*Progress In Electromagnetics Research B, Vol. 82, 189–210, 2018*

# $\mathcal{H}$-Matrix Arithmetic for Fast Direct and Iterative Method of Moment Solution of Surface-Volume-Surface EFIE for 3-D Radiation Problems

**Reza Gholami[1, *], Jamiu Mojolagbe[1], Anton Menshov[2], Farhad S. H. Lori[1], and Vladimir Okhmatovski[1]**

**Abstract**—Hierarchical ($\mathcal{H}$-) matrix based fast direct and iterative algorithms are presented for acceleration of the Method of Moment (MoM) solution of the Surface-Volume-Surface Electric Field Integral Equation (SVS-EFIE) formulated for scattering and radiation problems on homogeneous dielectric objects. As the SVS-EFIE features the product of the integral operator mapping the tangential equivalent electric current on the surface of the scatterer to the volume polarization current and the integral operator mapping the volume polarization current to the tangential component of the scattered electric field, its MoM discretization produces the product of non-square matrices. Formation of the non-square $\mathcal{H}$-matrices for the MoM discretized integral operators is described. The algorithms for arithmetics pertinent to the product of the non-square $\mathcal{H}$-matrices are explained. The memory and CPU time complexity scaling of the required $\mathcal{H}$-matrix operations are analyzed in details and verified numerically. The numerical validation of the proposed algorithm is provided for both the low-loss dielectric objects as well as for the high-loss biological tissues found in the bioelectromagnetics applications. The numerical experiments demonstrate a significant reduction of memory usage and a considerable speedup for CPU time compared to naïve MoM, thus, enabling solution of the large-scale scattering and radiation problems with the SVS-EFIE.

## 1. INTRODUCTION

Electromagnetic field interactions with the biological tissues occur in many natural and staged scenarios ranging from involuntary human body exposures to the natural and man-made sources of RF and microwave radiation [1] to the near field radiation of the mobile [2], wearable and implanted antennas [3] or biomedical imaging systems [4]. Understanding, prediction, and control of such interactions is becoming of paramount importance in the design of the novel body mounted communication systems [5] and sensor networks [6], setting up the Internet-of-Things connected home and office spaces [7], development of smart antennas for 5G systems [8], and many other practical applications [9].

The limiting capabilities of the non-invasive experimental techniques for studies of field-to-body interactions motivate development of computational methods [10] for accurate virtual prototyping of such phenomenon. Due to significant inhomogeneity of the biological tissues, the methods of computational electromagnetics (CEM) based on the direct discretization of the Maxwell Equations such as Finite-Difference Time-Domain (FDTD) [11] and discretization of the wave equations such as the Finite Element Method (FEM) [12] have become the most popular approaches for bioelectromagnetics (BioEM) analysis with various commercial tools available [13, 14]. While FDTD and FEM offer great

*Received 19 October 2018, Accepted 28 November 2018, Scheduled 7 December 2018*
* Corresponding author: Reza Gholami (gholamir@myumanitoba.ca).
[1] Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada. [2] Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA.

versatility and simplicity in implementation and parallelization, they are plagued by fundamental limitations of error accumulation when propagation of wave phenomena over electrically large distances is analyzed [15] and necessity to discretize the space outside the regions of interest [12]. Integral equations (IEs) of CEM discretized with Method of Moments (MoM) [16] establish computational frameworks free of these disadvantages. The discretization of the IEs, however, results in the dense matrix equations which require development of sophisticated fast algorithms [15] in order to reduce the CPU time and memory complexity associated with their solution. The detailed comparisons in terms of the achieved accuracy and used computational resources between the differential and integral equations based approaches in application to solution of the BioEM problems can be found in [17].

The development of fast algorithms aiding iterative solutions of the dense matrix equations [18, 19] which result from the discretization of the IEs has reached a certain maturity [15] with Multi-Level-Fast-Multipole-Method (MLFMM) [20–23], FFT-based methods [24–26], and Adaptive Cross Approximation (ACA) [27, 28] methods dominating the field. The grand challenge in use of the fast iterative algorithms though is in development of appropriate preconditioning schemes which ensure sufficiently rapid convergence under conditions of multiscale discretization, high disparity in the material properties, and broad range of frequencies. To date development of such preconditioning schemes has been met with only partial success and remains a rather 'black art' than science [29].

The recent efforts in circumventing this grand challenge posed by the lack of robustness in the iterative schemes for the solution of the dense matrix equations have been in two areas. The first is the construction of well-conditioned IE formulations and their appropriate discretization schemes [30], and the second is the development of the fast methods for direct solution of the dense matrix equations [31–38]. The non-iterative nature of the latter allows them to remain largely insensitive to the deterioration of the IEs conditioning stemming from the multiscaling and other factors and, hence, provide robust computational schemes under conditions where traditional iterative schemes fail.

In this work, we develop a hierarchical ($\mathcal{H}$-) matrix based MoM frameworks [31, 32] for solution of the Surface-Volume-Surface Electric Field Integral Equation (SVS-EFIE) which we recently introduced for analysis of the scattering problems on dielectric objects [39–43]. The SVS-EFIE is a class of Single-Source Surface Integral Equations (SSSIE) [44, 45] which reduces the number of unknowns by half compared to the traditional surface integral-equation formulations [46, 47]. It features only one product of electric-field type integral operators and allows for a mixed potential formulation free of hyper-singular integrals under MoM discretization. These benefits of the SVS-EFIE come at an extra cost of computing field translations from the scatterer surface to its volume and then from its volume back to the surface. In the BioEM applications, computation of the field throughout the volume of the lossy body tissues is typically required for determining of the specific absorption rate, depth of field penetration, and localization of the high field concentration areas making SVS-EFIE particularly suitable [43] for solution of such problems. Due to field translation to the volume of the object of interest, discretization of both surface and volume field quantities takes place in the SVS-EFIE. As such, the memory requirements and computational complexity of dense matrix operations and storage for naïve MoM solution of the SVS-EFIE become prohibitive for tackling of practically important problems. To reduce the computational and memory costs, an $\mathcal{H}$-matrix based fast direct and iterative solvers are developed in this work to accelerate MoM solution of the SVS-EFIE for 3-D scattering problems on dielectric objects.

## 2. FORMULATIONS AND EQUATIONS OF SVS-EFIE AND ITS MOM SOLUTION

In our previous work [40, 43], we introduced a new type of an SSSIE by combining the ideas of the traditional volume equivalence principle and the theory of SSIEs. The SVS-EFIE formulation for 3-D scattering problem is given by Eq. (1), where $\boldsymbol{r}$ is an observation location on the scatterer boundary $\partial V$; $\boldsymbol{r}'$ is the position-vector of the total electric field inside the scatterer; and complex relative permittivity $\epsilon$ is $\epsilon = \varepsilon + \sigma/(i\omega\varepsilon_0)$. In Eq. (1), $\hat{\mathbf{t}}$ is the tangential vector to the scatterer surface $\partial V$, and $\omega$ is the cyclic frequency.

$$i\omega\mu_0\hat{\mathbf{t}}\cdot\left(-\oint_{\partial V}\bar{\bar{G}}_\epsilon(\boldsymbol{r},\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')ds''+k_0^2\,(\epsilon-1)\int_V\bar{\bar{G}}_0(\boldsymbol{r},\boldsymbol{r}')\cdot\oint_{\partial V}\bar{\bar{G}}_\epsilon(\boldsymbol{r}',\boldsymbol{r}'')\cdot\boldsymbol{J}(\boldsymbol{r}'')ds''dv'\right)=\hat{\mathbf{t}}\cdot\boldsymbol{E}^{\mathrm{inc}}(\boldsymbol{r}),\quad \boldsymbol{r}\in\partial V. \tag{1}$$

It is convenient to express Eq. (1) in the integral-operator form

$$-\underbrace{\left(\mathcal{T}_{\epsilon,\nabla\Phi}^{\partial V,\partial V}+\boldsymbol{\mathcal{T}}_{\epsilon,A}^{\partial V,\partial V}\right)}_{\mathcal{T}_{\epsilon}^{\partial V,\partial V}}\circ\boldsymbol{J}+\underbrace{\left(\boldsymbol{\mathcal{T}}_{0,\nabla\varphi}^{\partial V,V}+\boldsymbol{\mathcal{T}}_{0,a}^{\partial V,V}\right)}_{\mathcal{T}_{0}^{\partial V,V}}\circ\underbrace{\left(\boldsymbol{\mathcal{T}}_{\epsilon,\nabla\Phi}^{V,\partial V}+\mathcal{T}_{\epsilon,A}^{V,\partial V}\right)}_{\mathcal{T}_{\epsilon}^{V,\partial V}}\circ\boldsymbol{J}=\hat{\mathbf{t}}\cdot\boldsymbol{E}^{\mathrm{inc}},\tag{2}$$

where $\mathcal{T}_{\epsilon}^{\partial V,\partial V}$, $\mathcal{T}_{0}^{\partial V,V}$, and $\mathcal{T}_{\epsilon}^{V,\partial V}$ are surface-to-surface, volume-to-surface, and surface-to-volume operators, respectively. Operators $\mathcal{T}_{\epsilon}^{\partial V,\partial V}$, $\mathcal{T}_{0}^{\partial V,V}$, and $\mathcal{T}_{\epsilon}^{V,\partial V}$ in Eq. (2) are composed as a sum of two integral operators corresponding to the scalar and vector potentials as detailed in [40].

In order to solve SVS-EFIE in Eq. (1) with MoM, the volume of the scatterer $V$ is discretized with a 3-D mesh consisting of $N$ tetrahedron elements, and its surface $\partial V$ is discretized with a 2-D mesh consisting of $M$ triangle elements. Discretization of the unknown tangential weighting function $\boldsymbol{J}$ defined on the boundary $\partial V$ is performed over $P$ Rao-Wilton-Glisson (RWG) basis functions [48] as

$$\boldsymbol{J}(\boldsymbol{r}'')\cong\sum_{m''=1}^{P}I_{m''}\boldsymbol{t}_{m''}(\boldsymbol{r}'').\tag{3}$$

Following the standard MoM procedure (described in detail in [40]), the SVS-EFIE is reduced to the set of linear algebraic equations with respect to the unknown coefficients vector $I$ in the expansion of the unknown surface weighting function in Eq. (3). The MoM matrix equation structure is shown in Fig. 1.

The surface-to-surface impedance matrix $Z_{\epsilon}^{\partial V,\partial V}$ corresponding to the operator $\mathcal{T}_{\epsilon}^{\partial V,\partial V}$ is a $P\times P$ square matrix since the surface $\partial V$ serves as support for both the domain and range of $\mathcal{T}_{\epsilon}^{\partial V,\partial V}$. The $x$, $y$, and $z$ components of $\boldsymbol{Z}_{0}^{\partial V,V}$, translating the volume polarization current to the tangential component of the scattered electric field and $\boldsymbol{Z}_{\epsilon}^{V,\partial V}$ translating the unknown tangential weighting source function to the total field inside the scatterer are handled separately and result in $P\times 3N$ and $3N\times P$ sized matrices. These matrices have rectangular structures since their continuous counterparts have different range and domain supports: $\partial V$ and $V$ for $\mathcal{T}_{0}^{\partial V,V}$ and $V$ and $\partial V$ for $\mathcal{T}_{\epsilon}^{V,\partial V}$, respectively. In Fig. 1, $\Gamma$ is the Gram matrix with elements of $\Gamma_{n,n'}=V_n$, if $n=n'$ and zero, otherwise, $n=1,\ldots,N$. Here, $V_n$ is the volume of the $n$th tetrahedron element.

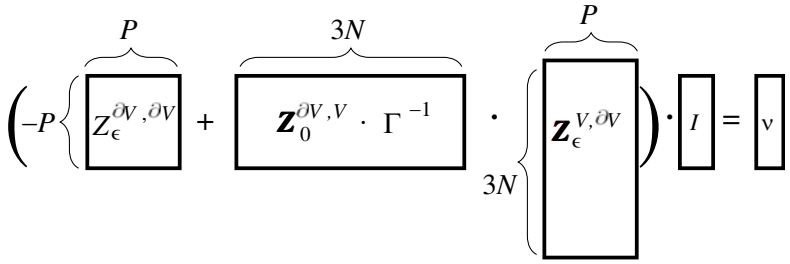


**Figure 1.** System of linear equations structure resulting from MoM discretization of the SVS-EFIE (1). Here, $P$ is the number of RWG basis functions on the boundary $\partial V$ and $N$ is the number of tetrahedron elements in the volume $V$.

## 3. $\mathcal{H}$-MATRIX ACCELERATION OF MOM SOLUTION

Blocks of the MoM matrices corresponding to far interactions can be compressed using ACA [27, 28] or other low-rank factorization methods [49, 50] allowing for significant reduction of storage requirements and fast arithmetic in $\mathcal{H}$-matrix format, including $\mathcal{H}$-LU decomposition and matrix-vector product (MVP) for the fast direct and fast iterative solution of the SVS-EFIE, respectively. Since the SVS-EFIE in Eq. (2) involves a product of integral operators with range and domain supports being different, the $\mathcal{H}$-matrix based MoM solution of the SVS-EFIE in Eq. (1) requires operations with rectangular $\mathcal{H}$-matrices and separate hierarchical interaction trees as discussed in the following subsections.

