Modeling Pyramidal Neurons Using Bidomain BEM and Hierarchical Matrix Approximation

Nahian I. Hasan, Vanine Sabino, Amanda Jeanne Walenciak, Yang Liu, and Luis J. Gomez

***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 paper, we present a novel bidomain boundary integral equation-based method 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. A comprehensive analysis shows that this method can be easily extended to model a group of neurons that were previously computationally intractable.**

***Index Terms*— Action potential, Bidomain BEM, Hierarchical Matrix Approximation, HODLR, L2/3Pyramidal Neuron**

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 brain activity. To determine the effects of E-fields on the brain, an important first step is to characterize their effects on single neurons and small networks of neurons. Here, we introduce a scalable direct solver accelerated boundary element bidomain approach to determine the effects of the E-field on neurons and small networks of neurons.

Standard approaches to analyze neural activity induced by E-fields use computational E-field dosimetry results as a forcing function (e.g., activation functions [1] or quasi-potentials [2]) that drives a filamentary cable model of a neuron. The underlying assumption is that a cell and its surrounding cells do not affect the E-field observed in the extracellular space, as such, the device E-field is introduced as an extracellular source driving the circuit [1]–[6]. This source is then incorporated into cable models of neuron cells as distributed current or voltage injections. However, this assumption is known to be incorrect. Local extracellular E fields around neurons are significantly modified by transmembrane ion currents [7]. Additionally, it was recently found that surrounding cells distort the device-induced E-fields, thereby changing the amount of E-field required to activate a neuron(i.e., neuron activation thresholds) [8]–[10].

Recently, [9] proposed using boundary element fast multipole method (BEM-FMM) [11], [12] to solve for the E-fields in the presence of cells and use these corrected E-fields as an activating function. Their approach offers an improvement over standard methods, however, it neglects changes in cell membrane electrical properties resulting from ionic channels opening and closing. Another approach, uses finite element method (FEM) to solve the full coupling between the cell and its surroundings [1], [13]–[15]. While these FEM-based methods capture the full coupling between the cells and surrounding tissue, they require volumetric meshes that bridge individual neurons (micrometers) to whole-brain (centimeters) scales. The generation of multi-scale meshes can make these bidomain FEM methods impractical or intractable for more complex simulations. As a way to avoid volumetric meshes, we introduced a bidomain boundary element method (BEM) for determining the neuron's response to an impinging E-field [10]. This approach avoids the multi-scale limitations of FEM while retaining full coupling. Unfortunately, the execution of this solver requires the solution of a dense matrix system of equations for each time-step, and a typical simulation has thousands of time-steps. As such, the bidomain BEM requires an intractable level of computational resources for analysis of realistic scenarios.

In this paper, we extend the bidomain BEM method by accelerating it using a direct solver. The dense BEM matrix is highly compressible, and we utilize the hierarchical off-diagonal low-rank (HODLR) approximation technique [16] to represent the interactions between different sub-groups of charges on the membrane surface with a minimal number of basis functions. This allows a precomputation of the inverse of the HODLR compressed BEM matrix for a large system and becomes a simple matrix-vector product routine during the computation and update of the membrane voltage. Our approach is the first scalable tool that can model E-field effects on neuron cells without including severe simplifications. We provide results demonstrating the scalability of our technique along with results from layer (L) 2/3 pyramidal neurons and

Nahian I. Hasan, Elmore School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906, USA,(e-mail: nahianhasan1994@gmail.com).

Vanine Sabino, Elmore School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906, USA,(e-mail: vsabinod@purdue.edu).

Amanda Jeanne Walenciak, Elmore School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906, USA,(e-mail: awalenci@purdue.edu).

Yang Liu, Applied Mathematics & Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, California 94720, USA,(e-mail: liuyangzhuan@lbl.gov).

Luis J. Gomez, Elmore School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47906, USA,(e-mail: ljgomez@purdue.edu).

benchmark the results for a large group of neurons.

II. Methods

A. Adjoint BEM Method

We consider a cell inside an inhomogeneous conductive medium stimulated by electrodes or coils (Fig.[1]). The E-field generated by the electrodes and coils is assumed to have a low-frequency spectrum (≤100 kHz), allowing for the validity of quasi-stationary assumptions [17]. The electrodes are driven by a known current $\mathbf{J}(\mathbf{r};t)\cdot\hat{\mathbf{n}}(\mathbf{r}) = \mathbf{J}(\mathbf{r})P_e(t)\cdot\hat{\mathbf{n}}(\mathbf{r})$ that is only non-zero on the electrodes (i.e., cathodes and anodes). Here, $P_e(t)$ is the driving current pulse waveform, and $\hat{\mathbf{n}}(\mathbf{r})$ is the outward-pointing unit normal vector. The coil is driven by a known current $\mathbf{J}_{\mathbf{coil}}(\mathbf{r};t) = \mathbf{J}(\mathbf{r})P_c(t)$, where $P_c(t)$ is its driving pulse waveform. The coil generates an E-field $\mathbf{E}_{\mathbf{coil}}(\mathbf{r};t) = \mathbf{E}_{\mathbf{coil}}(\mathbf{r})P_c'(t)$ in empty spaces, where $P_c'(t)$ is the time derivative of the current pulse to the coil. This results in a charge distribution $\rho(\mathbf{r};t)$ accumulated on boundaries between media with distinct conductivities. This accumulated charge results in a secondary E-field $\mathbf{E}_\rho(\mathbf{r};t)$. In addition, the cell has a thin membrane that mediates the exchange of ions between its intracellular and extracellular spaces. This membrane has a voltage drop of $V_m^{(t)}(\mathbf{r})$ across it that results in an E field $\mathbf{E}_{\mathbf{m}}(\mathbf{r};t)$. The adjoint double layer is derived by imposing continuity of the normal component of the conduction current at boundaries between media with distinct conductivities and the cell membrane as

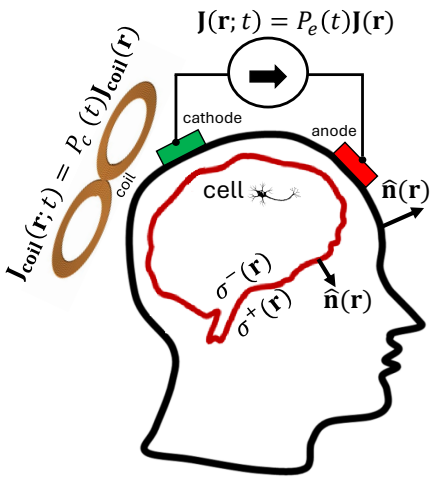


Fig. 1: A cell inside the inhomogeneous conductive medium (head) being stimulated by electrodes (cathode and anode) and a coil.

$$\hat{\mathbf{n}}(\mathbf{r})\cdot\sigma^+(\mathbf{r})\mathbf{E}^+(\mathbf{r};t) = \hat{\mathbf{n}}(\mathbf{r})\cdot\sigma^-(\mathbf{r})\mathbf{E}^-(\mathbf{r};t) \tag{1a}$$

for $\{\mathbf{r}\notin electrodes\}$ and

$$\mathbf{J}(\mathbf{r};t)\cdot\hat{\mathbf{n}}(\mathbf{r}) = \hat{\mathbf{n}}(\mathbf{r})\cdot\sigma^-(\mathbf{r})\mathbf{E}^-(\mathbf{r};t) \tag{1b}$$

for $\{\mathbf{r}\in electrodes\}$.

Here, $\sigma^{+/-}(\mathbf{r})$ and $\mathbf{E}^{+/-}(\mathbf{r};t)$ are the conductivities and E-fields just outside/inside of the boundary, respectively. Equations (1a) and (1b) are writen more compactly as

$$\begin{aligned}\mathbf{J}(\mathbf{r};t)\cdot\hat{\mathbf{n}}(\mathbf{r}) + \big(\sigma^+(\mathbf{r})-\sigma^-(\mathbf{r})\big)\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E}_{\mathbf{coil}}(\mathbf{r};t) = \\ -\sigma^+(\mathbf{r})\big(\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E}_\rho^+(\mathbf{r};t)+\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E}_{\mathbf{m}}^+(\mathbf{r};t)\big) \\ +\sigma^-(\mathbf{r})\big(\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E}_\rho^-(\mathbf{r};t)+\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E}_{\mathbf{m}}^-(\mathbf{r};t)\big)\end{aligned} \tag{2}$$

by setting $\sigma^+(\mathbf{r}) = 0$ wherever $\mathbf{r}\in electrodes$. The three components of the total E-field inside the head are

$$\mathbf{E}_{\mathbf{coil}}(\mathbf{r};t) = -\frac{\mu_0}{4\pi}P_c'(t)\int\frac{\mathbf{J}(\mathbf{r}')}{||\mathbf{r}-\mathbf{r}'||}d\mathbf{r}', \tag{3a}$$

$$\mathbf{E}_\rho(\mathbf{r};t) = -\frac{1}{4\pi}\nabla\int\frac{\rho(\mathbf{r}';t)}{||\mathbf{r}-\mathbf{r}'||}d\mathbf{r}', \tag{3b}$$

$$\mathbf{E}_{\mathbf{m}}(\mathbf{r};t) = -\hat{\mathbf{n}}(\mathbf{r})\cdot\nabla\int V_m(\mathbf{r}';t)\nabla'\frac{1}{||\mathbf{r}-\mathbf{r}'||}d\mathbf{r}', \tag{3c}$$

where μ_0 is the permeability in free space. A more detailed and generalized mathematical derivation of the bidomain BEM can be found in [10].

