Modeling Pyramidal Neurons Using Bidomain BEM and Hierarchical Matrix Approximation

Nahian I. Hasan, Vanine Sabino, and Luis J. Gomez
Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA

***Abstract*—Electromagnetic brain stimulation uses electrodes or coils to induce electric fields (E-fields) in the brain and affect its activity. Our understanding of the precise effects of the device-induced E-fields on neural activity is limited. In this work, we present a novel boundary integral equation that enables the modeling of fully coupled E-fields from both neurons and stimulation devices. This boundary element approach is accelerated using fast direct solvers to allow for the analysis of realistic scenarios. We present examples, indicating the ability of our solver to analyze rat L2/3 pyramidal neurons derived from the blue brain project.**

I. Introduction

In electromagnetic brain stimulation, coils or electrodes driven by low-frequency current pulses induce electric (E) fields in the brain. Correspondingly, these E-fields affect neural activity. It is crucial to characterize the effects that E-fields have on individual (and a network of) neurons to determine the stimulating parameters (i.e., pulse waveform, intensity, and device placement). Recently, a boundary element method (BEM) was developed for determining the neuron's response to an impinging E-field [1]. Unlike commonly used approaches, this solver accounts for neighboring neurons, includes neuron geometry, and incorporates local E-field distortions due to neural activity, thereby increasing the predictive power of existing methods. However, the BEM method requires the solution of a dense system of equations at each time step, limiting its use to simple scenarios. We extend the BEM method by accelerating it using a direct solver, developing a methodology to convert filamentary wire models of cells into surface meshes, and including an extended set of gate models to enable the modeling of E-field effects on morphologically realistic neuron cells. Our approach is the first scalable tool that can model E-field effects on neuron cells without including severe simplifications.

Standard approaches to analyze neural activity induced by E-fields use computational E-field dosimetry results as a forcing function for a cable equation solver [2]. These approaches typically assume that the extracellular E fields remain unaffected by induced neuronal ion currents. However, this assumption is known to be incorrect, as local extracellular E fields around neurons are significantly modified by transmembrane ion currents [3]. In response to these limitations, a fully coupled neuron, brain, and stimulating device bidomain finite element method (FEM) was proposed [4]. While these FEM-based methods captured the full coupling between the cells and surrounding tissue, they require volumetric meshes that bridge individual neuron (micrometer) to whole brain (centimeter) scales. The generation of multi-scale meshes can make these bidomain FEM methods impractical or intractable for more complex simulations. A bidomain boundary integral equation (BEM) approach was proposed in [1] to avoid the multi-scale limitations of FEM while retaining full coupling. The bidomain BEM does not require volumetric meshes but needs the solution of dense systems of equations.

In the following sections, we first describe the bidomain BEM. Second, we detail how we bypass the need to solve dense systems of equations by using a fast direct solver. Third, we describe the cell membrane models and methods to convert a filamentary wire cell model into a 3D mesh. Finally, we provide results demonstrating the scalability of our technique along with results from layer (L) 2/3 pyramidal neurons.

II. Methods

A. Adjoint BEM Method

The details of the 'adjoint BEM' can be found in [1]. Here, we provide a brief overview. Brain stimulation devices operate at frequencies below 100 kHz, allowing for the validity of quasi-stationary assumptions. The total E-field generated in the head $\mathbf{E}(\mathbf{r};t)$, has three components: an E-field generated by the device in free-space $\mathbf{E_{inc}}(\mathbf{r};t)$; an E-field $\mathbf{E}_\rho(\mathbf{r};t)$, due to charge $\rho(\mathbf{r};t)$ accumulated on tissue and cell boundaries; and an E-field $\mathbf{E_m}(\mathbf{r};t)$, due to double layer sources that model the voltage jump $\mathrm{V_m}(\mathbf{r};t)$ across the thin cell membrane. The adjoint double layer finds the charges by imposing continuity of the normal component of the conduction current at tissue boundaries, given by the equation $\hat{\mathbf{n}}(\mathbf{r};t)\cdot\sigma^+(\mathbf{r})\mathbf{E}^+(\mathbf{r};t) = \hat{\mathbf{n}}(\mathbf{r};t)\cdot\sigma^-(\mathbf{r})\mathbf{E}^-(\mathbf{r};t)$. Here $\sigma^{+/-}(\mathbf{r})$ and $\mathbf{E}^{+/-}(\mathbf{r};t)$ are the conductivities and E-fields just outside/inside of the boundary, respectively, and $\hat{\mathbf{n}}$ is the outward pointing normal. We can write the boundary condition as