### 3.1. Multilevel Geometry Partitioning

The $\mathcal{H}$-matrix based MoM algorithm starts with a multilevel partitioning of discretized domains. In this work, we use geometric based bisection [32] that performs well for relatively uniform meshes. Since two different domains $\partial V$ and $V$ are discretized, two separate cluster trees are created: binary tree $T_S$ for the test functions and binary tree $T_V$ for the basis functions (Fig. 2). To construct a surface cluster tree $T_S$, we start from the full index set of RWG basis/testing functions $S_1^{(0)}$ (the superscript represents the level and the subscript represents the ID of the set within the level). Next, we determine the bounding box for this set and partition it in half along the largest dimension to form disjoint children clusters $S_1^{(1)}$ and $S_2^{(1)}$. The partitioning process is repeated recursively until the number of basis/testing functions becomes less than the predetermined leafsize $n_{\min}$ parameter which controls the depth of the cluster tree. The leafsize $n_{\min}$ is usually chosen empirically, and it is problem/hardware dependent. The cluster tree for the volume domain $T_V$ is constructed similarly to $T_S$ with the piece-wise basis/testing functions on the tetrahedron elements being partitioned as opposed to the RWGs. As depicted in Fig. 2, both surface and volume basis/testing functions are hierarchically divided into an $L$-level tree of clusters. In the subsequent discussion, the level $\ell$ relative of the cluster $S^{(j)}$ at the $j$th level is denoted as $R_\ell(S^{(j)})$.[†]

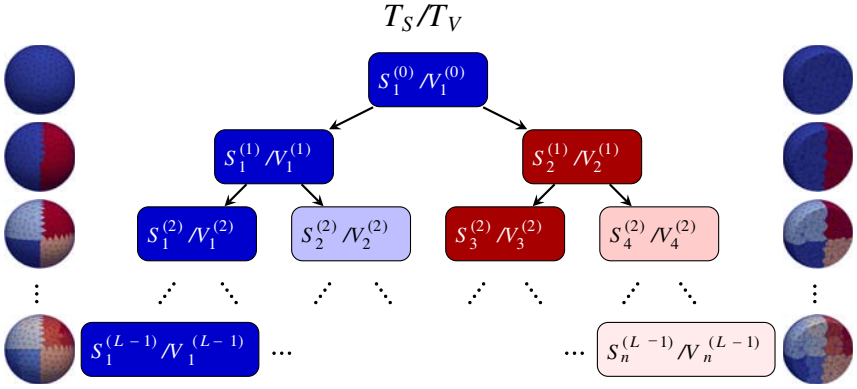


**Figure 2.** Hierarchical partitioning of RWG basis/testing functions for surface mesh and piece-wise basis/testing functions for volume mesh into an $L$-level hierarchy of clusters. $T_S$ and $T_V$ represent surface and volume cluster trees, respectively.

### 3.2. $\mathcal{H}$-Matrix Structure for the SVS-EFIE Operators

To approximate the impedance matrices of the MoM discretized SVS-EFIE integral operators, the hierarchical matrix structure for each impedance matrix is constructed. The process starts by taking the root elements of the cluster trees corresponding to the range (observer) and domain (source) of the MoM discretized integral operator. The distance between the source and observer, and the size of the individual clusters are calculated, and their interaction is classified as admissible or inadmissible according to the following criterion [31]

$$\min\{\text{diam}(\mathcal{B}_{\text{obs}}), \text{diam}(\mathcal{B}_{\text{src}})\} \leq \eta \ \text{dist}(\mathcal{B}_{\text{obs}}, \mathcal{B}_{\text{src}}) \tag{4}$$

where $\mathcal{B}_{\text{obs}}$ and $\mathcal{B}_{\text{src}}$ are the bounding boxes of the observer and source clusters, respectively; $\text{diam}(\cdot)$ and $\text{dist}(\cdot,\cdot)$ denote the Euclidean diameter and minimum distance between these bounding boxes. In Eq. (4), parameter $\eta$ is a positive real number that controls the amount of admissible blocks and accuracy of the solver. The interactions between the test/basis functions in the clusters are considered admissible (corresponding matrix block is rank-deficient), if the distance between the bounding boxes

[†] Terminology: Relatives of a cluster at all levels $\ell$ higher than its own level $j$ (i.e., $\ell < j$) are called *ancestors*. For example, for cluster $S_1^{(2)}$ in Fig. 2, its relative $R_0(S_1^{(2)})$ at level 0 is $S_1^{(0)}$, e.g., $R_0(S_1^{(2)}) = S_1^{(0)}$. Similarly, level 1 relative $R_1(S_1^{(2)})$ of $S_1^{(2)}$ is $S_1^{(1)}$. Relatives of a cluster at its own level (i.e., $\ell = j$) $R_2(S_1^{(2)}) = \{S_1^{(2)}, S_2^{(2)}\}$ are the cluster itself and its *sibling* $R_2(S_1^{(2)}) = S_2^{(2)}$. Also, relatives of a cluster at levels $\ell$ lower than its own level (i.e., $\ell > j$) are called *descendants*, e.g., $R_3(S_1^{(2)}) = \{S_1^{(3)}, S_2^{(3)}\}$.

is sufficiently large in Eq. (4). Otherwise, the interactions at this level are classified as inadmissible and interactions between all the children of both the source and observer clusters are examined on admissibility recursively until the leaf level is reached. To demonstrate the described process, the construction of $\mathcal{H}$-matrix structure for the volume-to-surface operator $\boldsymbol{Z}_0^{\partial V,V}$ of SVS-EFIE is visualized in Fig. 3. Here, the spheroid model [17] is chosen as an example, and four-level surface and volume cluster trees for its test and basis functions are constructed. As demonstrated in Fig. 3, to construct an $\mathcal{H}$-matrix structure for $\boldsymbol{Z}_0^{\partial V,V}$, $T_S$ is chosen as the observer tree $T_{\text{obs}}$ and $T_V$ chosen as the source tree $T_{\text{src}}$. The process starts from the interaction between the whole set of surface testing functions $S_1^{(0)}$ and volume basis functions $V_1^{(0)}$ and continues recursively by applying admissibility criterion in Eq. (4).

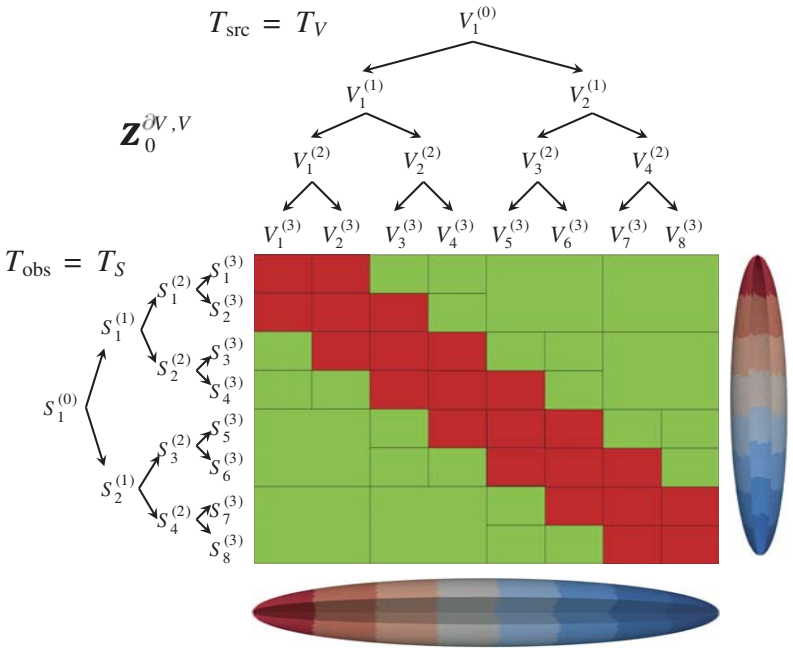


**Figure 3.** $\mathcal{H}$-matrix representation of the volume-to-surface MoM impedance matrix $\boldsymbol{Z}_0^{\partial V,V}$ arising from $T_{S\times V}$ interaction tree. The structure is constructed using an spheroid model with its surface tree $T_S$ as observer tree and volume tree $T_V$ as source tree with tree depth $L = 3$. Red blocks represent inadmissible blocks and green blocks represent admissible blocks.

To solve the SVS-EFIE, $\mathcal{H}$-matrix structures have to be constructed for each of the three operators (Fig. 1):

- $Z_\epsilon^{\partial V,\partial V}$: $T_S$ is chosen for both source $T_{\text{src}}$ and observer $T_{\text{obs}}$ trees. Therefore, as shown in Fig. 4, the corresponding surface-to-surface square $\mathcal{H}$-matrix arising from $T_{S\times S}$ interaction tree has the total size of $P \times P$.
- $\boldsymbol{Z}_0^{\partial V,V}$: For volume-to-surface interactions, $T_{\text{src}} = T_V$ and $T_{\text{obs}} = T_S$. The corresponding rectangular $\mathcal{H}$-matrix arising from $T_{S\times V}$ interaction tree is shown in Fig. 4 and has the total size of $P \times 3N$.
- $\boldsymbol{Z}_\epsilon^{V,\partial V}$: For surface-to-volume interactions, $T_{\text{src}} = T_S$ and $T_{\text{obs}} = T_V$. Therefore, as shown in Fig. 4, the volume-to-surface rectangular $\mathcal{H}$-matrix arising from $T_{V\times S}$ interaction tree has the total size of $3N \times P$.

### 3.3. Filling $\mathcal{H}$-Matrices of the SVS-EFIE Operators

Figure 4 depicts the constructed $\mathcal{H}$-matrices, where red blocks represent inadmissible blocks that will be stored without any approximation in a full-matrix ($\mathcal{F}$) format. For the green blocks corresponding

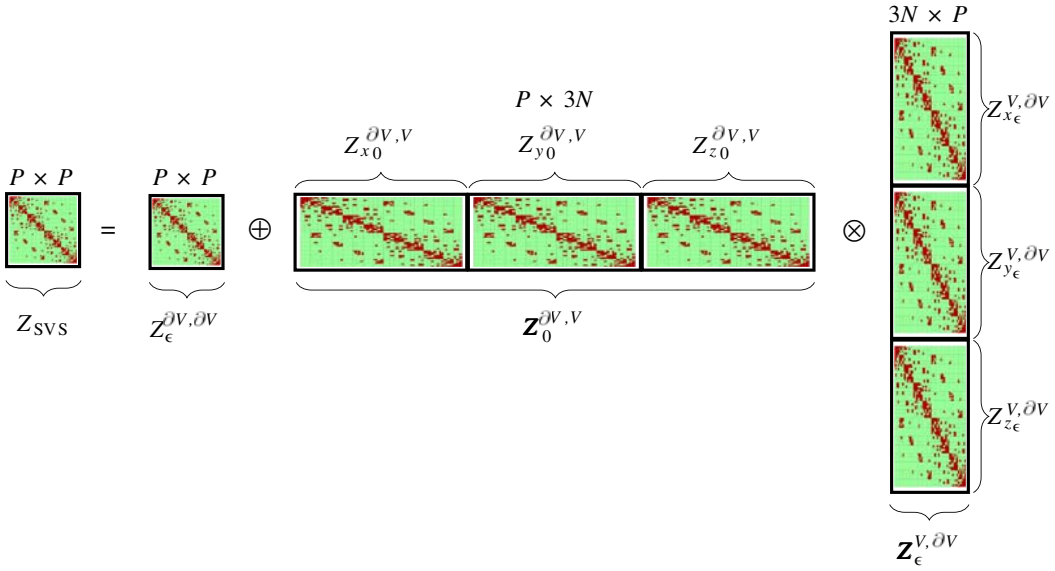


**Figure 4.** Assembly of $Z_{\text{SVS}}$ MoM $\mathcal{H}$-matrix from the individual $\mathcal{H}$-matrices of SVS-EFIE operators: $Z_\epsilon^{\partial V,\partial V}$ arising from $T_{S\times S}$ interaction tree, $\boldsymbol{Z}_0^{\partial V,V}$ arising from $T_{S\times V}$, and $\boldsymbol{Z}_\epsilon^{V,\partial V}$ arising from $T_{V\times S}$ via formatted multiplication and addition (8) required by a $\mathcal{H}$-LU based direct solution of the SVS-EFIE.

to admissible interactions, ACA algorithm [27] is used to obtain the blocks in a compressed low-rank ($\mathcal{R}$) format:

$$\underbrace{\mathcal{R}}_{m\times n, k(\tau_{\text{ACA}})} = \underbrace{A}_{m\times k(\tau_{\text{ACA}})} \times \underbrace{B^H}_{n\times k(\tau_{\text{ACA}})} \tag{5}$$

where $m$ and $n$ are the row and column sizes of the block, and $k(\tau_{\text{ACA}})$ is the revealed rank with the accuracy of the approximation controlled by the ACA truncation threshold $\tau_{\text{ACA}}$. The complexity of the ACA is linearly proportional to the size of the block and reveals rank $k(\tau_{\text{ACA}})$ [27].