Both the membrane voltage and charge density are unknowns. However, the membrane voltage is related to the membrane current $I_m(\mathbf{r};t)$, which again is proportional to the charges as

$$\frac{d}{dt}V_m(\mathbf{r};t) = \frac{1}{c_m}\big(I_m(\mathbf{r};t) - I_{ion}(\mathbf{r};t;V_m)\big), \tag{4a}$$

$$I_m(\mathbf{r};t) = \big(\frac{1}{\sigma^+}-\frac{1}{\sigma^-}\big)^{-1}\rho(\mathbf{r};t). \tag{4b}$$

Here $I_{ion}(\mathbf{r};t;V_m)$ is the total flow of current due to the exchange of charged ions between the intracellular and extracellular space, c_m is the membrane capacitance. The ionic currents can be determined from the membrane voltage and equivalent circuit models of the cell membrane. To find the ionic currents from the membrane voltage, we utilize distinct circuit models for the soma, axon, and apical and basal dendrites given in Section II-F.

To determine the membrane voltage, current, and charges, we approximate the boundaries of the cell and surrounding media as a triangle mesh consisting of N_T triangles and N_V nodes. All quantities are approximated using piece-wise linear nodal elements on the triangle mesh as $\rho(\mathbf{r};t) = \sum_{\tau=1}^{N_t}\sum_{i=1}^{N_V}\rho_i^{(\tau)}B_{i,\tau}(\mathbf{r};t)$, $I_{ion}(\mathbf{r};t) = \sum_{\tau=1}^{N_t}\sum_{i=1}^{N_V}{I_{ion}}_i^{(\tau)}B_{i,\tau}(\mathbf{r};t)$, $V_m(\mathbf{r};t) = \sum_{\tau=1}^{N_t}\sum_{i=1}^{N_V}{V_m}_i^{(\tau)}B_{i,\tau}(\mathbf{r};t)$, $I_m(\mathbf{r};t) = \sum_{\tau=1}^{N_t}\sum_{i=1}^{N_V}{I_m}_i^{(\tau)}B_{i,\tau}(\mathbf{r};t)$, where $B_{i,\tau}(\mathbf{r};t) = L_\tau(t)M_i(\mathbf{r})$, $M_i(\mathbf{r})$ is the nodal element corresponding to the ith node of the mesh and is a hat basis function centered at $t=\tau\Delta t$ and with support $(\tau-1)\Delta t < t \le \tau\Delta t$ and N_t is the number of time-steps. It is assumed that we know the state of all quantities at the initial time $t=0$.

We use (4a) and (2) to determine all quantities at time $(\tau+1)\Delta t$ from all quantities at time $\tau\Delta t$. A semi-implicit approximation of (4a) is made and evaluated at each node of the mesh as

$${\mathbf{V}_{\mathbf{m}}}^{(\tau+1)} \approx {\mathbf{V}_{\mathbf{m}}}^{(\tau)} + \frac{\Delta t}{c_m}\big(\frac{{\mathbf{I}_{\mathbf{m}}}^{(\tau)}+{\mathbf{I}_{\mathbf{m}}}^{(\tau+1)}}{2} - {\mathbf{I}_{\mathbf{ion}}}^{(\tau)}\big), \tag{5}$$

where $\mathbf{V_m}^{(\tau)} = (V_{m_1}^{(\tau)}, V_{m_2}^{(\tau)}, \ldots, V_{m_{N_V}}^{(\tau)})$, $\mathbf{I_m}^{(\tau)} = (I_{m_1}^{(\tau)}, I_{m_2}^{(\tau)}, \ldots, I_{m_{N_V}}^{(\tau)})$, and $\mathbf{I_{ion}}^{(\tau)} = (I_{ion_1}^{(\tau)}, I_{ion_2}^{(\tau)}, \ldots, I_{ion_{N_V}}^{(\tau)})$. The right-hand side of (5) and (4b) are used to cast (1) as an integral equation with $\rho^{(\tau+1)} = (\rho_1^{(\tau+1)}, \rho_2^{(\tau+1)}, ..., \rho_{N_V}^{(\tau+1)})$ as unknown. A standard Galerkin approach is applied to the resulting equation, and this results in the following linear system of equations

$$\mathbf{A}\rho^{(\tau+1)} = \mathbf{b}^{(\tau+1)}, \tag{6}$$

where

$$\begin{aligned}\mathbf{A}_{ij} = &-\frac{\sigma^+ + \sigma^-}{2}\int M_i(\mathbf{r})M_j(\mathbf{r})d\mathbf{r} \\ &+ \frac{\sigma^+ - \sigma^-}{4\pi}\int M_i(\mathbf{r})\hat{\mathbf{n}}(\mathbf{r})\cdot\nabla\int\frac{M_j(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|}d\mathbf{r}'d\mathbf{r} \\ &+ \frac{\Delta t}{2c_m}\frac{\sigma_{ext}\sigma_{mem}}{\sigma_{mem}-\sigma_{ext}}\mathbf{N}_{\mathbf{m}ij},\end{aligned} \tag{7a}$$

$$\begin{aligned}\mathbf{b}_i^{(\tau+1)} = &P_e((\tau+1)\Delta t)\int M_i(\mathbf{r})\mathbf{J}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r})d\mathbf{r} \\ &-(\sigma^+ - \sigma^-)P_c'((\tau+1)\Delta t)\int M_i(\mathbf{r})\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{E_{coil}}(\mathbf{r})d\mathbf{r} \\ &-\mathbf{N}_{\mathbf{m}ij}\big(\mathbf{V_m}^{(\tau)} + \frac{\Delta t}{2c_m}(\mathbf{I_m}^{(\tau)} - 2\mathbf{I_{ion}}^{(\tau)})\big),\end{aligned} \tag{7b}$$

$$\begin{aligned}\mathbf{N}_{\mathbf{m}ij} = &\frac{\sigma^+ - \sigma^-}{4\pi}\int\hat{\mathbf{n}}(\mathbf{r})\times\nabla M_i(\mathbf{r})\cdot \\ &\int_{membrane}\frac{\hat{\mathbf{n}}(\mathbf{r}')\times\nabla' M_j(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|}d\mathbf{r}'d\mathbf{r}.\end{aligned} \tag{7c}$$

Here σ_{mem} is the cell membrane conductance, σ_{ext} is the conductivity of the region surrounding the cell. The entries of the matrix $\mathbf{A}$ that correspond to 3rd-order neighbors are computed using analytical [10], [18], [19] formulas for the inner integral and a 8th-order accurate quadrature rule for the outer one. In other words, if the column and row indexes correspond to nodal elements with corresponding nodes connected on the mesh by a path of three edges or fewer, they are computed to high accuracy. All other matrix entries are approximated using single-point triangle centroid quadratures. The system of equations is solved to determine $\rho^{(\tau+1)}$. Once $\rho^{(\tau+1)}$ is known, the membrane current coefficients $\mathbf{I_m}^{(\tau+1)}$ are determined using (4b), and $\mathbf{V_m}^{(\tau+1)}$ using (5). Finally, $\mathbf{I_{ion}}^{(\tau+1)}$ is computed from $\mathbf{V_m}^{(\tau+1)}$ using (12) whose evaluation is described in section II-F.

The storage of $\mathbf{A}$ requires order N_V^2 memory. Correspondingly, this limits the complexity of scenarios that can be studied with the standard Bidomain BEM. Here we compressed the matrix $\mathbf{A}$ and stored it using a hierarchical off-diagonal low-rank (HODLR) format requiring only order $N_V log(N_V)$ memory. Furthermore, we describe an approach to compute its LU factors in the HODLR format, enabling its fast-direct solution.

B. HODLR compression of A

To lower the memory complexity of storing the matrix $\mathbf{A}$, its LU factors are stored in a HODLR format. The HODLR matrix $\mathbf{H} \approx \mathbf{A}$ is stored to a prescribed accuracy $||\mathbf{A} - \mathbf{H}|| < \epsilon_H||\mathbf{A}||$, where ϵ_H is a user specified error level (here chosen as 10^{-7}) and $||\cdot||$ is the Frobenius matrix norm. A two-level HODLR matrix ($\mathbf{H}$) is shown in Fig. [2]. In Fig. [2], compression is achieved by only storing full-diagonal sub-matrices (shown in red) and compressing the others. Several methods can be used to compress the off-diagonal submatrices; here we used adaptive cross-approximation (ACA) [20]–[22], and its details are given in the next section. Next, the hierarchical structure of the HODLR matrix is described in more detail.

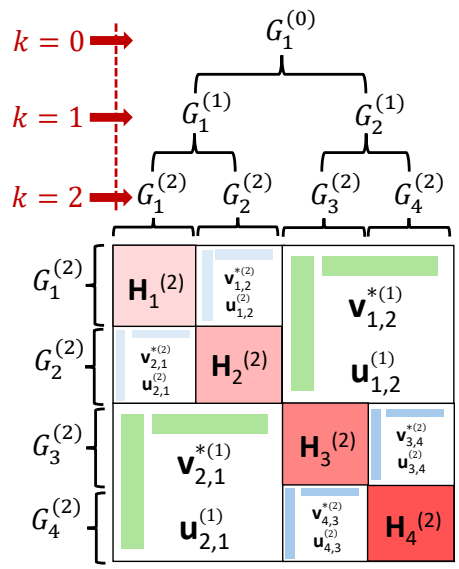


Fig. 2: HODLR matrix for 2 levels of hierarchical partitioning. The kth level has 2^k groups of nodes. The off-diagonal admissible blocks are computed using a low-rank approximation technique. At the last level ($k = 2$), the diagonal inadmissible dense blocks $[\mathbf{H_1^{(2)}}, \mathbf{H_2^{(2)}}, \mathbf{H_3^{(2)}}, \mathbf{H_4^{(2)}}]$ are shown in solid red colors.