$$\big[\sigma^+(\mathbf{r})-\sigma^-(\mathbf{r})\big]\hat{\mathbf{n}}(\mathbf{r})\mathbf{E_{inc}}(\mathbf{r};t) = \sigma^-(\mathbf{r})\big[\hat{\mathbf{n}}\cdot\mathbf{E}_\rho^-(\mathbf{r};t)+ \hat{\mathbf{n}}(\mathbf{r})\mathbf{E_m^-}(\mathbf{r};t)\big] - \sigma^+(\mathbf{r})\big[\hat{\mathbf{n}}\cdot\mathbf{E}_\rho^+(\mathbf{r};t)+\hat{\mathbf{n}}(\mathbf{r})\mathbf{E_m^+}(\mathbf{r};t)\big], \tag{1a}$$

$$\mathbf{E}_\rho(\mathbf{r};t) = -\nabla\int\frac{\rho(\mathbf{r};t)}{||\mathbf{r}-\mathbf{r}'||}\mathrm{d}\mathbf{r}', \tag{1b}$$

$$\mathbf{E_m}(\mathbf{r};t) = -\nabla\int\mathrm{V_m}(\mathbf{r};t)\hat{\mathbf{n}}\cdot\nabla'\frac{1}{||\mathbf{r}-\mathbf{r}'||}\mathrm{d}\mathbf{r}'. \tag{1c}$$

To solve both the membrane voltage and charge (unknown on the membrane) the following equations can be used -

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{V_m}(\mathbf{r};t) = \frac{1}{\mathrm{c_m}}\big(\mathrm{I_m}(\mathbf{r};t)-\mathrm{I_{ion}}(\mathbf{r};t)\big), \tag{2a}$$

$$\mathrm{I_m}(\mathbf{r};t) = \left(\frac{1}{\sigma^+}-\frac{1}{\sigma^-}\right)^{-1}\rho(\mathbf{r};t) \tag{2b}$$

$$\mathrm{I_{ion}}(\mathbf{r};t) = \sum_{i=1}^{N} \mathrm{r_i}^{-1}(\mathbf{r};t;\mathrm{V_m})\big(\mathrm{V_m}(\mathbf{r};t) - \mathrm{V_i}\big). \tag{2c}$$

Here, $\mathrm{I_m}(\mathbf{r};t)$ is the current across the membrane, $\mathrm{I_{ion}}(\mathbf{r};t)$ is the total flow of current due to the exchange of charges between the intra-cellular and extra-cellular region through the membrane ionic gates, $\mathrm{c_m}$ is the membrane capacitance, $\mathrm{r_i}^{-1}(\mathbf{r};t;\mathrm{V_m})$ is the resistivity of each ion type, $\mathrm{V_i}$ is the equilibrium transmembrane potential. To solve (1), the following semi-implicit approximation of state equation (2a) is used

$$\mathrm{V_m}(\mathbf{r};t+\Delta t) \approx \frac{\Delta t}{\mathrm{c_m}}\left(\frac{\mathrm{I_m}(\mathbf{r};t) + \mathrm{I_m}(\mathbf{r};t+\Delta t)}{2} - \mathrm{I_{ion}}(\mathbf{r};\mathrm{t})\right). \tag{3}$$

The bidomain approach assumes that we know all quantities at time t and determines quantities at time $t+\Delta t$. First, the charges are determined by using the right-hand side of (3) and the relation $\mathrm{I_m}(\mathbf{r};t+\Delta t) = \left(\frac{1}{\sigma^+} - \frac{1}{\sigma^-}\right)^{-1}\rho(\mathbf{r};t+\Delta t)$ to solve (1) for $\rho(\mathbf{r};t+\Delta t)$ using a Galerkin approach with nodal elements. Once $\rho(\mathbf{r};t+\Delta t)$ is known, this is used to determine $\mathrm{I_m}(\mathbf{r};t)$ and $\mathrm{V_m}(\mathbf{r};t)$. The resistivities of each membrane channel $\mathrm{r_i}^{-1}(\mathbf{r};t;\mathrm{V_m})$ are then updated using the values of $\mathrm{V_m}(\mathbf{r};t)$. It is assumed that the time-step is small enough that the conductivity can be assumed to be well-approximated by the previous time-step one at the half-time step. Therefore, an explicit scheme is used to update $\mathrm{r_i}^{-1}$.