## 4. $\mathcal{H}$-MATRIX FAST ITERATIVE AND DIRECT SOLUTIONS

After $\mathcal{H}$-matrix approximations are constructed for each discretized integral operator entering SVS-EFIE, the resultant system of linear algebraic equations

$$\left(Z_\epsilon^{\partial V,\partial V} \oplus \boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}\right) \otimes I = \mathcal{V}, \tag{6}$$

has to be solved, with $\otimes$ and $\oplus$ being the operations of formatted multiplication and addition described later in this section. In this paper, we consider both $\mathcal{H}$-GMRES iterative and $\mathcal{H}$-LU direct solution approaches. To solve the system using iterative $\mathcal{H}$-GMRES method, at each iteration the MVPs $Z_{\text{SVS}} \otimes I$ are performed via formatted multiplication $i = \boldsymbol{Z}_\epsilon^{V,\partial V} \otimes I$ followed by formatted multiplication $\boldsymbol{Z}_0^{\partial V,V} \otimes i$ and its addition to the result of the formatted multiplication $Z_\epsilon^{\partial V,\partial V} \otimes I$, as follows:

$$Z_{\text{SVS}} \otimes I = Z_\epsilon^{\partial V,\partial V} \otimes I + \boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V} \otimes I. \tag{7}$$

In contrast to the iterative method, the construction of the $Z_{\text{SVS}}$ combining all three approximated integral operators entering the SVS-EFIE is required to solve the system directly. After the final $Z_{\text{SVS}}$ MoM $\mathcal{H}$-matrix is assembled as (Fig. 4)

$$Z_{\text{SVS}} = Z_\epsilon^{\partial V,\partial V} \oplus \boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}, \tag{8}$$

$\mathcal{H}$-LU decomposition followed by $\mathcal{H}$-back-substitution is applied in order to solve the system of linear equations using the $\mathcal{H}$-matrix arithmetic [31]. To elaborate, let's consider how resultant $P \times P$ $\mathcal{H}$-matrix $Z_{\text{SVS}}$ is formed through combining the $\mathcal{H}$-matrices for the individual SVS-EFIE operators using approximate $\mathcal{H}$-arithmetic (Fig. 4).

The key steps of constructing $Z_{\mathrm{SVS}}$ are given in Algorithms 1 and 2. The procedure starts from a call to a recursive function *MulAdd* ($Z_\epsilon^{\partial V,\partial V}$, $\boldsymbol{Z}_0^{\partial V,V}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}$, $\partial V$, $\partial V$, $V$) (see line 1 in Algorithm 1) with SVS-EFIE $\mathcal{H}$-matrices, where $\partial V \in T_S$ and $V \in T_V$ are the root-level clusters in Fig. 2. This function performs recursive block matrix-matrix multiplication of $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$ and adds it to $Z_\epsilon^{\partial V,\partial V}$ in order to form the resultant $Z_{\mathrm{SVS}}$.

As long as all input matrices for this function are in $\mathcal{H}$-format, the function calls itself for the corresponding sub-blocks (line 8, Algorithm 1). However, if at a certain level a block $r_1 \times r_2$ of $Z_\epsilon^{\partial V,\partial V}$ ($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$) is in $\mathcal{R}$-format (i.e., it is a leaf), while the other function arguments $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ are still in $\mathcal{H}$-format, the function *MulAdd* (...) calls itself only with the required part ${Z'}_\epsilon^{\partial V,\partial V}|_{r_i\times r_j}$ (corresponding to $(r_i \times r_j)$ child of $(r_1 \times r_2)$) of $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$ in $\mathcal{R}$-format that is returned through *GetPart*$\mathcal{R}$ (...) function (line 10, Algorithm 1). Subsequently, the child $Z_\epsilon^{\partial V,\partial V}|_{r_i\times r_j}$ of $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$ will be overwritten by the matrix ${Z'}_\epsilon^{\partial V,\partial V}|_{r_i\times r_j}$ (line 12, Algorithm 1) resulting from the recursive call to multiplication and addition function *MulAdd* (...) (line 11, Algorithm 1).

If $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{F}$ (lines 17–28, Algorithm 1), the resultant matrix of $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ has to be stored in $\mathcal{F}$-format. Hence, a naïve multiplication and addition can be performed (line 28, Algorithm 1) after conversion of the arguments $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ to the $\mathcal{F}$-format (see lines 19, 21, 24, 26 in Algorithm 1). The recursive function $\mathcal{H}$-*Approx*$\mathcal{F}$ ($M \leftarrow M'''$) is depicted in Fig. 5(a) for an example of an $\mathcal{H}$-matrix with depth $L = 2$. Here, $\mathcal{H}$-matrix $M'''$ is converted to a block $M$ in $\mathcal{F}$-format by converting $\mathcal{R}$-blocks to $\mathcal{F}$ ($M'' \leftarrow M'''$) and coarsening the structure in the next two recursive steps ($M \leftarrow M' \leftarrow M''$).

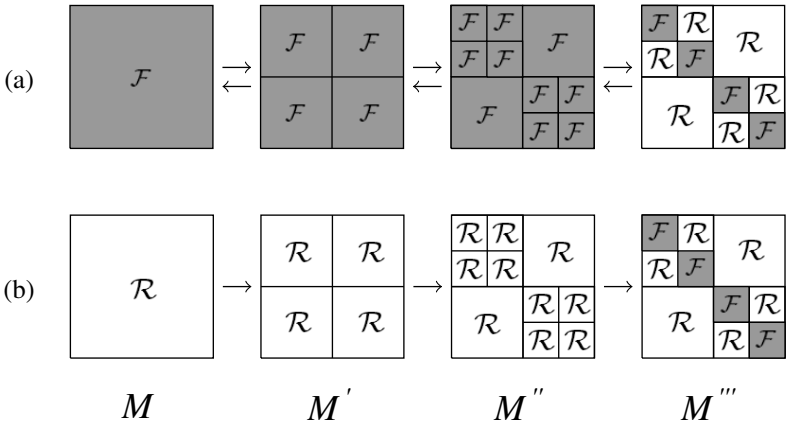


**Figure 5.** (a) Recursive function $\mathcal{H}$-*Approx*$\mathcal{F}$ ($M \rightarrow M'''$) converting an $\mathcal{F}$-block $M$ to an $\mathcal{H}$-matrix $M'''$. This procedure can also be implemented in the opposite order as $\mathcal{F}$-*Approx*$\mathcal{H}$ ($M''' \rightarrow M$). (b) Recursive function $\mathcal{H}$-*Approx*$\mathcal{R}$ ($M \rightarrow M'''$) converting an $\mathcal{R}$ block $M$ to an $\mathcal{H}$-matrix $M'''$. If $\mathcal{H}$-matrix $M'''$ has depth of $L$, then $L$-step recursion (e.g., $M \rightarrow M''$) and one final step (e.g., $M'' \rightarrow M'''$) are needed for both cases (a) and (b). Here, to simplify the depiction, the depth of $M'''$ is considered to be $L = 2$ which leads to 3 steps conversion.

If $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$ is in $\mathcal{H}$- or $\mathcal{R}$-format, and the leaf level is reached for any of $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ or $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ (lines 29–30, Algorithm 1), their product will be computed and stored as $\mathcal{R}$ or $\mathcal{F}$. Here (line 30, Algorithm 1), the function *FmtMulAdd* (see Algorithm 2) performing formatted multiplication (at leaves) and addition [31] is called:

- If any of $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ or $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{R}$ (lines 2–9, Algorithm 2), the result of the multiplication is stored as $\mathcal{R}$. Hence, if $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ is $\mathcal{H}$ (line 3, Algorithm 2), recursive function $\mathcal{H}mul\mathcal{R}$ (...) depicted in Fig. 6 is called to perform $\mathcal{H} \otimes \mathcal{R}$ and store the result as $\mathcal{R}$ (line 4, Algorithm 2). Otherwise (lines 5–6, Algorithm 2), recursive function $\mathcal{R}mul\mathcal{H}$ (...) is called to perform $\mathcal{R} \otimes \mathcal{H}$ that is similar to $\mathcal{H}$mul$\mathcal{R}$ (...) in Fig. 6. As it is shown in Fig. 6, these two functions include the

truncation step $Trun^{\mathcal{R}+\mathcal{R}}_{k''\leftarrow k+k'}(\dots)$ which is depicted in Fig. 7(a). This operation can be performed by truncating the singular values of the matrix block stored in an $\mathcal{R}$-format in Eq. (5) that can be done fast using reduced singular value decomposition (rSVD) [51, 7.1.1].

- If neither $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ nor $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ are in $\mathcal{R}$-format (i.e., one of them is in $\mathcal{F}$-format), the multiplication result matrix $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2}$ has to be stored in $\mathcal{F}$-format (lines 10–18, Algorithm 2).

**Algorithm 1** Multiplication and addition of $\mathcal{H}$-matrices

**Inputs:** matrices $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$, and $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$, clusters $r_1, r_2 \in \partial V$ and $s \in V$.
**Output:** $Z_{\text{SVS}} = Z_\epsilon^{\partial V,\partial V} \oplus (\boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V})$

1: *MulAdd*($Z_\epsilon^{\partial V,\partial V}|_{\partial V\times\partial V}$, $\boldsymbol{Z}_0^{\partial V,V}|_{\partial V\times V}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{V\times\partial V}$, $\partial V$, $\partial V$, $V$)
2: **function** *MulAdd*($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$, $r_1$, $r_2$, $s$)
3: **if** ($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \{\mathcal{H}$ or $\mathcal{R}\}$) **and** ($\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{H}$) **and** ($\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{H}$) **then**
4: **for all** $r_i$ children of $r_1$ **do**
5: **for all** $r_j$ children of $r_2$ **do**
6: **for all** $s_k$ children of $s$ **do**
7: **if** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{H}$ **then**
8: *MulAdd*($Z_\epsilon^{\partial V,\partial V}|_{r_i\times r_j}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_i\times s_k}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s_k\times r_j}$, $r_i$, $r_j$, $s_k$)
9: **else if** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{R}$ **then**
10: $Z_\epsilon'^{\,\partial V,\partial V}|_{r_i\times r_j}$ = *GetPartR*($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$)
{*GetPartR* returns a part of matrix $Z_\epsilon^{\partial V,\partial V}$ in $\mathcal{R}$-format corresponding to $r_i \times r_j$ child of $r_1 \times r_2$}
11: *MulAdd*($Z_\epsilon'^{\,\partial V,\partial V}|_{r_i\times r_j}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_i\times s_k}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s_k\times r_j}$, $r_i$, $r_j$, $s_k$)
12: *UpdatePartR*($Z_\epsilon^{\partial V,\partial V}|_{r_i\times r_j}$, $Z_\epsilon'^{\,\partial V,\partial V}|_{r_i\times r_j}$)
{*UpdatePartR* rewrites part of matrix $Z_\epsilon^{\partial V,\partial V}$ corresponding to $r_i \times r_j$ with $Z_\epsilon'^{\,\partial V,\partial V}$}
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: **else if** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{F}$ **then**
18: **if** $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{H}$ **then**
19: $\mathcal{H}$-*Approx*$\mathcal{F}$($\mathcal{F} \leftarrow \boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$) {Fig. 5(a)}
20: **else if** $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{R}$ **then**
21: convert $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ to $\mathcal{F}$
22: **end if**
23: **if** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{H}$ **then**
24: $\mathcal{H}$-*Approx*$\mathcal{F}$($\mathcal{F} \leftarrow \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$)
25: **else if** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{R}$ **then**
26: convert $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ to $\mathcal{F}$
27: **end if**
28: $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} += \boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ {naïve}
29: **else if** ($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \{\mathcal{H}$ or $\mathcal{R}\}$) **and** ($\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \notin \mathcal{H}$ **or** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \notin \mathcal{H}$) **then**
30: *FmtMulAdd*($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$, $r_1$, $r_2$, $s$)
31: **end if**
32: **return**

Hence, for any of $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ or $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ in $\mathcal{H}$-format, they will be converted to the $\mathcal{F}$ using $\mathcal{H}$-*Approx*$\mathcal{F}$ ($M \leftarrow M'''$) in Fig. 5(a) and then naïve multiplication is performed on the two $\mathcal{F}$-blocks.