The construction begins by generating an L-level binary tree partitioning of the row/column indices based on their corresponding node coordinates. At the first level of the tree, nodes are divided into two groups by choosing a bisecting plane and assigning nodes that are on the same side of it to the same group. The bisecting plane is chosen to cut the axis-aligned minimum bounding box in half along its longest dimension, also known as the KD-tree partitioning. At the second level, each of the two groups from the first level is divided using the same method, resulting in a partition into four groups. This process continues recursively for L levels, where each level doubles the number of groups. At the final level, the mesh is divided into 2^L groups, ensuring a hierarchical partition structure. The groups are arranged hierarchically to maintain inheritance relationships. Specifically, each group at level k consists of two child groups of level $k+1$. Furthermore, the ith group from level k is formed by merging the groups $2i-1$ and $2i$ from level $k+1$ and is referred to as its parent. (Note: Without loss of generality, we assume that the nodes are numbered in an ordered way with an increasing node index corresponding to an increasing group index. In other words, at the Lth partitioning level, the indexes of the nodes

in the ith group are $G_i^{(L)} = \{N_{i-1}^{(L)}+1, N_{i-1}^{(L)}+2, \ldots, N_i^{(L)}\}$. We define $N_0^{(L)} = 0$ and $N_{2^L}^{(L)} = N_V$ and $N_i^{(L)}$ to be the index of the last element of the ith group at level L. As a result, the nodes in the ith group at level k have indexes $G_i^{(k)} = \{N_{2^{L-k}(i-1)}^{(L)}+1, N_{2^{L-k}(i-1)}^{(L)}+2, ..., N_{2^{L-k}i}^{(L)}\}$. This numbering scheme ensures that each group at a higher level is a direct aggregation of its corresponding child groups, preserving a structured hierarchy throughout the partitioning process.

The HODLR submatrices are selected and formatted according to simple rules. At each level, matrices corresponding to row and column indices of two groups sharing the same parent are stored using ACA. In other words, $\mathbf{H}(G_{2i}^{(k)}, G_{2i-1}^{(k)}) = \mathbf{u}_{(2i),(2i-1)}^{(k)} \mathbf{v}^{*(k)}_{(2i),(2i-1)} \approx \mathbf{A}(G_{2i}^{(k)}, G_{2i-1}^{(k)})$ and $\mathbf{H}(G_{2i-1}^{(k)}, G_{2i}^{(k)}) = \mathbf{u}_{(2i-1),(2i)}^{(k)} \mathbf{v}^{*(k)}_{(2i-1),(2i)} \approx \mathbf{A}(G_{2i-1}^{(k)}, G_{2i}^{(k)})$, where $k \in \{1, 2, \ldots, L\}$, $i \in \{1, 2, \ldots, 2^{k-1}\}$. At the lowest level, submatrices where the row and column indices belong to the same group are stored in a dense format. In other words, $\mathbf{H}(G_i^{(L)}, G_i^{(L)}) = \mathbf{H}_i^{(L)} = \mathbf{A}(G_i^{(L)}, G_i^{(L)})$. For example, in Fig. [2], at level 1, submatrices $\mathbf{H}(G_1^{(1)}, G_2^{(1)})$ and $\mathbf{H}(G_2^{(1)}, G_1^{(1)})$ are compressed and its factors are shown in green. At level 2, compression applies to submatrices $\mathbf{H}(G_1^{(2)}, G_2^{(2)})$, $\mathbf{H}(G_2^{(2)}, G_1^{(2)})$, $\mathbf{H}(G_3^{(2)}, G_4^{(2)})$ and $\mathbf{H}(G_4^{(2)}, G_3^{(2)})$ as shown in blue. Additionally, at the second level, all diagonal submatrices are stored in full uncompressed format as shown in red. For interested readers, a 3-level HODLR matrix and its LU decomposition are provided in the supplementary file.

C. Low-rank approximation of off-diagonal blocks

The HODLR compression relies on special properties of the $\mathbf{A}$. In particular, it is well known that its off-diagonal blocks have an ϵ-rank that scales as $log(\epsilon)$ under the quasi-stationary assumption [23]. The ϵ-rank of a matrix is the smallest number of singular values required to approximate it to ϵ accuracy (i.e., r is the ϵ-rank if r is the smallest integer such that $\|\mathbf{A} - \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^*\| < \epsilon\|\mathbf{A}\|$, where $\mathbf{U}_r$, $\mathbf{S}_r$, $\mathbf{V}_r$ are a singular value decomposition consisting of the the first r singular vectors and values. A similar decomposition can be found from linearly independent columns and rows of $\mathbf{A}$. For example, the interpolatory decomposition or CUR $\mathbf{A} \approx \mathbf{C}_{\mathbf{r_i}}\mathbf{O}_{\mathbf{r_i}}^{\dagger}\mathbf{R}_{\mathbf{r_i}}$, where $\mathbf{C}_{\mathbf{r_i}}$ are r_i linearly independent columns, $\mathbf{C}_{\mathbf{r_i}}$ are r_i linearly independent rows, $\mathbf{O}_{\mathbf{r_i}}$ is a matrix composed of the overlap matrix between the r_i rows and columns,and $\dagger$ denotes pseudo inverse. Whenever the rows and columns are chosen to maximize the determinant of $\mathbf{O}_{\mathbf{r_i}}$, this decomposition has been shown to produce practically nearly optimal compression. Furthermore, unlike the SVD, its construction in principle does not require the evaluation of the whole matrix. Here we employ the ACA algorithm to compress off-diagonal blocks of the HODLR matrix without needing to compute the full off-diagonal blocks. The ACA iteratively generates an interpolatory decomposition of a matrix $\mathbf{A}$, increasing the rank of the approximation by 1 at each iteration by appending a column and row, which will maximize the correction based on the current approximation. The ACA [24] was introduced to compress BEM matrices and has been very successful [21], [25], [26]. In particular, the ACA factorizes the rows and column of the matrix to write an equivalent form of the interpolatory decomposition involving only two factors $\mathbf{H}(G_{2i-1}^{(k)}, G_{2i}^{(k)}) \approx \mathbf{u}_{(2i-1),(2i)}^{(k)} \mathbf{v}^{(k)*}_{(2i-1),(2i)}$.

D. Computing the LU of the HODLR matrix

The system of equations described in (6) must be solved at each time step. Here we describe the method we use to decompose $\mathbf{H}$ into lower-triangular $\mathbf{L}$ and upper-triangular $\mathbf{U}$ matrices $\mathbf{H} = \mathbf{LU}$ in the HODLR format and the inverse $\mathbf{H}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}$. The charge distribution at each time-step can be determined using the inverse LU factors as $\rho^{(\tau+1)} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{b}^{(\tau+1)}$. The construction of the $\mathbf{U}^{-1}$ and $\mathbf{L}^{-1}$ only requires order $r_{max}^2 N_V log^2(N_V)$ computations, where r_{max} is the maximum compression rank; thereby reducing the computational run-time. The algorithm for computing the LU decomposition of an HODLR matrix is mentioned in [27]. Specifically, at level L, the LU factorization $\mathbf{H}_i^{(L)} = \mathbf{L}_i^{(L)}\mathbf{U}_i^{(L)}$ of each of the diagonal dense blocks are computed using LAPACK routines. The LU factorization for parent groups is computed using the following formula

$$\begin{aligned}
\mathbf{H}(G_i^{(k)}, G_i^{(k)}) =& \mathbf{L}_i^{(k)}\mathbf{U}_i^{(k)} \\
\mathbf{L}_i^{(k)} =& \begin{bmatrix} \mathbf{L}_{2i-1}^{(k+1)} & 0 \\ \mathbf{u}_{(2i),(2i-1)}^{(k+1)} \tilde{\mathbf{v}}^{*(k+1)}_{(2i),(2i-1)} & \mathbf{L}_{2i}^{(k+1)} \end{bmatrix} \\
\mathbf{U}_i^{(k)} =& \begin{bmatrix} \mathbf{U}_{2i-1}^{(k+1)} & \tilde{\mathbf{u}}_{(2i-1),(2i)}^{(k+1)} \mathbf{v}^{*(k+1)}_{(2i-1),(2i)} \\ 0 & \mathbf{U}_{2i}^{(k+1)} \end{bmatrix}.
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{v}}^{*(k+1)}_{(2i),(2i-1)} = \mathbf{v}^{*(k+1)}_{(2i),(2i-1)} \left[\mathbf{U}_{2i-1}^{(k+1)}\right]^{-1}$ and $\tilde{\mathbf{u}}_{(2i-1),(2i)}^{(k+1)} = \left[\mathbf{L}_{2i-1}^{(k+1)}\right]^{-1} \mathbf{u}_{(2i-1),(2i)}^{(k+1)}$. (Note: The algorithm as explained has a computational cost of order $r_{max}^2 N_V log^2(N_V)$ [16], [27]. Fig. [3] illustrates the LU factorization of the two-level matrix of Fig. [2]. At level 2, the LU factors are computed for each of the red blocks (Fig. [3A]). At the first level, the LU factorization of each group is computed by concatenating its children group LUs and generating new compressed off-diagonal blocks (Fig. [3B]). The final structure of the LU factors is shown at the bottom of the figure.