The solution of this equation at each time step requires a CPU run-time that scales with the number of mesh nodes squared (a typical simulation involves 20,000-time steps). Consequently, the memory and CPU requirements limit the applicability of this method. As an extension, we initially employed the fast multipole method (FMM) to compress the BEM matrix, significantly reducing the solver's memory requirements. However, solving the BEM system iteratively at each time step required considerably more time. For instance, conducting 2,000-time steps with 1,000 unknowns took the FMM-accelerated version 3.7 hours, compared to just 10 seconds using the dense BEM system on a standard laptop. Given that a typical morphologically realistic mesh consists of 16,830 nodes (Fig. 1), the FMM acceleration proved too slow. To address this, we used the hierarchical matrix (H-matrix) approximation method to compress the dense BEM matrix inverse.

B. Compressing BEM Matrix

To compress the resultant full dense matrix from the bidomain BEM formulation of the neuron membrane dynamics, we use the hierarchical off-diagonal low rank (HODLR) approximation method. In this new representation, the membrane nodes are clustered into groups and arranged into a binary tree-like structure consisting of multiple levels. At any level of partitioning, the inadmissible blocks (usually diagonal blocks and are not compressible) are stored as dense matrices, and the admissible blocks (off-diagonal blocks) are approximated using a low-rank approximation method. For efficient compression, we clustered the nodes on the cell membrane and the surrounding bath separately using an octree data structure based on physical proximity. At any level (l), the admissible blocks are approximated using the adaptive cross-approximation (ACA) whereas the inadmissible blocks are computed analytically. The dense BEM matrix has a storage complexity of $O(n^2)$ whereas the HODLR matrix has a complexity of approximately $O(knlogn)$ with k being the highest rank of the off-diagonal blocks, and n is the number of nodes. The inverse of the HODLR matrix is stored in the pre-computation stage to be utilized later in updating the membrane charge density ($\rho(\mathbf{r};t)$) and membrane voltage ($\mathrm{V_m}(\mathbf{r};t)$) at any time ($t$) using simple matrix-vector product. An HM-toolbox [5] is used to generate the HODLR matrix.

C. Simulation Setup

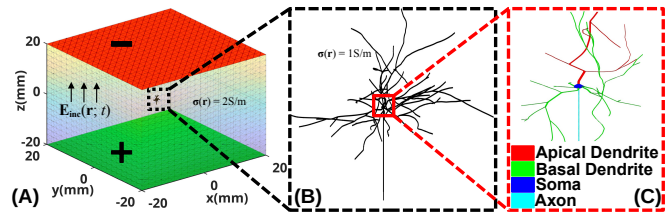

Fig. 1. The L2/3 pyramidal neuron within an inhomogeneous media is stimulated by an E-field $\mathbf{E_{inc}}(\mathbf{r};t)$. The intra-cellular and the extra-cellular spaces have conductivity of 1 S/m and 2 S/m, respectively. The cell has four compartment types (soma, axon, apical dendrite, and basal dendrite).

A filamentary cable model of the L2/3 pyramidal neuron cell was obtained from the Blue Brain Project [6]. The model consists of four types of compartments: soma, axon, apical dendrite, and basal dendrite (Fig. 1C). Each compartment has a set of ion channels and equations governing the membrane mechanisms as shown in Fig. 2. We coded the cell membrane circuits and ion gates into MATLAB to model the cells.

Each compartment model consists of compartment nodes and edges. To mesh a neuron, a temporary triangle mesh of a circle is added near the ends of each edge at first (Fig. 3B). Second, individual meshes of each edge are made by meshing the convex hull of each of its circles and individual meshes for each node are made by meshing the convex hull containing all of the circles corresponding to the closest circle of each coincident edge (Fig. 3C). Finally, all edge and node meshes are combined into a solid mesh by removing all circles and merging the individual meshes. This results in a watertight mesh (Fig. 3D). For stability, we modified the algorithm [7] by first mapping all points of circles to a unit sphere centered about the node before constructing its convex hull mesh.