**Algorithm 2** Formatted multiplication (at leaves) and addition

**Inputs:** matrices $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$, and $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$, clusters $r_1, r_2 \in T_S$ and $s \in T_V$.
**Output:** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} = Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \oplus (\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2})$
**Require:** $(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \{\mathcal{H} \text{ or } \mathcal{R}\})$ **and** $(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \notin \mathcal{H}$ **or** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \notin \mathcal{H})$

**function** *FmtMulAdd*($Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$, $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$, $r_1$, $r_2$, $s$)
FORMATTED MULTIPLICATION:
**if** $(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{R})$ **or** $(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{R})$ **then** {the result of multiplication $Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}$ is stored as $\mathcal{R}$}
  **if** $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{H}$ **then**
    $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} = \mathcal{H}mul\mathcal{R}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}, \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2})$ {Fig. 6}
  **else if** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{H}$ **then**
    $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} = \mathcal{R}mul\mathcal{H}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}, \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2})$ {Fig. 6}
  **else**
    $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} = \boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \cdot \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$
  **end if**
**else if** $(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{F})$ **or** $(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{F})$ **then** $\{Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}$ is stored as $\mathcal{F}\}$
  **if** $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \in \mathcal{H}$ **then**
    $\mathcal{H}$-*Approx*$\mathcal{F}(\mathcal{F} \leftarrow \boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s})$ {Fig. 5(a)}
  **end if**
  **if** $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2} \in \mathcal{H}$ **then**
    $\mathcal{H}$-*Approx*$\mathcal{F}(\mathcal{F} \leftarrow \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2})$ {Fig. 5(a)}
  **end if**
  $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} = \boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \cdot \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$
**end if**
FORMATTED ADDITION:
**if** $(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{H})$ **and** $(Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{F})$ **then**
  $\mathcal{H}$-*Approx*$\mathcal{F}(Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} \rightarrow Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2})$ {Fig. 5(a)}
**else if** $(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{H})$ **and** $(Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{R})$ **then**
  $\mathcal{H}$-*Approx*$\mathcal{R}(Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} \rightarrow Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2})$ {Fig. 5(b)}
**end if**
**if** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{H}$ **then**
  $Trun_{k''\leftarrow k+k'}^{\mathcal{H}+\mathcal{H}}(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}, Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2})$ {Fig. 7(b)}
**else** $\{Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{R}\}$
  **if** $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{F}$ **then**
    convert $Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2}$ to $\mathcal{R}$
  **end if**
  $Trun_{k''\leftarrow k+k'}^{\mathcal{R}+\mathcal{R}}(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}, Z_\epsilon'^{\,\partial V,\partial V}|_{r_1\times r_2})$ {Fig. 7(a)}
**end if**
**return** $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$

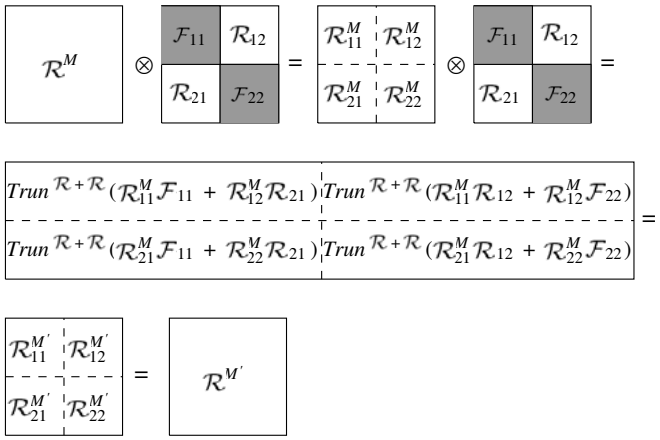


**Figure 6.** Recursive function $\mathcal{R}mul\mathcal{H}$ X$(\mathcal{R},\mathcal{H})$ performing formatted multiplication of $\mathcal{R}$- and $\mathcal{H}$-blocks and storing the result in $\mathcal{R}$-format. The recursive function $\mathcal{H}mul\mathcal{R}$ $(\mathcal{H},\mathcal{R})$ is defined similarly with an opposite order of operations. Here, to simplify the depiction, the depth of $\mathcal{H}$-matrix is considered to be $L = 1$. In general, $\mathcal{H}$-blocks of various depths occur.

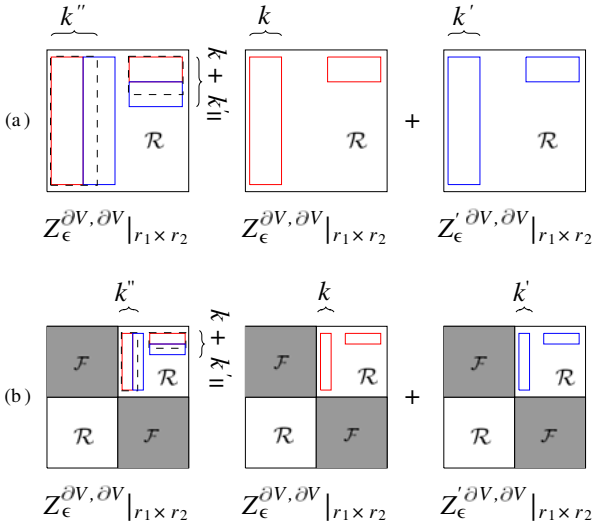


**Figure 7.** Procedures to re-compress the addition of two matrices. (a) Function $Trun^{\mathcal{R}+\mathcal{R}}_{k''\leftarrow k+k'}(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}, Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}, \tau_{\mathcal{H}})$ for addition and truncation of two matrices in $\mathcal{R}$-format. (b) Recursive function $Trun^{\mathcal{H}+\mathcal{H}}_{k''\leftarrow k+k'}(Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}, Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}, \tau_{\mathcal{H}})$ for addition and truncation of two $\mathcal{H}$-matrices. Again, to simplify the depiction, the depth of $\mathcal{H}$-matrices in (b) is considered to be $L = 1$. The resultant compressed blocks with revealed rank $k''$ are represented as dashed blocks.

As it is shown in Algorithm 2, after formatted multiplication the formatted addition is performed. Before adding the multiplication result matrix $Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}$ to the matrix of $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$, the format of $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2}$ is checked (lines 19–31, Algorithm 2):

- If $Z_\epsilon^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{H}$ the structure of $Z_\epsilon'^{\partial V,\partial V}|_{r_1\times r_2}$, which is in either $\mathcal{F}$ or $\mathcal{R}$ format, is converted to the hierarchical structure of $Z_\epsilon^{\partial V,\partial V}$ using $\mathcal{H}$-$Approx\mathcal{F}$ $(M \to M''')$ or $\mathcal{H}$-$Approx\mathcal{R}$ $(M \to M''')$ in Fig. 5, respectively (lines 20 and 22, Algorithm 2). After matching the structures, we add these two $\mathcal{H}$-matrices and then compress the new created $\mathcal{H}$-matrix using $Trun^{\mathcal{H}+\mathcal{H}}_{k''\leftarrow k+k'}(\dots)$ function (line 25, Algorithm 2). As shown in Fig. 7(b), this recursive function adds two $\mathcal{H}$-matrices and then re-

compresses $\mathcal{R}$-blocks from rank $k+k'$ to rank $k''$ using rSVD [51, 7.1.1] based on the predefined $\mathcal{H}$-arithmetic tolerance $\tau_{\mathcal{H}}$ in the new $\mathcal{H}$-matrix.

- If $Z_{\epsilon}^{\partial V,\partial V}|_{r_1\times r_2} \in \mathcal{R}$, the structure of $Z_{\epsilon}^{'\ \partial V,\partial V}|_{r_1\times r_2}$ is converted to $\mathcal{R}$ when it is in $\mathcal{F}$-format (line 28, Algorithm 2). Then, two $\mathcal{R}$-blocks of $Z_{\epsilon}^{\partial V,\partial V}|_{r_1\times r_2}$ and $Z_{\epsilon}^{'\ \partial V,\partial V}|_{r_1\times r_2}$ are added and compressed through $Trun_{k''\leftarrow k+k'}^{\mathcal{R}+\mathcal{R}}(\dots)$ function with $\mathcal{H}$-arithmetic tolerance $\tau_{\mathcal{H}}$ in Fig. 7(a) (line 30, Algorithm 2).

## 5. MEMORY AND COMPUTATIONAL COMPLEXITY ANALYSIS FOR $\mathcal{H}$-MATRIX SVS-EFIE

To derive the expressions for computational complexity and storage requirements, we define the sparsity constant [31] $C_{\text{sp}}^{S\times V}$ for a given interaction tree $T_{S\times V}$, as

$$C_{\text{sp}}^{S\times V} := \max\left\{ \overbrace{\max_{\substack{r_1\in T_S,\\ \ell=0,\dots,L}} \#\{s\in T_V : r_1\times s\in\mathcal{L}(T_{S\times V},\ell)\}}^{\text{(a)}}, \overbrace{\max_{\substack{s\in T_V,\\ \ell=0,\dots,L}} \#\{r_1\in T_S : r_1\times s\in\mathcal{L}(T_{S\times V},\ell)\}}^{\text{(b)}} \right\} \tag{9}$$

where the term (a) in Eq. (9) is the maximum number of interaction blocks $r_1\times s$ at the leaf level associated with an observer cluster $r_1\in T_S$, and the term (b) in Eq. (9) is the maximum number of interaction blocks $r_1\times s$ at the leaf level associated with a source cluster $s\in T_V$ among all levels $\ell=0,\dots,L$ of interaction tree $T_{S\times V}$. Here, $\mathcal{L}(T_{S\times V},\ell)$ is the set of leaves at the $\ell$th level, and $L$ is the number of levels of $T_{S\times V}$. The largest of these two counts is the sparsity constant $C_{\text{sp}}^{S\times V}$ of $T_{S\times V}$.

As an example, for the $\mathcal{H}$-matrix structure in Fig. 3 the number of interaction blocks $r_1\times s$ associated with the observer cluster $r_1=S_1^{(3)}$ at level 3 is:

$$\#\left\{s\in T_V : S_1^{(3)}\times s\in\mathcal{L}(T_{S\times V},3)\right\} = \#\left\{S_1^{(3)}\times\left\{V_1^{(3)},V_2^{(3)},V_3^{(3)},V_4^{(3)}\right\}\right\} = 4 \tag{10}$$

Also, at the same level $\ell=3$, the number of interaction blocks $r_1\times s$ associated with the observer cluster $r_1=S_3^{(3)}$ is:

$$\#\left\{s\in T_V : S_3^{(3)}\times s\in\mathcal{L}(T_{S\times V},3)\right\} = \#\left\{S_3^{(3)}\times\left\{V_1^{(3)},V_2^{(3)},V_3^{(3)},V_4^{(3)},V_5^{(3)},V_6^{(3)}\right\}\right\} = 6 \tag{11}$$

After computing this counts for all observer clusters $r_1\in T_S$ in each level, the maximum number of these counts among all four levels of this $\mathcal{H}$-matrix is 6 for term (a) in Eq. (9). Analogously, for term (b) in Eq. (9), this maximum number among all source clusters $s\in T_V$ in each level of the $\mathcal{H}$-matrix is 6. So, the sparsity constant $C_{\text{sp}}^{S\times V}$ for $\mathcal{H}$-matrix in Fig. 3 is maximum between the two counts (a) and (b) in Eq. (9), which is 6.

Since in the interaction tree $T_{V\times S}$ the domain and range are merely switched compared to the interaction tree $T_{S\times V}$, and its sparsity constant is:

$$C_{\text{sp}}^{V\times S} = C_{\text{sp}}^{S\times V} = C_{\text{sp}} \tag{12}$$

To simplify the notation, $\mathcal{M}(\cdot)$ will be used for asymptotic memory requirement in $\mathcal{H}$-matrix format, while $\mathcal{M}^{\mathcal{R}}(\cdot)$ and $\mathcal{M}^{\mathcal{F}}(\cdot)$ are for the memory required by all $\mathcal{R}$- and $\mathcal{F}$-blocks, respectively.

### 5.1. Memory Complexity for SVS-EFIE $\mathcal{H}$-Matrices

The memory usage to store volume-to-surface $\boldsymbol{Z}_0^{\partial V,V}$ and surface-to-volume $\boldsymbol{Z}_{\epsilon}^{V,\partial V}$ discretized operators can be derived analogously to the ones for the square $\mathcal{H}$-matrix [31, 32].

To compute the storage for the rectangular $\mathcal{H}$-matrix $\boldsymbol{Z}_0^{\partial V,V}$ of the size $P \times 3N$, the storage of $\mathcal{R}$- and $\mathcal{F}$-blocks is analyzed, separately. Since all $\mathcal{F}$-blocks exist only at the leaf level and the number of them scales linearly with the size of the matrix [31], the storage $\mathcal{M}^{\mathcal{F}}$ is estimated as $\mathcal{O}(n_{\min}^2(N+P))$, with $n_{\min} \times n_{\min}$ being the size of the full leaf blocks.

