: 10.1016/S0006-3495(67)86598-6.

[68] Rahmouni L, Adrian SB, Cools K, Andriulli FP. Conforming discretizations of boundary element solutions to the electroencephalography forward problem. *Comptes Rendus Physique*. 2018 Jan-Feb; 19(1-2):7-25. doi: https://doi.org/10.1016/j.crhy.2018.02.002.

[69] Wilton D, Rao S, Glisson A, Schaubert D, Al-Bundak O, and Butler C. "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains". *IEEE Transactions on Antennas and Propagation*, vol. 32(3): 276-281, March 1984. doi: 10.1109/TAP.1984.1143304.

[70] Van Oosterom A, Strackee J. "The Solid Angle of a Plane Triangle". *IEEE Transactions on Biomedical Engineering*, vol. 30(2):25-126, Feb. 1983. doi: 10.1109/TBME.1983.325207

[71] Iacono MI, Neufeld E, Akinnagbe E, Bower K, Wolf J, Vogiatzis Oikonomidis I, et al. (2015) MIDA: A Multimodal Imaging-Based Detailed Anatomical Model of the Human Head and Neck. *PLoS ONE* 10(4): e0124126. doi:10.1371/journal.pone.0124126

[72] Htet AT, Burnham EH, Noetscher GM, Pham DN, Nummenmaa A, Makarov SN. Collection of CAD human head models for electromagnetic simulations and their applications. *Biomedical Physics & Engineering Express*. 2019b: 6(5):1-13. doi: https://doi.org/10.1088/2057-1976/ab4c76.

[73] Smythe WR. *Static and Dynamic Electricity*, McGraw Hill, 1950, New York, Second Ed., Chapter 11.

[74] Balanis CA. *Advanced Engineering Electromagnetics*, Wiley, New York, 2012. 2nd Ed., ISBN-10: 0470589485.

[75] Navarro de Lara LI, Rad LG, Makarov SN, Stockmann J, Wald LL, Nummenmaa A. Simulations of a birdcage coil B1+ field on a human body model for designing a 3T multichannel TMS/MRI head coil array. *Conf. Proc IEEE Eng Med Biol Soc*. 2018 07; 2018:4752-4755.PMID: 30441411.

[76] Navarro de Lara L, Golestanirad L, Makarov SN, Stockmann JP, Wald LL, Nummenmaa A. Evaluation of RF interactions between a 3T birdcage transmit coil and transcranial magnetic stimulation coils using a realistically shaped head phantom. *Magn Reson Med*. 2020 Jan 23. doi: 10.1002/mrm.28162.

[77] Van Bladel J. *Electromagnetic Fields*. 2nd Ed. IEEE Press Series on Electromagnetic Wave Theory. 2007. Wiley. ISBN-13: 978-0471263883.

[78] 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; **62**(4):2222–2231. Online (Jan. 2020): http://www.humanconnectomeproject.org/

[79] Lee EG, Duffy W, Hadimani RL, Waris M, Siddiqui W, Islam F, et al. Investigational effect of brain-scalp distance on the efficacy of transcranial magnetic stimulation treatment in depression. *IEEE Trans Magn*. 2016;52:52–5. https://doi.org/10.1109/TMAG.2015.2514158.

[80] Lee EG, Rastogi P, Hadimani RL, Jiles DC, Camprodon JA. Impact of non-brain anatomy and coil orientation on inter- and intra-subject variability in TMS at midline. *Clin Neurophysiol*. 2018 Sep;129(9):1873-1883. doi: 10.1016/j.clinph.2018.04.749. Appendix A. Supplementary data.

[81] The Population Head Model Repository. 2016. IT'IS Foundation website. doi: 10.13099/ViP-PHM-V1.0. Online: https://www.itis.ethz.ch/virtual-population/regional-human-models/phm-repository/

[82] Saturnino, G.B., Thielscher, A., Madsen, K.H., Knoesche, T.R., Weise, K., 2019. A principled approach to conductivity uncertainty analysis in electric field calculations.

Neuroimage 188 (1), 821–834. https://doi.org/10.1016/j.neuroimage.2018.12.053.

[83] *Database of Tissue Properties*. IT'IS Foundation website. 2019. doi: 10.13099/VIP21000-04-0. Online: https://itis.swiss/virtual-population/tissue-properties/database/

[84] Yousry TA, Schmid UD, Alkadhi H, Schmidt D, Peraud A, Buettner A, et al. Localization of the motor hand area to a knob on the precentral gyrus. A new landmark. *Brain* 1997;**120**:141e57. https://doi.org/10.1093/brain/120.1.141

[85] SimNIBS Example Dataset. Online: https://simnibs.github.io/simnibs/build/html/dataset.html

[86] Thielscher A. Private communication. Dec. 2019.

[87] Reijonen J, Säisänen L, Könönen M, Mohammadi A, Julkunen P. The effect of coil placement and orientation on the assessment of focal excitability in motor mapping with navigated transcranial magnetic stimulation. *J. Neurosci Methods*. 2019 Nov 13:108521. doi: 10.1016/j.jneumeth.2019.108521.

[88] Miranda PC, Correia L, Salvador R, Basser PJ. Tissue heterogeneity as a mechanism for localized neural stimulation by applied electric fields. *Phys Med Biol*. 2007 Sep 21;**52**(18):5603-17.

[89] Salvador R, Silva S, Basser PJ, Miranda PC. Determining which mechanisms lead to activation in the motor cortex: a modeling study of transcranial magnetic stimulation using realistic stimulus waveforms and sulcal geometry. *Clin Neurophysiol*. 2011 Apr;**122**(4):748-58. doi: 10.1016/j.clinph.2010.09.022.

[90] Di Lazzaro V, Oliviero A, Profice P, Saturno E, Pilato F, Insola A, Mazzone P, Tonali P, Rothwell JC. Comparison of descending volleys evoked by transcranial magnetic and electric stimulation in conscious humans. *Electroencephalogr Clin Neurophysiol* 1998;**109**: 397–401. doi: 10.1016/S0924-980X(98)00038-1.

[91] Di Lazzaro V, Ziemann U, Lemon RN. State of the art: Physiology of transcranial motor cortex stimulation. *Brain Stimul*. 2008 Oct;1(4):345-62. doi: 10.1016/j.brs.2008.07.004.

[92] Di Lazzaro V, Ziemann U. The contribution of transcranial magnetic stimulation in the functional evaluation of microcircuits in human motor cortex. *Front Neural Circuits*. 2013 Feb 13;7:18. doi: 10.3389/fncir.2013.00018.

[93] Volz L, Hamada M, Rothwell JC, Grefkes C. What Makes the Muscle Twitch: Motor System Connectivity and TMS-Induced Activity. *Cereb Cortex*. 2015 Sep; **25**(9):2346-53. doi: 10.1093/cercor/bhu032.

[94] Niemann N, Wiegel P, Kurz A, Rothwell JC, Leukel C. Assessing TMS-induced D and I waves with spinal H-reflexes. *J Neurophysiol*. 2018 Mar 1;**119**(3):933-943. doi: 10.1152/jn.00671.2017.

[95] Nummenmaa A, McNab JA, Savadjiev P, Okada Y, Hämäläinen MS, Wang R, Wald LL, Pascual-Leone A, Wedeen VJ, Raij T. Targeting of White Matter Tracts with Transcranial MagneticStimulation. *Brain Stimul*. 2014 Jan-Feb;**7**(1):80-4. doi: 10.1016/j.brs.2013.10.001.

[96] Day BL, Dressler D, Maertens de Noordhout A, Marsden CD, Nakashima K, Rothwell JC, Thompson PD. Electric and magnetic stimulation of human motor cortex: surface EMG and single motor unit responses. *J. Physiol* **412**: 449–473, 1989. doi:10.1113/jphysiol.1989.sp017626.

[97] Bungert A, Antunes A, Espenhahn S, Thielscher A. Where does TMS Stimulate the Motor Cortex? Combining Electrophysiological Measurements and Realistic Field Estimates to Reveal the Affected Cortex Position. *Cereb Cortex*. 2017 Nov 1;27(11):5083-5094. doi: 10.1093/cercor/bhw292. PMID: 27664963.

[98] Diana M, Raij T, Melis M, Nummenmaa A, Leggio L, and Bonci A. Rehabilitating the addicted brain with transcranial magnetic stimulation. *Nat. Rev. Neurosci.* 2017; 18: 685-93. doi: 10.1038/nrn.2017.113

[99] Wang B, Grill WM, Peterchev AV. Coupling Magnetically Induced Electric Fields to Neurons: Longitudinal and Transverse Activation. *Biophys J*. 2018 Jul 3;115(1):95-107. doi: 10.1016/j.bpj.2018.06.004.

[100] Aberra AS, Peterchev AV, Grill WM. Biophysically realistic neuron models for simulation of cortical stimulation. *J Neural Eng*. 2018 Dec;15(6):066023. doi: 10.1088/1741-2552/aadbb1. PMID: 30127100.

[101] Aberra AS, Wang B, Grill WM, Peterchev AV. Simulation of transcranial magnetic stimulation in head model with morphologically-realistic cortical neurons. *Brain Stimul*. 2020; 13(1):175-89. pii: S1935-861X(19)30409-7. doi: 10.1016/j.brs.2019.10.002.

[102] Schmidt S, Bathe-Peters R, Fleischmann R, Rönnefarth M, Scholz M, Brandt SA. Nonphysiological factors in navigated TMS studies; confounding covariates and valid intracortical estimates. *Hum Brain Mapp*. 2015 Jan;36(1):40-9. doi: 10.1002/hbm.22611.

[103] Fang X, Liu M, Lu C, Zhao Y, Liu X, Current status and potential application of navigated transcranial magnetic stimulation in neurosurgery: a literature review. *Chin. Neurosurg J*. 2019 5(12) doi:10.1186/s41016-019-0159-6.

[104] Goetz SM, Kozyrkov IC, Luber B, Lisanby SH, Murphy DLK, Grill WM, Peterchev AV. Accuracy of robotic coil positioning during transcranial magnetic stimulation. *J Neural Eng*. 2019 Sep 17;16(5):054003. doi: 10.1088/1741-2552/ab2953.

[105] Makarov SN, Wartman WA, Noetscher GM, Fujimoto K, Zaidi T, Burnham EH, Aapo A. Degree of Improving TMS Focality through a Geometrically Stable Solution of an Inverse TMS Problem. *Neuroimage*, under review.

[106] Di Lazzaro V, Ranieri F, Profice P, Pilato F, Mazzone P, Capone F, Insola A, Oliviero A. Transcranial direct current stimulation effects on the excitability of corticospinal axons of

the human cerebral cortex. Brain Stimul. 2013 Jul;6(4):641-3. doi: 10.1016/j.brs.2012.09.006. Epub 2012 Oct 6. PMID: 23085442.

[107] Bashir S, Perez JM, Horvath JC, Pascual-Leone A. Differentiation of motor cortical representation of hand muscles by navigated mapping of optimal TMS current directions in healthy subjects. J Clin Neurophysiol. 2013 Aug;30(4):390-5. doi: 10.1097/WNP.0b013e31829dda6b. PMID: 23912579; PMCID: PMC3740163.

[108] Raffin E, Pellegrino G, Di Lazzaro V, Thielscher A, Siebner HR. Bringing transcranial mapping into shape: Sulcus-aligned mapping captures motor somatotopy in human primary motor hand area. *Neuroimage*. 2015 Oct 15;120:164-75. doi: 10.1016/j.neuroimage.2015.07.024.

[109] Weise K, Numssen O, Thielscher A, Hartwigsen G., Knösche T. R. A novel approach to localize cortical TMS effects. *Neuroimage* 209, 116486 (2020).

[110] Balderston NL, Roberts C, Beydler EM, Deng ZD, Radman T, Luber B, Lisanby SH, Ernst M, Grillon C. A generalized workflow for conducting electric field-optimized, fMRI-guided, transcranial magnetic stimulation. *Nat Protoc*. 2020 Sep 30. doi: 10.1038/s41596-020-0387-4.

[111] Fox PT, Narayana S, Tandon N, Sandoval H, Fox SP, Kochunov P, et al. Column-based model of electric field excitation of cerebral cortex. *Hum Brain Mapp* 2004;22:1e14. https://doi.org/10.1002/hbm.20006.

[112] Krieg TD, Salinas FS, Narayana S, Fox PT, Mogul DJ. PET-based confirmation of orientation sensitivity of TMS-induced cortical activation in humans. *Brain Stimul* 2013;6:898e904. https://doi.org/10.1016/j.brs.2013.05.007.

[113] Laakso I, Murakami T, Hirata A, Ugawa Y. Where and what TMS activates: Experiments and modeling. *Brain Stimul*. 2018 Jan-Feb;11(1):166-174. doi: 10.1016/j.brs.2017.09.011. Epub 2017 Sep 27. PMID: 29030110.

[114] Laakso I, Hirata A. Fast multigrid-based computation of the induced electric field for transcranial magnetic stimulation. *Phys Med Biol*. 2012 Dec 7;57(23):7753-65. doi: 10.1088/0031-9155/57/23/7753.

[115] Stenroos M, Koponen LM. Real-time computation of the TMS-induced electric field in a realistic head model. *Neuroimage*. 2019 Dec;203:116159. doi: 10.1016/j.neuroimage.2019.116159.

[116] Dubbioso R, Raffin E, Karabanov A, Thielscher A, Siebner HR. Centre-surround organization of fast sensorimotor integration in human motor hand area. *Neuroimage*. 2017 Sep;158:37-47. doi: 10.1016/j.neuroimage.2017.06.063.

[117] Puonti O, Saturnino GB, Madsen KH, Thielscher A. Value and limitations of intracranial recordings for validating electric field modeling for transcranial brain stimulation. *NeuroImage*. 2020 Mar;208:116431. doi: 10.1016/j.neuroimage.2019.116431.

[118] Fischl B. FreeSurfer. *NeuroImage*. 2012;62(2):774-81. doi: 10.1016/j.neuroimage.2012.01.021. PubMed PMID: 22248573.

[119] *Structural Brain Mapping Group. Computational Anatomy Toolbox (CAT)*. Univ. of Jena, Germany. Accessed 04/05/21. Online: http://www.neuro.uni-jena.de/wordpress/publications/

[120] Seiger R, Ganger S, Kranz GS, Hahn A, Lanzenberger R. Cortical Thickness Estimations of FreeSurfer and the CAT12 Toolbox in Patients with Alzheimer's Disease and Healthy Controls. *J Neuroimaging*. 2018 Sep;28(5):515-523. doi: 10.1111/jon.12521.

[121] *Axilum Robotics*. University of Strasbourgh. France. Onine: ttp://www.axilumrobotics.com/en/ Collaborative robot (cobot) data online: http://www.axilumrobotics.com/en/tms-cobot-features/

[122] Righart R, Schmidt P, Dahnke R, Biberacher V, Beer A, Buck D, Hemmer B, Kirschke JS, Zimmer C, Gaser C, Mühlau M. Volume versus surface-based cortical thickness measurements: A comparative study with healthy controls and multiple sclerosis patients. PLoS One. 2017 Jul 6;12(7):e0179590. doi: 10.1371/journal.pone.0179590.

[123] Katoch N, Choi BK, Sajib SZK, Lee E, Kim HJ, Kwon OI, Woo EJ. Conductivity Tensor Imaging of In Vivo Human Brain and Experimental Validation Using Giant Vesicle Suspension. IEEE Trans Med Imaging. 2019 Jul;38(7):1569-1577. doi: 10.1109/TMI.2018.2884440.

[124] Pascual-Leone A, Rubio B, Pallardó F, Catalá MD. Rapid-rate transcranial magnetic stimulation of left dorsolateral prefrontal cortex in drug-resistant depression. *Lancet*. 1996 Jul 27;**348**(9022):233-7.

[125] Geller V, Grisaru N, Abarbanel JM, Lemberg T, Belmaker RH. Slow magnetic stimulation of prefrontal cortex in depression and schizophrenia. *Prog Neuropsychopharmacol Biol Psychiatry*. 1997 Jan;**21**(1):105-10.

[126] Koenigs M, Grafman J. The functional neuroanatomy of depression: Distinct roles for ventromedial and dorsolateral prefrontal cortex. *Behav Brain Res*. 2009 Aug 12;**201**(2):239-43. doi: 10.1016/j.bbr.2009.03.004.

[127] Gomez LJ, M, Peterchev AV. Fast computational optimization of TMS coil placement for individualized electric field targeting. *NeuroImage*. 2021 Mar;228:117696. doi: 10.1016/j.neuroimage.2020.117696.