Figure 3D shows an example mesh generated from a 1,387-compartment model of an L2/3 pyramidal cell used in this study. Each triangle is assigned a membrane type. The neuron consists of 16,830 nodes with 33,660 triangular facets. The dimension of the neuron is of the order of $10^{-4}\mathrm{m}^3$ and has a membrane capacitance of 1 $\mu\mathrm{F/cm}^2$. The neuron is submerged inside a cubical bath with a dimension of $40 \times 40 \times 40$ mm^3. Two sides of the bath are modeled as cathode (-) and anode (+) terminal to apply an external electric stimulus to the neuron (Fig. 1A). The bath consists of 2,168 nodes

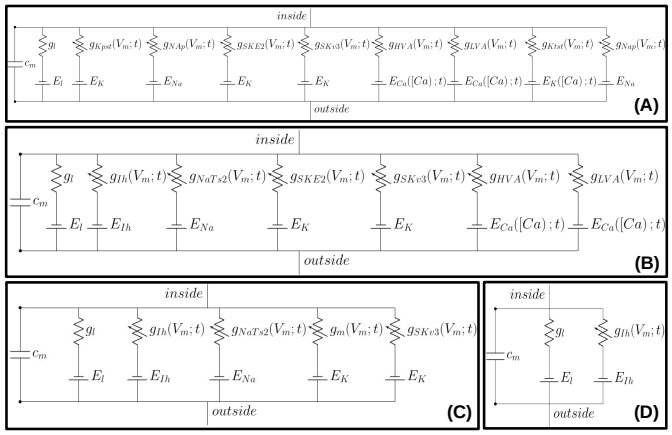

Fig. 2. Equivalent circuit of the axon (A), soma (B), apical dendrite (C), and basal dendrite (D). Here, the g terms denote the conductance of the specific ion channel and the E terms model the electrochemical gradient driving the ion movement between extra-cellular (outside) and intra-cellular (inside) regions.

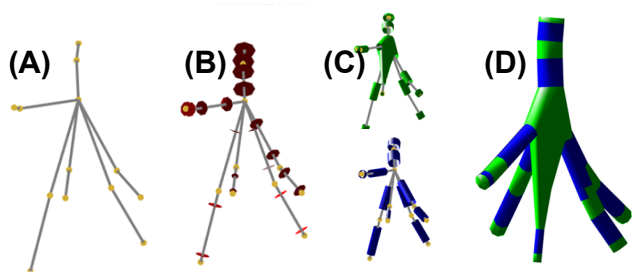

Fig. 3. Meshing neuron. (A) node locations and edges of the neuron. (B) Adding temporary circle triangle mesh (red). (C) Constructing a mesh by connecting all circle meshes of each compartment and then removing the circle meshes (D) Merging meshes to construct a surface mesh of neuron.

with 4,332 triangular facets. The HODLR matrix consists of 8 hierarchical levels where the off-diagonal blocks are computed using ACA with a convergence tolerance of 10^{-7}. The neuron and the bath have conductivities of 1 S/m and 2 S/m, respectively. The electrodes are driven with an external current of amplitude 14.4 mA, resulting in a nearly uniform incident field of $\mathbf{E_{inc}}(\mathbf{r};t) \approx 72\mathrm{V/m}$ for the first 5 ms. The transmembrane voltage is computed with a temporal resolution of 1 μs and a total of 40,000 time steps.

III. Results

A. HODLR Matrix

A compressed form of the HODLR matrix is shown in Fig. 4 using $l = 8$ for both membrane nodes and bath nodes. The charges on the neuron membrane weakly interact with the charges on the bath surface. Therefore, the neuron-bath or bath-neuron coupling is represented using only ranks of 3 and 9, respectively. Additionally, the interactions of charges on the membrane itself are highly compressible. For example, Fig. 4A highlights the interaction between two sets of membrane nodes (green and red block with the corresponding nodes shown in Fig. 4B). The ranks of these compressible matrix blocks $(3,845 \times 12,986)$ are 219 and 218. Figure 4A shows the inadmissible dense blocks in blue. The interactions among the charges on the bath boundary are less compressible than the membrane nodes (Fig. 4C).

