

A Butterfly-Accelerated Volume Integral Equation Solver for Broad Permittivity and Large-Scale Electromagnetic Analysis

Sadeed B. Sayed, *Member, IEEE*, Yang Liu, *Member, IEEE*, Luis J. Gomez, *Member, IEEE*, and Abdulkadir C. Yucel, *Senior Member, IEEE*

Abstract—A butterfly-accelerated volume integral equation (VIE) solver is proposed for fast and accurate electromagnetic (EM) analysis of scattering from heterogeneous objects. The proposed solver leverages the hierarchical off-diagonal butterfly (HOD-BF) scheme to construct the system matrix and obtain its approximate inverse, used as a preconditioner. Complexity analysis and numerical experiments validate the $O(N \log^2 N)$ construction cost of the HOD-BF-compressed system matrix and $O(N^{1.5} \log N)$ inversion cost for the preconditioner, where N is the number of unknowns in the high-frequency EM scattering problem. For many practical scenarios, the proposed VIE solver requires less memory and computational time to construct the system matrix and obtain its approximate inverse compared to a \mathcal{H} matrix-accelerated VIE solver. The accuracy and efficiency of the proposed solver have been demonstrated via its application to the EM analysis of large-scale canonical and real-world structures comprising of broad permittivity values and involving millions of unknowns.

Index Terms—Butterfly algorithm, direct solver, fast solver, preconditioners, volume integral equation.

I. INTRODUCTION

THE analysis of electromagnetic (EM) scattering from heterogeneous scatterers is traditionally performed via volume integral equation (VIE) solvers [1]–[3]. These solvers discretize the VIE and obtain a dense system of linear equations with N unknowns. Oftentimes, these solvers' applicability to the large-scale problems is limited due to $O(N^2)$ or $O(N^3)$ memory and CPU time requirements for the iterative or direct solution of the dense VIE system, respectively. On the other hand, differential equation solvers require the solution of a sparse system of linear equations and consequently demand less computational resources compared to VIE solvers when applied to the large-scale problems. That said, unlike VIE solvers, differential equation solvers do not implicitly satisfy the radiation conditions and can suffer from the numerical dispersion.

Manuscript received March 3, 2021. This work was supported by Nanyang Technological University under a Start-Up Grant. (*Corresponding author: Abdulkadir C. Yucel.*)

S. B. Sayed and A. C. Yucel are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. (e-mails: sadeed.sayed@ntu.edu.sg, acyucel@ntu.edu.sg).

Y. Liu is with Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA. (e-mail: liuyangzhan@lbl.gov).

L. J. Gomez is with the division of Electrical and Computer Engineering, Purdue University, USA. (e-mail: ljgomez@purdue.edu).

So far, a plethora of methods has been developed to lower the memory and CPU time requirements of VIE solvers [4]–[12]. These methods were primarily developed to expedite the matrix-vector multiplications during the iterative solution of the VIE system. They leverage analytic or algebraic compression techniques, including the fast multipole method (FMM) [4], [5], fast Fourier transform [6]–[8], [13], low-rank compression [9], hierarchical matrices as kernel-free FMM [10], [11], and, more recently, tensor decompositions [12]. Despite the success of these methods in expediting matrix-vector multiplications during iterative solution, the discretized VIE systems are often ill-conditioned. This results in a prohibitively large iteration count during the iterative solutions of the VIE systems. In particular, the systems are ill-conditioned in scenarios where the scatterers have large permittivity [14]–[18], negative permittivity [19]–[22], or large electrical size. This results in several challenges for using VIE in the EM analysis of biomedical procedures applied to high-permittivity tissues [23], EM analysis of the structures covered by negative permittivity plasma [24], and EM analysis of electrically-large photonic devices [25]. For these scenarios, fast matrix-vector multiplication schemes have to be used in conjunction with effective preconditioners. Alternatively, fast direct methods can be utilized.

Existing fast direct solvers for the VIE include Huygens' equivalence principle-based algorithms [26], [27] yielding an $O(N^2)$ complexity, and low-rank-based algorithms such as \mathcal{H} matrices [28], [29], \mathcal{H}^2 matrices [30]–[32], and skeletonization [20], [33], [34]. These low-rank-based algorithms can attain $O(N)$ or $O(N \log N)$ complexities for static or low-frequency scattering problems, but tend to become less efficient as the electrical size of the scatterer increases. Low-rank-based direct solvers have also been developed for surface integral equations (SIE) [10], [11], [35], but low-rank algorithms tend to perform better for VIE solvers. This is due to the fact that the density of discretization elements per wavelength is typically much larger for the VIE compared to that for the SIE.

In contrast to the low-rank algorithms, here we consider another class of algebraic compression techniques called butterfly algorithms [36]–[40] for fast iterative and/or direct solution of the VIE. For a $m \times n$ matrix, its butterfly representation can be treated as an extension of the fast Fourier transform into an $O(\log n)$ -factor multiplicative matrix decomposition that is well-suited to compress highly-oscillatory operators [41]–[43]. Each butterfly factor contains $O(n)$ number of small

dense blocks whose sizes are oftentimes referred to as butterfly ranks. When combined with hierarchical matrix techniques, they have been leveraged for the direct or iterative solutions of 2-dimensional [37], [44]–[46], 3-dimensional [47]–[49] SIE, and differential equations [50] for electrically-large scattering problems. In general, the butterfly-accelerated SIE solvers can be classified into the strong-admissibility condition-based ones (such as hierarchical LU [47]) and weak-admissibility conditioned-based ones (such as hierarchically off-diagonal butterfly (HOD-BF) algorithm [45]). The former compresses only matrix blocks for the well-separated interactions, while the latter compresses the matrix blocks for both nearby and well-separated interactions [51]. They can be treated as the butterfly extensions of the strong-admissibility \mathcal{H} [28] and weak-admissibility hierarchically off-diagonal low-rank (HOD-LR) [52]–[54] matrices, respectively.