[128] Daneshzand M, Makarov S, Navarro de Lara L, Guerin B, McNab J, Rosen B, Hämäläinen M, Raij T, Nummenmaa A. Rapid computation of TMS-induced E-fields using a dipole-based magnetic stimulation profile approach. *NeuroImage*. 2021. Accepted.

[129] Weise K, Poßner L, Müller E, Gast R, Knösche TR, Pygpc: A sensitivity and uncertainty analysis toolbox for Python, *SoftwareX*, vol. 11, 2020, 100450, ISSN 2352-7110, https://doi.org/10.1016/j.softx.2020.100450.

[130] Huang Y, Datta A, Bikson M, Parra LC. Realistic volumetric-approach to simulate transcranial electric stimulation-ROAST-a fully automated open-source pipeline. *J Neural Eng.* 2019 Jul 30;16(5):056006. doi: 10.1088/1741-2552/ab208d.

[131] Wenger C, Miranda PC, Salvador R, Thielscher A, Bomzon Z, Giladi M, Mrugala MM, Korshoej AR. A Review on Tumor-Treating Fields (TTFields): Clinical Implications Inferred From Computational Modeling. *IEEE Rev Biomed Eng*. 2018;11:195-207. doi: 10.1109/RBME.2017.2765282. Epub 2018 Feb 13. PMID: 29993870.

[132] Anderson DN, Osting B, Vorwerk J, Dorval AD, Butson CR. Optimized programming algorithm for cylindrical and directional deep brain stimulation electrodes. *J Neural Eng*. 2018 Apr;15(2):026005. doi: 10.1088/1741-2552/aaa14b.

[133] Vorwerk J, Brock AA, Anderson DN, Rolston JD, Butson CR. A retrospective evaluation of automated optimization of deep brain stimulation parameters. *J Neural Eng*. 2019 Nov 6;16(6):064002. doi: 10.1088/1741-2552/ab35b1

[134] Duffley G, Anderson DN, Vorwerk J, Dorval AD, Butson CR. Evaluation of methodologies for computing the deep brain stimulation volume of tissue activated. *J Neural Eng.* 2019 Oct 29;16(6):066024. doi: 10.1088/1741-2552/ab3c95.

[135] Makarov SN, Hämäläinen M, Okada Y, Noetscher GM, Ahveninen J., Nummenmaa A Boundary Element Fast Multipole Method for Enhanced Modeling of Neurophysiological Recordings. *IEEE Trans on Biomed Eng*., vol. 68, no. 1, pp. 308-318, Jan. 2021, doi: 10.1109/TBME.2020.2999271.

[136] M. Cheney, D. Isaacson, and J. C. Newell, "Electrical Impedance Tomography," *SIAM Review*, vol. 41, no. 1, pp. 85-101, Mar., 1999.

[137] M.A. Nitsche, L.G. Cohen, E.M. Wassermann, A. Priori, N. Lang, A. Antal, W. Paulus, F. Hummel, P.S. Boggio, F. Fregni, A. Pascual-Leone, "Transcranial Direct Current Stimulation: State of the Art 2008", *Brain Stimulation*, vol. 1, pp. 206-223, 2008.

[138] Helmholtz H. Ű̋ber einige Gesetze der Vertheilung elektrischer Ströme in körperlichen Leitern mit Anwendung auf die thierisch-elektrischen Versuche. *Ann. der Phys. (und Chemie)*. 1853;89: 353-377. 3rd ser.

[139] Rush S, Driscoll DA. EEG electrode sensitivity--an application of reciprocity. *IEEE Trans Biomed Eng*. 1969;16(1):15-22. doi:10.1109/tbme.1969.4502598.

[140] Vladimirov, V. S., 1971, *Equations of Mathematical Physics*, Marcel Dekker, New York, ISBN-10 0824717139.

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

[142] *TT-Toolbox*. *FGMRES*. 2016. V Oseledets, S Dolgov, V Kazeev, D Savostyanov, O Lebedeva, P Zhlobich, T Mach, L Song. Online: https://github.com/oseledets/TT-Toolbox/blob/master/solve/fgmres.m

[143] Klee M, Plonsey R. Stimulation of spheroidal cells-The role of cell shape. 1976. *IEEE Trans Biomed Eng*. Jul;23(4):347-54. PMID: 1278928.

[144] Cartee LA, Plonsey R. The transient subthreshold response of spherical and cylindrical cell models to extracellular stimulation. *IEEE Trans Biomed Eng*. 1992 Jan;39(1):76-85. DOI: 10.1109/10.108130.

[145] Krassowska W, Neu JC. Response of a single cell to an external electric field. *Biophys J*. 1994 Jun;66(6):1768-76. doi: 10.1016/S0006-3495(94)80971-3.

[146] Pourtaheri N, Ying W, Kim JM, Henriquez CS. Thresholds for transverse stimulation: fiber bundles in a uniform field. *IEEE Trans Neural Syst Rehabil Eng*. 2009 Oct;17(5):478-86. doi: 10.1109/TNSRE.2009.2033424.

[147] Rattay F. Analysis of the electrical excitation of CNS neurons. *IEEE Trans. Biomed. Eng*. 1998;45 (6): 766–772. doi:10.1109/10.67861.

[148] Ruohonen J, Panizza M, Nilsson J, Ravazzani P, Grandori F, Tognola G. Transverse-field activation mechanism in magnetic stimulation of peripheral nerves. *Electroencephalogr Clin Neurophysiol*. 1996 Apr;101(2):167-74. doi: 10.1016/0924-980x(95)00237-f.

[149] Jiang, Jimmy & Truong, Dennis & Esmaeilpour, Zeinab & Huang, Yu & Badran, Bashar & Bikson, Marom. (2019). Enhanced tES and tDCS computational models by meninges emulation. Journal of Neural Engineering. 17. 10.1088/1741-2552/ab549d.

[150] Khadka N, Bikson M. Role of skin tissue layers and ultra-structure in transcutaneous electrical stimulation including tDCS. Phys Med Biol. 2020 Sep 11. doi: 10.1088/1361-6560/abb7c1. Epub ahead of print. PMID: 32916670.

[151] Farcito S, Puonti O, Montanaro H, Saturnino GB, Nielsen JD, Madsen CG, Siebner HR, Neufeld E, Kuster N, Lloyd BA. "Accurate anatomical head segmentations: a data set for biomedical simulations," 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019, pp. 6118-6123, doi: 10.1109/EMBC.2019.8857041.

[152] Sim4Life. Zurich MedTech AG. Online: https://zmt.swiss/sim4life/

[153] Wartman WA, Davids M, Daneshzand M, Burnham EH, Nummenmaa A, and Makarov SN. High Resolution Computational Modeling of Transcranial Stimulation using the MIDA Head Model. *10th International IEEE/EMBS Conference on Neural Engineering.* Virtual conference. May 4-6, 2021.

[154] Wilton D. Private communication, 2021

Appendix A: Code Segments

AA.1.Neighbor Integrals

AA.1.a. potint2.m

```
function [Int] = potint2(r1, r2, r3, normal, ObsPoint)
%   This function computes potential integrals grad(1/r) for a single triangle
%   The integrals are not divided by the area
%   Vectorized for an arbitrary number of observation points
%
%   Copyright SNM 2004-2020

%   Test (comparison with Wang et. al., 2003):
% %     clear all;
% %     format long;
% %     r1 = [62.5 25.0 0];
% %     r2 = [62.5 25.0 2];
% %     r3 = [62.5 37.5 0];
% %     ObsPoint = [62.5 0.0 0.0];
% %     tempv           = cross(r2-r1, r3-r1);  %   correct normal sign!
% %     temps           = sqrt(tempv(1)^2 + tempv(2)^2 + tempv(3)^2);
% %     normal          = tempv/temps;
% %     Area            = temps/2;
% %     %   Analytical integral
% %      [coeff, weights, IndexF] = tri(25, 10);
% %      Int = [0 0 0];
% %      for m = 1:length(coeff)
% %          Point = coeff(1, m)*r1 + coeff(2, m)*r2 + coeff(3, m)*r3;
% %          R = ObsPoint - Point;
% %          Int = Int - Area*weights(m)*R/(sqrt(sum(R.*R)))^3;
% %      end
% %      Int
% %      Int = potint2(r1, r2, r3, normal, ObsPoint)

    N           = size(ObsPoint, 1);
    I           = zeros(N, 3);
```

```
S           = zeros(N, 9);
Int         = zeros(N, 3);
BetaTerm    = zeros(N, 3);

%   Create r+ and r- coordinates
temp = [r2 r1 r3 r1 r3 r2];
r    = repmat(temp, N, 1);

%   Create l coordinates
normabsl1 = sqrt(sum((r2-r1).^2));
normabsl2 = sqrt(sum((r3-r1).^2));
normabsl3 = sqrt(sum((r3-r2).^2));
temp = [(r2-r1)./normabsl1 (r3-r1)./normabsl2 (r3-r2)./normabsl3];
l    = repmat(temp, N, 1);

%   Create unit normal to the edges of the triangle
u(1:3) = +cross((r2-r1)./normabsl1, normal);
u(4:6) = -cross((r3-r1)./normabsl2, normal);
u(7:9) = +cross((r3-r2)./normabsl3, normal);
u      = repmat(u, N, 1);

%   Create projection vector of the observation point
NORM = repmat(normal, N, 1);
ndot = sum(ObsPoint.*NORM, 2);
p    = ObsPoint - repmat(ndot, 1, 3).*NORM;

%   Is the projection point on the triangle edge or its continuation?
%   Calculate midpoints of the edges of a triangle
temp     = 0.5*[r1+r2 r1+r3 r2+r3];
midpoint = repmat(temp, N, 1);
%   Calculate vectors from observation point project to midpoints
vector = midpoint - [p p p];
  %   Move the observation  point (projection) from the edge
factor = 1e-6;
factor1 = factor*normabsl1;
factor2 = factor*normabsl2;
factor3 = factor*normabsl3;
check1 = sum(vector(:, 1:3).*u(:, 1:3), 2);
check2 = sum(vector(:, 4:6).*u(:, 4:6), 2);
check3 = sum(vector(:, 7:9).*u(:, 7:9), 2);
```

```
index  = (abs(check1)<factor1); ObsPoint(index, :) =  ObsPoint(index, :) - factor1*u(index, 1:3);
index  = (abs(check2)<factor2); ObsPoint(index, :) =  ObsPoint(index, :) - factor2*u(index, 4:6);
index  = (abs(check3)<factor3); ObsPoint(index, :) =  ObsPoint(index, :) - factor3*u(index, 7:9);
ndot   = sum(ObsPoint.*NORM, 2);
p      = ObsPoint - repmat(ndot, 1, 3).*NORM;

% Calculation of the analytical formula
count = 0;
for c1 = 0:2
    %   Distance of observation point perpendicular to the plane with triangle
    temporary   = ObsPoint - r(:, [1:3] + 3*(count + 1));
     %%  Changes compared to potint.m
    distanceobs = sum(NORM.*temporary, 2);
     %%  End of changes compared to potint.m

    %   Calculate p+ and p-
    d1 = sum(NORM.*r(:, [1:3] + 3*count), 2);
    d2 = sum(NORM.*r(:, [1:3] + 3*(count + 1)), 2);

    pplus  = r(:, [1:3] + 3*count)       - NORM.*repmat(d1, 1, 3);
    pminus = r(:, [1:3] + 3*(count + 1)) - NORM.*repmat(d2, 1, 3);

    %   Calculate l+ and l-
    lplus  = sum(l(:, [1:3] + 3*c1).*(pplus - p), 2);
    lminus = sum(l(:, [1:3] + 3*c1).*(pminus - p), 2);

    %   Perpendicular distance from projection vector to edge
    P0 = abs( sum(u(:, [1:3] + 3*c1).*(pminus-p), 2) );

    %   Distances to l+ and l- from projection vector
    PPLUS  = sqrt(P0.*P0 + lplus.*lplus);
    PMINUS = sqrt(P0.*P0 + lminus.*lminus);

    %   Vector containing line P0 measured
    PHAT = (pminus - p - repmat(lminus, 1, 3).*l(:, [1:3] + 3*c1))./repmat(P0, 1, 3);

    %   Distances to l+ and l- from observation point
    RPLUS  = sqrt(PPLUS.*PPLUS + distanceobs.*distanceobs);
    RMINUS = sqrt(PMINUS.*PMINUS + distanceobs.*distanceobs);
    R0     = sqrt(P0.*P0 + distanceobs.*distanceobs);
```

```
        %  Changes compared to potint.m
        %   A value of one term of the analytic sum 1/R
        d1      = atan(P0.*lplus./(R0.*R0 + abs(distanceobs).*RPLUS));
        d2      = atan(P0.*lminus./(R0.*R0 + abs(distanceobs).*RMINUS));
        d3      = log((RPLUS + lplus)./(RMINUS + lminus));

        % End of changes compared to potint.m

        dotPHATu= sum(PHAT.*u(:, [1:3] + 3*c1), 2);

        %  Changes compared to potint.m
        BetaTerm(:, c1+1) = dotPHATu.*(d1 - d2);
        S(:, 1 + 3*c1) = d3.*u(:, 1 + 3*c1);
        S(:, 2 + 3*c1) = d3.*u(:, 2 + 3*c1);
        S(:, 3 + 3*c1) = d3.*u(:, 3 + 3*c1);
        % End of changes compared to potint.m
        count = count + 2;
    end

    %  Changes compared to potint.m
    Beta = sum(BetaTerm, 2);

    I(:, 1) = S(:, 1) + S(:, 4) + S(:, 7);
    I(:, 2) = S(:, 2) + S(:, 5) + S(:, 8);
    I(:, 3) = S(:, 3) + S(:, 6) + S(:, 9);

    %   find sign of distanceobs
    Sign = sign(distanceobs);

    %   value of integral for 1/R
    Int(:, 1) = -normal(1)*Sign.*Beta - I(:, 1);
    Int(:, 2) = -normal(2)*Sign.*Beta - I(:, 2);
    Int(:, 3) = -normal(3)*Sign.*Beta - I(:, 3);

    %   Contribution is zero when the projection point on the edge (Wilton et al. 1984, p. 279)
    Int(isnan(Int)) = 0;  Int(isinf(Int)) = 0;
    % End of changes compared to potint.m
end
```

AA.1.b. potint4.m

```
% This function calculates n*grad(1/r) at a given observation point obsPoint
% given a triangle with vertices r1, r2, and r3 and normal vector normal.
% It uses the solid-angle approximation of Van Oosterom and Strackee 1983
% to quickly compute the normal component of the field in the vicinity of
% the triangle.
% r1: Nx3 first triangle vertex location for N triangles
% r2: Nx3 second triangle vertex location for N triangles
% r3: Nx3 third triangle vertex location for N triangles
% normal: Nx3 triangle normal vectors for N triangles
% obsPoint: Mx3 list of observation points at which electric field should be evaluated
% Int: MxN matrix of integral contributions to each point.  Right-multiply by column
%  vector of triangle weights (e.g. charges) to obtain total contribution to each
%  observation point.

% Copyright William Wartman 2020
function [Int] = potint4(r1, r2, r3, obsPoint)
     N = size(r1, 1);        %N triangles
     M = size(obsPoint, 1);  %M observation points

    Int = zeros(M, N);
    for j = 1:size(obsPoint, 1)
        %Observation point to vertex vectors for observation point j (Nx3)
        R1 = r1 - obsPoint(j,:);
        R2 = r2 - obsPoint(j,:);
        R3 = r3 - obsPoint(j,:);

        %Vector magnitudes of R1, R2, R3 (Nx1)
        R1norm = vecnorm(R1, 2, 2);
        R2norm = vecnorm(R2, 2, 2);
        R3norm = vecnorm(R3, 2, 2);

        %Triple product of R1, R2, R3 (Nx1)
        numerator = dot(R1, cross(R2, R3, 2), 2);

        %Nx1
        denominator = (R1norm .* R2norm .* R3norm) + R3norm .* dot(R1, R2, 2) + R2norm .* dot(R1, R3, 2) + R1norm .*
dot(R2, R3, 2);
```

```
        %Nx1
        omega = 2*atan2(numerator, denominator);

        % M observation points by N solid angles to have proper dimensions for charge weighting!
        Int(j,:) = transpose(omega);
    end
end
```