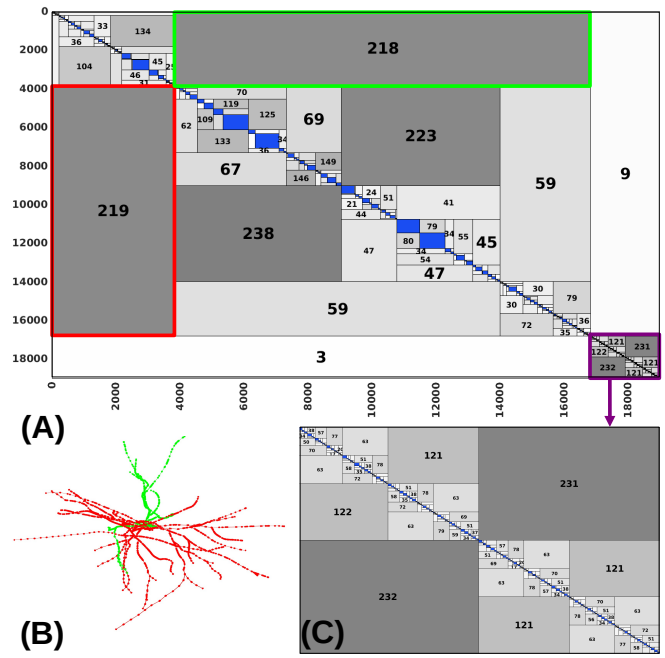

Fig. 4. (A) HODLR representation of the BEM matrix (dense blocks in blue). (B) Two sets of membrane nodes with weak interaction as shown in green and red blocks $(3,845 \times 12,986)$ in (A) require only 219 and 218 ranks. (C) HODLR subblock for the interaction of charges on the bath boundary.

B. Action Potential Propagation

Figure 5 shows the action potential at three locations along the axon, soma, and basal dendrite (shown in the inset). The

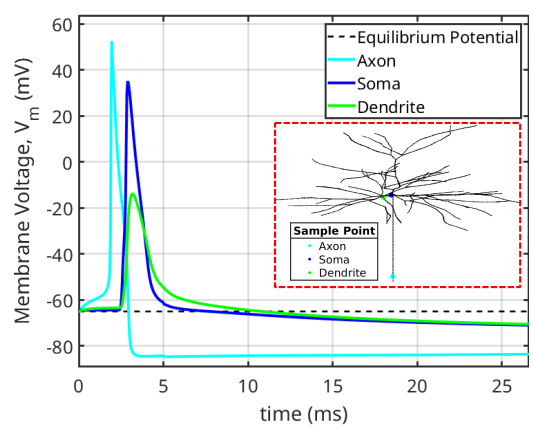

Fig. 5. Propagation of action potential ($\mathrm{V_m}$) through the axon, soma, and dendrites. The inset shows the corresponding spatial sample locations.

action potential starts at one end of the axon and propagates through the axon, soma, and dendrites. The membrane becomes hyper-polarized (below the equilibrium potential of -65 mV) before transitioning to a complete resting potential. Figure 6 shows a graphical representation of action potential

propagation through the membrane at different time steps. The action potentials in the soma and axon, shown in Fig. 5 and Fig. 6, agree with what would be expected based on NEURON environment simulations not presented here. This indicates that our BEM solver can simulate realistic neuron morphology.

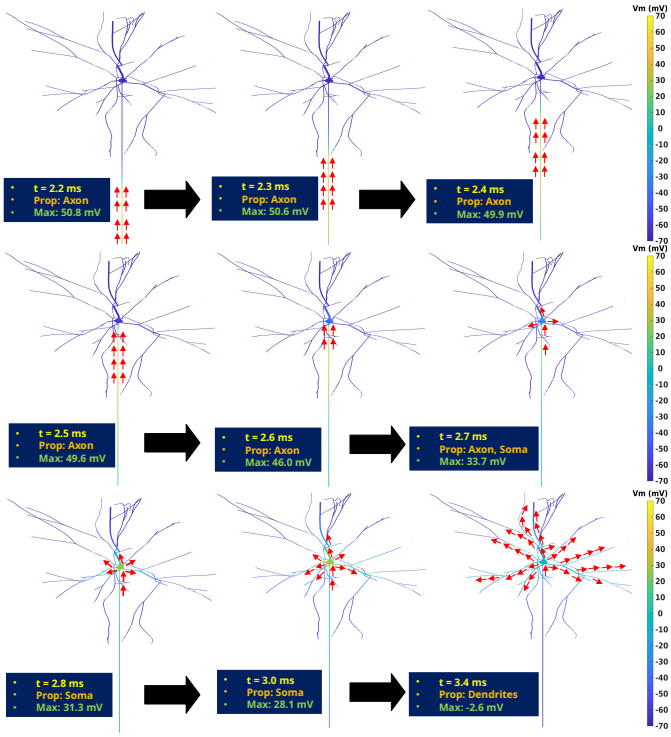

Fig. 6. The action potential starts propagating from the axon and traverses through the axon towards the soma. Next, it propagates along the dendrites to nearby neurons. The insets show the time stepping (t), propagating compartment ($prop$), and the maximum amplitude of the action potential. The red arrow follows the maximum transmembrane voltage.