On the other hand, rank deficient $b$th $\mathcal{R}$-block of size $m(\ell) \times n(\ell)$ at the $\ell$th level is stored as $AB^H$ in Eq. (5), where matrix $A$ is of size $m(\ell) \times k_b^{\partial V,V}$, and matrix $B^H$ is of size $k_b^{\partial V,V} \times n(\ell)$. Therefore, its storage is $k_b^{\partial V,V}(m(\ell)+n(\ell))$. Since, there are at most $2^\ell C_{\text{sp}}$ interaction blocks at the $\ell$th level in the case of bisection based partitioning and considering sparsity constant definition in Eq. (9), the complexity to store all $\mathcal{R}$-blocks for $\boldsymbol{Z}_0^{\partial V,V}$ can be approximated as

$$\mathcal{M}^{\mathcal{R}}\left(\boldsymbol{Z}_0^{\partial V,V}\right) \leq \sum_{\ell=0}^{L_S} 2^\ell C_{\text{sp}} k_{\max}^{\partial V,V} m(\ell) + \sum_{\ell=0}^{L_V} 2^\ell C_{\text{sp}} k_{\max}^{\partial V,V} n(\ell), \tag{13}$$

where $L_S$ is the number of levels in $T_S$ and $L_V$ the number of levels in $T_V$ for the $\boldsymbol{Z}_0^{\partial V,V}$ interaction tree $T_{S\times V}$, $b = 1, \ldots, N^{\mathcal{R}}$, and $k_{\max}^{\partial V,V}$ is the maximum rank revealed (e.g., by ACA) among all $N^{\mathcal{R}}$ $\mathcal{R}$-blocks of $\boldsymbol{Z}_0^{\partial V,V}$. Here, we assume $k_{\max}^{\partial V,V}$ to only weakly increase across the levels, hence, implying quasi-dynamic simulation scenarios, in which the structure does not exceed several wavelengths in size. Since $m(\ell) = P/2^\ell$ and $n(\ell) = 3N/2^\ell$ are the sizes of the row and column of each $\mathcal{R}$-block at the $\ell$th level, respectively, and $L_S = \mathcal{O}(\log P)$ and $L_V = \mathcal{O}(\log N)$, Eq. (13) is simplified to

$$\mathcal{M}^{\mathcal{R}}\left(\boldsymbol{Z}_0^{\partial V,V}\right) \leq C_{\text{sp}} k_{\max}^{\partial V,V} \left( P \sum_{\ell=0}^{L_S} 1 + 3N \sum_{\ell=0}^{L_V} 1 \right) = \mathcal{O}(k_{\max}^{\partial V,V} P \log P) + \mathcal{O}(k_{\max}^{\partial V,V} N \log N). \tag{14}$$

Therefore, the storage $\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V})$ is dominated by the storage of its $\mathcal{R}$-blocks in Eq. (14)

$$\begin{aligned} \mathcal{M}\left(\boldsymbol{Z}_0^{\partial V,V}\right) = \mathcal{M}^{\mathcal{R}}\left(\boldsymbol{Z}_0^{\partial V,V}\right) + \mathcal{M}^{\mathcal{F}}\left(\boldsymbol{Z}_0^{\partial V,V}\right) &= \mathcal{O}(k_{\max}^{\partial V,V}(P\log P + N \log N)) + \mathcal{O}(n_{\min}^2(N+P)) \\ &= \mathcal{O}(k_{\max}^{\partial V,V} P \log P) + \mathcal{O}(k_{\max}^{\partial V,V} N \log N), \end{aligned} \tag{15}$$

where for simplicity $k_{\max}^{\partial V,V}$ is assumed to be of $\mathcal{O}(n_{\min})$. Note that since the $\mathcal{H}$-matrix structure of $\boldsymbol{Z}_\epsilon^{V,\partial V}$ is a transpose of $\boldsymbol{Z}_0^{\partial V,V}$, asymptotically its storage is the same

$$\mathcal{M}\left(\boldsymbol{Z}_\epsilon^{V,\partial V}\right) = \mathcal{O}(k_{\max}^{V,\partial V} P \log P) + \mathcal{O}(k_{\max}^{V,\partial V} N \log N). \tag{16}$$

For $P \times P$ matrix $Z_\epsilon^{\partial V,\partial V}$, the expression for the memory requirement can be derived by simplifying Eq. (15) for a square $\mathcal{H}$-matrix case:

$$\mathcal{M}(Z_\epsilon^{\partial V,\partial V}) = \mathcal{O}(k_{\max}^{\partial V,\partial V} P \log P). \tag{17}$$

where $k_{\max}^{\partial V,\partial V}$ is the maximum rank revealed (e.g., by ACA) among all $\mathcal{R}$-blocks of $\boldsymbol{Z}_\epsilon^{\partial V,\partial V}$.

### 5.2. Computational Complexity of $\mathcal{H}$-GMRES Based Iterative Solver

The number of operations $\mathcal{N}_{\text{MVP}}$ (complexity) of the MVP for an $\mathcal{H}$-matrix can be bounded by the memory required to store the $\mathcal{H}$-matrix itself [32, Lemma 2.5]

$$\mathcal{N}_{\text{MVP}}(Z) \leq \mathcal{M}(Z), \quad Z \in \{Z_\epsilon^{\partial V,\partial V}, \boldsymbol{Z}_\epsilon^{V,\partial V}, \boldsymbol{Z}_0^{\partial V,V}\} \tag{18}$$

Therefore, the computational complexity for $\mathcal{H}$-GMRES based solver for SVS-EFIE is composed of three MVPs with the corresponding discretized integral operators:

$$\begin{aligned} \mathcal{N}_{\mathcal{H}\text{-GMRES}} &= N_{\text{it}} \left( \mathcal{N}_{\text{MVP}}(\boldsymbol{Z}_\epsilon^{V,\partial V}) + \mathcal{N}_{\text{MVP}}(\boldsymbol{Z}_0^{\partial V,V}) + \mathcal{N}_{\text{MVP}}(Z_\epsilon^{\partial V,\partial V}) \right) \\ &= N_{\text{it}} \left( \mathcal{O}(k_{\max} P \log P) + \mathcal{O}(k_{\max} N \log N) \right), \end{aligned} \tag{19}$$

where $N_{\text{it}}$ is the number of iterations required for GMRES to converge to a prescribed tolerance.

### 5.3. Computational Complexity of $\mathcal{H}$-LU Based Direct Solver

As discussed in Section 4, $\mathcal{H}$-LU based direct solver has three steps, the setup of $\mathcal{H}$-matrix $Z_{\text{SVS}}$ involving formatted multiplication and addition of $\mathcal{H}$-matrices $Z_\epsilon^{\partial V,\partial V} \oplus \boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}$, $\mathcal{H}$-LU decomposition, and backsubstitution. Below, we estimate computational complexity of these steps.

#### *5.3.1. Formatted Multiplication* $\boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}$

As shown in Eq. (8), the complexity to setup $Z_{\text{SVS}}$ can be estimated through formatted multiplication and addition of $\mathcal{H}$-matrices. To simplify the analysis, at first the truncation functions $Trun_{k''\leftarrow k+k'}^{\mathcal{H}+\mathcal{H}}(\dots)$ and $Trun_{k''\leftarrow k+k'}^{\mathcal{R}+\mathcal{R}}(\dots)$ in Algorithm 2 will be considered as a simple copying of the data without any re-compression, so $k''=k+k'$ in Fig. 7. For this scenario, the formatted multiplication of two sub-blocks $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ in Algorithm 1, line 30 (see also lines 4 and 6 of Algorithm 2) becomes the exact multiplication $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ in $\mathcal{H}$-matrix format.

So, in Algorithm 2, each leaf block $r_1 \times r_2$ at the $j$th level resulting from the multiplication $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ is performed with

$$\left(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}\right)|_{r_1\times r_2} = \sum_{\ell=0}^{\iota}\left(\sum_{s\in\mathcal{U}(r_1\times r_2,\ell)} \hat{\boldsymbol{Z}}_0^{\partial V,V}|_{r_1\times s} \times \hat{\boldsymbol{Z}}_\epsilon^{V,\partial V}|_{s\times r_2}\right), \tag{20}$$

where $\iota \in [0, L]$ which $L$ is the depth of $\mathcal{H}$-matrices $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$. Also,

$$\begin{aligned}\mathcal{U}(r_1\times r_2,\ell) = \Big\{ & s\in T_V^{(\ell)} : \{R_\ell(r_1)\times s \in T_{S\times V} \text{ and } s\times R_\ell(r_2)\in\mathcal{L}(T_{V\times S})\} \\ & \text{or } \{R_\ell(r_1)\times s \in \mathcal{L}(T_{S\times V}) \text{and } s\times R_\ell(r_2)\in T_{V\times S}\}\Big\},\end{aligned} \tag{21}$$

and

$$V = \dot{\bigcup_{\ell=0,\dots,\iota}} \; \dot{\bigcup_{s\in\mathcal{U}(r_1\times r_2,\ell)}} s. \tag{22}$$

In Eq. (21), $R_\ell(r_1)$ is the relative of cluster $r_1$ (ancestor ($\ell < j$), self ($\ell = j$), or descendant ($\ell > j$)) at $\ell$th level, and $\mathcal{L}(T_{V\times S})$ is a leaf of the $T_{V\times S}$ interaction tree. Hence, in Eq. (20) sub-block $\hat{\boldsymbol{Z}}_0^{\partial V,V}|_{r_1\times s}$, or sub-block $\hat{\boldsymbol{Z}}_\epsilon^{V,\partial V}|_{s\times r_2}$, or both are in the leaf level of the corresponding interaction tree and are stored as $\mathcal{R}$- or $\mathcal{F}$-blocks.

Since $\mathcal{F}$-blocks have the size of at most $n_{\min}\times n_{\min}$, $\mathcal{H}$-matrix-matrix product with them requires $n_{\min}$ MVPs, while the exact $\mathcal{H}$-matrix-matrix product with an $\mathcal{R}$-block involves $k$ MVPs. Therefore, to compute $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ in Eq. (20) at any level $\ell$ will require at most $\max(k_{\max}, n_{\min})$ MVPs, where $k_{\max}$ is the maximum rank revealed for $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$. Without loss of generality if $k_{\max} \geq n_{\min}$, the complexity $\mathcal{N}$ to compute the product of sub-blocks $\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}$ can be estimated as

$$\begin{aligned}\mathcal{N}\left(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s\in\mathcal{L}(T_{S\times V},\ell)} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2\in(T_{V\times S},\ell)}\right) &\leq k_{\max}^{\partial V,V}\mathcal{N}_{\text{MVP}}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2\in(T_{V\times S},\ell)}) \\ &\overset{(18)}{\leq} k_{\max}^{\partial V,V}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2\in(T_{V\times S},\ell)}), \quad k_{\max}^{\partial V,V} \geq n_{\min}\end{aligned} \tag{23}$$

Similarly,

$$\mathcal{N}\left(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s\in(T_{S\times V},\ell)} \times \boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2\in\mathcal{L}(T_{V\times S},\ell)}\right) \leq k_{\max}^{V,\partial V}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s\in(T_{S\times V},\ell)}), \quad k_{\max}^{V,\partial V} \geq n_{\min} \tag{24}$$

The overall complexity to multiply $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$ can be calculated by analyzing the interaction tree $T_{S\times V} \times T_{V\times S} = T_{S\times S}$ resulting from the multiplication. By considering all resultant leaf blocks

$r_1 \times r_2 \in \mathcal{L}(T_{S\times S}, j)$ at the $j$th level of Eq. (20), $j = 1, \ldots, L_S$, the complexity estimate is

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \leq \sum_{j=0}^{L_S} \sum_{r_1\times r_2 \in \mathcal{L}(T_{S\times S},j)} \times \sum_{\ell=0}^{L} \sum_{s\in\mathcal{U}(r_1\times r_2,\ell)} \mathcal{N}\left(\hat{\boldsymbol{Z}}_0^{\partial V,V}|_{r_1\times s} \times \hat{\boldsymbol{Z}}_\epsilon^{V,\partial V}|_{s\times r_2}\right) \tag{25}$$

where $L$ is the depth of $\mathcal{H}$-matrices $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$. By substituting Eqs. (23) and (24) into Eq. (25), it can be written as:

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \leq \sum_{j=0}^{L_S} \sum_{r_1\times r_2 \in \mathcal{L}(T_{S\times S},j)} \sum_{\ell=0}^{L} \sum_{s\in\mathcal{U}(r_1\times r_2,\ell)} \left(k_{\max}^{V,\partial V}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}) + k_{\max}^{\partial V,V}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2})\right), \tag{26}$$

Considering the definition in Eq. (22) for the union of all the volume clusters $s$ contributing to formation of the destination block $r_1\times r_2$ the operation count $\sum_{\ell=0}^{L}\sum_{s\in\mathcal{U}(r_1\times r_2,\ell)}(k_{\max}^{V,\partial V}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times s}) + k_{\max}^{\partial V,V}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{s\times r_2}))$ in the summation over the volume clusters $s$ in Eq. (26) can be simplified by considering the root cluster $V$ containing all volume basis/testing functions, yielding

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \leq \sum_{j=0}^{L_S} \sum_{r_1\times r_2 \in \mathcal{L}(T_{S\times S},j)} \left(k_{\max}^{V,\partial V}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times V}) + k_{\max}^{\partial V,V}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{V\times r_2})\right) \tag{27}$$

On the other hand, for each level $j$, the total number of leaf blocks $r_1 \times r_2 \in \mathcal{L}(T_{S\times S}, j)$ in the resultant interaction tree $T_{S\times S}$ is bounded by $2^j C_{\text{sp}}^{S\times S}$ in case of bisection based partitioning and considering sparsity constant definition in Eq. (9). Here, $2^j$ is the total number of observer clusters $r_1$ or source clusters $r_2$ at the $j$th level of interaction tree $T_{S\times S}$. Therefore, in Eq. (27) the number of operations (estimated via memory use) in summations $\sum_{r_1\times r_2\in\mathcal{L}(T_{S\times S},j)}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{r_1\times V})$ and $\sum_{r_1\times r_2\in\mathcal{L}(T_{S\times S},j)}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{V\times r_2})$ required for computation of the resultant leaf clusters $r_1\times r_2$ at level $j$ can be replaced by the operation count $C_{\text{sp}}^{S\times S}(\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{\partial V\times V}) + \mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{\partial V\times V}))$ over the root cluster $\partial V$ containing all the surface RWG basis/testing functions. So, Eq. (27) can be rewritten as:

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \leq \sum_{j=0}^{L_S} C_{\text{sp}}^{S\times S}\left(k_{\max}^{V,\partial V}\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{\partial V\times V}) + k_{\max}^{\partial V,V}\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{V\times\partial V})\right) \tag{28}$$

Here, $\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V}|_{\partial V\times V})$ and $\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}|_{V\times\partial V})$ can simply be replaced by $\mathcal{M}(\boldsymbol{Z}_0^{\partial V,V})$ and $\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V})$ denoting the memory use for storage of the $\mathcal{H}$-matrices $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$, respectively. Then, by recalling that the depth of the surface tree is $L_S = \mathcal{O}(\log P)$ and introducing the maximum rank $k_{\max}$ for blocks of either $\boldsymbol{Z}_0^{\partial V,V}$ or $\boldsymbol{Z}_\epsilon^{V,\partial V}$ (i.e., $k_{\max} = \max(k_{\max}^{\partial V,V}, k_{\max}^{V,\partial V})$), Eq. (28) becomes

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \leq L_S C_{\text{sp}}^{S\times S} k_{\max}\left(\mathcal{M}(\boldsymbol{Z}_\epsilon^{V,\partial V}) + \mathcal{M}(\boldsymbol{Z}_0^{\partial V,V})\right). \tag{29}$$

Recalling the estimates for memory use Eqs. (15) and (16) we finally get the estimate for operation count in the *exact* multiplication of the $\mathcal{H}$-matrices

$$\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) = \mathcal{O}\left(k_{\max}^2 P\log^2 P\right) + \mathcal{O}\left(k_{\max}^2 N\log N\log P\right). \tag{30}$$

Note that the *formatted* multiplication $\boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V}$ requires the truncation based on the prescribed $\mathcal{H}$-arithmetic tolerance $\tau_{\mathcal{H}}$ of small singular values resulting from addition of the $\mathcal{R}$-blocks. For that, the rSVD with the complexity of $\mathcal{O}(n(\ell)+m(\ell))k^2 + O(k^3)$ [51, 7.1.1] is used throughout the Algorithm 2 leaving the complexity estimate of Eq. (30) correct for the formatted multiplication as well (i.e., $\mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \times \boldsymbol{Z}_\epsilon^{V,\partial V}) \simeq \mathcal{N}(\boldsymbol{Z}_0^{\partial V,V} \otimes \boldsymbol{Z}_\epsilon^{V,\partial V})$) provided $k_{\max}$ is relatively small compared to $P$ and $N$ and can be considered to be independent of them.

*5.3.2. Formatted Addition* $\boldsymbol{Z}_{\epsilon}^{\partial V,\partial V} \oplus (\boldsymbol{Z}_{0}^{\partial V,V} \otimes \boldsymbol{Z}_{\epsilon}^{V,\partial V})$

Next, the complexity of the formatted addition of two $\mathcal{H}$-matrices of the size $P \times P$ required to form $Z_{\mathrm{SVS}}$ by adding $Z_{\epsilon}^{\partial V,\partial V}$ and the result of $\boldsymbol{Z}_{0}^{\partial V,V} \otimes \boldsymbol{Z}_{\epsilon}^{V,\partial V}$ in Eq. (8) has to be considered. For formatted addition of the square $\mathcal{H}$-matrices, the estimate for number of operations is $O(k_{\max}^2 P \log P) + O(k_{\max}^3 P)$ [31]. The total complexity for setting up the $Z_{\mathrm{SVS}}$ is composed of the complexity of the formatted multiplication in Eq. (30) and the formatted addition as

$$\mathcal{N}(Z_{\mathrm{SVS}}) = \mathcal{O}\left(k_{\max}^2 P \log^2 P\right) + \mathcal{O}\left(k_{\max}^2 N \log N \log P\right) + O(k_{\max}^3 P) \tag{31}$$

One can see that for the problems of electrically moderate sizes with bounded rank, the complexity for setting up the $Z_{\mathrm{SVS}}$ is dominated by the complexity of the formatted multiplication in Eq. (30).

*5.3.3. $\mathcal{H}$-LU Decomposition and Back Substitution*

To solve the matrix equations, $\mathcal{H}$-LU decomposition and back substitution are applied to $Z_{\mathrm{SVS}}$. Since $Z_{\mathrm{SVS}}$ is a square $\mathcal{H}$-matrix of the size $P \times P$, the time complexity for the $\mathcal{H}$-LU decomposition and back substitution are $\mathcal{O}(k_{\max}^3 P \log^2 P)$ and $\mathcal{O}(k_{\max}^3 P \log P)$, respectively [31].

## 6. NUMERICAL RESULTS

To validate the accuracy of the $\mathcal{H}$-matrix SVS-EFIE solver, we performed comparison of its numerical solutions against the analytic Mie series [52] and the results obtained using FEKO commercial solver [53] for several scattering problems. The performance of the SVS-EFIE solver for the timing and memory requirements is shown to corroborate the asymptotic complexities derived in Section 5. In all the numerical results, we considered the tolerance for $\mathcal{H}$-matrix compression $\tau_{\mathcal{H}}$ to be the same as the truncation tolerance of the ACA algorithm $\tau_{\mathrm{ACA}}$. So, for brevity $\tau = \tau_{\mathrm{ACA}} = \tau_{\mathcal{H}}$ is used throughout Section 6. The optimal values of the admissibility criterion $\eta$ and leafsize $n_{\min}$ are usually problem dependent and chosen empirically. The $\mathcal{H}$-matrix parameters in the paper are the optimized values for $\eta = \{1, 2, 3, 4, 5\}$ and $n_{\min} = \{8, 16, 32, 64\}$ sets in all numerical examples. It was observed that all combinations of these parameters are working well for the proposed solver. However, to balance between memory use and CPU time we run the numerical example for different parameters and empirically find that $\eta = 4$ and $n_{\min} = 16$ provide better time and memory performance for certain numerical examples of this paper.

### 6.1. Dielectric Sphere

In the first example, we consider scattering on a dielectric sphere of radius $R = 0.1\,\mathrm{m}$ and relative dielectric permittivity $\epsilon = 1.5$ excited by a radial dipole with the dipole moment $I\ell = 1[\mathrm{A} \cdot \mathrm{m}]$.

To demonstrate the behavior of the error in $\mathcal{H}$-matrix accelerated MoM solution of SVS-EFIE and its reduction with the decrease of ACA truncation threshold $\tau$, we consider dielectric sphere excitation by a $z$-directed electric dipole at 3 GHz situated above the north pole of the sphere 0.4 m away from the origin. The behavior of the average relative error ($\delta$) of both $\mathcal{H}$-matrix accelerated MoM solution and direct MoM solution of the SVS-EFIE [40] with respect to the Mie series solution is depicted as a function of $\tau$ in Fig. 8 for three different mesh densities, $h = \lambda/10$, $\lambda/20$, and $\lambda/30$, $h$ being the characteristic size of mesh elements. The results demonstrate that $\mathcal{H}$-matrix accelerated MoM solution with $\tau \leq 10^{-3}$ does not produce any substantial increase in accuracy compared to the direct MoM solution. On the other hand, the compression ratio (CR) of the overall MoM memory use in SVS-EFIE is shown for different $\tau$ in Table 1, where

$$\mathrm{CR} = \left(1 - \frac{\text{Mem. } \mathcal{H}\text{-matrix SVS-EFIE}}{\text{Mem. direct MoM SVS-EFIE [40]}}\right) \times 100\%. \tag{32}$$

From Fig. 8 and Table 1 it can be seen that the proposed solver has a good compression for $\tau = 10^{-3}$ with an acceptable error level. So, the ACA truncation threshold is set to $\tau = 10^{-3}$ for the other examples of this paper.

**Table 1.** Memory compression ratio (32) for the $\mathcal{H}$-matrix accelerated MoM solution of the SVS-EFIE for the dielectric sphere excitation problem with different mesh densities ($h = \lambda/10,\ \lambda/20$, and $\lambda/30$).

| $\tau$ | $CR\%$ ($h = \lambda/10$) | $CR\%$ ($h = \lambda/20$) | $CR\%$ ($h = \lambda/30$) |
|---|---|---|---|
| $10^{-1}$ | 62 | 87 | 93 |
| $10^{-2}$ | 48 | 82 | 90 |
| $10^{-3}$ | 31 | 75 | 86 |
| $10^{-4}$ | 7 | 55 | 80 |
| $10^{-5}$ | 0 | 20 | 63 |

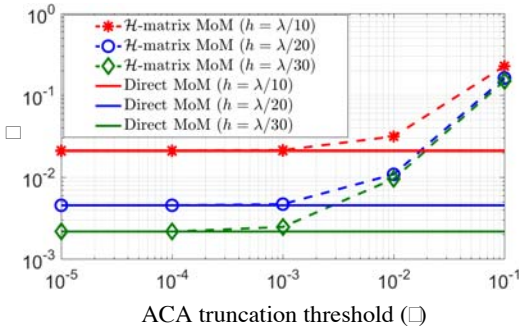


**Figure 8.** Average relative error $\delta$ of both $\mathcal{H}$-matrix accelerated MoM solution for dielectric sphere excitation problem with different mesh densities ($h = \lambda/10,\ \lambda/20$, and $\lambda/30$) with respect to the Mie series solution as a function of ACA truncation tolerance $\tau$. The error in the direct MoM solution of SVS-EFIE [40] is shown for reference.

To analyze the $\mathcal{H}$-matrix acceleration performance for MoM solution with large number of degrees of freedom (DoF), we examine the memory usage by the $\mathcal{H}$-matrices and CPU time used for pertinent $\mathcal{H}$-matrix operations in comparison with those in the direct MoM solution [40]. For this study, the dielectric sphere with mesh density $h = \lambda/10$ over the frequency range from 4 GHz to 10 GHz is considered.

In Fig. 9(a), the memory usage for $Z_\epsilon^{\partial V,\partial V}$ is demonstrated as a function of number of RWG basis functions $P$. Its $\mathcal{H}$-matrix MoM scaling behaves as $\mathcal{O}(P \log P)$ confirming the theoretical estimates in Eq. (17). In addition, the memory requirement of $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$ is also depicted in Fig. 9(b) with respect to the total number of surface and volume DoFs $X = 3N + P$. It can be seen that the numerically observed memory scaling of $\mathcal{O}(N \log N) + \mathcal{O}(P \log P)$ for $\mathcal{H}$-matrix MoM confirms the theoretical estimates in Eqs. (15) and (16).

Next, in Figs. 9(c) and (d), the time required to fill the compressed $\mathcal{H}$-matrices is plotted. Fig. 9(c) shows the time to fill $Z_\epsilon^{\partial V,\partial V}$ that behaves as $\mathcal{O}(P \log P)$ for $\mathcal{H}$-matrix solver. Also, the CPU time to fill $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$ is plotted as a function of total number of surface and volume DoFs $X = 3N + P$ in Fig. 9(d). One can observe fill time complexities of $\mathcal{O}(N \log N) + \mathcal{O}(P \log P)$ which is of the same order as the memory use complexity in Eqs. (15) and (16). In addition, the set up time for creating $Z_{\text{SVS}}$ as a part of the $\mathcal{H}$-LU direct method is depicted in Fig. 9(e) with the complexity of $\mathcal{O}(N \log N \log P) + \mathcal{O}(P \log^2 P)$ which confirms the theoretical analysis in Eq. (31). The time complexities for $\mathcal{H}$-LU decomposition and back-substitution are also plotted in Fig. 9(f) with scaling of $O(P \log^2 P)$ and $O(P \log P)$, respectively [31].

For the same scattering problem, the computational time using $\mathcal{H}$-GMRES iterative method is plotted in Fig. 9(g). In the first study, the number of iterations is fixed to $N_{\text{it}} = 1000$ and the time is plotted in blue triangles as a function of number of DoFs leading to $N_{\text{it}}(\mathcal{O}(N \log N) + \mathcal{O}(P \log P))$ scaling. In the second study, to show the convergence behavior of $\mathcal{H}$-GMRES iterative SVS-EFIE, the

iterative solver tolerance is set to $10^{-6}$, and the solution time is depicted in red crosses in the same figure.

In this study, the dielectric sphere having $\epsilon = 1.5$ has a relatively small electrical radius ranging from $R = 1.6\lambda$ to $R = 4\lambda$ in the considered frequency range from 4 GHz to 10 GHz. Therefore, the maximum rank will be approximately constant for all $\mathcal{H}$-matrices [50] in the entire frequency range. It can be seen from Figs. 9(a)–(f) that the memory requirement and CPU time scale according to the constant rank assumption.