This paper proposes a butterfly-accelerated VIE solver for the EM scattering analyses of large-scale scatterers comprising broad permittivity values. The proposed solver leverages the HOD-BF scheme to compress the blocks in the VIE system and obtain the approximate inverse of the system matrix for forming a preconditioner. During the iterative solution of the VIE system, the matrix-vector multiplications are accelerated using the HOD-BF-compressed blocks while the iteration count is dramatically dropped to single digits with the preconditioner, even for highly ill-conditioned problems. Compared to the strong-admissibility solvers [47], [48], the proposed solver has simpler butterfly arithmetic, smaller leading constants in complexity, and significantly better parallelization performance. The HOD-BF scheme leveraged in the proposed solver was first introduced as a sequential algorithm to solve 2-dimensional SIE with $O(N \log^2 N)$ memory and $O(N^{1.5} \log N)$ CPU time requirements [45]; the same complexity can also be attained for 3-dimensional SIE despite of the non-uniform rank patterns due to weak-admissibility [55]. To the best of our knowledge, neither the HOD-BF scheme nor other butterfly schemes has been applied to the 3-dimensional VIE before, and consequently the complexity analysis of HOD-BF scheme has never been performed for the VIE. Unlike the HOD-LR scheme that scales poorly for high-dimensional problems, e.g., $O(N^{5/3})$ memory and $O(N^{7/3})$ CPU for 3D VIEs, we show that HOD-BF scheme's complexity is dimension independent. Furthermore, the advantages of the HOD-BF scheme compared to the low-rank-based \mathcal{H} matrix scheme have never been shown. All the numerical experiments are carried out using a distributed-memory parallel implementation of HOD-BF, briefly explained in this study. Without such implementation, the demonstration of the advantages of HOD-BF compared to the HOD-LR and \mathcal{H} matrix schemes would not be possible. These are the novelties introduced in this paper.

Here, we theoretically and numerically analyze the rank scaling and complexity of the proposed HOD-BF-accelerated VIE solver. Furthermore, we extensively compare the performance of the proposed VIE solver with that of a low-rank-based \mathcal{H} matrix-accelerated VIE solver. The numerical analysis shows that the proposed solver achieves $O(N \log^2 N)$ memory and CPU time scaling for constructing the HOD-BF-

compressed system matrix as well as $O(N \log^2 N)$ memory and $O(N^{1.5} \log N)$ CPU time scaling for inverting the system matrix to obtain the preconditioner. Compared to the \mathcal{H} matrix-accelerated VIE solver, the proposed solver requires 1.4/3.5 times less CPU time and 3.8/3.9 times less memory for the construction/inversion when used for the EM analysis of a dielectric sphere involving 0.9 M unknowns. Similar comparison performed on a NASA almond involving 1.6 M unknowns shows that the proposed simulator requires 5.0/3.2 times less CPU time and 3.8/2.7 times less memory for the construction/inversion compared to the \mathcal{H} matrix-accelerated VIE solver. Moreover, the proposed solver with the preconditioner requires maximum five iterations during the iterative solution of an highly ill-conditioned VIE system generated for the EM scattering analysis of negative permittivity sphere involving 1.7 M unknowns.

The rest of the paper is organized as follows: Section II describes the VIE formulation. Section III presents the construction of the HOD-BF-compressed system matrix via butterfly algorithm, inversion of the compressed system matrix to obtain preconditioner, complexity estimates, and parallelization strategy. Section IV provides numerical results to demonstrate the scalability, accuracy, and efficiency of the proposed butterfly-accelerated VIE solver. The conclusions are drawn in Section V. Throughout this paper, a time-harmonic variation of field quantities is assumed and suppressed.

II. VIE FORMULATION

Consider a time-harmonic EM field with incident electric and magnetic fields, $\mathbf{E}^{\text{inc}}(\mathbf{r})$ and $\mathbf{H}^{\text{inc}}(\mathbf{r})$, at frequency f . The field impinges on a heterogeneous scatterer V residing in a background medium with permittivity ε_0 , permeability μ_0 , and intrinsic impedance $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$. Here the background medium is assumed to be free-space. Using the volume equivalence principle, the VIE for the electric flux density $\mathbf{D}(\mathbf{r})$ is obtained as [1]

$$\mathbf{D}(\mathbf{r})/\varepsilon(\mathbf{r}) - j\omega\eta_0\mathcal{L}[\kappa\mathbf{D}](\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}); \mathbf{r} \in V, \quad (1)$$

where

$$\mathcal{L}[\mathbf{X}](\mathbf{r}) = -jk_0 \int_V dv' \left[\mathbf{I} + \frac{1}{k_0^2} \nabla \nabla \cdot \right] \mathbf{X}(\mathbf{r}) \mathbf{g}(\mathbf{r}, \mathbf{r}'). \quad (2)$$

Here $\omega = 2\pi f$ is the angular frequency, \mathbf{r} and \mathbf{r}' are the observation and source points, $k_0 = \omega\sqrt{\mu_0\varepsilon_0} = 2\pi/\lambda_0$ is the wave number, λ_0 is the wavelength, $\varepsilon(\mathbf{r})$ is the dielectric permittivity, $\kappa(\mathbf{r}) = (\varepsilon(\mathbf{r}) - \varepsilon_0)/\varepsilon(\mathbf{r})$ is the dielectric contrast, and $\mathbf{g}(\mathbf{r}, \mathbf{r}') = e^{-jk_0|\mathbf{r}-\mathbf{r}'|}/4\pi|\mathbf{r}-\mathbf{r}'|$ is the Green's function.