C. Compression and Speed

The computation of the full dense BEM matrix requires 5.38 Gigabytes (GB) of computational memory, whereas the compressed HODLR matrix requires only 0.24 GB (a compression ratio of 22.4 times). The set-up time before compression and the HODLR compression are 77.92 min and 47.93 min, respectively. Finally, it takes 191.08 min (0.29s/step) and 19.6 min (0.03s/step) for the uncompressed and compressed BEM matrix, respectively, to compute the transmembrane voltage across 40,000-time steps (10 times improvement).

D. Effect of External Bath and Hierarchical Tree Structure

In this section, we analyze the effect of the dimension of the bath ($n \times n \times n$) and the number of HODLR hierarchical levels (l) on the computational memory, maximum HODLR rank, delay in the initialization of action potential, and the preprocessing time (Fig. 7). The required memory, the maximum rank in the HODLR matrix, and the preprocessing time increase as a function of the bath dimension shown in Fig. 7A, 7B and 7D, respectively. The bath dimension negligibly affects

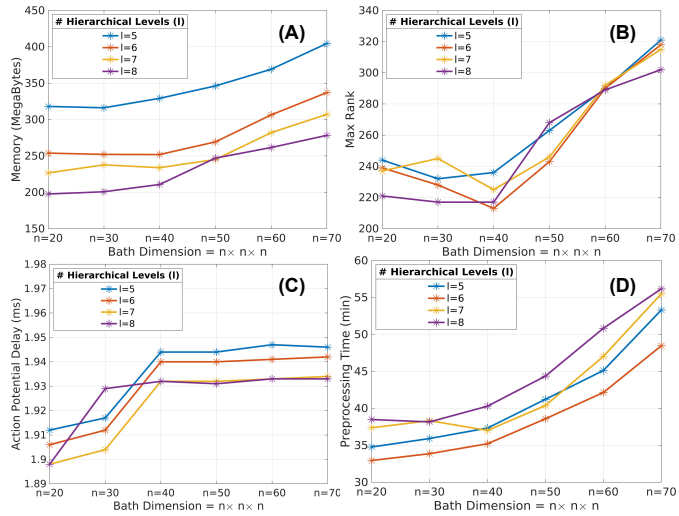

Fig. 7. Effects of the bath size ($n \times n \times n$) and the number of HODLR hierarchical levels (l) on the computational memory of HODLR compression (A), the HODLR maximum rank (B), initialization of the action potential (C), and total compression time (D).

the initialization of the action potential Fig. 7C. On the other hand, the computational memory and the maximum HODLR rank decrease as a function of HODLR hierarchical levels shown in Fig. 7A and 7B, respectively. The HODLR hierarchical levels negligibly affect the initialization of action potential (Fig. 7C) and the preprocessing time (Fig. 7D). Neither the bath dimension nor the tree levels affect the initialization and the total computational time of transmembrane voltage.

IV. Conclusion

By leveraging direct solvers, the Bidomain BEM can analyze the response of an L2/3 neuron. This tool will enable the study of geometric factors to stimulation. Furthermore, analyzing a larger neuron or multiple neurons under the same bath is possible due to the compression of dense BEM matrix using HODLR matrix representation.

References

[1] D. M. Czerwonky, A. S. Aberra, and L. J. Gomez, "A boundary element method of bidomain modeling for predicting cellular responses to electromagnetic fields," *Journal of Neural Engineering*, 2023.

[2] B. Wang, W. M. Grill, and A. V. Peterchev, "Coupling magnetically induced electric fields to neurons: longitudinal and transverse activation," *Biophysical journal*, vol. 115, no. 1, pp. 95–107, 2018.

[3] R. Plonsey, R. C. Barr, and A. Bioelectricity, "Quantitative approach," *Cham, Switzerland: Springer*, 2007.

[4] A. Agudelo-Toro and A. Neef, "Computationally efficient simulation of electrical activity at cell membranes interacting with self-generated and externally imposed electric fields," *Journal of neural engineering*, vol. 10, no. 2, p. 026019, 2013.

[5] S. Massei, L. Robol, and D. Kressner, "hm-toolbox: Matlab software for hodlr and hss matrices," *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. C43–C68, 2020.

[6] A. Larkman and A. Mason, "Correlations between morphology and electrophysiology of pyramidal neurons in slices of rat visual cortex. i. establishment of cell classes," *Journal of Neuroscience*, vol. 10, no. 5, pp. 1407–1414, 1990.

[7] G. W. Hart, "Sculptural forms from hyperbolic tessellations," in *2008 IEEE International Conference on Shape Modeling and Applications*. IEEE, 2008, pp. 155–161.