To compute the inverse of each of the LU factors at the lowest level, the inverse of the $\mathbf{L}_i^{(L)}$ and $\mathbf{U}_i^{(L)}$ factors is computed using standard methods. Then, the other levels are computed as

$$\begin{aligned}
\left[\mathbf{L}_i^{(k)}\right]^{-1} =& \begin{bmatrix} \left[\mathbf{L}_{2i-1}^{(k+1)}\right]^{-1} & 0 \\ \tilde{\tilde{\mathbf{u}}}_{(2i),(2i-1)}^{(k+1)} \tilde{\tilde{\mathbf{v}}}^{*(k+1)}_{(2i),(2i-1)} & \left[\mathbf{L}_{2i}^{(k+1)}\right]^{-1} \end{bmatrix} \\
\left[\mathbf{U}_i^{(k)}\right]^{-1} =& \begin{bmatrix} \left[\mathbf{U}_{2i-1}^{(k+1)}\right]^{-1} & \vdots\tilde{\mathbf{u}}_{(2i-1),(2i)}^{(k+1)} \vdots\tilde{\mathbf{v}}^{*(k+1)}_{(2i-1),(2i)} \\ 0 & \left[\mathbf{U}_{2i}^{(k+1)}\right]^{-1} \end{bmatrix}.
\end{aligned} \tag{9}$$

where $\tilde{\tilde{\mathbf{v}}}^{*(k+1)}_{(2i),(2i-1)} = \tilde{\mathbf{v}}^{*(k+1)}_{(2i),(2i-1)} \left[\mathbf{L}_{2i-1}^{(k+1)}\right]^{-1}$, $\tilde{\tilde{\mathbf{u}}}_{(2i),(2i-1)}^{(k+1)} = -\left[\mathbf{L}_{2i}^{(k+1)}\right]^{-1} \mathbf{u}_{(2i),(2i-1)}^{(k+1)}$, $\vdots\tilde{\mathbf{v}}^{*(k+1)}_{(2i-1),(2i)} =$

$$\mathbf{v}^{*(k+1)}_{(2i-1),(2i)}\left[\mathbf{U}^{(k+1)}_{2i}\right]^{-1} \quad \text{and} \quad \dot{\mathbf{u}}^{(k+1)}_{(2i-1),(2i)} = -\left[\mathbf{U}^{(k+1)}_{2i-1}\right]^{-1}\tilde{\mathbf{u}}^{(k+1)}_{(2i-1),(2i)}.$$

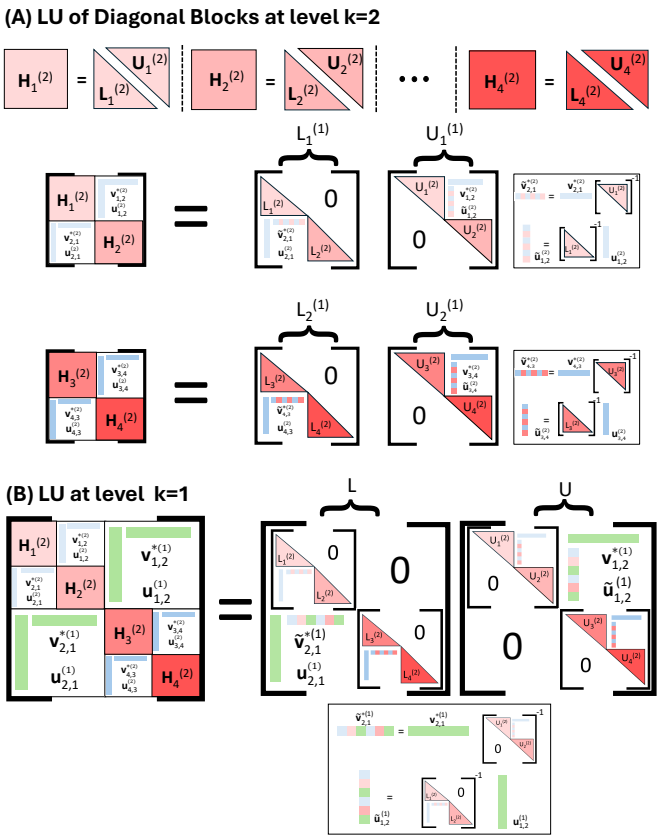


Fig. 3: LU decomposition of the HODLR matrix. (A) Diagonal inadmissible blocks at level $k = 2$ are factorized using direct LU decomposition. (B) LU decomposition at level $k = 1$.

A summary of the bidomain BEM algorithm is shown in Fig. 4, which depicts the algorithmic flow chart including HODLR compression and updating the gating variables.

E. Meshing a Neuron

A filamentary cable model of the L2/3 pyramidal neuron cell was obtained from the Blue Brain Project [28]. The cable model consists of vertices and the cable path, which is indicated by edges that connect the vertices. The width of each cable segment is indicated for each edge along with its membrane type, which is either soma, axon, apical dendrite, or basal dendrite. To solve the bidomain equation, we must generate a triangle mesh of the neuron surface from the filamentary cable description. This is done by using the algorithm described in [29] with a slight modification when generating the convex hulls. First, a temporary triangle mesh of a circle with a diameter equal to the cable diameter of its corresponding edge is added near each of its ends (Fig. [5B]). Second, individual meshes of edge segments are generated by meshing the convex hull of its two circles. Third, meshes of the vertex junctions are made by meshing the convex hull of all of the nearest circles of coincident edges (Fig. [5C]). Finally, all edge and vertex junction meshes are merged into

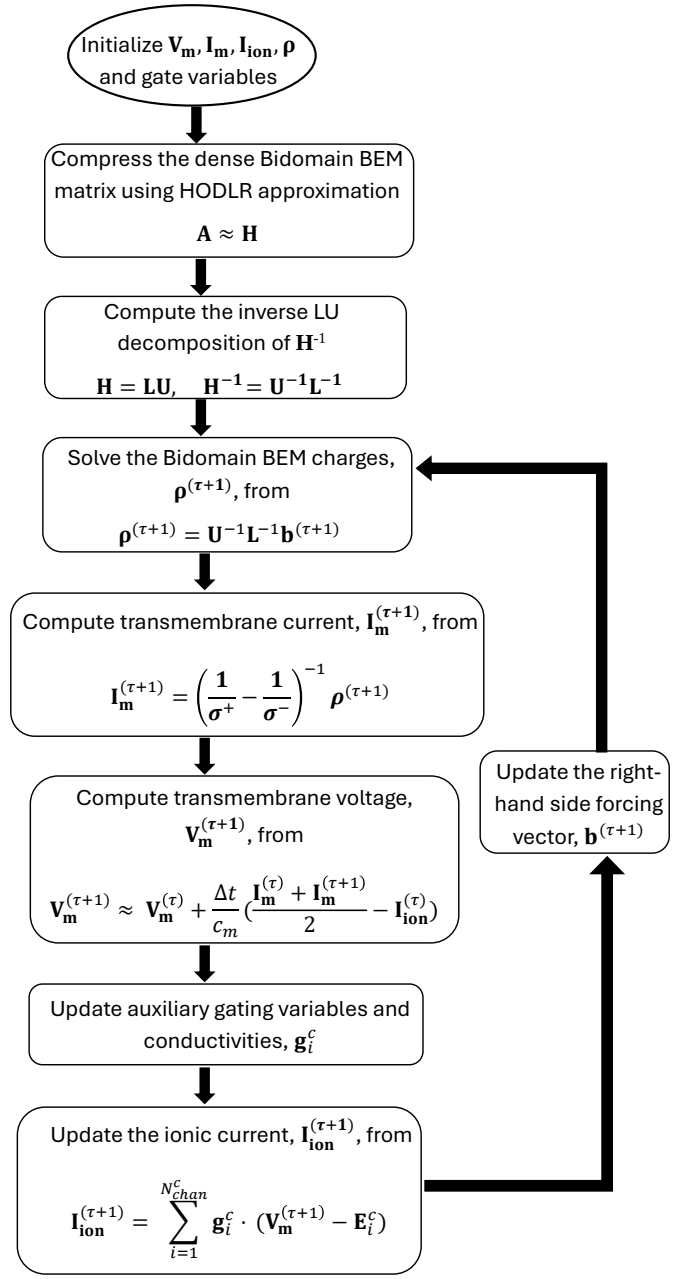


Fig. 4: A flowchart of the bidomain BEM algorithm. The algorithm repeats for a user-defined number of time steps.

a single mesh, and repeated triangles (i.e., internal circles) are removed. This results in a watertight mesh (Fig. [5D]). For stability, we modified the algorithm [29] by not generating the convex hull of the vertex junction circle convex hull, but by first mapping all points of circles to a unit sphere centered about its vertex before constructing its convex hull mesh. Fig. [5D] 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.