AA.1.c. potint4b.m

```
% This function calculates n*grad(1/r) at a given observation point obsPoint
% given a triangle with vertices r1, r2, and r3 and normal vector normal.
% It uses the solid-angle approximation of Van Oosterom and Strackee 1983
% to quickly compute the normal component of the field in the vicinity of
% the triangle.
% r1: Nx3 first triangle vertex location for N triangles
% r2: Nx3 second triangle vertex location for N triangles
% r3: Nx3 third triangle vertex location for N triangles
% obsPoint: Mx3 list of observation points at which electric field should be evaluated
% Int: MxN matrix of integral contributions to each point.  Right-multiply by column
%  vector of triangle weights (e.g. charges) to obtain total contribution to each
%  observation point.

% Copyright William Wartman 2020

function [Int] = potint4b(r1, r2, r3, obsPoint)
    %Vectorize operation for triangles and observation points simultaneously
     N = size(r1, 1);        %N triangles
     M = size(obsPoint, 1);  %M observation points

    %Dimension 1: triangle index.  Dimension 2: 3. Dimension 3: Observation point index
    r1Exp = repmat(r1, 1, 1, M);
    r2Exp = repmat(r2, 1, 1, M);
    r3Exp = repmat(r3, 1, 1, M);

    obsPointExpA = zeros(1, 3, M);
    obsPointExpA(1, :, :) = transpose(obsPoint);
    obsPointExp = repmat(obsPointExpA, N, 1, 1);
    %Vectors from observation points to triangle vertices (N by 3 by M)
    R1 = r1Exp - obsPointExp;
    R2 = r2Exp - obsPointExp;
    R3 = r3Exp - obsPointExp;

    %Norms of vectors (N by 1 by M)
    R1norm = vecnorm(R1, 2, 2);
    R2norm = vecnorm(R2, 2, 2);
    R3norm = vecnorm(R3, 2, 2);

    %N by 1 by M
    numerator = dot(R1, cross(R2, R3, 2), 2);

    %N by 1 by M
    denominator = (R1norm .* R2norm .* R3norm) + R3norm .* dot(R1, R2, 2) + R2norm .* dot(R1, R3, 2) + R1norm .*
dot(R2, R3, 2);

    omega = 2*atan2(numerator, denominator);

    %Squeeze out the middle dimension and shape omega properly to be scaled
    %by a column vector of triangle charges
    if(N ~= 1)
        Int = transpose(squeeze(omega));
    else
        % If there is only one triangle, the first dimension is 1 and thus
        % is squeezed out along with the second dimension, leaving a row
        % vector as desired.
        Int = squeeze(omega);
    end

end
```

AA.1.d. meshneighborints_1b.m

```
function [integralxc, integralyc, integralzc, ...
          integralxe, integralye, integralze, ...
          EC] = meshneighborints_1b(P, t, normals, Area, Center, RnumberE, ineighborE, numThreads)
%   Accurate integration for electric field/electric potential on neighbor facets
%   Copyright SNM 2017-2020
    tic
    N = size(t, 1);
    integralxe      = zeros(RnumberE, N);    %   exact Ex integrals for array of neighbor triangles
    integralye      = zeros(RnumberE, N);    %   exact Ey integrals for array of neighbor triangles
    integralze      = zeros(RnumberE, N);    %   exact Ez integrals for array of neighbor triangles

    integralxc      = zeros(RnumberE, N);    %   center-point Ex integrals for array of neighbor triangles
    integralyc      = zeros(RnumberE, N);    %   center-point Ey integrals for array of neighbor triangles
    integralzc      = zeros(RnumberE, N);    %   center-point Ez integrals for array of neighbor triangles

    gauss       = 25;   %   number of integration points in the Gaussian quadrature
                        %   for the outer potential integrals
                        %   Numbers 1, 4, 7, 13, 25 are permitted
    %   Gaussian weights for analytical integration (for the outer integral)
    if gauss == 1;  [coeffS, weightsS, IndexS]  = tri(1, 1); end;
    if gauss == 4;  [coeffS, weightsS, IndexS]  = tri(4, 3); end;
    if gauss == 7;  [coeffS, weightsS, IndexS]  = tri(7, 5); end;
    if gauss == 13; [coeffS, weightsS, IndexS]  = tri(13, 7); end;
    if gauss == 25; [coeffS, weightsS, IndexS]  = tri(25, 10); end;
    W           = repmat(weightsS', 1, 3);

    %   Main loop for analytical double integrals (parallel, 24 workers)
    %   This is the loop over columns of the system matrix
    tic
    parpool(numThreads);
    parpoolStartTime = toc
    tic
    parfor n = 1:N                  %   inner integral; (n =1 - first column of the system matrix, etc.)
        r1      = P(t(n, 1), :);    %   [1x3]
        r2      = P(t(n, 2), :);    %   [1x3]
        r3      = P(t(n, 3), :);    %   [1x3]
        index   = ineighborE(:, n); %   those are non-zero rows of the system matrix for given n
        ObsPoints   = zeros(RnumberE*IndexS, 3);    %   to compute RnumberE outer integrals numerically
        I           = zeros(RnumberE, 3);           %   for rhe field
        %   Accurate electric-field integrals
        for q = 1:RnumberE
            num = index(q);
            for p = 1:IndexS
                ObsPoints(p+(q-1)*IndexS, :)  = coeffS(1, p)*P(t(num, 1), :) +  coeffS(2, p)*P(t(num, 2), :) +
coeffS(3, p)*P(t(num, 3), :);
            end
        end
        J = potint2(r1, r2, r3, normals(n, :), ObsPoints); %   Outer integral is computed analytically, for all inner
IntPoints
        for q = 1:RnumberE
            I(q, :) = sum(W.*J([1:IndexS]+(q-1)*IndexS, :), 1);
        end
        I(1, :)          = 0;            %   self integrals will give zero
        integralxe(:, n) = -I(:, 1);     %   accurate integrals, entries of non-zero rows of n-th column
        integralye(:, n) = -I(:, 2);     %   accurate integrals, entries of non-zero rows of n-th column
        integralze(:, n) = -I(:, 3);     %   accurate integrals, entries of non-zero rows of n-th column
        %   Center-point electric-field integrals
        temp    = repmat(Center(n, :), RnumberE, 1) - Center(index, :); %   these are distances to the
observation/target triangle
        DIST    = sqrt(dot(temp, temp, 2));                             %   single column
        I       = Area(n)*temp./repmat(DIST.^3, 1, 3);                  %   center-point integral, standard format
        I(1, :) = 0;                                                    %   self integrals will give zero
        integralxc(:, n) = -I(:, 1);     %   center-point integrals, entries of non-zero rows of n-th column
        integralyc(:, n) = -I(:, 2);     %   center-point integrals, entries of non-zero rows of n-th column
        integralzc(:, n) = -I(:, 3);     %   center-point integrals, entries of non-zero rows of n-th column
    end
    integralTime = toc

    tic
    delete(gcp('nocreate'));
    parpoolShutdownTime = toc

    tic
    %%  Define useful sparse matrices EC, PC (for GMRES speed up)
    N               = size(t, 1);
    const           = 1/(4*pi);
    integralc       = zeros(RnumberE, N);    %   normal integral component for array of neighbor triangles (center
point) - to speed up GMRES
    integrale       = zeros(RnumberE, N);    %   normal integral component for array of neighbor triangles (exact) - to
speed up GMRES
    for n = 1:N                  %   inner integral; (n =1 - first column of the system matrix, etc.)
        index = ineighborE(:, n); %   those are non-zero rows of the system matrix for given n
        integrale(:, n)  =       +(integralxe(:, n).*normals(index, 1) + ...
                                   integralye(:, n).*normals(index, 2) + ...
                                   integralze(:, n).*normals(index, 3));
        integralc(:, n)  =       +(integralxc(:, n).*normals(index, 1) + ...
                                integralyc(:, n).*normals(index, 2) + ...
                                integralzc(:, n).*normals(index, 3));
    end
    ii  = ineighborE;
    jj  = repmat([1:N], RnumberE, 1);
    EC  = sparse(ii, jj, const*(-integralc + integrale));                %   almost symmetric
    save('integrals_base', 'integrale', 'integralc');
    ECConstructionTime = toc
end
```

AA.1.e. meshneighborints_2.m

```
function [EC] = meshneighborints_2(P, t, normals, Area, Center, RnumberE, ineighborE, numThreads)
%   Accurate integration for electric field/electric potential on neighbor facets
%   Copyright WAW 2020
    N = size(t, 1);
    integralxc      = zeros(RnumberE, N);    %   center-point Ex integrals for array of neighbor triangles
    integralyc      = zeros(RnumberE, N);    %   center-point Ey integrals for array of neighbor triangles
    integralzc      = zeros(RnumberE, N);    %   center-point Ez integrals for array of neighbor triangles

    gauss       = 25;   %   number of integration points in the Gaussian quadrature
                        %   for the outer potential integrals
                        %   Numbers 1, 4, 7, 13, 25 are permitted
    %   Gaussian weights for analytical integration (for the outer integral)
    if gauss == 1;  [coeffS, weightsS, IndexS]  = tri(1, 1); end
    if gauss == 4;  [coeffS, weightsS, IndexS]  = tri(4, 3); end
    if gauss == 7;  [coeffS, weightsS, IndexS]  = tri(7, 5); end
    if gauss == 13; [coeffS, weightsS, IndexS]  = tri(13, 7); end
    if gauss == 25; [coeffS, weightsS, IndexS]  = tri(25, 10); end
    if gauss == 0;  [coeffS, weightsS, IndexS]  = tri(40); end
    %W           = repmat(weightsS', 1, 3);


    %   Main loop for analytical double integrals (parallel, 24 workers)
    %   This is the loop over columns of the system matrix
    tic
    parpool(numThreads);
    parpoolStartTime = toc
    tic
    integrale = zeros(N, RnumberE);
    parfor n = 1:N                  %   inner integral; (n =1 - first column of the system matrix, etc.)
        % Calculate observation points on this triangle
        ObsPoints = zeros(IndexS, 3);
        for p = 1:IndexS
            ObsPoints(p, :)  = coeffS(1, p)*P(t(n, 1), :) +  coeffS(2, p)*P(t(n, 2), :) +  coeffS(3, p)*P(t(n, 3), :);
        end

        % Get vertices of neighbor triangles acting on this triangle
        index = ineighborE(:,n);
        r1 = P(t(index, 1), :); %get first vertex of each neighbor triangle
        r2 = P(t(index, 2), :); %get second vertex of each neighbor triangle
        r3 = P(t(index, 3), :); %get third vertex of each neighbor triangle

        %Int_temp stores the contribution of each triangle (column) to each observation point (row)
        Int_temp = potint4b(r1, r2, r3, ObsPoints);
        Int_temp(:,1) = 0; %kill self-term
        %Now weight and sum each column of Int_temp properly to get a single row
        %weightsS: row vector containing contribution of each observation point to final triangle
        Int = weightsS*Int_temp; % Exploiting dimensions of weightsS and Int_temp to ensure proper product occurs
        integrale(n, :) = Int;

        %   Center-point electric-field integrals
        temp    = repmat(Center(n, :), RnumberE, 1) - Center(index, :); %   these are distances to the
observation/target triangle
        DIST    = sqrt(dot(temp, temp, 2));                             %   single column
        I       = Area(n)*temp./repmat(DIST.^3, 1, 3);                  %   center-point integral, standard format
        I(1, :) = 0;                                                    %   self integrals will give zero
        integralxc(:, n) = -I(:, 1);    %   center-point integrals, entries of non-zero rows of n-th column
        integralyc(:, n) = -I(:, 2);    %   center-point integrals, entries of non-zero rows of n-th column
        integralzc(:, n) = -I(:, 3);    %   center-point integrals, entries of non-zero rows of n-th column

    end
    integralTime = toc
    tic
    delete(gcp('nocreate'));
    parpoolShutdownTime = toc

    tic
    %% Properly weight integrale with the self-triangle area instead of the neighbor-triangle area
    area_neighbor = Area(transpose(ineighborE));
    area_self = repmat(Area, 1, RnumberE);
    integrale = integrale .* area_self ./ area_neighbor;

    %%  Define useful sparse matrices EC, PC (for GMRES speed up)
    const           = 1/(4*pi);
    integralc       = zeros(RnumberE, N);    %   normal integral component for array of neighbor triangles (center
point) - to speed up GMRES

    for n = 1:N                  %   inner integral; (n =1 - first column of the system matrix, etc.)
        index = ineighborE(:, n); %   those are non-zero rows of the system matrix for given n
        integralc(:, n)  =       +(integralxc(:, n).*normals(index, 1) + ...
                                   integralyc(:, n).*normals(index, 2) + ...
                                   integralzc(:, n).*normals(index, 3));
    end

    ii  = ineighborE;
    jj  = repmat([1:N], RnumberE, 1);
    EC  = sparse(ii, jj, const*(-integralc + transpose(integrale)));               %   almost symmetric

    ECConstructionTime = toc
    save('integrals_test', 'integrale', 'integralc');
end
```

AA.2. Coincident Facet Resolution

```
function [P, t, normals, centroids, areas, Indicator, condin, condout, contrast] = clean_coincident_facets(P, t,
normals, centroids, areas, Indicator, tissue, condin, condout, contrast)
%   This function searches for facets that are duplicates of other facets, or otherwise have centroids that are
%   too close together to be properly treated by BEM-FMM.  It removes one copy of each of these duplicate facets
%   to ensure that the algorithm executes properly.

%   Copyright WAW/SNM 2019-2020

    %---Find nearest neighbors for every facet---
    disp('  Evaluating nearest neighbors ...');
    [index, DIST] = knnsearch(centroids, centroids, 'k', 2);
    %Now find entries where DIST is zero
    index_trimmed = index(DIST(:,2) < 100*eps,:);
    index_trimmed = sort(index_trimmed, 2, 'descend');  %Higher index first in each row
    index_trimmed = sortrows(index_trimmed); %Order rows lowest to highest
    k = reshape(1:size(index_trimmed, 1), [],1); %Every other row is a duplicate
    index_trimmed(mod(k,2) == 1,:) = [];     %Delete duplicate rows

    %Now find the facets associated with each index
    facetList1 = t(index_trimmed(:,1), :);
    facetList2 = t(index_trimmed(:,2), :);

    %---Check that coincident facets have identical vertices---
    disp('  Evaluating coincident facets ...');
    coincidentFacetsCounter = 0;
    coincidentCentroidsCounter = 0;
    coincidentFacets = zeros(size(facetList1,1), 2);
    coincidentCentroids = zeros(size(facetList1,1), 2);
    for j = 1:size(facetList1,1)
        %Can probably do this outside the for loop - assign vertices1 and
        %vertices2 exactly as they are, but replace 'j' with ':'.  If it turns
        %out that one pair of rows doesn't match, take int(rownumber)/3+1 to
        %find offending entry in facetList.
        vertices1 = [P(facetList1(j, 1), :); P(facetList1(j, 2), :); P(facetList1(j,3),:)];
        vertices2 = [P(facetList2(j, 1), :); P(facetList2(j, 2), :); P(facetList2(j, 3),:)];
        %Make sure the vertices are listed in the same order
        % (Need to do this intelligently if we want to take this outside the
        % for loop)
        vertices1 = sortrows(vertices1);
        vertices2 = sortrows(vertices2);
        %Check for coincident vertices
        if(all(abs(vertices1 - vertices2) < eps))
            coincidentFacetsCounter = coincidentFacetsCounter + 1;
            coincidentFacets(coincidentFacetsCounter,:) = index_trimmed(j,:);
        else
            coincidentCentroidsCounter = coincidentCentroidsCounter + 1;
            coincidentCentroids(coincidentCentroidsCounter,:) = index_trimmed(j,:);
        end
    end
    %Clean out unused rows
%     coincidentFacets(coincidentFacets == 0) = [];
%     coincidentCentroids(coincidentCentroids == 0) = [];
    coincidentFacets( (coincidentFacets(:, 1) == 0) & (coincidentFacets(:, 2) == 0), : )= [];
    coincidentCentroids( (coincidentCentroids(:, 1) == 0) & (coincidentCentroids(:, 2) == 0), : )= [];
    disp(['  Found ' num2str(coincidentFacetsCounter) ' duplicate facets']);
    if(coincidentCentroidsCounter ~= 0)
        % For coincident centroids but not coincident facets: pull the offending centroids' vertices apart by a very small distance.
        warning(['Found ' num2str(coincidentCentroidsCounter) ' facets with coincident centroids that do not have coincident vertices.  Resolving by perturbing vertices of both meshes']);
        centroids(coincidentCentroids(:, 2), :) = centroids(coincidentCentroids(:, 2), :) - 1e-8 * normals(coincidentCentroids(:, 2), :);
    end

    %---Update conductivity information for duplicated facets---
    disp('  Resolving duplicate facets ...');
    if(~isempty(coincidentFacets))
        keepFacet = coincidentFacets(:,1);
        deleteFacet = coincidentFacets(:,2);
        condout(keepFacet) = condin(deleteFacet);
        contrast(keepFacet) = (condin(keepFacet) - condout(keepFacet))./(condin(keepFacet) + condout(keepFacet));
        contrast(isnan(contrast)) = 0; %

        %---Now, delete duplicated facets and all associated information---
        areas(coincidentFacets(:,2),:) = [];
        centroids(coincidentFacets(:,2),:) = [];
        Indicator(coincidentFacets(:,2),:) = [];
        normals(coincidentFacets(:,2),:) = [];
        t(coincidentFacets(:,2),:) = [];
        condin(coincidentFacets(:,2),:) = [];
        condout(coincidentFacets(:,2),:) = [];
        contrast(coincidentFacets(:,2),:) = [];
    end

    %Remove unreferenced vertices
    [P, t] = fixmesh(P, t, 0);
end
```