To solve the VIE, the scatterer is discretized by a volumetric tetrahedral mesh having N faces. Then, the electric flux in V is approximated via Schaubert Wilton Glisson (SWG) basis functions $\mathbf{f}_n(\mathbf{r})$ as [1]

$$\mathbf{D}(\mathbf{r}) = \sum_{n=1}^N D_n \mathbf{f}_n(\mathbf{r}); \mathbf{r} \in V. \quad (3)$$

Note that one SWG is defined for each of the N faces of the tetrahedral mesh, and each basis function $\mathbf{f}_n(\mathbf{r})$ has support on V_n , the volume of the tetrahedron which the face N belongs to.

Substituting (3) into (1) and applying Galerkin testing yields the discretized VIE linear system of equations as

$$\mathcal{Z}\mathcal{I} = \mathcal{V}, \quad (4)$$

where

$$\begin{aligned} \mathcal{Z}(m, n) &= \left\langle \mathbf{f}_m(\mathbf{r}), \frac{\mathbf{f}_n(\mathbf{r})}{\varepsilon(\mathbf{r})} - j\omega\eta\mathcal{L}(\kappa(\mathbf{r})\mathbf{f}_n(\mathbf{r})) \right\rangle_m \\ \mathcal{I}(n) &= D_n \\ \mathcal{V}(m) &= \left\langle \mathbf{f}_m(\mathbf{r}), \mathbf{E}^{\text{inc}}(\mathbf{r}) \right\rangle_m; \quad n, m = 1, \dots, N. \end{aligned} \quad (5)$$

Here, \mathcal{Z} is the system matrix, \mathcal{I} is the unknown coefficient vector, and \mathcal{V} is the excitation vector. Furthermore, $\kappa(\mathbf{r}) = (\varepsilon(\mathbf{r}) - \varepsilon_0)/\varepsilon(\mathbf{r})$ and $\langle \mathbf{a}, \mathbf{b} \rangle_m = \int_{V_m} \mathbf{a} \cdot \mathbf{b} d\mathbf{r}$ denotes the standard inner product of \mathbf{a} with \mathbf{b} in V_m .

The linear system (4) is solved with preconditioned iterative solvers as

$$\mathcal{Z}_{\chi_{\text{fact}}}^{-1} \mathcal{Z}_{\chi_{\text{con}}} \mathcal{I} = \mathcal{Z}_{\chi_{\text{fact}}}^{-1} \mathcal{V}. \quad (6)$$

Here $\mathcal{Z}_{\chi_{\text{con}}}$ is a compressed representation of \mathcal{Z} with tolerance χ_{con} using e.g., \mathcal{H} [28], HOD-LR [52], or the proposed HOD-BF [45] techniques. $\mathcal{Z}_{\chi_{\text{fact}}}^{-1}$ is the compressed representation of \mathcal{Z}^{-1} with tolerance $\chi_{\text{fact}} \geq \chi_{\text{con}}$. We will drop the subscripts of $\mathcal{Z}_{\chi_{\text{con}}}$ and $\mathcal{Z}_{\chi_{\text{fact}}}^{-1}$ in the rest of this paper. The preconditioned linear system (6) can be solved via an iterative solver (such as the transpose-free quasi-minimal residual method (TFQMR) [56]) by setting a stopping criteria χ_{sol} .

III. HOD-BF ACCELERATION AND PRECONDITIONER

The HOD-BF scheme first generates a hierarchical partitioning of the system matrix and then obtains the compressed off-diagonal blocks representing non-overlapping interactions as the butterfly format. The compressed blocks permit rapid matrix-vector multiplications and inversion, particularly for high-frequency regimes and 3-dimensional scatterers. In what follows, we briefly summarize the construction of the system matrix via HOD-BF-compressed blocks and inversion of the compressed system matrix to obtain preconditioner.

A. Construction of HOD-BF-compressed system matrix

The construction begins by recursively subdividing the scatterer into two sub-scatterers with approximately equal numbers of unknowns using a tree clustering algorithm (such as KD) [44], until the subscatterer contains a prescribed number of unknowns. This procedure results in a complete binary tree \mathfrak{T}_H of L_H levels with root level 0 and leaf level L_H . Each node τ at level l is an index set $\tau \subset \{1, \dots, N\}$ associated with the corresponding subscatterer. For a non-leaf node τ at level l with children τ_1 and τ_2 , $\tau = \tau_1 \cup \tau_2$ and $\tau_1 \cap \tau_2 = \emptyset$. For a non-root node τ , its parent is denoted p_τ . Let $\mathcal{D}_\tau = \mathcal{Z}(\tau, \tau)$. For the root τ , $\mathcal{D}_\tau = \mathcal{Z}$; for each leaf node τ , \mathcal{D}_τ is directly computed as dense blocks; off-diagonal blocks are compressed using the butterfly representation described below. As an example, Fig. 1 shows a HOD-BF-compressed matrix with $L_H = 3$ levels.