F. Gate Equations

Each compartment has a set of ion channels and equations governing the membrane mechanisms. The equivalent

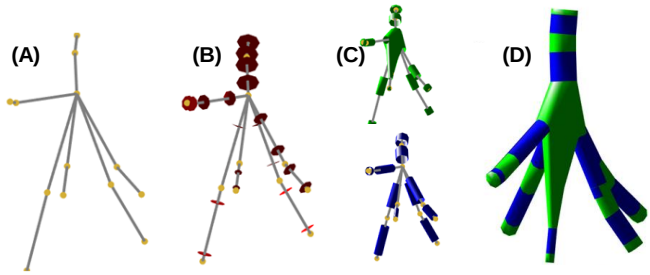
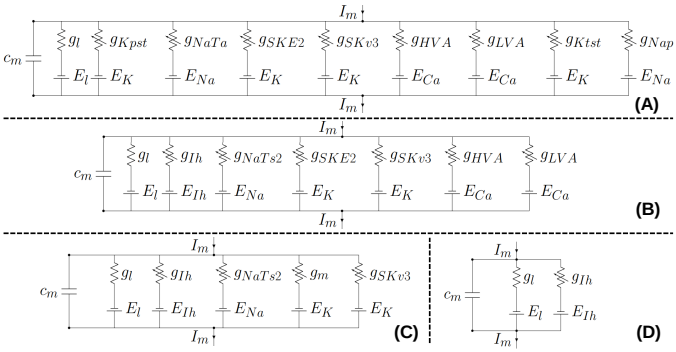


Fig. 5: Meshing neuron. (A) node locations and edges of the neuron. (B) Adding a temporary circle triangle mesh (red). (C) Constructing a mesh by connecting all circle meshes of each compartment, and then removing the circle meshes. (D) Merging meshes to construct a surface mesh of the neuron.

circuit models for each type of compartment are shown in Fig. [6]. Each ionic channel has its conductance and a voltage source that models the electrochemical gradient driving the ion movement between the extracellular and intracellular regions. For example, the axon, soma, apical dendrite and basal dendrite has ion channels with conductivities $\mathbf{g}^{axon}, \mathbf{g}^{soma}, \mathbf{g}^{apical}$, and $\mathbf{g}^{basal}$, respectively. The corresponding voltage sources are $\mathbf{E}^{axon}, \mathbf{E}^{soma}, \mathbf{E}^{apical}$, and $\mathbf{E}^{basal}$, respectively. A detailed description of each ion channel is provided in the supplementary file.



$\mathbf{g}^{axon} = \{g_l, g_{Kpst}, g_{NaTa}, g_{SKE2}, g_{SKv3}, g_{HVA}, g_{LVA}, g_{Ktst}, g_{Nap}\}$
$\mathbf{E}^{axon} = \{E_l, E_K, E_{Na}, E_K, E_K, E_{Ca}, E_{Ca}, E_K, E_{Na}\}$
$\mathbf{g}^{soma} = \{g_l, g_{Ih}, g_{NaTs2}, g_{SKE2}, g_{SKv3}, g_{HVA}, g_{LVA}\}$
$\mathbf{E}^{soma} = \{E_l, E_{Ih}, E_{Na}, E_K, E_K, E_{Ca}, E_{Ca}\}$
$\mathbf{g}^{apical} = \{g_l, g_{Ih}, g_{NaTs2}, g_m, g_{SKv3}\}$, $\mathbf{E}^{apical} = \{E_l, E_{Ih}, E_{Na}, E_K, E_K\}$
$\mathbf{g}^{basal} = \{g_l, g_{Ih}\}$, $\mathbf{E}^{basal} = \{E_l, E_{Ih}\}$

Fig. 6: Equivalent circuit of the axon (A), soma (B), apical dendrite (C), and basal dendrite (D). Here, the $g_{\{\cdot\}}$ denote the conductance of the specific ion channel, and the $E_{\{\cdot\}}$ is the voltage source that models the electrochemical gradient driving the specific ion movement through that channel between extracellular and intracellular regions.

The ionic current for each compartment type is

$$\mathbf{I_{ion}} = \sum_{i=1}^{N^c_{chan}} \mathbf{g}^c_i \cdot \left(\mathbf{V_m} - \mathbf{E}^c_i\right), \tag{10}$$

where $c \in \{soma, apical, basal, axon\}$ and N^c_{chan} is the number of parallel ion channels in the specific compartment. In the most general sense, the time dependence of each channel is fully described by the following channel parameters $\bar{g}^c_i$, γ_i, η_i, β_i, α_i, and τ_i summarized in Table IV in the supplementary file. In general, these parameters can be used to compute the elements of conductance vector $\mathbf{g}^c_i$ as $g^c_i(\mathbf{r};t;V_m) = \bar{g}^c_i n_i(\mathbf{r};t;V_m)^{\gamma_i}\bar{n}_i(\mathbf{r};t;V_m)^{\eta_i}$, where n_i and $\bar{n}_i$ are the probability of the ith ion channel being open and close, respectively. Both n_i and $\bar{n}_i$ follow the same differential equation

$$\frac{dn_i(\mathbf{r};t)}{dt} = \alpha_i(V_m(\mathbf{r};t))(1 - n_i(\mathbf{r};t)) - \beta_i(V_m(\mathbf{r};t))n_i(\mathbf{r};t), \tag{11}$$

where α_i and β_i are channel dependent parameters. Solving (11), the value of n_i (or, $\overline{n}_i$) during time-step $(t+\Delta t)$ is found from its value at t by the following time-stepping scheme

$$n_i(\mathbf{r};t+\Delta t) \approx n_i(\mathbf{r};\infty) + \left[n_i(\mathbf{r};t) - n_i(\mathbf{r};\infty)\right] exp\left\{\frac{-\Delta t}{\tau_i(\mathbf{r};t)}\right\}, \tag{12}$$

where $n_i(\mathbf{r};\infty)$ is the limiting probability of the ith ion channel being open for a steady value of $V_m(\mathbf{r};t)$ as t approaches ∞ and its value for each channel is given in Table I and Table III in the supplementary file. This time-stepping scheme solves the gating probability (11) assuming constant coefficients for the time interval between t and $t + \Delta t$.

G. Simulation Setup

To model the electrical stimulation of a neuron inside a head compartment, the neuron is submerged inside a cubic bath. The size of the neuron is of the order of 10^{-4}m^3 and the bath is $40 \times 40 \times 40$ mm^3. The neuron and the bath have conductivities of 1 S/m and 2 S/m. The electrodes are placed on the head surface, two sides of the bath as shown in Fig. [7] with the cathode in green and the anode in red. The electrodes are driven with an external current of amplitude 16 mA, duration of 5 ms (starting at 0.2 ms), and its distributions on both the cathode and anode is assumed uniform. The resulting E-field is nearly uniform and $\mathbf{E}(\mathbf{r};t) \approx 80\text{V/m}$ along the z-direction (from cathode to anode) for 5 ms starting at 0.2 ms. The bath consists of $2,168$ nodes with $4,332$ triangular facets, and the neuron consists of 16,830 nodes with 33,660 triangular facets; resulting in $N_V = 18,998$ nodes and $N_T = 37,992$ triangular facets in total. The HODLR matrix consists of $L = 9$ hierarchical levels, where the off-diagonal blocks are computed using ACA with a convergence tolerance of $\epsilon_H = 10^{-7}$, respectively. The transmembrane voltage is computed with a temporal resolution of 1 μs and a total of $N_t = 40,000$ time steps. The bath and cell are groups using distinct oct-tree hierarchies with 8 levels each to ensure that the first level separates the cell and bath into one group. This partition enables us to determine the strength and rank of the coupling between the cell and the bath. The equilibrium potential for the neuron membrane is set to -74.5 mV.

III. Results

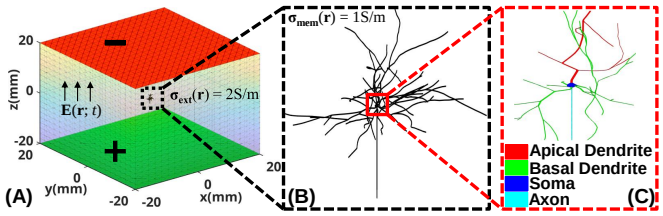


Fig. 7: The L2/3 pyramidal neuron is placed within an inhomogeneous medium. The intracellular and the extracellular spaces have conductivities of 1 S/m and 2 S/m, respectively. The cell has four compartment types (soma, axon, apical dendrite, and basal dendrite).

A. HODLR Matrix Ranks

A compressed form of the HODLR matrix is shown in Fig. [8], using 8 levels for both membrane nodes and bath nodes with $L = 9$. The charges on the neuron membrane generate an E-field that can be decomposed into a relatively small number of basis vectors—i.e., the interaction between membrane charges and evaluation points on the bath surface is of low rank. For example, the neuron-bath and bath-neuron couplings are represented using ranks of only 3 and 9, respectively.

Additionally, E-fields generated by charges at different group locations are also of low rank. For instance, the green and red submatrices in Fig. [8] correspond to interactions between the bottom and top halves of the cell membrane (Fig. [8B]). These submatrices have sizes of $3,845 \times 12,986$ and ranks of 219 and 218, resulting in a compression ratio of 2980. Fig. [8A] highlights the inadmissible dense blocks in blue. The submatrix of the bath is less compressible than that of the membrane (Fig. [8C]), likely because the membrane is a wire-like structure with fewer unknowns per unit volume compared to the solid bath boundary. The maximum rank required across all sub-blocks is 238.

B. Compression Ratio and Computational Run-time

All benchmarking results were obtained using an AMD EPYC 7502 32-core processor (2495.535 MHz) CPU. Fig. [9A] shows the required computational memory as a function of the number of neurons under stimulation, along with the corresponding compression ratio, defined as the ratio of memory required by the uncompressed BEM matrix to that of the HODLR-compressed version. Fig. [9B] displays the setup time for HODLR compression.

For a single neuron (18,998 nodes and 37,992 triangular facets), computing the full dense BEM matrix requires 5.38 GB of memory, while the compressed HODLR matrix requires only 0.24 GB, yielding a 22.4× compression. The setup times for the uncompressed and HODLR matrices are 77.92 and 32.88 minutes, respectively. Over 40,000 time steps, the transmembrane voltage computation takes 191.08 minutes (0.29 seconds/step) using the dense BEM matrix and only 19.6 minutes (0.03 seconds/step) using the HODLR matrix, resulting in a 10× speedup.