AA.3. Observation Point Material Assignment

```
function [id] = assign_tissue_type_volume(obsPoints, n, Center, Indicator)
    %This function determines the tissue type that exists at user-specified
    %observation points based on proximity to model surfaces.
    % obsPoints (Mx3): Observation points whose tissue types are to be determined
    % n (Nx3): normal vectors of model facts
    % Center (Nx3): centroids of model facets
    % Indicator (N): tissue codes of model facets

    % id (M): tissue codes for observation points
    %Copyright WAW/SNM 2020

    %Preallocate
    distTriangles = zeros(size(obsPoints, 1), max(Indicator));
    pointInsideShell = -ones(size(obsPoints, 1), max(Indicator));
    %Construct matrices telling which surfaces enclose/don't enclose which observation points
    for j = 1:max(Indicator)
        eligibleCenters = Center(Indicator == j, :);
        eligibleN = n(Indicator == j, :);

        %For the current tissue, find the triangle closest to each observation point
        [nearestTriangles, distTriangles(:,j)] = knnsearch(eligibleCenters, obsPoints, 'K', 1);

        %Get vectors from observation points to nearest triangles
        r = eligibleCenters(nearestTriangles, :) - obsPoints;

        %If the dot product of [vector from point to triangle center] and [triangle normal] is positive, the point is
inside the current shell.
        pointInsideShell(:,j) = dot(r, eligibleN(nearestTriangles, :), 2) > 0;
    end

    %Knowing which observation points are within which surface, find the tissue type at the observation point
    id = zeros(size(obsPoints, 1), 1);
    for j = 1:size(obsPoints, 1)
        %Catch the case where the observation point lies completely outside the model
        if(all(pointInsideShell(j,:) == 0))
            id(j) = 0;
            continue;
        end

        %The tissue type assigned to the point is the tissue type of the closest outside shell
        insideIndices = find(pointInsideShell(j,:));
        tempDistances = distTriangles(j, insideIndices);
        [~,insideIndicesIndex] = min(tempDistances);
        id(j) = insideIndices(insideIndicesIndex);
    end
end
```

Appendix B: TMS-Focused BEM-FMM Toolkit Manual

This appendix is excerpted from Appendix A to [1].

AB.1. Use, System Requirements, and Third-Party Components

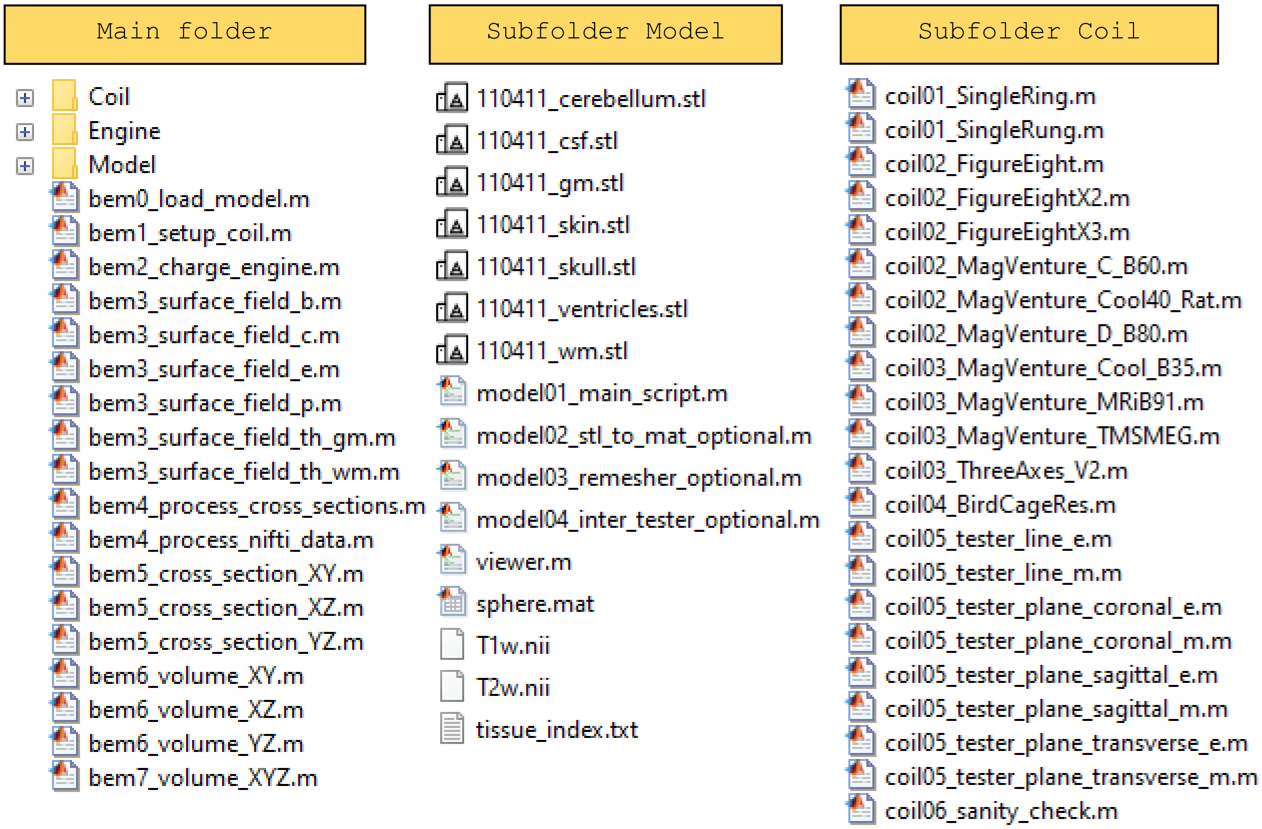
The toolkit is intended for academic use only. The software platform is MATLAB 2019a or newer (Windows/Linux). While the Windows implementation is stable and fast, the Linux implementation of the method may require extra recompilation of the FMM distributables [66] as described in the FMM software manual.

The following toolboxes (usually supplied with the MATLAB Academic Package) are required: Image Processing Toolbox (for NIfTI data processing), Partial Differential Equations or Antenna Toolbox (for model remeshing), and Statistics and Machine Learning Toolbox (for geometrical search of nearest neighbors used in the volumetric fields plots). Those toolboxes are not absolutely necessary, but the TMS toolkit must be modified to operate without them, and its performance will somewhat degrade. The FMM engine [66] and example setups with SimNIBS segmentation [46] of Human Connectome Project subjects 101309, 110411, 117122, 120111, 122317, 122620, 124422, 128632, 130013, 131722, 138534, 149337, 149539, 151627, 160123, and 198451 [78], as well as example setups with the SimNIBS Ernie model [43] and the IT'IS Foundation's MIDA model [71], have been included with permission in the redistributable software package. A GitHub folder [54] contains the base code for Human Connectome Project subject 110411. An additional Dropbox folder contains example setups for all 18 head models referenced above.

AB.2. Toolkit Organization

Multiple copies of the toolkit, each using a different head model, are available at the Dropbox repository. For purposes of this walkthrough, we focus specifically on the "light" toolkit containing Connectome Subject 110411 (`TMS_Master_Spring_2020`). The other toolkit copies differ primarily in the contents of their `Model` subfolders and in the coil positioning transformations applied in `bem1_setup_coil.m`.

Each toolkit contains a number of short MATLAB or MATLAB-compatible scripts organized within three subfolders – `Model`, `Coil`, and `Engine` – and a number of scripts located in the main folder, as shown in Fig. A1.



Main folder

Coil
Engine
Model
bem0_load_model.m
bem1_setup_coil.m
bem2_charge_engine.m
bem3_surface_field_b.m
bem3_surface_field_c.m
bem3_surface_field_e.m
bem3_surface_field_p.m
bem3_surface_field_th_gm.m
bem3_surface_field_th_wm.m
bem4_process_cross_sections.m
bem4_process_nifti_data.m
bem5_cross_section_XY.m
bem5_cross_section_XZ.m
bem5_cross_section_YZ.m
bem6_volume_XY.m
bem6_volume_XZ.m
bem6_volume_YZ.m
bem7_volume_XYZ.m

Subfolder Model

110411_cerebellum.stl
110411_csf.stl
110411_gm.stl
110411_skin.stl
110411_skull.stl
110411_ventricles.stl
110411_wm.stl
model01_main_script.m
model02_stl_to_mat_optional.m
model03_remesher_optional.m
model04_inter_tester_optional.m
viewer.m
sphere.mat
T1w.nii
T2w.nii
tissue_index.txt

Subfolder Coil

coil01_SingleRing.m
coil01_SingleRung.m
coil02_FigureEight.m
coil02_FigureEightX2.m
coil02_FigureEightX3.m
coil02_MagVenture_C_B60.m
coil02_MagVenture_Cool40_Rat.m
coil02_MagVenture_D_B80.m
coil03_MagVenture_Cool_B35.m
coil03_MagVenture_MRiB91.m
coil03_MagVenture_TMSMEG.m
coil03_ThreeAxes_V2.m
coil04_BirdCageRes.m
coil05_tester_line_e.m
coil05_tester_line_m.m
coil05_tester_plane_coronal_e.m
coil05_tester_plane_coronal_m.m
coil05_tester_plane_sagittal_e.m
coil05_tester_plane_sagittal_m.m
coil05_tester_plane_transverse_e.m
coil05_tester_plane_transverse_m.m
coil06_sanity_check.m

Fig. A1. Low-level organization chart of the toolkit.

The main folder and three subfolders are organized as follows:

The main folder contains all major computational scripts which define coil/head position, perform computations, and output electric fields both on surfaces and in volume. If NIfTI data are available, surface meshes and fields can be registered against NIfTI slices using the built-in NIfTI viewer.

The subfolder `Model` contains the head model that will be used for analysis. It also contains tools for remeshing (coarsening or refining) the head model and for performing necessary precomputations, such as double potential integrals for neighbor facets.

The subfolder `Coil` is devoted to coil definition, construction of the coil wire and CAD models, and, optionally, separate coil testing/optimization.

The subfolder `Engine` contains computational scripts and functions serving different purposes, including the BEM-FMM engine.

All scripts can be changed/modified and rearranged to organize parametric loops if necessary. The scripts of the main folder can be executed at any time for the default configuration.

AB.3. Default Application Example

In the following computation example, we will consider Connectome subject model #110411 and the MRi-B91 TMS-MRI coil model targeting the hand area of the primary motor cortex, located above the precentral gyrus of the right hemisphere (the hand knob area). The tissue conductivity values are those of SimNIBS TMS software package, though others may be used. The coil is driven with a time-varying current of $\frac{dI}{dt} = 9.4e7\ Amperes/sec.$

AB.4. Coil Selection and Analysis (subfolder `Coil`)

AB.4.a. Coil selection

Start with subfolder `Coil`. Here, dedicated MATLAB scripts generate coil models (both wire and CAD), with *one* script per coil. The coil conductor centerline is defined manually using either an analytical formula or a set of points in three dimensions. After that, the corresponding script automatically generates the volumetric computational wire grid coil model and the coil CAD model. The default coil axis is the *z*-axis. Run scripts `coil01*` through `coil03*`. This generates several coil models, some of which are shown in Fig. A2.

Finally, run the script `coil03_MagVenture_MRiB91.m`. This will generate and save the MRI compatible TMS coil model MRi-B91 from MagVenture, Denmark shown in Fig. A2a. This model, centered at the origin of the Cartesian coordinate system, will be used in further analysis below.

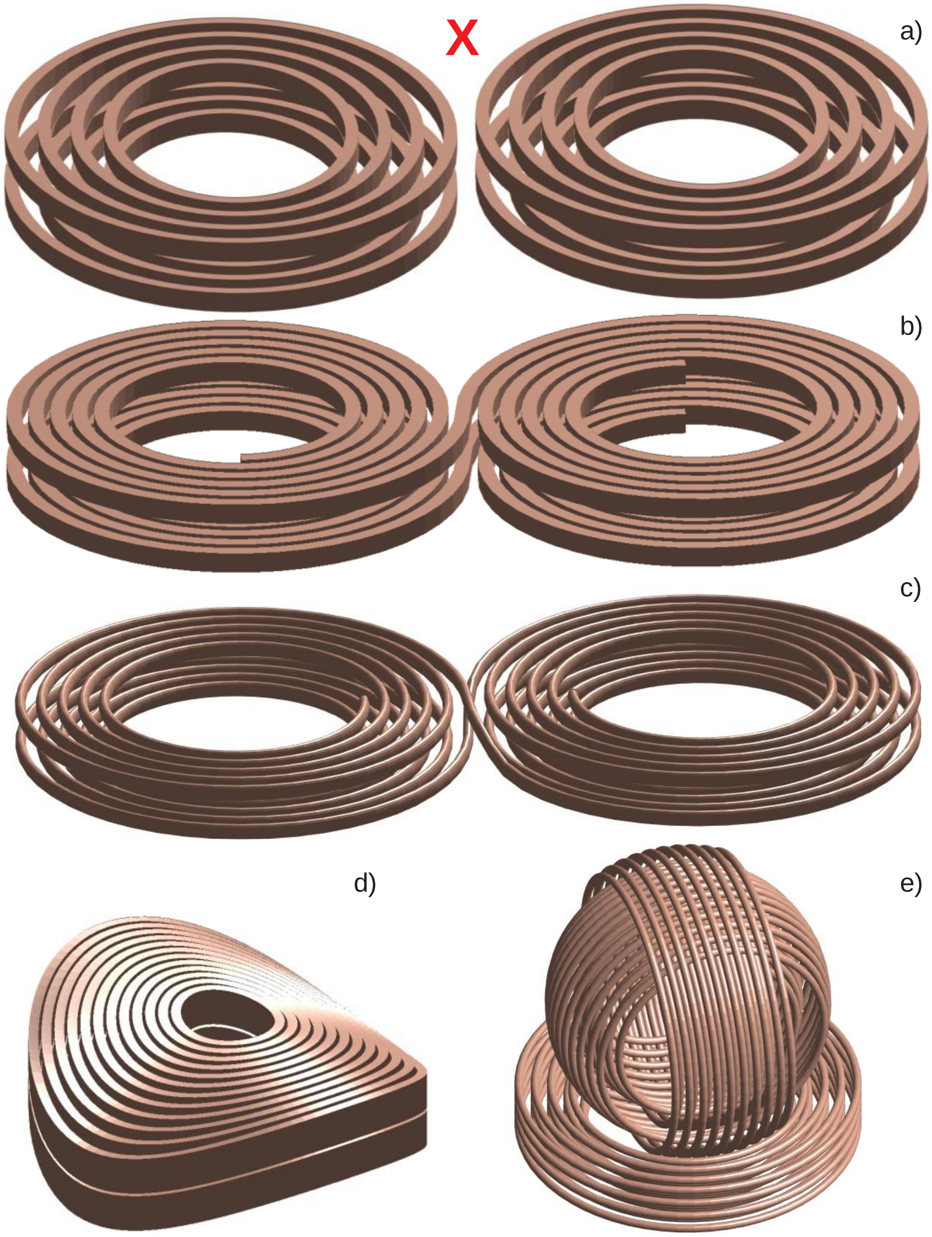


Fig. A2. Some solid CAD models created using the MATLAB-based coil geometry generator. Fig. A2a is a simplified MRi-B91 TMS-MRI coil model (MagVenture, Denmark) with elliptical conductors of a rectangular cross-section used in this example; Fig. A2b is a simplified MagPro C-B60 coil model (MagVenture, Denmark); Fig. A2c is a generic double figure-eight spiral coil model with an elliptical cross-section and two bootstrapped interconnections; Fig. A2d is a simplified Cool-40 Rat small animal coil model (MagVenture, Denmark); Fig. A2e is a three-axis multichannel TMS coil array radiator [75]. The red "X" denotes the default coil model.