Specifically, let τ_1 and τ_2 be two siblings in \mathfrak{T}_H on level l . These two sibling nodes correspond to two off-diagonal

blocks $\mathcal{B}_{\tau_1} = \mathcal{Z}(\tau_1, \tau_2)$ and $\mathcal{B}_{\tau_2} = \mathcal{Z}(\tau_2, \tau_1)$. As an example consider the butterfly compression of the $m \times n$ block $\mathcal{B} = \mathcal{B}_{\tau_1}$ with $o = \tau_1$ and $s = \tau_2$, then $\mathcal{B} = \mathcal{Z}(o, s)$ is compressed as a butterfly with $L = L_H - l$ levels. Let \mathfrak{T}_o and \mathfrak{T}_s denote subtrees of \mathfrak{T}_H with L levels, rooted at nodes o and s respectively. The butterfly compression requires the *complementary low-rank property*: for any level $0 \leq l \leq L$, any node τ at level l of \mathfrak{T}_o and any node ν at level $L-l$ of \mathfrak{T}_s , the subblock $\mathcal{Z}(\tau, \nu)$ is numerically low-rank with rank $r_{\tau, \nu}$ bounded by a small number r called the butterfly rank. As will be discussed in Section III-C, we do not necessarily require constant butterfly ranks to attain a quasi-linear representation. Given the complementary low-rank property, we can compress any subblock $\mathcal{Z}(\tau, \nu)$ above as a low-rank product in an interpolative form:

$$\mathcal{Z}(\tau, \nu) \approx \mathcal{Z}(\tau, \bar{\nu}) \mathcal{V}_{\tau, \nu}. \quad (7)$$

where $\bar{\nu}$ represents the skeleton columns, $\mathcal{Z}(\tau, \bar{\nu})$ is the skeleton matrix, and $\mathcal{V}_{\tau, \nu}$ is the interpolation matrix. The approximation in Eq. (7) is typically computed with a compression tolerance χ_{con} such that $|\mathcal{Z}(\tau, \nu) - \mathcal{Z}(\tau, \bar{\nu}) \mathcal{V}_{\tau, \nu}|_F = O(\chi_{\text{con}}) |\mathcal{Z}(\tau, \nu)|_F$. This low-rank product can be computed from the interpolative decomposition (ID) [57]. Moreover, the interpolation matrices $\mathcal{V}_{\tau, \nu}$ for non-leaf nodes ν can be defined in a nested fashion as

$$\mathcal{V}_{\tau, \nu} = \mathcal{W}_{\tau, \nu} \begin{bmatrix} \mathcal{V}_{p_\tau, \nu_1} & \\ & \mathcal{V}_{p_\tau, \nu_2} \end{bmatrix}. \quad (8)$$

where $\mathcal{W}_{\tau, \nu}$ are referred to as the transfer matrices. $\mathcal{W}_{\tau, \nu}$ is computed from ID: $\mathcal{Z}(\tau, \bar{\nu}) \mathcal{W}_{\tau, \nu} \approx [\mathcal{Z}(\tau, \bar{\nu}_1), \mathcal{Z}(\tau, \bar{\nu}_2)]$

Once all the interpolation and transfer matrices are computed, the butterfly factorization of \mathcal{B} can be written as

$$\mathcal{Z}(o, s) \approx \mathcal{B}^L \mathcal{W}^L \mathcal{W}^{L-1} \dots \mathcal{W}^1 \mathcal{V}^0. \quad (9)$$

Here, the outer factor $\mathcal{V}^0 = \text{diag}(\mathcal{V}_{o, \nu_1}, \dots, \mathcal{V}_{o, \nu_{2L}})$ consists interpolation matrices at the leafs of \mathfrak{T}_s , and the block-diagonal inner factors $\mathcal{W}^l, l = 1, \dots, L$ have blocks \mathcal{W}_τ for all nodes τ at level $l-1$ of \mathfrak{T}_o

$$\mathcal{W}_\tau = \begin{bmatrix} \text{diag}(\mathcal{W}_{\tau_1, \nu_1}, \dots, \mathcal{W}_{\tau_1, \nu_{2L-l}}) \\ \text{diag}(\mathcal{W}_{\tau_2, \nu_1}, \dots, \mathcal{W}_{\tau_2, \nu_{2L-l}}) \end{bmatrix} \quad (10)$$

Here, $\nu_1, \nu_2, \dots, \nu_{2L-l}$ are the nodes at level $L-l$ of \mathfrak{T}_s and τ_1, τ_2 are children of τ . Lastly, the block-diagonal outer factor $\mathcal{B}^L = \text{diag}(\mathcal{Z}(\tau_1, \bar{s}), \dots, \mathcal{Z}(\tau_{2L}, \bar{s}))$ for all leafs $\tau_1, \tau_2, \dots, \tau_{2L}$ of \mathfrak{T}_o . Note that there exist several equivalent forms to Eq. (9) [38]–[40], [55]. Upon factorizing subblocks in \mathcal{Z} in the form of Eq. (9), we obtain a compressed system matrix $\mathcal{Z}_{\chi_{\text{con}}}$. The storage and application costs of a matrix-vector product using $\mathcal{Z}_{\chi_{\text{con}}}$ scale as $O(n \log n)$. As a result, the overall HOD-BF construction requires $O(N \log^2 N)$ memory and CPU resources (see Section III-C for more details). The HOD-BF-compressed system matrix can then be used for fast matrix-vector multiplication during the iterative solution of the VIE.

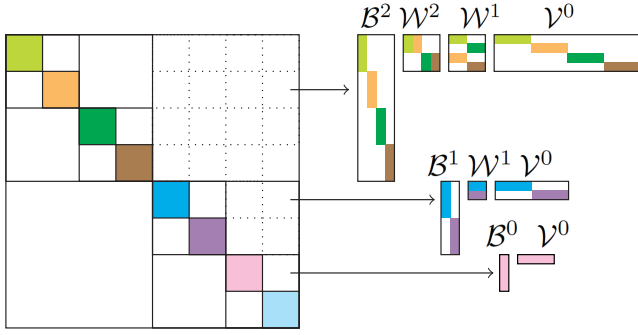


Fig. 1. Example of a $L_H = 3$ -level HOD-BF-compressed matrix. At the leaf level, the diagonal blocks D_τ are stored as dense blocks; at levels $l > L_H$, the off-diagonal blocks are compressed as $L = L_H - l$ -level butterfly representations. The HOD-BF matrix is distributed over 8 MPI processes each denoted by one color. For each τ , D_τ and B_τ are assigned to the same set of processes. B_τ is distributed over multiple processes using parallel layouts of [55].