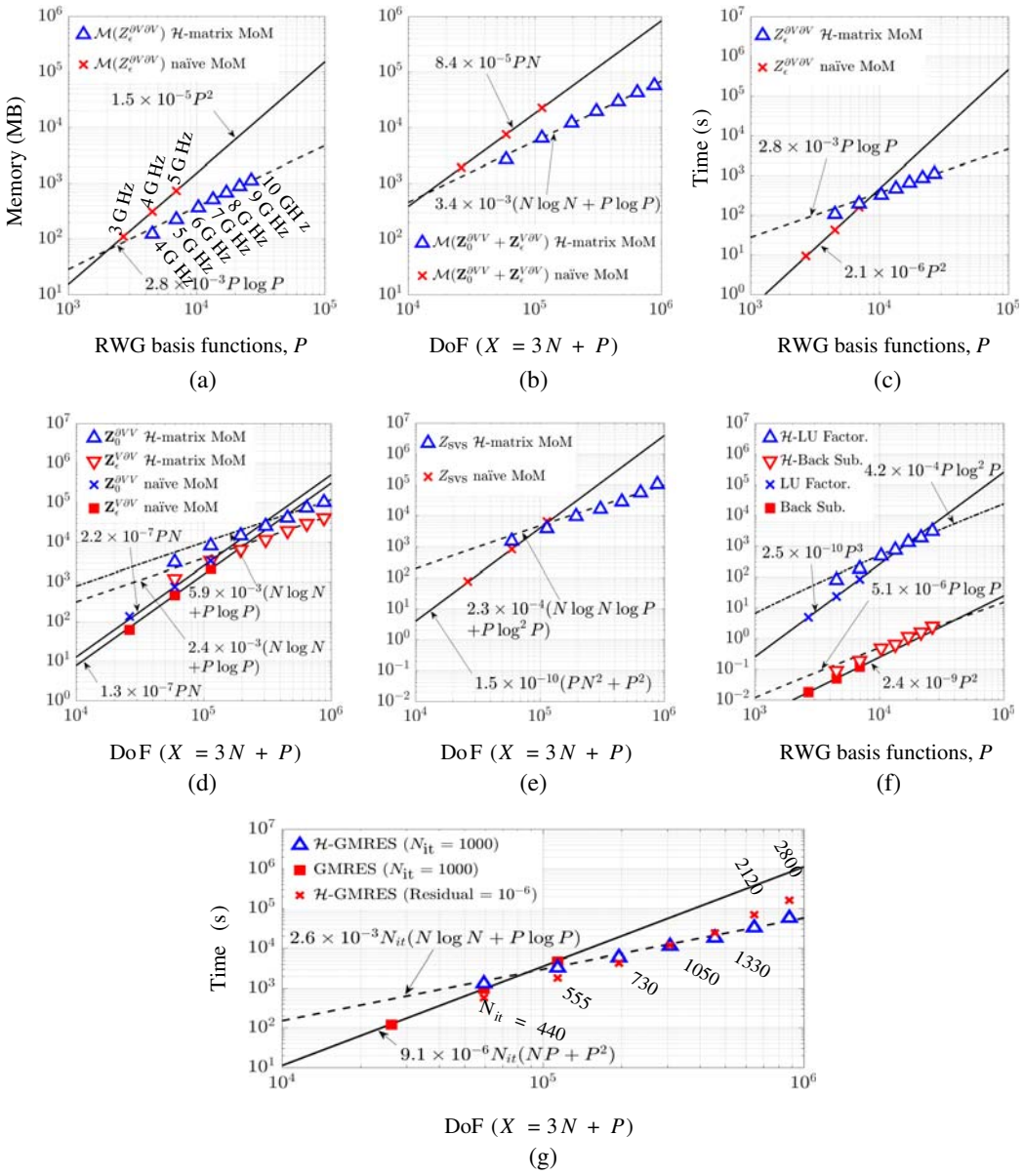


**Figure 9.** Scaling behavior of both $\mathcal{H}$-matrix (dashed lines) and direct (solid lines) MoM SVS-EFIE for the dielectric sphere with mesh density $h = \lambda/10$ across the frequency range from 4 GHz to 10 GHz. (a) Memory for $Z_\epsilon^{\partial V,\partial V}$. (b) Memory for $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$. (c) Set up time for $Z_\epsilon^{\partial V,\partial V}$. (d) Set up time for $\boldsymbol{Z}_0^{\partial V,V}$ and $\boldsymbol{Z}_\epsilon^{V,\partial V}$. (e) Set up time for $Z_{\mathrm{SVS}}$ as a part of $\mathcal{H}$-LU-based direct solver. (f) Solution time for $\mathcal{H}$-LU-based direct method ($\mathcal{H}$-LU factorization and back substitution). (g) Solution time for $\mathcal{H}$-GMRES-based iterative method (fixed $N_{\mathrm{it}} = 1000$ and fixed residual $10^{-6}$ cases).

### 6.2. Dielectric NASA Almond

In order to show the behaviour of SVS-EFIE solution for a non-smooth geometry featuring a sharp corner, we consider an accelerated MoM solution for the analysis of a dipole radiation near NASA almond [54] target at 3 GHz. The target has a relative permittivity $\epsilon = 12$ and a size of 0.252 m, 0.0976 m, and 0.0325 m along $x$, $y$, and $z$ axii, respectively. The model is excited by the electric dipole directed along the short axis of the almond ($z$ axis in Fig. 10) and situated on the long axis ($x$ axis in Fig. 10) 0.8528 m away from almond's tip. The electric dipole moment is $I\ell = 1[\mathrm{A}\cdot\mathrm{m}]$. In this example, the number of triangles in the MoM discretization on the boundary $\partial V$ is $M = 11,994$, the number of edges is $P = 17,991$, and the number of tetrahedrons in the volume $V$ is $N = 191,440$ which leads to $592,311$ number of volumetric and surface DoFs. Distribution of the magnitude of the total electric field using accelerated SVS-EFIE is depicted in Fig. 10. In order to validate the $\mathcal{H}$-matrix MoM solution of SVS-EFIE, MoM solution via surface equivalence principle (SEP) for the same radiation problem is obtained using FEKO commercial software [53]. The magnitudes of the total electric fields along the $x$ and $y$ axes of the almond are depicted in the same Fig. 10 for both the $\mathcal{H}$-matrix based MoM solution of SVS-EFIE and MoM solution from FEKO with an observed good agreement between the two.

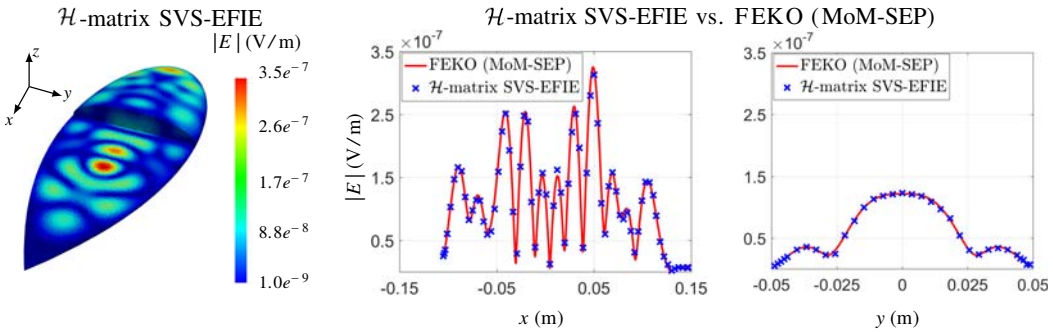


**Figure 10.** Magnitude of the total electric field at 3 GHz inside the dielectric NASA almond model with the relative dielectric permittivity $\epsilon = 12$. A 3-D representation of the total electric field obtained from $\mathcal{H}$-matrix SVS-EFIE is in the left plot. The distribution of the magnitude of the total electric field along the $x$ and $y$ axes for the $\mathcal{H}$-matrix SVS-EFIE and FEKO solutions are shown in the right plots.

### 6.3. Austin Benchmark Suite for Computational Bioelectromagnetics

Another example considers the scattering problems on a human head sized sphere and a human body sized spheroid from Austin Benchmark Suite [17]. Both models are filled with a tissue-equivalent homogeneous material [55] with frequency dependent relative dielectric permittivity and conductivity as listed in Table 2. The sphere has a radius of $R = 0.108$ m, and the spheroid model has minor and major axes of 0.344 m, and 1.76 m, respectively, with its major axis aligned with the $z$-axis. The scattering problem is solved at three frequencies in the UHF band: 402 MHz, 900 MHz, and 2.45 GHz.

**Table 2.** Values of tissue-equivalent material relative dielectric permittivity and conductivity [55].

| 402 MHz | | 900 MHz | | 2.45 GHz | |
|---|---|---|---|---|---|
| $\epsilon_r$ | $\sigma$ | $\epsilon_r$ | $\sigma$ | $\epsilon_r$ | $\sigma$ |
| 44.7 | 0.87 | 41.5 | 0.97 | 39.2 | 1.80 |

#### *6.3.1. Homogeneous Human Head-Sized Sphere*

The model is excited by the same dipole used for dielectric sphere example in Section 6.1 that is situated above the north pole of the sphere 0.4 m away from the origin. Distribution of the computed field is

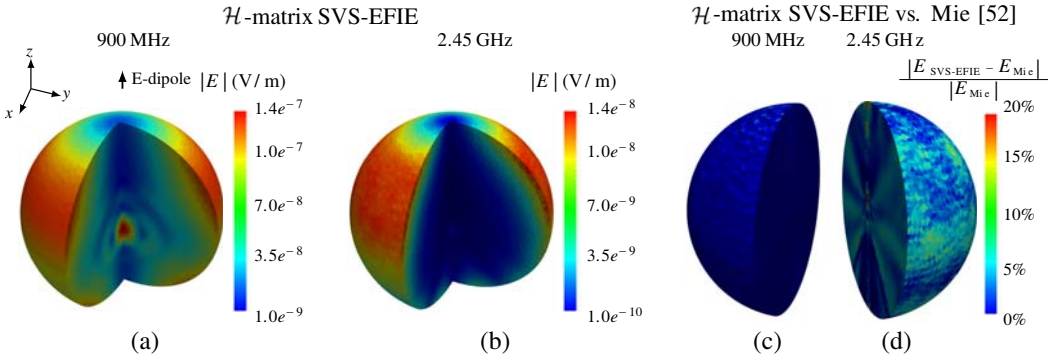


**Figure 11.** Magnitude of the total electric field inside the human head-sized sphere produced by a $z$-directed electric dipole situated at $x' = 0\,\text{m}$, $y' = 0\,\text{m}$, $z' = 0.4\,\text{m}$ obtained by the $\mathcal{H}$-matrix solver at (a) 900 MHz (b) 2.45 GHz. The relative error distribution with respect to the Mie series solution is shown in (c) and (d).

depicted in Fig. 11 at 900 MHz and 2.45 GHz. In this example, the number of triangles in the MoM discretization on the boundary $\partial V$ is $M = 17,262$, the number of edges is $P = 25,893$, and the number of tetrahedrons in the volume $V$ is $N = 263,309$ that leads to total $815,820$ number of DoFs. The numerical solutions of the radiation problem obtained using the proposed method at 900 MHz and 2.45 GHz are shown in Figs. 11(a) and (b), respectively. The relative error between the $\mathcal{H}$-matrix SVS-EFIE solution and the Mie series solution is depicted in Figs. 11 (c) and (d) for 900 MHz and 2.45 GHz, respectively. The average relative error in the solution at 900 MHz is below 0.006 with a standard deviation of 0.002. Also, the solution of the same problem at 2.45 GHz has an average error of 0.049 with a standard deviation of 0.018.

### *6.3.2. Homogeneous Human Body-Sized Spheroid*

The spheroid model is excited by a uniform $z$-polarized plane-wave traveling in the $+x$-direction. In this example, the number of triangles in the MoM discretization on the boundary $\partial V$ is $M = 15,804$; the number of edges is $P = 23,706$; and the number of tetrahedrons in the volume $V$ is $N = 206,431$ which leads to $642,999$ number of DoFs. Distribution of the magnitude of the total electric field using $\mathcal{H}$-matrix accelerated MoM solution of SVS-EFIE is depicted in Fig. 12 for 402 MHz and 900 MHz. The magnitude of the total electric field along the $x$ and $z$ axes are also depicted in the same Fig. 12 for the case of $\mathcal{H}$-matrix based MoM solution of SVS-EFIE and FEKO's MoM solution with an observed good agreement between the two.