AB.4.b. Coil fields in free space

After the coil model MRi-B91 has been selected, a number of scripts in subfolder `Coil` will allow us to examine the coil's electric and magnetic fields. These are line plots (`coil05_tester_line_e/m.m`, Fig. A3a,b) as well as high-resolution 2D contour plots (`coil05_tester_plane_coronal_e/m.m` etc.) for any component of the electric and/or magnetic field in the coronal, sagittal, and transverse planes. The coil's time-varying current, dI/dt, must be specified at the beginning of every script. For the magnetic field, only the steady-state current, I_0, is required. Scripts `coil05_tester_plane_coronal_e/m.m` may be run to obtain Fig. A3c,d. Note that these scripts also define the observation plane window. This window will not be reused for head-coil computations of the main folder.

While performing mathematical (mesh generation, FMM computations) and graphical operations, these scripts call several functions from the subfolder `Engine`. A coil mesh generator script and a field computation script may be further combined into one script and augmented with a parametric loop to enable coil analysis and design (cf., for example,[53]).

AB.4.c. Changing coil geometry/optimizing coil fields

All geometry parameters are to be given in the respective coil scripts. This coil is constructed as many elliptical coaxial rings of a finite cross-section where the ring axis is the *z*-axis. The script introduces the coil geometry by defining intersections of the conductor centerlines with the *xz*- and *yz*-planes, respectively. Parameters at the beginning of the script `coil03_MagVenture_MRiB91.m` define the conductor's characteristics. Rectangular (`flag 2`) and elliptical (`flag 1`) cross-sections are permitted.

Both the computational wire grid and the coil CAD model are generated by the function `meshcoil.m`. This function is specifically applicable to a particular coil geometry consisting of a (large) number of concentric loops; it creates the wire coil model all at once. The input are intersection points of the loop centerlines with *xz*- and *yz* planes. Either a Litz wire model (parameter `sk = 0`) or a skin-layer model (parameter `sk = 1`) may be used. In the former case, the current distribution across a conductor's cross-section is nearly uniform. In the latter case, the wire grid is situated close to the surface of the conductor. The density of the wire grid depends on the cross-section triangulation; it is controlled by parameter `M` – the number of cross-section subdivisions. The grid resolution in the direction of the conductor centerline is controlled by the original centerline discretization.

Function `meshwire.m` of subfolder `Engine` is more general than `meshcoil.m`. This function creates the wire mesh for an arbitrary *single* conductor. Either closed loops (`coil01_SingleRing.m`) or open conductors (`coil01_SingleRung.m`) may be generated. The computational wire grid coil model consists of straight, short, infinitely-thin current filaments or segments. The current filaments are defined as short straight lines joining centroids of triangles of the cross-sectional mesh, which are replicated along the conductor's centerline as many times as required. The cross-section is always perpendicular to the conductor's centerline, so the filaments are always parallel to the conductor's centerline.
In either case, the computational coil grid is the structure `strcoil` with the following fields:

```
Pwire – nodes of elementary wires inside the conductor
Ewire – edges (start & end points) of elementary wires inside the conductor
Swire - weights of elementary wire segments given total current of 1A
```

Weights are required to ensure that the total current through the conductor's cross-section is 1 A.

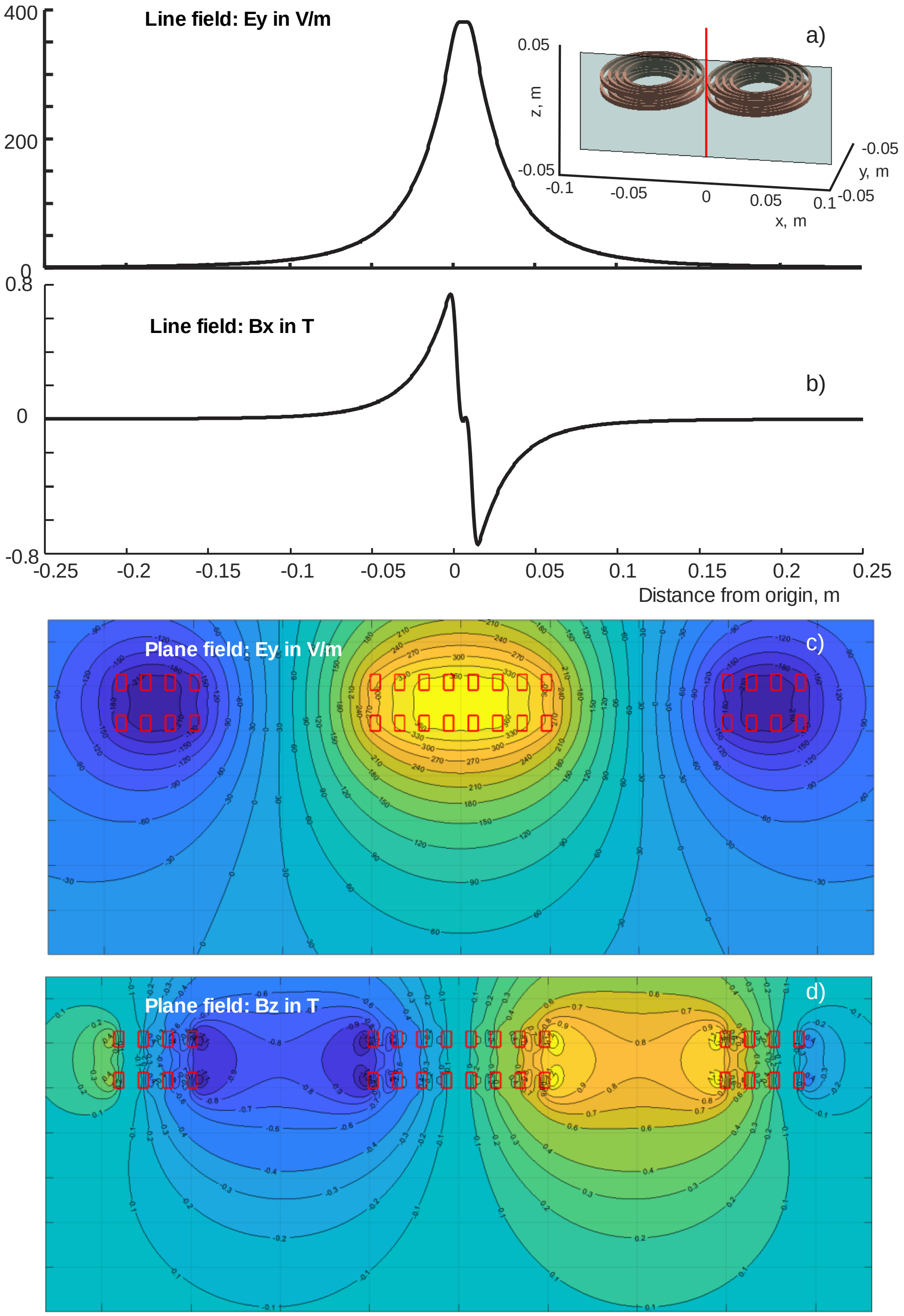


Fig. A3. Coil evaluation example. Electric (a,c) and magnetic (b,d) fields on a line or in a plane with 0.25 M observation points for the MRi-B91 TMS-MRI coil (MagVenture). Conductor cross-section is marked in red. The coil is driven with a time-varying current of $\frac{dI}{dt} = 9.4e7\ Amperes/sec$. *An equivalent definition would be a conductor current of 5 kA and a CW frequency of 3 kHz.*

A solid CAD model (as opposed to a wire mesh model) for the coil conductor can be constructed using the function `meshsurface.m` of subfolder `Engine`. This script creates a structured triangular surface mesh (comprising array of nodes `P` and array of facets `t`) for the conductor's side surface. The coil CAD model should properly define the normal vectors of the triangular surface patches and the corresponding triangle orientation. Once converted to `*.stl` format using MATLAB's built-in function `stlwrite`, this coil model may be used in FEM-based software packages (e.g. ANSYS Electronics Desktop). Fig. A4 shows a detailed concept of the combined wire/CAD coil model using in this software.

At present, the coil geometry modeler is restricted to predominantly flat or moderately bent conductor loops or nearly planar curves. H-coils with sharp conductor bends in all three planes may be constructed if necessary, but only using the circular conductor cross-section.

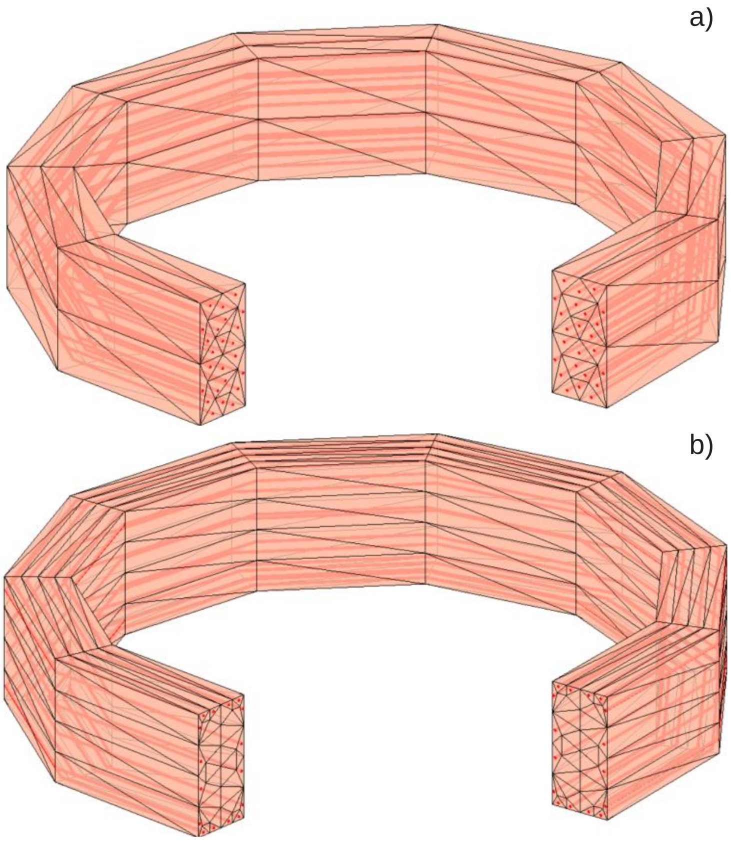


Fig. A4. Concept and construction of the coil model. Filaments of current (red) within conductor's surface CAD model are shown. a) – Uniform current distribution (Litz wire); b) – modeling the skin effect (a solid conductor at a high frequency).

AB.5. Head Model Import, Remeshing, and Analysis (subfolder `Model`)

AB.5.a. Two acceptable model formats

The head model files should always be located in the dedicated folder `Model` with contents shown in Fig. A1. The primary set are *.stl (stereolithography) files for every individual brain compartment in the form of a faceted shell. The *.stl files use triangular facets with normal vectors facing out of the shell. This is the standard output of the SimNIBS segmentation pipeline and other relevant software packages. The number of shells may be arbitrary. The script `model02_stl_to_mat_optional.m` converts *.stl files, either binary or ASCII, to equivalent MATLAB data files (using MATLAB's built-in function `stlread`) containing arrays of vertices $\boldsymbol{P}$, facets t, and normal vectors $\boldsymbol{n}$. Every MATLAB data file can further be inspected and visualized using the function `viewer.m` from the same subfolder (as shown in Fig. A5a below). Repeat this last operation for every brain compartment in the folder.

AB.5.b. Built-in head models

For computational studies that do not involve MRI data collection, the Dropbox location provides 16 realistic head models for 16 Connectome Project [78] subjects with isotropic voxel resolution of 0.7 mm. These are subjects #101309, 110411, 117122, 120111, 122317, 122620, 124422, 128632, 130013, 131722, 138534, 149337, 149539, 151627, 160123, and 198451. The datasets have been converted to surface models with the help of the SimNIBS 2.1 pipeline; every model includes seven brain compartments (skin, skull, CSF or cerebrospinal fluid, GM or gray matter, WM or white matter, ventricles, cerebellum). Every model has been checked and confirmed against the original NIfTI images and with regard to mesh manifoldness [72]. The default average cortical surface mesh edge length is 1.5 mm, the cortical nodal density is 0.55 nodes per mm^2, and the total number of facets is 0.9 M.

In addition to the Connectome Project head models, the Dropbox location also includes the default example model of the SimNIBS 2.1 pipeline, the Ernie model. This model is comparable in complexity to the Connectome models, with 0.9M facets and seven tissue meshes. The final model included is the MIDA model [71] with 11M facets and 117 tissues.

Any other surface model obtained from SimNIBS pipeline may be used in *.stl or *.mat (MATLAB) format. In particular, fifty CAD models, known as the Population Head Model Repository or PHM [79],[80] have been made available from the website of the IT'IS Foundation, Switzerland [81].

The default head geometry in the folder `Model` is subject 110411 with the following seven 2-manifold watertight enclosed brain compartments: white matter (WM), gray matter (GM), cerebrospinal fluid (CSF or inner skull), skull, skin, cerebellum, and ventricles. These brain compartments, with the exception of the cerebellum and ventricles, are shown in Fig. A5a.

AB.5.c. Processing NIfTI Data

NIfTI data (if available) should be located in the same subfolder `Model` as shown in Fig. A1. For example, the Connectome Project database contains T1 and T2 NIfTI data for every subject, which were made available with permission. The default application example uses subject 110411.

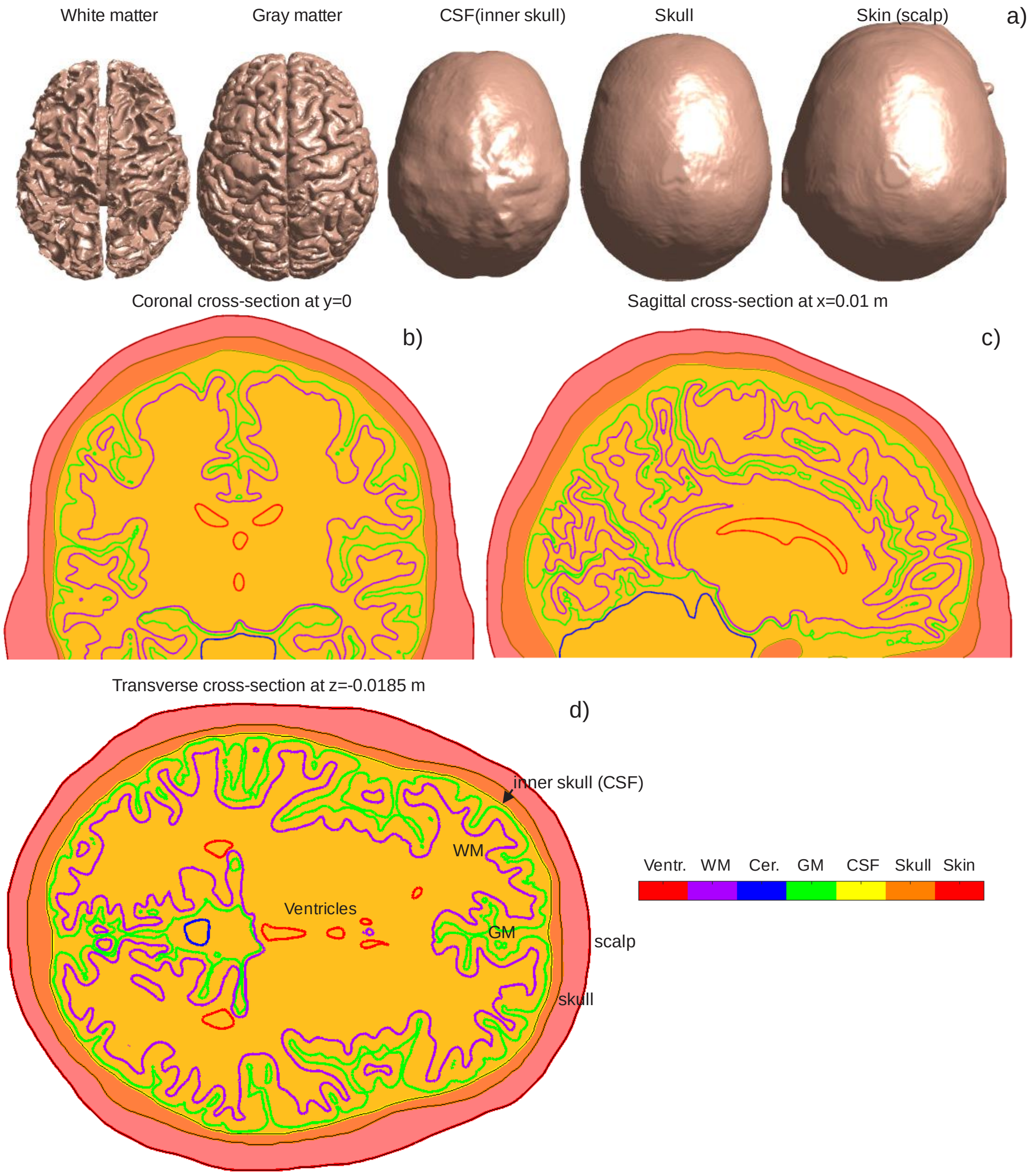


Fig. A5. a): Brain compartments of the default package head model # 110411: white matter (WM), gray matter (GM), cerebrospinal fluid (CSF or inner skull), skull, and skin. b-d): Head cross-sections in three principal planes.