B. Inversion of HOD-BF-compressed system matrix

Once constructed, the inverse of the HOD-BF-compressed system matrix can be computed approximately and used as a preconditioner during the iterative solution of Eq. (4). The inversion algorithm has been previously described in [45], [50] and is briefly summarized as HODBF_invert in Algorithm 1.

For each non-leaf τ , $\mathcal{D}_\tau = [\mathcal{D}_{\tau_1}, \mathcal{B}_{\tau_1}; \mathcal{B}_{\tau_2}, \mathcal{D}_{\tau_2}]$. \mathcal{D}_τ^{-1} is computed by first recursively computing $\mathcal{D}_{\tau_1}^{-1}$ and $\mathcal{D}_{\tau_2}^{-1}$. Next, the butterfly block is updated as $\mathcal{B}_{\tau_i} \leftarrow \mathcal{D}_{\tau_i}^{-1} \mathcal{B}_{\tau_i}$ with both $\mathcal{D}_{\tau_i}^{-1}$ and \mathcal{B}_{τ_i} already compressed (see line 7). Finally the updated matrix $[\mathcal{I}, \mathcal{B}_{\tau_1}; \mathcal{B}_{\tau_2}, \mathcal{I}]$ is inverted using the butterfly extension of the Sherman-Morrison-Woodbury formula [58], named BF_SMW in Algorithm 2 (see line 8).

The operations at line 7 of Algorithm 1 and lines 3, 5 of Algorithm 2 require construction of a new butterfly representation out of combinations of existing butterfly representations. This type of operations can be accelerated by rapid multiplications of the operator with random vectors and reconstructing the butterfly from the results with tolerance χ_{fact} . This algorithm is named BF_random_matvec and requires $O(n^{3/2} \log n)$ CPU operations [55], and consequently the computational cost for \mathcal{Z}^{-1} scales as $O(N^{3/2} \log N)$.

Algorithm 1 HODBF_invert(\mathcal{A}): Inversion of a HOD-BF matrix.

Input: \mathcal{A} in HOD-BF form

Output: \mathcal{A}^{-1} in HOD-BF form

- 1: Let $\mathcal{D}_\tau = \mathcal{A}$ with τ denoting the root node.
- 2: **if** \mathcal{D}_τ dense **then**
- 3: Directly compute \mathcal{D}_τ^{-1}
- 4: **else**
- 5: $\mathcal{D}_\tau = [\mathcal{D}_{\tau_1}, \mathcal{B}_{\tau_1}; \mathcal{B}_{\tau_2}, \mathcal{D}_{\tau_2}]$
- 6: $\mathcal{D}_{\tau_i}^{-1} \leftarrow \text{HODBF_invert}(\mathcal{D}_{\tau_i}), i = 1, 2$
- 7: $\mathcal{B}_{\tau_i} \leftarrow \text{BF_random_matvec}(\mathcal{D}_{\tau_i}^{-1} \mathcal{B}_{\tau_i}), i = 1, 2$
- 8: $\mathcal{D}_\tau^{-1} \leftarrow \text{BF_SMW}\left(\begin{bmatrix} \mathcal{I} & \mathcal{B}_{\tau_1} \\ \mathcal{B}_{\tau_2} & \mathcal{I} \end{bmatrix}\right) \begin{bmatrix} \mathcal{D}_{\tau_1}^{-1} & \\ & \mathcal{D}_{\tau_2}^{-1} \end{bmatrix}$
- 9: **end if**

Algorithm 2 BF_SMW(\mathcal{A}): SMW inversion of a butterfly matrix.

Input: $\mathcal{A} - \mathcal{I}$ is a butterfly of L levels

Output: $\mathcal{A}^{-1} - \mathcal{I}$ is a butterfly of L levels

- 1: Split \mathcal{A} into four children butterflies of $L - 2$ levels: $\mathcal{A} = [\mathcal{A}_{11}, \mathcal{A}_{12}; \mathcal{A}_{21}, \mathcal{A}_{22}]$ using \mathfrak{T}_H
- 2: $\mathcal{A}_{22}^{-1} \leftarrow \text{BF_SMW}(\mathcal{A}_{22})$
- 3: $\mathcal{A}_{11} \leftarrow \text{BF_random_matvec}(\mathcal{A}_{11} - \mathcal{A}_{12} \mathcal{A}_{22}^{-1} \mathcal{A}_{21} - \mathcal{I})$
- 4: $\mathcal{A}_{11}^{-1} \leftarrow \text{BF_SMW}(\mathcal{A}_{11})$
- 5: $\mathcal{A}^{-1} - \mathcal{I} \leftarrow \text{BF_random_matvec}\left(\begin{bmatrix} \mathcal{I} & \mathcal{A}_{11}^{-1} \\ -\mathcal{A}_{22}^{-1} \mathcal{A}_{21} & \mathcal{I} \end{bmatrix} \begin{bmatrix} \mathcal{A}_{11}^{-1} & -\mathcal{A}_{12} \mathcal{A}_{22}^{-1} \\ \mathcal{A}_{22}^{-1} & \mathcal{I} \end{bmatrix} - \mathcal{I}\right)$

C. Rank scaling and complexity estimate

We first provide a simple analysis of the butterfly rank r of the off-diagonal blocks \mathcal{B}_τ in HOD-BF, and then summarize the complexity of the proposed HOD-BF-accelerated solver with preconditioner.