For a group of 8 neurons (136,808 nodes and 273,612 triangular facets), the HODLR matrix requires just 1.66 GB

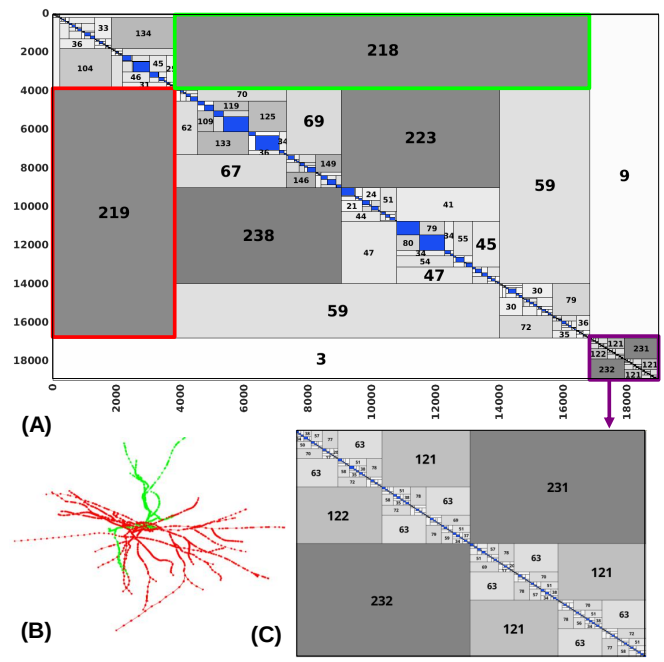


Fig. 8: (A) HODLR representation of the BEM matrix (dense blocks in blue). (B) Two sets of membrane nodes with weak interaction, shown in green and red blocks ($3,845 \times 12,986$) in (A), require only 219 and 218 ranks. (C) HODLR subblock for the interaction of charges on the bath boundary.

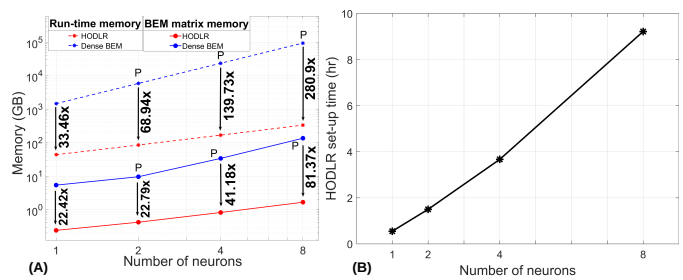


Fig. 9: (A) Computational run-time memory and the BEM matrix memory for the HODLR compression and the uncompressed dense BEM matrix as a function of the number of neurons. 'P' denotes the entries computed with the interpolation technique. Memory compression is shown as bold text. (B) Set-up time required by the HODLR compression method with respect to the number of neurons.

of memory compared to 135.07 GB for the dense BEM matrix—an 81.37× reduction. During the setup phase, peak RAM usage is 332.78 GB for HODLR and 93,478 GB for the dense BEM matrix, corresponding to a 280.90× compression. The total setup time for the 8-neuron configuration is 9.22 hours (Fig. [9B]).

C. Accuracy Benchmarks

Fig. [10] compares the predicted membrane voltage at representative points along the axon, soma, apical dendrite, and basal dendrite, as computed by the cable equation-based solver, the uncompressed BEM, and the HODLR-compressed

BEM.

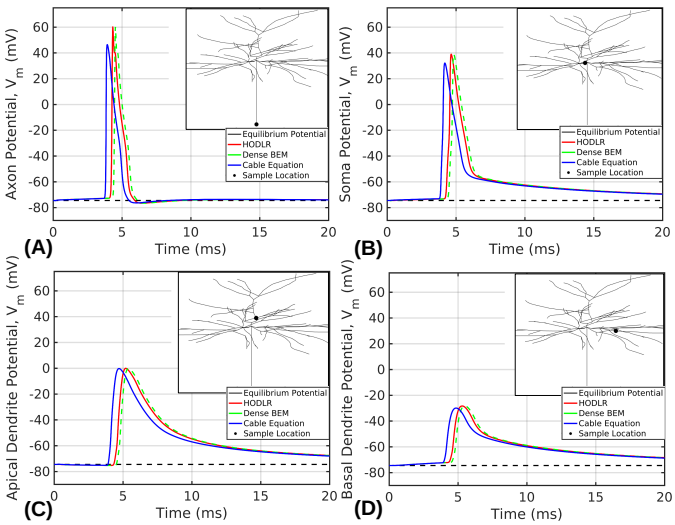


Fig. 10: Propagation of action potential (V_m) through the axon (A), soma (B), apical dendrite (C) and basal dendrite (D) computed from the uncompressed dense BEM matrix (green), the HODLR compression (red) and the cable equation approach (blue). The inset shows the corresponding spatial sample locations (black dot).

The action potentials are generated with an The uncompressed and HODLR-compressed BEM results are virtually indistinguishable, confirming the fidelity of the HODLR compression. However, the cable equation-based solution shows an earlier activation, approximately 0.40 ms ahead of the BEM results. At the axon (and soma), the cable equation also predicts a lower peak membrane voltage of 60.16 mV (and 38.98 mV), compared to 46.44 mV (and 32.14 mV) for both BEM-based solvers. This discrepancy likely stems from the simplified activation function used in cable models, as opposed to the more detailed electric field coupling captured by BEM. The action potential initiates at the distal axon and propagates through the axon, soma, and dendritic structures. The propagation speed is primarily influenced by the local cross-sectional area of the neuron. The consistent 0.40 ms delay across all sample locations in Fig. [10] suggests that the cable and BEM models exhibit the same propagation speed, despite the timing offset.

Fig. [11] provides additional detail at the axonal sample site. It shows the membrane voltage (V_m), transmembrane current (I_m), total channel conductance ($\sum_i g_i$), membrane charge density (ρ), and total ionic currents (I_{ion}). Notably, the transmembrane current, total ionic current, and membrane charge density each undergo two polarity reversals—once at action potential initiation and again at the peak membrane voltage. Individual ionic channel contributions are shown in Fig. [11A]. Some channels, such as NaTa, SKv3, and Kpst, contribute significantly to the total current (up to 708 A), while others, like Ktst and Nap, exhibit much smaller currents (up to 16 A). Calcium channels, the SKE2 channel, and leakage currents contribute negligibly, up to 1 A.

Fig. [12] illustrates the spatiotemporal propagation of the

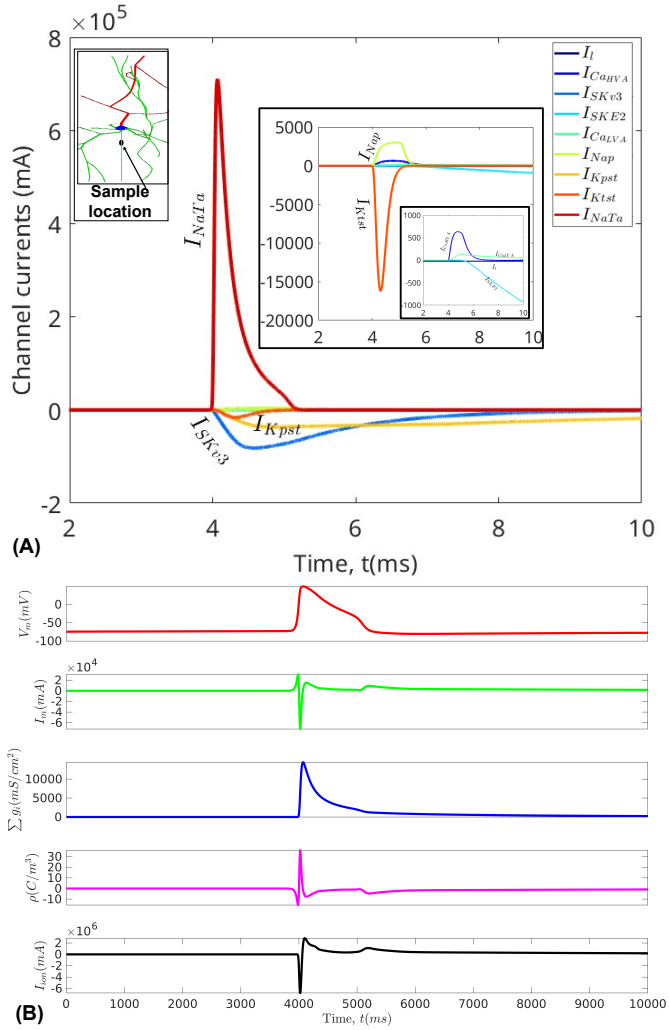


Fig. 11: (A) For the sample coordinate along the axon (inset), individual ionic channel currents are shown; (B) transmembrane voltage (V_m), transmembrane current (I_m), total channel conductance ($\sum_i g_i$), membrane charge density (ρ) and total ionic currents (I_{ion}) for the sample location in (A).

action potential across the neuronal membrane at successive time steps. The action potential initiates at the distal tip of the axon (Fig. [12A]), where the external electric field generated by the electrodes is optimally aligned with the axonal orientation. The depolarization then propagates upward through the soma (Fig. [12(B–C)]), before attenuating as it spreads into the dendritic branches (Fig. [12(D–F)]). This behavior closely aligns with predictions made by cable equation-based models and reflects known biophysical dynamics of action potential propagation in neurons.