AB.5.d. Model remeshing

A CM2 SurfRemesh® remeshing program from Computing Objects, France is included in the MATLAB package. This software enables creation of coarser and/or finer surface representations while minimizing the surface deviation error from the master mesh. MATLAB script `model03_remesher_optional.m` performs automated remeshing to any required maximum edge length, which should be given at the beginning of the script.

For example, the remeshing program generates a coarser model with the average cortical edge length of 1.9 mm and the average cortical nodal density of 0.32 nodes per mm^2 when the maximum edge length is chosen as 3 mm; the total number of facets is 0.4 M. On the other hand, the same program generates a finer model with the average cortical edge length of 0.99 mm and average cortical nodal density of 1.2 nodes per mm^2 when the maximum edge length is chosen as 1 mm; the total number of facets is 1.8 M. Fig. A6 shows the corresponding surface meshes for the gray matter shell along with the original segmentation. The red circle labels the targeted stimulation area close to the precentral gyrus crown. The remeshing procedure may require significant time.

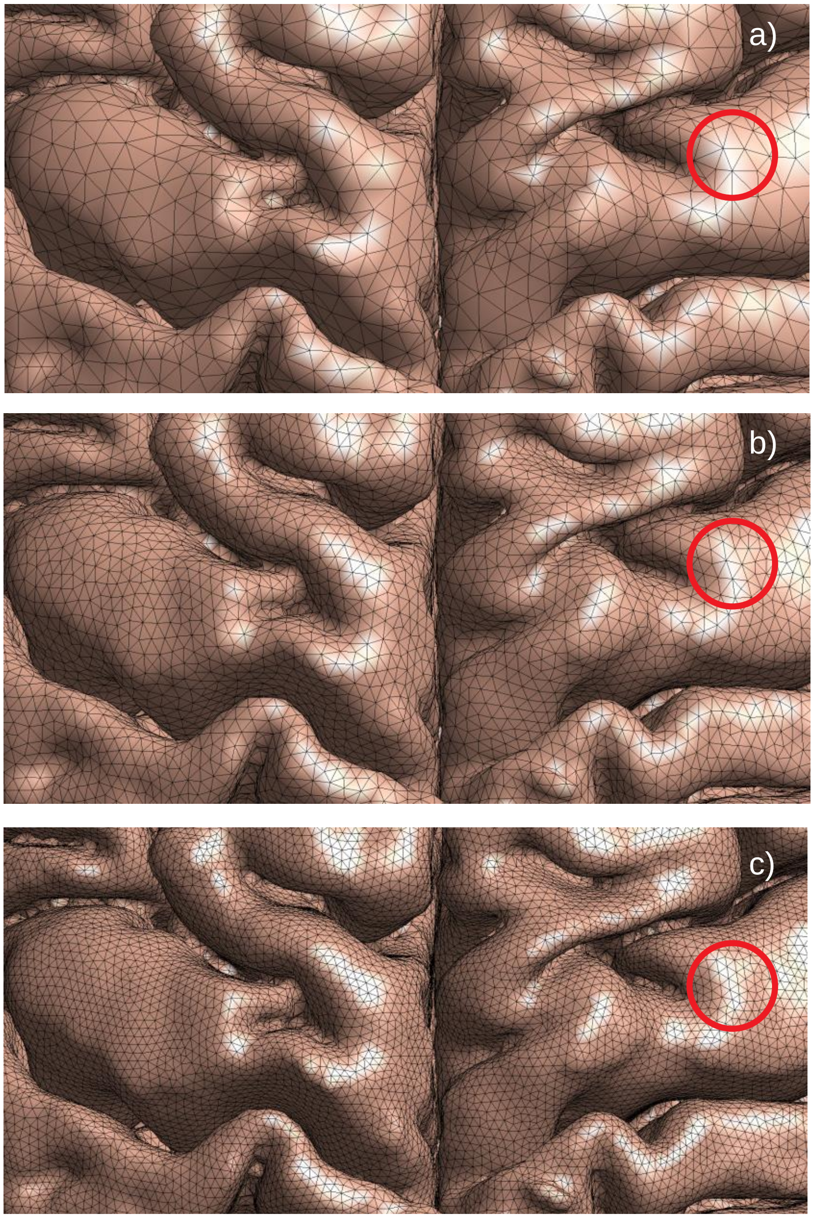


Fig. A6. a) – Coarser model with average cortical edge length of 1.9 mm and average cortical nodal density of 0.32 nodes per mm^2; b) – default meshing with average cortical surface mesh edge length of 1.5 mm and average cortical nodal density of 0.55 nodes per mm^2; c) – finer model with average cortical edge length of 0.99 mm and average cortical nodal density of 1.2 nodes per mm^2. The red circle labels the targeted stimulation area close to the precentral gyrus crown.

AB.5.e. Creating combined head mesh

The combined mesh for the entire head is created by appending individual meshes. This is done by running the script `model01_main_script.m`. The combined mesh is stored in the MATLAB data file `CombinedMesh.mat`. An additional data file, `CombinedMeshP.mat`, is generated in the same folder. This file contains precomputed double surface electrostatic integrals over triangles necessary for accurate BEM-FMM simulations. The default (minimum) number of neighbors is 3. Integrals are computed in parallel, using 15 cores by default. The `numThreads` variable on line 85 of `model01_main_script.m` may be adjusted depending on the computer configuration. Run the script `model01_main_script.m`.

The script `model01_main_script.m` reads from a tissue index file (always named `tissue_index.txt`) in the `Model` subfolder to determine which `*.mat` tissue files to assemble into the final model and what conductivity values should be assigned to each of those tissues. Each line of a tissue index file provides the following information: tissue name (for reference in subsequent scripts), tissue source file, tissue conductivity, and enclosing tissue. It then assigns initial conductivity information to each facet of each tissue: the facet's interior conductivity (in the opposite direction of the facet's normal vector), the facet's exterior conductivity (in the direction of the facet's normal vector), and the conductivity contrast across the facet.

This script also checks the combined mesh for duplicate facets and for facets whose centroids are too close to be treated with the BEM-FMM algorithm. For the Connectome models, there should be none of these complications, because tissues of these models surround and enclose each other without touching – they are hollow shells, where each shell segments a boundary between exactly two tissue types. For the MIDA model, however, the interior and exterior boundaries of every tissue are explicitly segmented. This means, for example, that the MIDA model's white matter and gray matter both independently segment their mutual boundary, producing a large number of duplicate facets. These duplicate facets would produce singularities that invalidate simulation results, so they are resolved as follows.

For each pair of duplicate facets, one is designated the facet to be deleted, and the other is designated the facet to be kept. The outer conductivity of the facet to be kept is set equal to the inner conductivity of the facet to be deleted, and associated conductivity contrast information is updated for the facet to be kept. The facet to be deleted, and all associated information, is then removed from the model. Fig. A7 below illustrates the results of this operation.

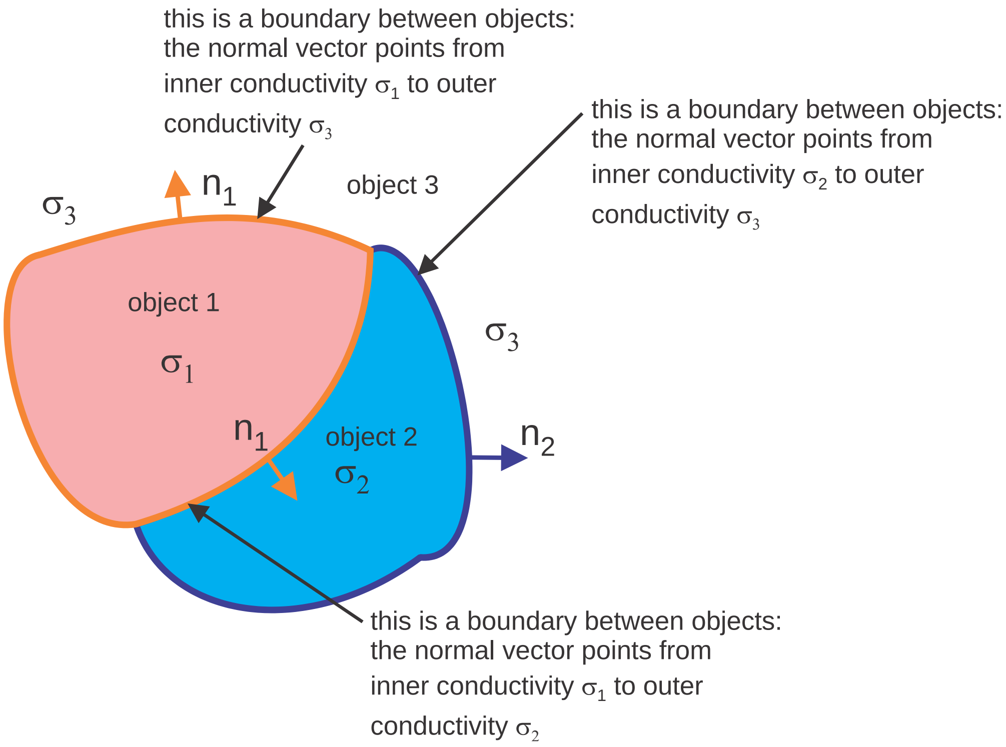

this is a boundary between objects: the normal vector points from inner conductivity σ1 to outer conductivity σ3
n1
object 3
this is a boundary between objects: the normal vector points from inner conductivity σ2 to outer conductivity σ3
σ3
object 1
σ1
σ3
n1
object 2
σ2
n2
this is a boundary between objects: the normal vector points from inner conductivity σ1 to outer conductivity σ2

Fig. A7. Object 3 (with interior conductivity σ_3) surrounds and encloses both Object 1 (with interior conductivity σ_1) and Object 2 (with interior conductivity σ_2), so Object 1 and Object 2 initially list σ_3 as the exterior conductivity for all facets in their respective meshes. Because Object 1 and Object 2 have each explicitly segmented their mutual interface, that interface initially contains coincident facets contributed by both objects. In this example, Object 2's copies of the interface facets have been removed, and Object 1's copies of the facets remain. Object 1's facets at the interface still list σ_1 as their interior conductivity, but have changed their exterior conductivity from σ_3 to σ_2.

AB.6. Computational Workflow (main folder)

The computations are performed in the main folder. The scripts of the main folder should be executed *sequentially*.

Go to the main folder and execute the script `bem0_load_model.m` first. This script imports head model data into the MATLAB workspace and sets the MATLAB path. It also imports the previously computed solution if available. Next (and before running the simulations), the coil position above the head as in Fig. A8 must be defined.

AB.6.a. Coil positioning/tissue properties

Run the script `bem1_setup_coil.m`. This script

(i) initializes the coil's time-varying current, dI/dt;

(ii) defines the steady-state current, I_0, which is necessary to compute the magnetic field;

(iii) defines coil position above the head by proper rotation and translation. The corresponding steps include coil rotation about its axis, tilt, and translation;

(iv) determines coil centerline or another observation line;

(v) displays the combined head-coil geometry (skin, skull, GM, or WM shell) as shown in Fig. A8; and

(vi) displays (nearest) intersection points between the tissues and the coil's centerline; also displays the corresponding distances from the bottom of the coil to the intersection points.

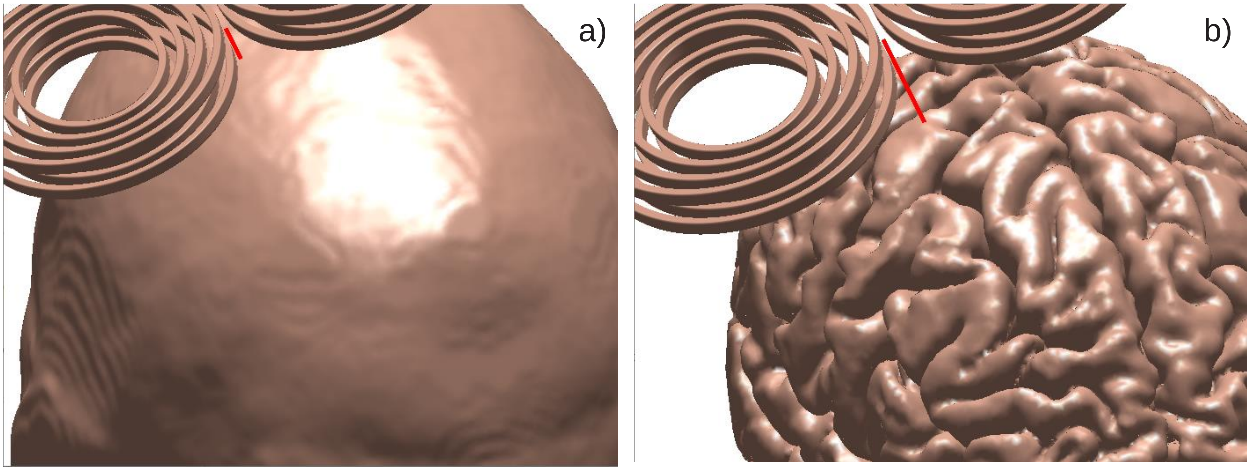


Fig. A8. Output of the script* `bem1_setup_coil` *for the same coil-model configuration: a) Coil position above the skin shell. The distance from the coil bottom to the skin shell along the coil centerline is 10.8 mm; b) – Coil position above the GM shell. The distance from the coil bottom to the GM shell along the coil centerline is 26.2 mm.

Coil position adjustment may be performed by running the script `bem1_process_data` multiple times. When performing mathematical (e.g. coil rotation) and graphical operations, this script calls several functions from subfolder `Engine`.

Specifically, coil positioning is done in three steps starting at line 25:

```
coilaxis        = [0 0 1];                   %   Transformation 1: rotation axis
theta           = 0;                         %   Transformation 1: angle to rotate
Nx = +0.45; Ny = 0.0; Nz = 1.0;              %   Transformation 2: New coil centerline
MoveX = +42e-3; MoveY = 0; MoveZ = 79.5e-3;%   Transformation 3: New coil position
```

AB.6.b. BEM-FMM engine

The next script to be executed is `bem2_charge_engine.m`. This script

(i) computes the primary field of the coil on every head interface (face nodes then face centers by interpolation) using the FMM;

(ii) computes the iterative solution of the BEM integral equation for the induced surface charge density using the FMM, precomputed near-field potential integrals, and MATLAB GMRES (generalized minimum residual method [141]); and

(iii) displays the time for every iteration step in the MATLAB command window and plots the entire convergence history when completed.

Fig. A9a shows the typical convergence rate (relative residual of the iterative solution); Fig. A9b is the same result when the charge conservation law is ignored; and Fig. A9c is the same result when the neighbor potential electrostatic integrals are replaced by the crude center-point approximation.

While the convergence without the charge conservation law might appear acceptable for this particular example, this solution typically converges to an incorrect result where charges accumulate at the sharp boundaries of the bottom of the head.

As for the near-field integration accuracy of the default example, three double potential integrals for three neighbor triangular patches (default value is given in the script `model02_mesh_combiner.m`) are computed precisely. For non-neighbor triangles, the center-point approximation is used for the double potential integrals and FMM. The number of neighbor integrals can be increased at the expense of a larger storage.