The asymptotic scaling of the butterfly ranks has been studied for various schemes, including the strong-admissibility-based \mathcal{H} matrix frameworks such as [37], [47], [48], $r = O(1)$ and weak-admissibility-based frameworks, e.g., HOD-BF. It was shown that $r = O(\log n)$ for 2-dimensional SIE [45], [46], and $r = O(n^{1/4})$ for 3-dimensional SIE [55]. In what follows, we show that when applied to blocks in the 3-dimensional VIE system matrix, $r = O(n^{1/3})$. Moreover, despite of the non-constant butterfly ranks, the off-diagonal block can still be compressed with $O(n \log n)$ memory and CPU complexities using the butterfly algorithm.

To begin with, Fig. 2 shows an example of scatterer pair (o, s) corresponding to one $m \times n$ block $\mathcal{Z}(o, s)$ with $L = 6$, the subscatterers at level $L/2 = 3$ of \mathfrak{T}_o and \mathfrak{T}_s are shown in Fig. 2(a), and the subblocks corresponding to these subscatterer pairs are shown in Fig. 2(b). Apparently, there are $2^{L/2} = 8$ subscatterers at level $L/2$ of both \mathfrak{T}_o and \mathfrak{T}_s , and a total of $2^L = 64$ subblocks $\mathcal{Z}(\tau, \nu)$ for nodes $\tau, \nu = 1, \dots, 8$ at level $L/2$. Each subscatterer contains $O(n^{1/6}) \times O(n^{1/6}) \times O(n^{1/6}) = O(n^{1/2})$ number of unknowns. Among the 2^L subblocks, the majority ($O(2^L)$ of them) representing interactions between well-separated subscatterer pairs have rank $r_{\tau, \nu}$ as $O(1)$ [36], [46], [59], [60]. There are only $O(2^{L/3}) \approx 4$ subblocks (highlighted in red) representing interactions between face-sharing subscatterer pairs. Note that here the big O notation is needed as in practise there can be pairs partially sharing one face (i.e., non-confirming partitioning in \mathfrak{T}_o and \mathfrak{T}_s). The ranks of these subblocks are determined by the number of unknowns residing on the common face, leading to $r_{\tau, \nu} = O(n^{1/6}) \times O(n^{1/6}) = O(n^{1/3})$. Here, $O(n^{1/6})$ represents the number of unknowns in one of three dimensions of a subscatterer. In addition, there can be $O(2^{L/3})$ subblocks representing interactions between edge-sharing or vertex-sharing subscatterer pairs whose ranks scale as $O(n^{1/6})$ or $O(\log n)$. These edge-sharing or vertex-sharing pairs can be safely ignored compared to the face-sharing pairs. As a result, the transfer matrices $\mathcal{W}_{\tau, \nu}$ in $\mathcal{W}^{L/2}$ require only $O(2^L) \times (O(1))^2 + O(2^{L/3}) \times (O(n^{1/3}))^2 = O(n)$

storage units. Similarly, it can be shown that any factor \mathcal{B}^L , \mathcal{W}^l , and \mathcal{V}^0 requires $O(n)$ memory and hence the total memory for one butterfly is $O(n \log n)$. For the same reason, the butterfly construction of $\mathcal{Z}(o, s)$ requires $O(n \log n)$ CPU operations. BE CAREFUL: the entry evaluation is $O(r^2)$, the id is $O(r^3)$, but it seems entry evaluation has very large prefactors. Moreover, the randomized butterfly construction BF_random_matvec used in the inversion algorithm also requires $O(n^{3/2} \log n)$ CPU operations despite of the non-constant butterfly rank [55]. Following the analysis in [45], the computational costs of the proposed HOD-BF-accelerated solver and preconditioner for VIE scale as $O(N \log^2 N)$ and $O(N^{3/2} \log N)$, respectively.

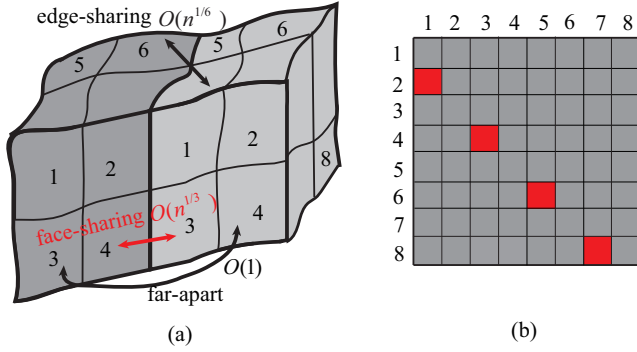


Fig. 2. (a) Subscatterers (labeled from 1 to 8) at level $L/2$ of both \mathfrak{T}_o and \mathfrak{T}_s with $L = 6$. (b) Subblocks at level $L/2$ of both \mathfrak{T}_o and \mathfrak{T}_s . The subblocks with rank $O(n^{1/3})$ are highlighted in red.

D. Parallelization

The proposed distributed-memory parallelization scheme assigns processors according to the binary tree \mathfrak{T}_H . Let τ denote the root of \mathfrak{T}_H . Assuming a total of P processors, the proposed scheme assigns all P processors to τ and approximately $P/2$ processors to each of τ_1 and τ_2 . Recall that \mathcal{D}_τ is partitioned into $[\mathcal{D}_{\tau_1}, \mathcal{B}_{\tau_1}; \mathcal{B}_{\tau_2}, \mathcal{D}_{\tau_2}]$. The first $P/2$ processors handle $\mathcal{D}_{\tau_1}, \mathcal{B}_{\tau_1}$, and the second $P/2$ processors handle $\mathcal{B}_{\tau_2}, \mathcal{D}_{\tau_2}$. The processor subdivision continues until the subtree rooted at one node of \mathfrak{T}_H is entirely handled by one processor. Let P_τ denote the number of processors in charge of \mathcal{D}_τ and \mathcal{B}_τ . The butterfly represented off-diagonal blocks $\mathcal{B}_{\tau_1}, \mathcal{B}_{\tau_2}$ are distributed using the partitioning scheme detailed in [55]. See Fig. 1 for an example of the parallel layout with 8 processors. According to this layout, Algorithms 1 and 2 are executed in parallel:

In Algorithm 1, the update operation (line 7) are handled by P_{τ_i} processors. As $\mathcal{D}_{\tau_i}^{-1}$ and \mathcal{B}_{τ_i} are handled by the same set of processors, no data redistribution is needed. Since there is no data dependency between nodes at each level, Algorithm 1 is highly parallelizable. On the other hand, the BF_SMW operation (line 8) is handled by P_τ processors. Due to the sequential nature of BF_SMW, i.e., lines 2-5 of Algorithm 2 are executed sequentially, the parallelization scheme works as follows: assuming that BF_SMW is executed on P processors, a data redistribution is performed at the splitting step (line 1) where each of $\mathcal{A}_{11}, \mathcal{A}_{12}, \mathcal{A}_{21}, \mathcal{A}_{22}$ is still distributed over P

processors following the layout of [55]. The splitting operation proceeds with $L/2$ levels until the children blocks become low-rank products. Note that during the splitting, the number of nonzero blocks in the children butterflies decreases and some of the P processors become idle towards the leaf level of the splitting operation. To alleviate such load imbalance, each nonzero block in a butterfly can be distributed over multiple processors. Following the splitting step, no more redistribution of butterflies is needed at lines 2-5. In both Algorithm 1 and 2, the essential building block BF_random_matvec is parallelized following [55]. Note that for line 5 of Algorithm 2, the matrix-vector multiplication results are not conformal to the layout of \mathcal{A}^{-1} , hence a data redistribution of the vectors is also required.

Overall, this scheme allows good load balance and reduced collective communication during the overall HOD-BF construction, inversion and application.

IV. NUMERICAL RESULTS

This section presents several numerical examples that demonstrate the accuracy and efficiency of the proposed butterfly-accelerated VIE solver when applied to the EM analysis of heterogeneous scatterers. Unless stated otherwise below, the scatterers are illuminated by an \hat{x} polarized plane-wave travelling along $-\hat{z}$ direction. All simulations are carried out on the Cori supercomputer of Haswell nodes; each node has two 16-core Intel Xeon E5-2698v3 processors and 128 GB of 2133 MHz DDR4 memory. Unless otherwise stated, each simulation is carried out on 16 nodes.

A. Validation

1) *Positive permittivity*: First, a five-layered sphere with layer thickness of 0.06 m is analyzed. The permittivity of each layer, from outer to inner, is $2\varepsilon_0, 3\varepsilon_0, 4\varepsilon_0, 5\varepsilon_0$ and $6\varepsilon_0$ [Fig. 3(a)]. The sphere is discretised by a tetrahedral mesh with $N = 407,842$, and the radar cross section (RCS) of the sphere is computed at $f = 750$ MHz using the proposed VIE solver with compression tolerance $\chi_{con} = 10^{-4}$, factorization tolerance $\chi_{fact} = 10^{-1}$, and TFQMR stopping criteria $\chi_{sol} = 10^{-3}$. The computed RCS is compared with the one obtained via (exact) Mie series solution [Fig. 3(b)]. Let x_i and \hat{x}_i , $i = 1, \dots, N_{rcs}$ denote the N_{rcs} RCS samples computed by the proposed solver and Mie series solution, respectively. The relative root-mean-square error (RMSE) is defined as $\sqrt{\sum_i (x_i - \hat{x}_i)^2 / N_{rcs} / \max_i \hat{x}_i}$, where x_i are in their original values rather than in dBsm. The relative RMSE between two solutions is 0.024. The good agreement between results shows that the proposed VIE solver with the chosen parameters achieve good accuracy. It should be noted that the proposed solver requires 2 and 78 TFQMR iterations with and without preconditioner, respectively. The details of the simulation performed by the proposed solver are provided in Table I.

Next, a three-layered sphere with a hollow spherical core (i.e., two shells) is analyzed. The sphere has a radius of 2.4 m and each shell has thickness of 0.03 m. The dielectric permittivity of each shell, from inner to outer, is $4\varepsilon_0$ and $2\varepsilon_0$, respectively [Fig. 4]. The tetrahedral mesh of two shells

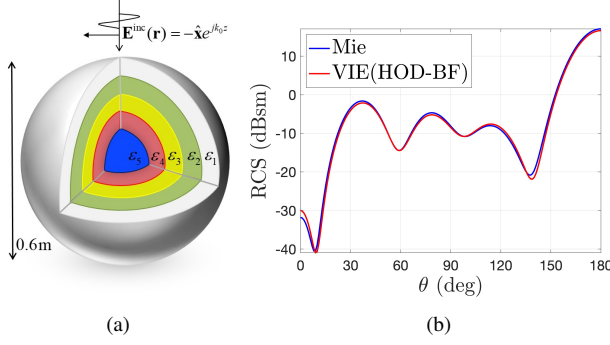


Fig. 3. (a) The structure considered for the analysis: a five-layered sphere with $\varepsilon_1 = 2\varepsilon_0$, $\varepsilon_2 = 3\varepsilon_0$, $\varepsilon_3 = 4\varepsilon_0$, $\varepsilon_4 = 5\varepsilon_0$, and $\varepsilon_5 = 6\varepsilon_0$. (b) The RCSs computed by the proposed HOD-BF-accelerated VIE solver and Mie series solution at $f = 750$ MHz.

consists of $N = 5,530,950$ faces. The RCS is computed at $f = 600$ MHz using the proposed VIE solver with $\chi_{con} = 10^{-4}$, $\chi_{fact} = 10^{-1}$, and $\chi_{sol} = 10^{-3}$. The computed RCS is compared with the one obtained via the Mie series solution [Fig. 4(b)]. The relative RMSE between two solutions is 0.0034. Once again, the results show very good agreement. The proposed VIE solver with and without preconditioner requires 7 and 643 TFQMR iterations, respectively. More details on the simulation are provided in Table I.