D. Activation threshold of a realistic neuron

In this section, we compare the activation thresholds predicted by the cable equation and the BEM approach for a realistic neuron model (Fig. [13]) for an equilibrium potential of -74.5 mV.

The orientation angle θ between the z-axis (i.e., the direction of the external electric field) and the neuron's axon is

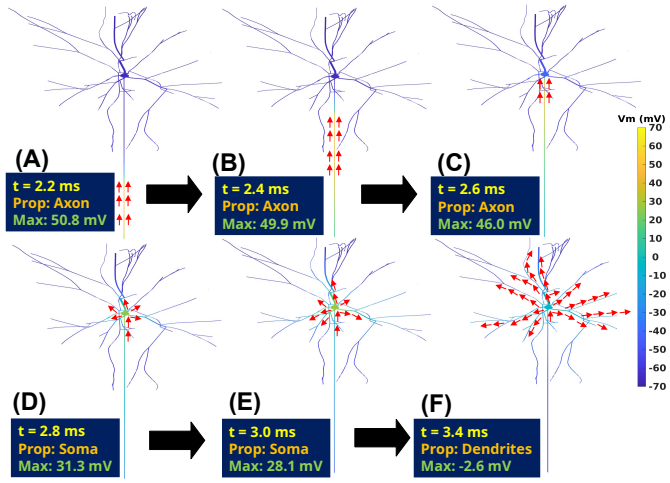


Fig. 12: The action potential starts propagating from one end of the axon (A) and traverses through the axon towards the soma (B-C). Next, it propagates along the dendrites to nearby neurons (D-F). The insets show the time stepping (t), propagating compartment ($prop$), and the maximum amplitude of the action potential. The red arrow follows the maximum transmembrane voltage.

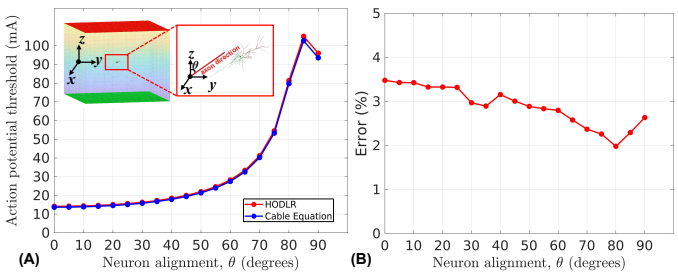


Fig. 13: Effect of neuron spatial orientation on the action potential (A). The action potential threshold increases as a function of the angle (θ) between the incident field and the axon for both the BEM and the cable-equation approach. (B) The relative error of the BEM threshold with respect to the cable-equation threshold increases as a function of the neuron spatial orientation (θ).

varied as a parameter (Fig. [13A]). When the neuron is fully aligned with the E-field ($\theta = 0°$), the activation thresholds are 14.16 mA for the BEM approach and 13.68 mA for the cable equation. When the neuron is oriented transversely to the E-field ($\theta = 90°$), the thresholds are 96.03 mA for BEM and 93.57 mA for the cable model. This corresponds to BEM-based thresholds that are 3.48% and 2.63% higher than those predicted by the cable equation for $\theta = 0°$ and $\theta = 90°$, respectively (Fig. [13B]). Across the range of orientations, the BEM approach consistently yields slightly higher activation thresholds. The smallest difference is 1.98% at $\theta = 80°$, and the largest is 3.48% at $\theta = 0°$.

E. Multiple Neurons

In this section, we analyze both compression efficiency and action potential behavior in simulations involving multiple neurons. Fig. [14A] illustrates a scenario in which a second neuron (N_2) gradually moves closer to a target neuron (N_1). The corresponding HODLR-compressed matrix is shown in Fig. [14B]. The E-field that the neurons generate on each other

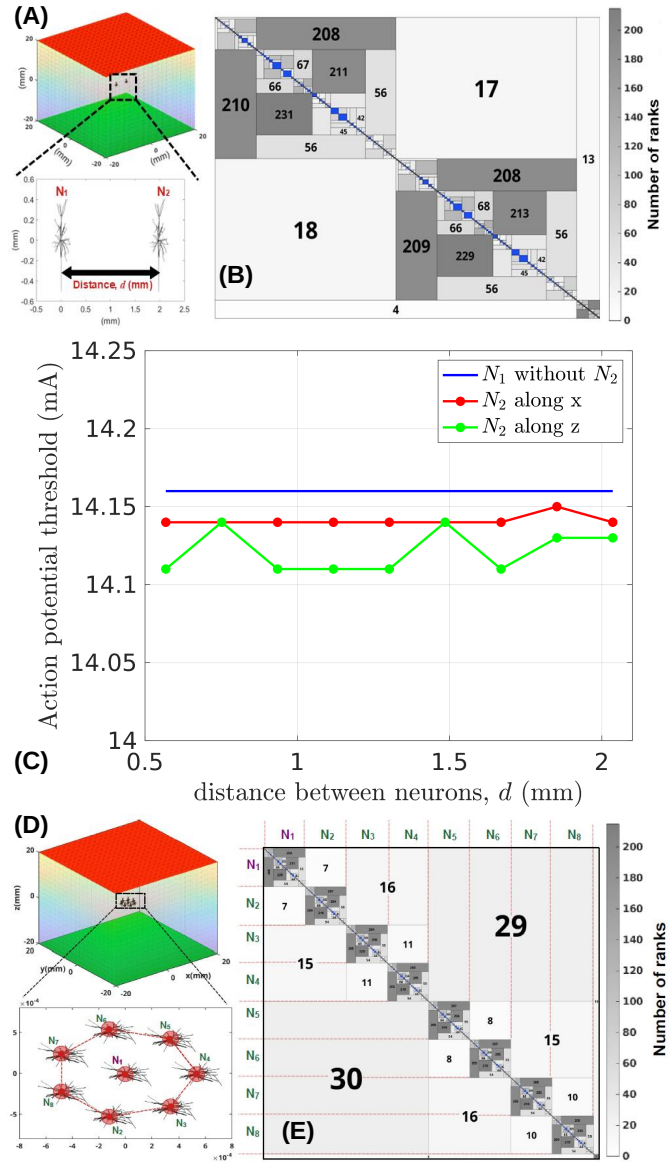


Fig. 14: (A) Two neurons at a distance, d, under the stimulation inside the bath. (B) HODLR compressed matrix for the two-neuron scenario. (C) The action potential threshold of the observed neuron (N_1) is lower when the second neuron (N_2) moves along the z-direction (parallel to the stimulating E-field) than along the x-direction. (D) A group of 7 neurons (N_2-N_8) surrounds an observed neuron (N_1) in a regular heptagon pattern. (E) The resultant HODLR matrix of the group of 8 neurons.

is highly compressible, requiring a rank of only 17 or 18. Fig. [14C] shows how the presence of N_2 affects the activation threshold of N_1 as N_2 approaches either along the x-direction or the z-direction. The results indicate that the action potential threshold of N_1 is only minimally affected by N_2's proximity. However, the threshold is slightly lower when N_2 approaches

along the z-direction compared to the x-direction. This may be because the external coil E-field impinges directly on N_2 when it moves along x, while N_1 partially shields it when it moves along z.

A similar analysis is performed for a group of 8 neurons (Fig. [14(D–E)]). Here, the observed neuron N_1 is surrounded by 7 other neurons (N_2–N_8), placed at the vertices of a regular heptagon (Fig. [14D]), with all neurons aligned along the direction of the stimulating E-field. The resulting HODLR matrix (Fig. [14E]) again shows highly compressible interactions between the neurons. As in the two-neuron case, the action potentials of all neurons remain unaffected by the presence of nearby cells.

These results demonstrate that our HODLR-based BEM framework effectively handles multi-neuron simulations, maintaining accuracy while offering significant compression.

IV. Discussion

Scalable bidomain modeling of neuron stimulation by a device-induced E-field using the BEM approach has been introduced by leveraging fast direct solvers. The computation and storage of the dense BEM matrix for a realistic neuron are computationally intractable. Additionally, updating the neuron membrane voltage at every time step requires recomputing the dense BEM matrix, limiting the applicability of the standard bidomain method. By utilizing the hierarchical off-diagonal low-rank (HODLR) form, the dense BEM matrix was precomputed and stored. The HODLR form reveals that interactions between charges on the membrane and the head surface, as well as interactions between charges on the same neuron, are highly compressible. An 8-level HODLR structure is sufficient to compute the BEM matrix without compromising the accuracy of action potential generation. A multi-neuron simulation scenario shows that the action potential of any given neuron is negligibly affected by nearby neurons.

Validation results indicate that using an 8-level hierarchical binary partitioning of the membrane nodes compresses the dense BEM matrix by 22.42 times for an L2/3 pyramidal neuron. Additionally, runtime memory is reduced by 33.46 times for the same neuron. With our existing hardware resources, simulations of up to 8 neurons show a BEM matrix compression of 81.37 times and a runtime memory reduction of 280.9 times. Moreover, it takes only 0.55 hours and 9.22 hours to precompute and store the BEM matrix in HODLR form for a single neuron and a group of 8 neurons, respectively. Our results show that by compressing the inverse of the BEM system of equations, it is feasible to solve for multiple cells using the bidomain model. The runtime memory of the solvers can likely be further reduced by performing computations in batches or, for direct solvers, by avoiding recursive implementations. For HODLR, there is a three-order difference in required storage between the runtime processes and the actual BEM matrix. This suggests that, by reducing runtime memory usage, thousands of cells could be modeled on a standard computing cluster. Computational time increases linearly with the number of cells, but this can be improved by parallelizing the generation of the HODLR matrices. Finally, the coupling ranks between cells and the surrounding medium, or between neighboring cells, tend to be low. This indicates potential for further improvement through matrix factorization using Schur complements, which could eliminate the need to solve for the surrounding media boundary at every time step. With these improvements, using HODLR solvers to analyze networks of thousands of neurons using BEM could become practical.