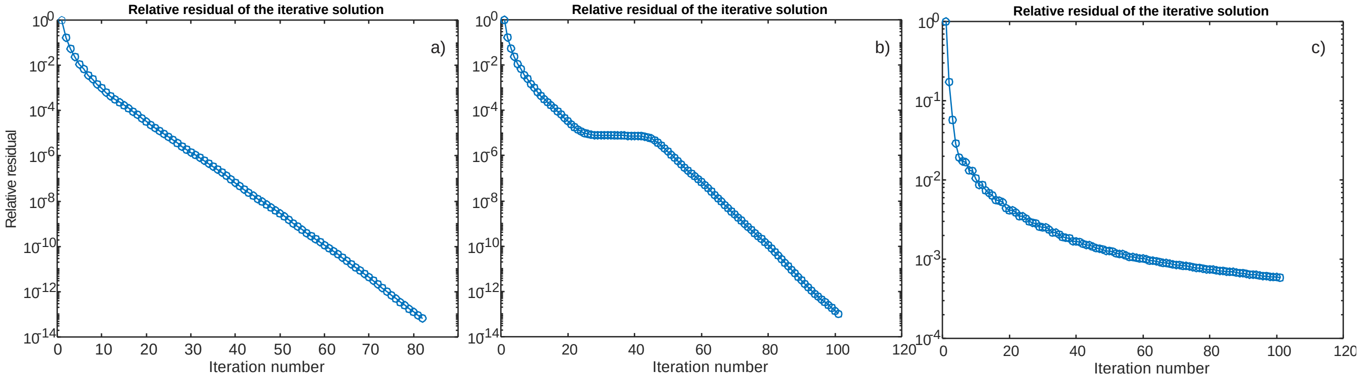


Fig. A9. a) –typical convergence rate of the solution for the default example; b) –the same result when the charge conservation law is ignored; and c) –the same result when the neighbor potential electrostatic integrals are replaced by the center-point approximation.

AB.6.c. Surface charge averaging

For practical purposes, it is convenient to introduce weighted surface charge averaging (i.e., to low pass filter the surface charge density). The default option averages over the target facet and its three immediate topological neighbor triangles. After the solution is obtained, we substitute in the script `bem2_charge_engine.m`

```
c = (c.*Area + sum(c(tneighbor).*Area(tneighbor), 2))./(Area +
sum(Area(tneighbor), 2));
```

This rule can be modified if necessary.

When performing mathematical (FMM) operations, the scripts of this folder call original and derived FMM functions from the subfolder `Engine`.

AB.7. Fields Output (main folder)

AB.7.a. Fields just inside/outside tissue interfaces

The script `bem3_surface_field_e` computes (via the FMM) and displays the generally discontinuous electric field just inside or outside any head compartment. It can be the total, normal, or the tangential field. The tangential field component is continuous through the interface. The script `bem3_surface_field_m` performs a similar operation for the magnetic field, which indeed remains continuous across boundaries. Scripts `bem3_surface_field_c.m` and `bem3_surface_field_p.m` perform the same operation for surface charge density and surface potential. Fig. A10 shows the output of `bem3_surface_field_m.m` and `bem3_surface_field_e.m` for the gray matter interface. The total electric field just inside the gray matter shell is plotted.

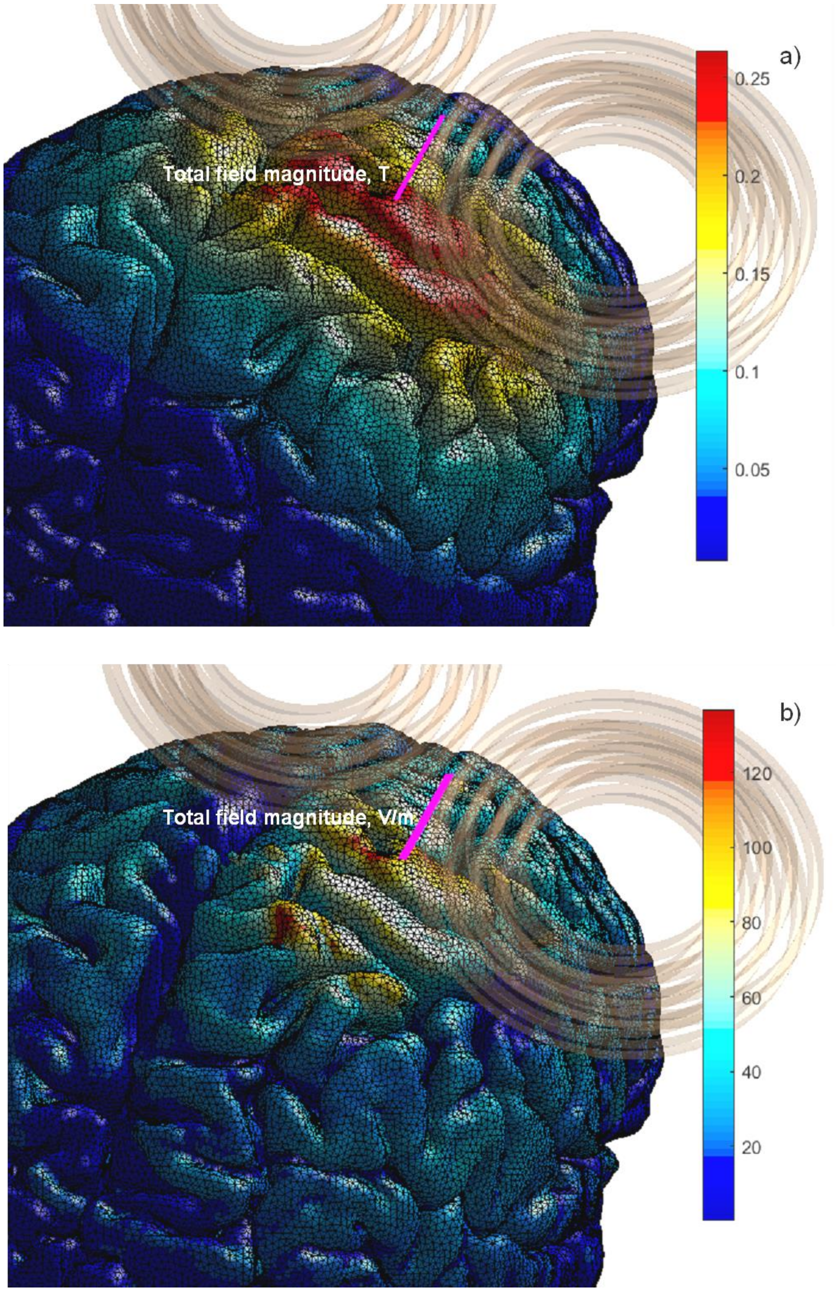


Fig. A10. a) – Magnitude of total magnetic field on the gray matter surface. b) – Magnitude of total electric field just inside the gray matter shell (the top level of the cortical layer).

AB.7.b. Normal fields just inside/outside tissue interfaces

Due to the geometry and electrophysiological characteristics of cortical neurons, the electric field component perpendicular to the cortical layer might be significant for neurostimulation. The scripts `bem3_surface_field_th_gm/wm.m` display the normal field component just inside or just outside the gray matter and white matter surfaces, respectively. Those components are directly obtained from the already-known surface-charge solution without extra computations.

To better estimate the focal area for the normal component at the interfaces, a small blue sphere is drawn at the center of each GM/WM facet where the absolute field value is in the range 80-100% of the maximum field value observed just inside (or outside) the shell. The parameter of 80% is the field threshold margin; it is assigned in the script `bem1_setup_coil.m` as

```
margin = 0.80;
```

and can be changed at any time if necessary.

Fig. A11 shows the corresponding display for the normal fields just inside the GM and WM shells, respectively, for the default example. One can see that the normal coil field just inside the WM shell appears to be quite focal in this particular case; however, the focal area is located not directly underneath the coil.

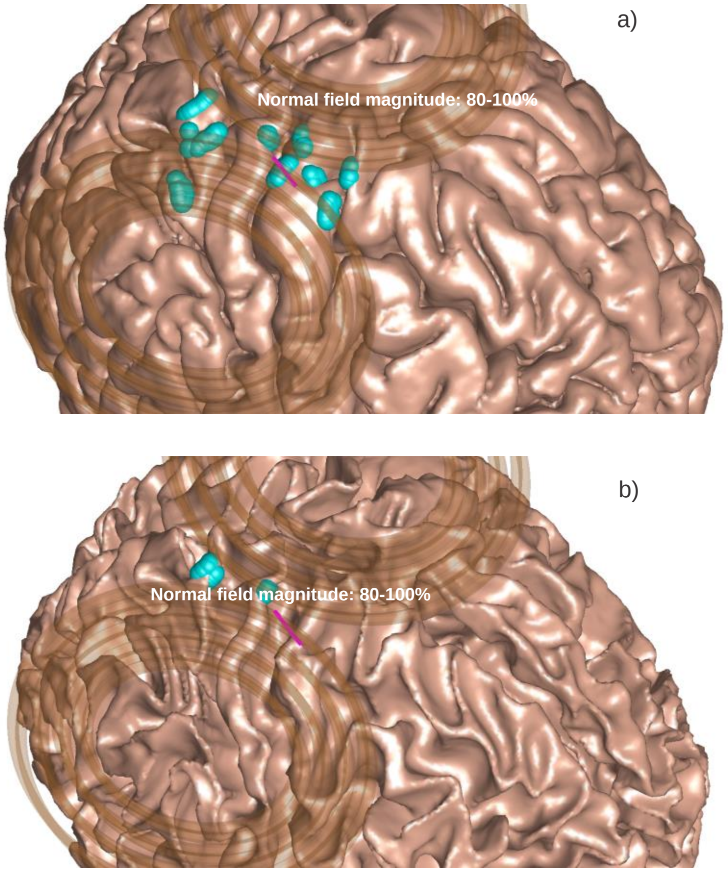

a)
Normal field magnitude: 80-100%
b)
Normal field magnitude: 80-100%

Fig. A11. Focal area of the normal field just inside GM (a) and WM (b). The blue spheres indicate facets whose normal field magnitudes are 80-100% of the corresponding maximum field magnitude.

AB.7.c. Segmentation surface cross-section precomputation

Next, run the script `bem4_process_cross_sections.m`. This script pre-computes segmentation (triangular surface mesh) cross-sections that will be used in the field output plots. The following computations are reasonably fast (real time).

If NIfTI data are available (e.g. `T1w.nii`), run the script `bem4_nifti_import.m` to import the data and find the required voxel resolution. Subsequent visualizations will superimpose the normal field onto the corresponding NIfTI slices.

AB.7.d. Field focal location in three principal planes

With precomputations complete and optional NIfTI data loaded, run the scripts `bem5_cross_section_XY/XZ/YZ.m` to view the surface field superimposed on the corresponding NIfTI slices. The default planes have been selected such that they pass through the position at which the field maximum occurs. The NIfTI field viewer runs in real time and is assembled to display a sequence of slices (a movie) if desired.

Figs. 33a-35a of the main paper text show the display for the focal area of the normal fields just within the WM shell. Here, the focal area is represented in the form of pink balls residing at the interfaces superimposed onto the corresponding NIfTI slices in three principal planes. This focal area is defined as the set of all facets whose normal E-field magnitude is 80% or higher of the maximum observed normal E-field magnitude.

AB.7.e. Volumetric fields in principal planes

The scripts `bem6_volume_XY/XZ/YZ.m` compute and output the electric field (any of its Cartesian components or a magnitude) in the three principal planes. The plane position and its size are specified at the beginning of every script. Since the previously computed mesh cross-sections are used again, these scripts should be made compatible with the previous scripts `bem5_cross_section_XY/XZ/YZ.m`. To calculate (but not visualize) the electric field in a three-dimensional grid, use the script `bem7_volume_xyz.m`.

The volumetric field computations require significantly more time since the potential integrals are no longer precomputed and must be calculated at the time of execution, depending on the position of a given observation point versus the nearest interface(s). The critical numerical parameter here is the dimensionless (vs. average triangle size) radius, R, of an integration sphere within which integration of the surface charge density is performed. Its default value is 2; higher numbers (e.g. $R = 5$) will provide better field accuracy but simultaneously slow down the computations.

Figs. 33b-35b of the main paper text illustrate the total electric field (magnitude) distribution for the default example in the three principal planes with the size of 20×20 mm each. The default field resolution (given in the same scripts) is 100 micrometers. These planes are centered at the positions of the maximum normal field just inside the white matter shell shown in Figs. 33a-35a of the main text. Direct overlap with the NIfTI data can thus be established.

AB.8. Control of Numerical Accuracy

For a given surface mesh resolution, the numerical accuracy of the method is controlled by the following parameters:

1. `RnumberE`, found in `Model\model01_main_script.m`: Number of neighbor potential electrostatic double surface integrals (electric field + electric potential) to be computed precisely. The default number is 3. The maximum number is unlimited, but is subject to memory constraints. Numbers above 16 do not affect the overall solution accuracy significantly.

2. `prec`, found in `Engine\bemf4_surface_field_electric.m` and in all other FMM-related sxripts: Intrinsic FMM precision (Gimbutas et al., 2019). The default value is 0.1-0.01 (10-1%). Values smaller than 0.01 do not affect the overall solution accuracy significantly.

3. `iter`, found in `bem2_charge_engine.m` in the main folder: Number of GMRES iterations used. The default value is 14. The maximum number is unlimited, but is subject to speed constraints. Numbers above 20 do not affect the overall solution accuracy.

4. `tneighbor`, found in `Model\model01_main_script.m`: Number of neighbor facets for averaging the computed surface charge density after the solution had been obtained. The default number is 3.

5. `R`, found in `bem6_volume_XY/XZ/YZ.m` and in `bem7_volume_XYZ.m` (in the main folder): Dimensionless (vs. average triangular face size) radius R of an integration sphere within which precise integration of the surface charge density is performed when computing volumetric fields. The default value is 2. The maximum number is unlimited, but is subject to speed constraints.

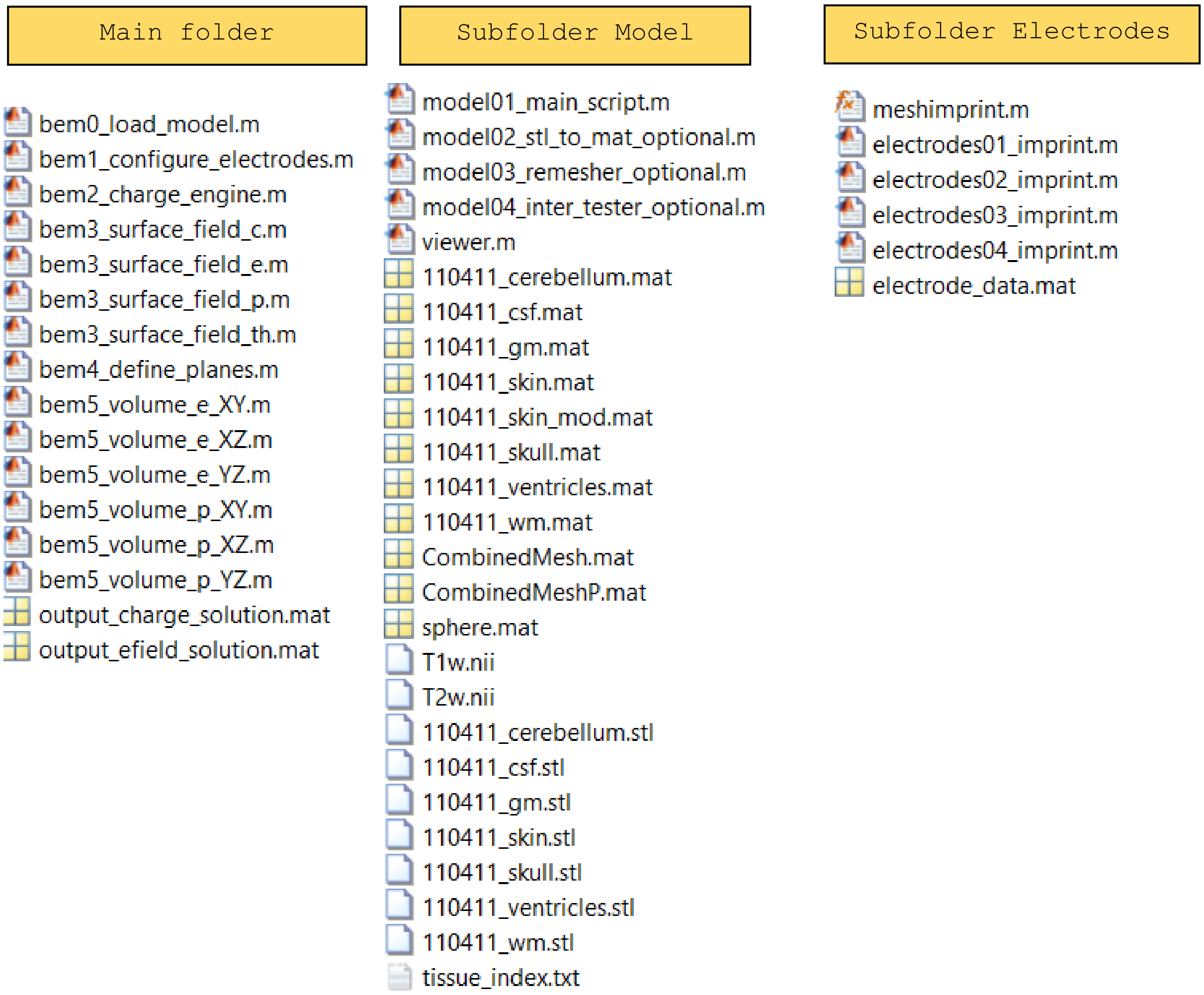
AB.9. Test of Numerical Accuracy

In application to TMS problems, the BEM-FMM algorithm was previously tested and validated against analytical and FEM numerical solutions by [49] and [50]; a very detailed and rigorous independent comparison study was further performed by [51].