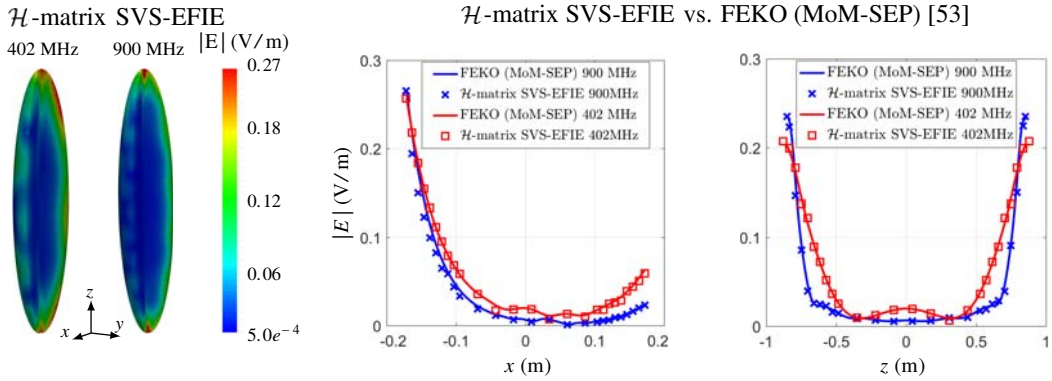


**Figure 12.** Magnitude of the total electric field inside the human body-sized spheroid produced by a uniform $\hat{z}$-polarized plane-wave traveling in the $+x$ direction at 402 MHz and 900 MHz obtained by the $\mathcal{H}$-matrix SVS-EFIE. The distribution of the magnitude of total electric field along the $x$ and $z$ axes for $\mathcal{H}$-matrix SVS-EFIE and FEKO solutions are shown in the right.

## 7. CONCLUSION

A new $\mathcal{H}$-matrix based direct and iterative algorithms are developed for the fast Method of Moments (MoM) solution of the Surface-Volume-Surface Electric Field Integral Equation (SVS-EFIE). The new computational framework allows for the iterative solution of 3-D scattering and radiation problems on homogeneous non-magnetic dielectrics with $O(N \log N)+O(P \log P)$ CPU time and memory complexity and their direct solution with $O(N \log N \log P) + O(P \log^2 P)$ CPU time and $O(N \log N) + O(P \log P)$ memory complexity for problems of moderate electrical size, $N$ being the number of tetrahedrons in the volume and $P$ being the number of surface unknowns produced by the MoM discretization. The $\mathcal{H}$-matrix arithmetics used to form and operate with the rectangular matrices produced by the MoM discretization of SVS-EFIE is described in details, as well as the associated computational and memory complexity estimates. The proposed accelerated method is shown to provide an efficient compression of the MoM impedance matrix which leads to significant reduction of the memory usage and the CPU time. Solution of the 3-D scattering problems of moderate electrical sizes which feature large number of associated degrees of freedom is demonstrated for both the dielectrics with low loss levels and the biological tissues exhibiting substantial loss. The method is particularly effective for the solution of scattering and radiation problems in bioelectromagnetics which require computation of the fields throughout the volume of the objects.

## ACKNOWLEDGMENT

The authors would like to thank Dr. S. Börm and Dr. L. Grasedyck for open source library $\mathcal{H}$-Lib, a modified version of which is used in this work.

## REFERENCES

1. Wu, T., T. S. Rappaport, and C. M. Collins, "The human body and millimeter-wave wireless communication systems: interactions and implications," *IEEE Int. Conf. Commun. (ICC)*, 2423–2429, Jun. 2015.
2. Fichte, L. O., "Interaction of biological tissue with electromagnetic waves in the RF range," *Asia-Pacific Conf. Env. Electromag. (CEEM)*, 10, Nov. 2015.
3. Agarwal, K. and Y. X. Guo, "Interaction of electromagnetic waves with humans in wearable and biomedical implant antennas," *Asia-Pacific Symp. on Electromag. Compat. (APEMC)*, 154–157, May 2015.
4. Ostadrahimi, M., P. Mojabi, S. Noghanian, L. Shafai, S. Pistorius, and J. LoVetri, "A novel microwave tomography system based on the scattering probe technique," *IEEE Trans. Instrum. Meas.*, Vol. 61, No. 2, 379–390, Feb. 2012.
5. Ferreira, D., P. Pires, R. Rodrigues, and R. F. S. Caldeirinha, "Wearable textile antennas: examining the effect of bending on their performance," *IEEE Antennas Propag. Mag.*, Vol. 59, No. 3, 54–59, Jun. 2017.
6. De Santis, V., M. Feliziani, and F. Maradei, "Safety assessment of UWB radio systems for body area network by the FD$^2$TD method," *IEEE Trans. Mag.*, Vol. 46, No. 8, 3245–3248, Aug. 2010.
7. Aguirre, E., J. Arpn, L. Azpilicueta, S. de Migue, V. Ramos, and F. Falcone, "Evaluation of electromagnetic dosimetry of wireless systems in complex indoor scenarios with human body interaction," *Prog. Electromagn. Res. B*, Vol. 43, 189–209, Sep. 2012.
8. Ojaroudiparchin, N., M. Shen, S. Zhang, and G. F. Pedersen, "A switchable 3-D-coverage-phased array antenna package for 5G mobile terminals," *IEEE Antennas Wireless Propag. Lett.*, Vol. 15, 1747–1750, Feb. 2016.
9. Thotahewa, K. M. S., J. M. Redoute, and M. R. Yuce, "SAR SA and temperature variation in the human head caused by IR-UWB implants operating at 4 GHz," *IEEE Trans. Microw. Theory Techn.*, Vol. 61, No. 5, 2161–2169, May 2013.
10. Mittra, M., *Computational Electromagnetics: Recent Advances and Engineering Applications*, 3rd Edition, Springer, New York, 2014.

11. Taflove, A. and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd Edition, Artech House, Boston, MA, 2005.
12. Jin, J. M., *The Finite Element Method in Electromagnetics*, 3rd Edition, Wiley-IEEE Press, 2015.
13. "Ansoft HFSS 11.1 user's guide," ANSYS Inc., Pittsburgh, PA, USA, 2009.
14. "XFDTD 7.3 user's manual," Remcom Inc., State College, PA, USA, 2012.
15. Chew, W. C., E. Michielssen, J. M. Song, and J. M. Jin, *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House, Norwood, MA, 2001.
16. Harrington, R. F., *Field Computation by Moment Methods*, Wiley-IEEE Press, 1993.
17. Massey, J. W., C. Liu, A. Menshov, and A. E. Yilmaz, "Bioelectromagnetic benchmarks," 2016, [Online]. Available: http://bit.ly/BioEM-Benchmarks.
18. Saad, Y. and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, 856–869, Jul. 1986.
19. Van der Vorst, H. A., "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, Vol. 13, No. 2, 631–644, 1992.
20. Coifman, R., V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, Vol. 35, No. 3, 7–12, Jun. 1993.
21. Ergül, O. and L. Gürel, *The Multilevel Fast Multipole Algorithm (MLFMA) for Solving Large-Scale Computational Electromagnetics Problems*, Wiley-IEEE Press, 2014.
22. Gumerov, N. A. and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*, Elsevier, Amsterdam, The Netherlands, 2006.
23. Aronsson, J. and V. Okhmatovski, "Vectorial low-frequency MLFMA for the combined field integral equation," *IEEE Antennas Wireless Propag. Lett.*, Vol. 10, 532–535, 2011.
24. Catedra, M. F., R. F. Torres, J. Basterrechea, and E. Gago, *The CG-FFT Method: Application of Signal Processing Techniques to Electromagnetics*, Artech House, Boston, MA, 1995.
25. Bleszynski, E., M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, Vol. 31, 1255–1251, Sep.–Oct. 1996.
26. Yang, K. and A. Yilmaz, "A three-dimensional adaptive integral method for scattering from structures embedded in layered media," *IEEE Trans. Geo. Remote Sens.*, Vol. 50, No. 4, 1130–1139, Apr. 2012.
27. Bebendorf, M. and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *Computing*, Vol. 70, No. 1, 1–24, Mar. 2003.
28. Zhao, K., M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Trans. Electromagn. Compat.*, Vol. 47, No. 4, 763–773, Nov. 2005.
29. Mitra, R. and C. Li, "The art and science of matrix preconditioning-a review," *IEEE Int. Conf. Comput. EM (ICCEM)*, 17–19, Feb. 2016.
30. Vico, F., L. Greengard, and M. Ferrando, "Decoupled field integral equations for electromagnetic scattering from homogeneous penetrable obstacles," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.06741.
31. Hackbusch, W., "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I. Introduction to $\mathcal{H}$-matrices," *Computing*, Vol. 62, No. 2, 89–108, 1999.
32. Grasedyck, L. and W. Hackbusch, "Construction and arithmetics of $\mathcal{H}$-matrices," *Computing*, Vol. 70, No. 4, 295–334, Aug. 2003.
33. Martinsson, P. G. and V. Rokhlin, "A fast direct solver for boundary integral equations in two dimensions," *J. Comput. Phyis.*, Vol. 205, No. 1, 1–23, May 2005.
34. Guo, H., J. Hu, and E. Michielssen, "On MLMDA/Butterfly compressibility of inverse integral operators," *IEEE Antennas Wireless Propag. Lett.*, Vol. 12, 31–34, 2013.

35. Shaeffer, J., "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns," *IEEE Trans. Antennas Propag.*, Vol. 56, No. 8, 2306–2313, Aug. 2008.
36. Brick, Y., V. Lomakin, and A. Boag, "Fast direct solver for essentially convex scatterers using multilevel non-uniform grids," *IEEE Trans. Antennas Propag.*, Vol. 62, No. 8, 4314–4324, Aug. 2014.
37. Corona, E., A. Rahimian, and D. Zorin, "A Tensor-Train accelerated solver for integral equations in complex geometries, *J. Comput. Phyis.*, Vol. 334, 145–169, Apr. 2017.
38. Oseledets, I. V., "Tensor-Train decomposition," *SIAM J. Sci. Comput.*, Vol. 33, No. 5, 2295–2317, 2011.
39. Menshov, A. and V. Okhmatovski, "New single-source surface integral equations for scattering on penetrable cylinders and current flow modeling in 2-D conductors," *IEEE Trans. Microw. Theory Techn.*, Vol. 61, No. 1, 341–350, Jan. 2013.
40. Hosseini, F. L. S., A. Menshov, R. Gholami, J. Mojolagbe, and V. Okhmatovski, "Novel single-source integral equation for scattering problems by 3D dielectric objects," *IEEE Trans. Antennas Propag.*, Vol. 66, No. 2, 797–807, Feb. 2018.
41. Zheng, S., R. Gholami, and V. Okhmatovski, "Surface-volume-surface electric field integral equation for solution of scattering problems on 3-D dielectric objects in multilayered media," *IEEE Trans. Microw. Theory Tech.*, 1–16, 2018.
42. Chen, Z., R. Gholami, J. Mojolagbe, and V. Okhmatovski, "Formulation of Surface-Volume-Surface-EFIE for solution of 3D scattering problems on composite dielectric objects," *IEEE Antennas Wireless Propag. Lett.*, Vol. 17, No. 6, 1043–1047, Jun. 2018.
43. Mojolagbe, J., R. Gholami, and V. Okhmatovski, "On complexity reduction in solution of scattering problems on well-conducting 3D objects with Surface-Volume-Surface EFIE," *Appl. Comput. Electromag. Conf. (ACES)*, 1–2, May 2018.
44. Swatek, D., "Investigation of single source surface integral equation for electromagnetic wave scattering by dielectric bodies," Ph.D. dissertation, Univ. Manitoba, Winnipeg, Canada, 1999.
45. Qian, Z. G., W. C. Chew, and R. Suaya, "Generalized impedance boundary condition for conductor modeling in surface integral equation," *IEEE Trans. Microw. Theory Techn.*, Vol. 55, No. 11, 2354–2364, Nov. 2007.
46. Müller, C., *Foundations of the Mathematical Theory of Electromagnetic Waves*, Springer-Verlag, Berlin, Heidelberg, New York, 1969.
47. Kishk, A. and L. Shafai, "Different formulations for numerical solution of single or multibodies of revolution with mixed boundary conditions," *IEEE Trans. Antennas Propag.*, Vol. 34, No. 5, 666–673, May 1986.
48. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, No. 3, 409–418, May 1982.
49. Martinsson, P.-G., V. Rokhlin, and M. Tygert, "A randomized algorithm for the decomposition of matrices," *Appl. Comput. Harmon. Anal.*, Vol. 30, No. 1, 47–68, Jan. 2011.
50. Brick, Y. and A. Yilmaz, "Fast multilevel computation of low-rank representation of $\mathcal{H}$-matrix blocks," *IEEE Trans. Antennas Propag.*, Vol. 64, No. 12, 5326–5334, Dec. 2016.
51. Börm, S., L. Grasedyck, and W. Hackbusch, *Hierarchical Matrices*, Technical Report 21, Max Planck Institute for Mathematics in the Sciences, 2006.
52. Chew, W. C., *Waves and Fields in Inhomogeneous Media*, Wiley-IEEE Press, 1999.
53. "FEKO user's manual," EM Software & Syst. Inc., Stellenbosch 7600, South Africa, 2014.
54. Woo, A. C., H. T. G. Wang, and M. J. Schuh, "Benchmark radar targets for the validation of computational electromagnetics programs," *IEEE Antennas Propag. Mag.*, Vol. 35, No. 1, 84–89, Feb. 1993.
55. *IEEE Recommended Practice for Determining the Peak Spatial-Average Specific Absorption Rate (SAR) in the Human Head from Wireless Communications Devices: Measurement Techniques*, IEEE Std 1528-2013 (Rev. IEEE Std 1528-2003), Sep. 2013.