All results presented here indicate that cable solvers can capture the key features of stimulation by a device-induced E-field. The BEM introduces subtle differences, likely due to the inclusion of geometric factors. Incorporating elliptical cross-sections for cell segments, more realistic representations of the soma, and the inclusion of dendritic spines could lead to further deviations from the cable model, and these effects warrant further study. As shown in our previous work [10], both the cable equation and BEM demonstrate a similar relationship between activation threshold and cell angle, suggesting that the key features of activation are preserved in the simplified cable model, with differences arising from more nuanced effects.

Multi-neuron scenarios have recently been shown to yield different activation thresholds than single-cell simulations [10], [30]. These scenarios cannot be accurately studied using standard cable approaches, which assume that the cell's activity and boundaries do not distort the device-induced E-fields. This work represents a significant step toward enabling the practical simulation of multi-neuron scenarios while preserving full geometric representations of cells and more accurately predicting the effects of E-fields on neural networks.

V. Conclusion

By leveraging direct solvers, we presented a systematic approach to compress the dense bidomain BEM matrix using the HODLR form, enabling full coupling of the neuron, the inhomogeneous medium, and the stimulating device for a realistic L2/3 rat pyramidal neuron. Compression and storage of the dense BEM matrix take only 0.55 hours, achieving compression ratios of 22.42 times for the matrix itself and 33.46 times for maximum runtime memory. Future work will explore the influence of geometric factors on stimulation and expand to multi-cell scenarios.

References

[1] A. Fellner, A. Heshmat, P. Werginz, and F. Rattay, "A finite element method framework to model extracellular neural stimulation," *Journal of Neural Engineering*, vol. 19, no. 2, p. 022001, 2022.

[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] P. J. Basser and B. J. Roth, "New currents in electrical stimulation of excitable tissues," *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 377–397, 2000.

[4] B. J. Roth and P. J. Basser, "A model of the stimulation of a nerve fiber by electromagnetic induction," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 6, pp. 588–597, 1990.

[5] J. Gomez-Tames, I. Laakso, T. Murakami, Y. Ugawa, and A. Hirata, "Tms activation site estimation using multiscale realistic head models," *Journal of Neural Engineering*, vol. 17, no. 3, p. 036004, 2020.

[6] B. Wang, A. S. Aberra, W. M. Grill, and A. V. Peterchev, "Modified cable equation incorporating transverse polarization of neuronal membranes for accurate coupling of electric fields," *Journal of Neural Engineering*, vol. 15, no. 2, p. 026003, 2018.

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

[8] G. M. Noetscher, D. Tang, A. R. Nummenmaa, C. S. Bingham, C. C. McIntyre, and S. N. Makaroff, "Estimations of charge deposition onto convoluted axon surfaces within extracellular electric fields," *IEEE Transactions on Biomedical Engineering*, vol. 71, no. 1, pp. 307–317, 2023.

[9] Z. Qi, G. M. Noetscher, A. Miles, K. Weise, T. R. Knösche, C. R. Cadman, A. R. Potashinsky, K. Liu, W. A. Wartman, G. N. Ponasso *et al.*, "Enabling electric field model of microscopically realistic brain," *Brain Stimulation*, vol. 18, no. 1, pp. 77–93, 2025.

[10] 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.

[11] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, 1987.

[12] ——, "A new version of the fast multipole method for the laplace equation in three dimensions," *Acta numerica*, vol. 6, pp. 229–269, 1997.

[13] 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.

[14] S. Joucla, A. Glière, and B. Yvert, "Current approaches to model extracellular electrical neural microstimulation," *Frontiers in computational neuroscience*, vol. 8, p. 13, 2014.

[15] W. Ying and C. S. Henriquez, "Hybrid finite element method for describing the electrical response of biological cells to applied fields," *IEEE transactions on biomedical engineering*, vol. 54, no. 4, pp. 611–620, 2007.

[16] S. Ambikasaran, *Fast algorithms for dense numerical linear algebra and applications*. Stanford University, 2013.

[17] S. S. Vaezi and L. J. Gomez, "Novel volume integral equation approach for low-frequency e-field dosimetry of transcranial magnetic stimulation," *IEEE Transactions on Magnetics*, 2024.

[18] R. D. Graglia, "On the numerical integration of the linear shape functions times the 3-d green's function or its gradient on a plane triangle," *IEEE transactions on antennas and propagation*, vol. 41, no. 10, pp. 1448–1455, 1993.

[19] D. Wilton, S. Rao, A. Glisson, D. Schaubert, O. Al-Bundak, and C. Butler, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *IEEE Transactions on Antennas and Propagation*, vol. 32, no. 3, pp. 276–281, 1984.

[20] M. Bebendorf and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *Computing*, vol. 70, pp. 1–24, 2003.

[21] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of emc problems," *IEEE transactions on electromagnetic compatibility*, vol. 47, no. 4, pp. 763–773, 2005.

[22] Y. Liu, W. Sid-Lakhdar, E. Rebrova, P. Ghysels, and X. S. Li, "A parallel hierarchical blocked adaptive cross approximation algorithm," *The International Journal of High Performance Computing Applications*, vol. 34, no. 4, pp. 394–408, 2020.

[23] H. Guo, Y. Liu, J. Hu, and E. Michielssen, "A butterfly-based direct integral-equation solver using hierarchical lu factorization for analyzing scattering from electrically large conducting objects," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 9, pp. 4742–4750, 2017.

[24] S. Rjasanow, "Adaptive cross approximation of dense matrices," in *Int. Association Boundary Element Methods Conf., IABEM*, 2002, pp. 28–30.

[25] S. Kurz, O. Rain, and S. Rjasanow, "The adaptive cross-approximation technique for the 3d boundary-element method," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 421–424, 2002.

[26] D. Wang, N. I. Hasan, M. Dannhauer, A. C. Yucel, and L. J. Gomez, "Fast computational e-field dosimetry for transcranial magnetic stimulation using adaptive cross approximation and auxiliary dipole method (aca-adm)," *NeuroImage*, vol. 267, p. 119850, 2023.

[27] W. Hackbusch *et al.*, *Hierarchical matrices: algorithms and analysis*. Springer, 2015, vol. 49.

[28] 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.

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

[30] K. Weise, S. N. Makaroff, O. Numssen, M. Bikson, and T. R. Knösche, "Statistical method accounts for microscopic electric field distortions around neurons when simulating activation thresholds," *Brain Stimulation*, 2025.

[31] J. Rettig, F. Wunder, M. Stocker, R. Lichtinghagen, F. Mastiaux, S. Beckh, W. Kues, P. Pedarzani, K. H. Schröter, and J. P. Ruppersberg, "Characterization of a shaw-related potassium channel family in rat brain." *The EMBO journal*, vol. 11, no. 7, pp. 2473–2486, 1992.

[32] M. Köhler, B. Hirschberg, C. Bond, J. M. Kinzie, N. Marrion, J. Maylie, and J. Adelman, "Small-conductance, calcium-activated potassium channels from mammalian brain," *Science*, vol. 273, no. 5282, pp. 1709–1714, 1996.

[33] J. Magistretti and A. Alonso, "Biophysical properties and slow voltage-dependent inactivation of a sustained sodium current in entorhinal cortex layer-ii principal neurons: a whole-cell and single-channel study," *The Journal of General Physiology*, vol. 114, no. 4, pp. 491–509, 1999.

[34] A. Korngreen and B. Sakmann, "Voltage-gated k+ channels in layer 5 neocortical pyramidal neurones from young rats: subtypes and gradients," *The Journal of Physiology*, vol. 525, no. 3, pp. 621–639, 2000.

[35] C. M. Colbert and E. Pan, "Ion channel properties underlying axonal action potential initiation in pyramidal neurons," *Nature neuroscience*, vol. 5, no. 6, pp. 533–538, 2002.

[36] M. H. Kole, S. Hallermann, and G. J. Stuart, "Single ih channels in pyramidal neuron dendrites: properties, distribution, and impact on action potential output," *Journal of Neuroscience*, vol. 26, no. 6, pp. 1677–1687, 2006.

[37] P. R. Adams, D. A. Brown, and A. Constanti, "M-currents and other potassium currents in bullfrog sympathetic neurones," *The Journal of Physiology*, vol. 330, no. 1, pp. 537–572, 1982.

[38] I. Reuveni, A. Friedman, Y. Amitai, and M. J. Gutnick, "Stepwise repolarization from ca2+ plateaus in neocortical pyramidal cells: evidence for nonhomogeneous distribution of hva ca2+ channels in dendrites," *Journal of Neuroscience*, vol. 13, no. 11, pp. 4609–4621, 1993.

[39] R. B. Avery and D. Johnston, "Multiple channel types contribute to the low-voltage-activated calcium current in hippocampal ca3 pyramidal neurons," *Journal of Neuroscience*, vol. 16, no. 18, pp. 5567–5582, 1996.

[40] A. Randall and R. Tsien, "Contrasting biophysical and pharmacological properties of t-type and r-type calcium channels," *Neuropharmacology*, vol. 36, no. 7, pp. 879–893, 1997.