Appendix C: TES-Focused BEM-FMM Toolkit Manual

AC.1. Computational Workflow Overview

The overall structure and workflow of the TES toolkit are very similar to those of the TMS toolkit described in Appendix B. Figure A12 below shows the toolkit organization. Note that the Coil subfolder has been replaced by an Electrodes subfolder, which contains computational scripts and functions for several electrode assemblies. In general, the same parameters and scripts are used for both this toolkit and the TMS toolkit; as such, this section will focus mainly on features unique to the TES toolkit.



Main folder

bem0_load_model.m
bem1_configure_electrodes.m
bem2_charge_engine.m
bem3_surface_field_c.m
bem3_surface_field_e.m
bem3_surface_field_p.m
bem3_surface_field_th.m
bem4_define_planes.m
bem5_volume_e_XY.m
bem5_volume_e_XZ.m
bem5_volume_e_YZ.m
bem5_volume_p_XY.m
bem5_volume_p_XZ.m
bem5_volume_p_YZ.m
output_charge_solution.mat
output_efield_solution.mat

Subfolder Model

model01_main_script.m
model02_stl_to_mat_optional.m
model03_remesher_optional.m
model04_inter_tester_optional.m
viewer.m
110411_cerebellum.mat
110411_csf.mat
110411_gm.mat
110411_skin.mat
110411_skin_mod.mat
110411_skull.mat
110411_ventricles.mat
110411_wm.mat
CombinedMesh.mat
CombinedMeshP.mat
sphere.mat
T1w.nii
T2w.nii
110411_cerebellum.stl
110411_csf.stl
110411_gm.stl
110411_skin.stl
110411_skull.stl
110411_ventricles.stl
110411_wm.stl
tissue_index.txt

Subfolder Electrodes

meshimprint.m
electrodes01_imprint.m
electrodes02_imprint.m
electrodes03_imprint.m
electrodes04_imprint.m
electrode_data.mat

Fig. A12. Low-level organization chart of the toolkit.

The computations are performed in the main folder. The scripts of the main folder should initially be executed sequentially but may subsequently be executed in any order since all the data will have already been computed.

The first script, `bem0_load_model.m`, imports head model data into the MATLAB workspace and sets the MATLAB path. It also imports the previously computed charge solution, if available.

The second script `bem1_configure_electrodes.m` loads pre-existing electrode configuration and initializes electrode voltages. The electrode configuration itself is defined and assembled in subfolder `Electrodes` via a separate MATLAB script and prior to performing any computations. Electrodes are accurately imprinted, and the skin surface mesh is modified accordingly. After every new geometrical electrode assembly, you must rerun the `model01_main_script.m` script in the `Model` folder to update the complete head mesh. Otherwise, incorrect results will be generated.

The next script to be executed is bem2_charge_engine.m. This script

(i) computes the iterative solution of the BEM integral equation for the induced surface charge density using the FMM, precomputed near-field potential integrals, and MATLAB GMRES (generalized minimum-residual method, Saad 2003);

(ii) displays the time for every iteration step in the MATLAB command window and plots the entire convergence history when completed;

(iii) computes the resulting surface electric fields (the principal component) and surface electric potentials via triangle subdivision (optional).

As for the near-field integration accuracy, the double potential integrals for three neighbor triangular patches (default value is given in the script `model01_main_script.m`) are computed precisely using the solid-angle approach. For non-neighbor triangles, the center-point approximation is used for the double potential integrals and FMM. The number of neighbor integrals can be increased at the expense of larger memory usage.

For practical purposes, it might be convenient to introduce weighted surface charge averaging (i.e., to low pass filter the surface charge density). One option is to average over the target facet and its three immediate topological neighbor triangles. After the solution is obtained, we might substitute in the script `bem2_charge_engine.m`

```
c = (c.*Area + sum(c(tneighbor).*Area(tneighbor), 2))./(Area +
sum(Area(tneighbor), 2));
```

This rule can be modified if necessary.
When performing mathematical (FMM) operations, the scripts of this folder call original and derived FMM functions from the subfolder `Engine`.

After the computations have been completed, the scripts `bem3_surface_field_c/e/p/th.m` display the surface charge density distribution, the surface electric field just inside/outside any interface, the continuous electric potential/voltage at the interfaces, and thresholded electric field just inside/outside any interface.

The next script is `bem4_define_planes.m`. This script defines three principal observation planes and prepares mesh cross-sections that will be used in the field output plots.

If NIfTI data are available (e.g. T1w.nii), they will be included in subsequent visualizations which will superimpose mesh cross-sections and/or fields onto the corresponding NIfTI slices.

The scripts bem5_volume_e/p_XY/XZ/YZ.m compute and output the electric field (any of its Cartesian components or a magnitude), and the electric potential in the three principal planes. The plane position and its size are specified in the script bem4_define_planes.m.
The volumetric field computations require more time since the potential integrals are no longer precomputed and must be calculated at the time of execution, depending on the position of a given observation point relative to the nearest interface(s). The critical numerical parameter here is the dimensionless (vs. average triangle size) radius, R, of an integration sphere within which integration of the surface charge density is performed. Its default value ranges between 2 and 5; higher numbers (e.g., $R = 10$) may provide better field accuracy but simultaneously slow down the computations.

A comparison with Ansys Maxwell Electronics Desktop FEM software solutions obtained using adaptive mesh refinement and very large tetrahedral meshes is performed in two separate folders: Version 0.3_example1_single_brick and Version 0.3_example2_composite_brick.

AC.2. Application Example 1: Two electrodes on the surface of a cube

In this example (folder Version 0.3_example1_single_brick), two voltage electrodes have been imprinted on one side of a conducting cube with a side length of 1 m. The major goal of this (somewhat simplified) example is to become familiar with code functionality and execution flow, and compare the results with another numerical solver – Ansys Electronics Desktop Maxwell. The main computational script bem2_charge_engine.m should execute in approximately 5 seconds (using a 2.4 GHz multicore server).

AC.2.a. Assigning cube conductivity

There is only one “tissue” mesh in this folder (subfolder Model): a brick named brick01.mat. It has a size of 1000 mm and approximately 35,000 facets. The average triangle quality (twice the ratio of the inradius to the circumradius) is 0.81, and the average edge length (mesh resolution) is 20.6 mm. Run viewer.m and inspect the cube mesh and its properties.
The assigned cube conductivity value is 0.1 S/m. This value is given in the editable tissue index file (named tissue_index.txt) in the same subfolder Model.

AC.2.b. Defining and imprinting electrodes

Go to subfolder Electrodes and run the script electrodes01_imprint.m. This script imprints two surface electrodes with the radius of 100 mm each separated by 500 mm on one (top) side of the brick.

AC.2.c. Assembling model and computing potential integrals

Now, go to subfolder `Model` and run the script `model01_main_script.m`. Reduce the number of cores for parallel computations (the `numThreads` variable) if necessary. This script has to be executed only *once*.

AC.2.d. Running simulations

After that, go to the main folder of the example. Run the first script, `bem0_load_model.m`. Next, open and run the second script `bem1_configure_electrodes.m`, which defines electrode voltages and a set of other parameters:

```
% Voltage (V) applied to each electrode
electrodeVoltages = [+1, -1];                % For electrode configuration 1
```

The next script to be executed is `bem2_charge_engine.m`. This script

(i) computes the iterative solution of the BEM integral equation for the induced surface charge density using the FMM, precomputed near-field potential integrals, and MATLAB GMRES (generalized minimum-residual method);

(ii) displays the time for every iteration step in the MATLAB command window and plots the entire convergence history when completed;

(iii) computes resulting surface electric potentials.

AC.2.e. Visualizing surface fields

The next scripts to execute are `bem3_surface_field_c/p.m`. These scripts display the surface charge distribution and the continuous surface electric potential.

AC.2.f. Comparing two numerical solutions

The next script to execute is `bem6_volume_comparison_XYZ.m`. This script computes the vector electric field within the cube for 0.125 M points uniformly distributed in space. The minimum offset from the boundary is 50 mm. Next, it compares the field with another numerical solution obtained using Ansys Electronics Desktop 2019 Maxwell FEM software with 5 adaptive meh refinement passes and with the final mesh of 2 M tetrahedra. The Ansys solution executes in 44 min on the same server.

For the present example, the script generates two metrics of error for the vector electric field: the relative 2-norm (maximum singular value)

$$Error2norm(\boldsymbol{E}_{BEM}, \boldsymbol{E}_{FEM}) = \|\boldsymbol{E}_{BEM} - \boldsymbol{E}_{FEM}\| / \|\boldsymbol{E}_{FEM}\|$$

and the vector field norm given by (symbol ‖·‖ now denotes the Euclidean vector norm)

$$ErrorV(\boldsymbol{E}_{BEM}, \boldsymbol{E}_{FEM}) = \frac{\sqrt{\sum_{k=1}^{K} \left\|\boldsymbol{E}_{k,BEM} - \boldsymbol{E}_{k,FEM}\right\|^2}}{\sqrt{\sum_{k=1}^{K} \left\|\boldsymbol{E}_{k,BEM}\right\| \left\|\boldsymbol{E}_{k,FEM}\right\|}}$$

Both electric-field errors appear to be **0.57%**.

AC.2.g. Visualizing volumetric fields

Run the script `bem4_define_planes.m` next to define the observation planes. The following scripts `bem5_volume_e/p_XY/XZ/YZ.m` will finally plot the volumetric electric fields and potential distributions in the observation planes.

AC.3. Application Example 2: Two electrodes on the surface of a cube

This example (folder `Version 0.3_example1_composite_brick`) is identical to the previous one, but a composite brick with approximately 71,000 facets is considered shown in Fig. A13.

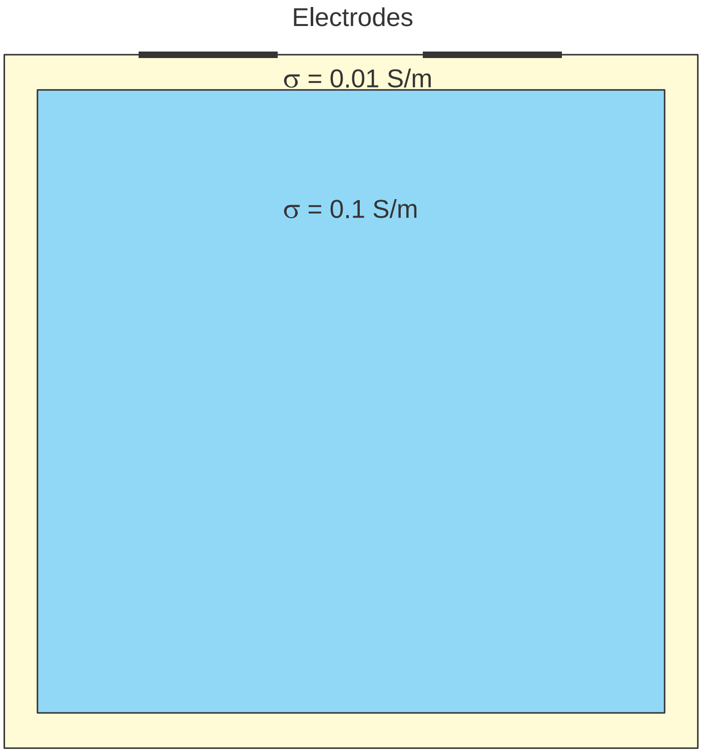


Fig. A13. Problem geometry and electrode position.

The code sequence of the previous example is executed in approximately 11 seconds (using a 2.4 GHz multicore server). The script `bem6_volume_comparison_XYZ.m` computes the vector electric field within the cube for 0.125 M points uniformly distributed in space. The minimum offset from the inner boundary in Fig. 5 is again 50 mm. Next, it compares the field with another numerical solution obtained using Ansys Electronics Desktop 2019 Maxwell FEM software with 9 adaptive meh refinement passes and with the final mesh of 2 M tetrahedra. Ansys solution executes in 1 h 42 min on the same server. The script generates both electric-field errors of **0.40%**.

AC.4. Application Example 3: Fronto-Medial Electrode Configuration for Connectome Subject 110411

In this example, we consider a realistic head model – Connectome subject 110411 – with seven head compartments [72]. The default average cortical surface mesh edge length is approximately 1.4 mm, the cortical nodal density is 0.55 nodes per mm^2, and the total number of facets is 0.9 M. The main computational script `bem2_charge_engine.m` should execute in less than 1-2 min (using a 2.4 GHz multicore server). However, the execution time is typically larger than for EEG/MEG or TMS modeling with the BEM-FMM.

AC.4.a. Defining electrodes

The corresponding example folder is `Version 0.3_example3_subject110411`. The electrode configuration itself is defined and assembled in subfolder `Electrodes` via a separate MATLAB script (`electrodes01_imprint.m`) and prior to performing any computations. Electrodes are accurately imprinted, and the skin surface mesh is modified accordingly. After every new geometrical electrode assembly, you must rerun the `model01_main_script.m` script in the `Model` folder to update the complete head mesh. Otherwise, incorrect results will be generated.

AC.4.b. Assigning conductivities of spherical compartments, assembling combined geometry, and computing potential integrals.

There are seven tissue meshes (skin, skull, CSF, GM, WM, cerebellum, ventricles) in this folder (subfolder `Model`); six of them are shown in Fig. A5. The meshes are inspected with the function `viewer.m` from the same subfolder.

The assigned conductivity values are the standard values used in the SimNIBS package [46]. These values are again given in the editable tissue index file (`tissue_index.txt`) in the same subfolder `Model`.

Now, run the script `model01_main_script.m`, which combines all partial meshes (including the skin mesh with embedded electrodes) together and computes necessary potential integrals. Reduce the number of cores for parallel computations (the `numThreads` variable) if necessary. This script has to be executed only *once*.

Next, go to the main folder of the example. Run the first script, `bem0_load_model.m`. After that, open and run the second script `bem1_configure_electrodes.m` to define electrode voltages and other parameters.

AC.4.c. Running simulations

The next script to be executed is `bem2_charge_engine.m`. This script

(i) computes the iterative solution of the BEM integral equation for the induced surface charge density using the FMM, precomputed near-field potential integrals, and MATLAB GMRES (generalized minimum-residual method, Saad 2003);

(ii) displays the time for every iteration step in the MATLAB command window and plots the entire convergence history when completed;

(iii) computes resulting surface electric fields (the principal component) and surface electric potentials via triangle subdivision (optional).

AC.4.d. Visualizing surface fields

After the computations have been completed, the scripts `bem3_surface_field_c/e/p/th.m` display the surface charge density distribution, the surface electric field just inside/outside any interface, the continuous electric potential/voltage at the interfaces, and thresholded electric field just inside/outside any interface or compartment. The compartment identifier is variable `tissue_to_plot`.

AC.4.e. Plotting volumetric fields

Run the script `bem4_define_planes.m` next to define the observation planes. The following scripts `bem5_volume_e/p_XY/XZ/YZ.m` will finally plot the volumetric electric fields and potential distributions in the observation planes. The plot resolution is controlled by variable `Ms`; the field threshold (for graphics) is controlled by variable(s) `th`; the number of plot levels is controlled by variable `levels`.