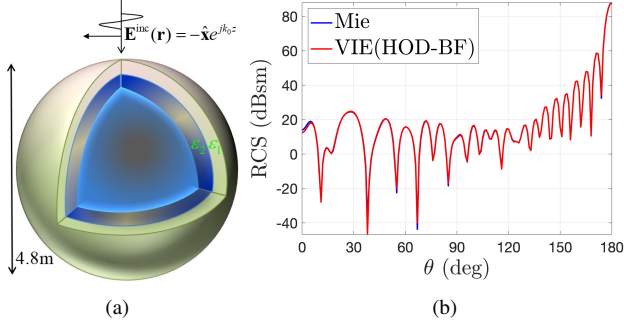


Fig. 4. (a) The structure considered for the analysis: a two-layered shell with $\varepsilon_1 = 2\varepsilon_0$, and $\varepsilon_2 = 4\varepsilon_0$. (b) The RCSs computed with the proposed HOD-BF-accelerated VIE solver and Mie series solution at $f = 600$ MHz.

TABLE I

TECHNICAL DATA FOR THE NUMERICAL EXAMPLES: LAYERED SPHERE, SHELL, AND SPHERE WITH NEGATIVE PERMITTIVITY.

Example	Layered	Shell	Negative Sphere
Max. dimension	$1.5\lambda_0$	$9.6\lambda_0$	$1.6\lambda_0$
N	407,842	5,530,950	896,001
Max. rank	1088	1331	1198
Memory for \mathcal{Z}	57 GB	324 GB	96 GB
Memory for \mathcal{Z}^{-1}	28 GB	321 GB	306 GB
Construction time	7.4 min	1.5 h	1.9 h
Inversion time	26 min	6.2 h	6.5 h
Solve time w/ precondition.	2.3 s	48.3 s	21.4 s
Iteration # w/ precondition.	2	7	6
Solve time w/o precondition.	42.8 s	32 min	46 min
Iteration # w/o precondition.	78	643	3447

2) *Negative permittivity*: In general, the VIE formulation results in poorly conditioned systems when applied to the

scattering problems involving negative permittivity. Iterative solution of such systems often do not converge with simple or no preconditioners. However, the use of HOD-BF with increased accuracy in factorization and solution provides stable and accurate results, even for the ill-conditioned systems. To demonstrate the stability and accuracy of the proposed simulator for such systems, a sphere of radius 0.3 m with permittivity $\varepsilon(\mathbf{r}) = (-4 - 0.2j)\varepsilon_0$ is analyzed. The sphere is discretized by a tetrahedral mesh consisting of $N = 896,001$ faces. The RCS is computed at $f = 800$ MHz using the proposed VIE solver with $\chi_{con} = 10^{-4}$, $\chi_{fact} = 10^{-3}$, and $\chi_{sol} = 10^{-5}$. The computed RCS is compared with the one obtained by the analytical Mie series solution [Fig. 5(b)]; The relative RMSE between two solutions is 0.0027. It is a well-known fact that the discretized VIE system becomes ill-conditioned for the EM analysis of the scatterers with negative permittivity and the iterative solution of the VIE system oftentimes cannot be obtained [21], [24], [61]. Here the proposed VIE solver with preconditioner obtains the solution after 6 TFQMR iterations. Note that the choice of $\chi_{fact} = 10^{-3}$ represents a trade-off between inversion time and solve time. This clearly shows that the proposed solver is less sensitive to the ill-conditioning due to negative permittivity and it is robust for the board permittivity EM analysis. This example was run on 16 nodes and the details of the simulation are provided in Table I.

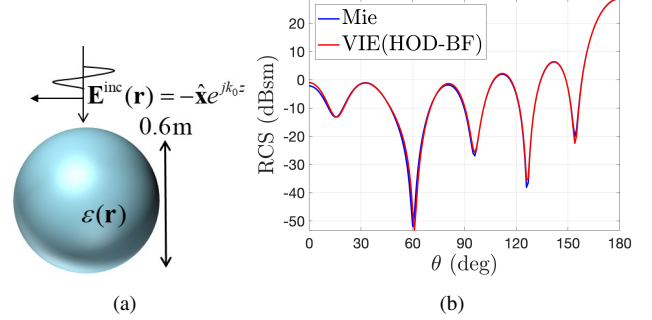


Fig. 5. (a) The structure considered for the analysis: a sphere with negative permittivity, $\varepsilon(\mathbf{r}) = (-4 - 0.2j)\varepsilon_0$. (b) The RCSs computed via the proposed HOD-BF-accelerated VIE solver and Mie series solution at $f = 800$ MHz.

B. Computational Complexity

In this subsection, several examples were considered to verify the predicted CPU and memory scaling of the proposed HOD-BF-accelerated VIE solver. In all examples, the scatterers are discretized by the tetrahedral mesh with elements comparable to $\lambda_0/10$. The performance of the proposed HOD-BF-accelerated VIE solver is compared with that of an \mathcal{H} matrix-accelerated VIE solver with strong admissibility, and an HOD-LR-accelerated VIE solver. For the \mathcal{H} matrix-accelerated solver, all low-rank matrix blocks corresponding to $d_{a,b} > \alpha \max(r_a, r_b)$ are compressed, where r_a and r_b are the radii of the spheres enclosing subscatterers a and b , $d_{a,b}$ is the distance between the centers of subscatterers, and $\alpha = 2$. We used a cobblestone distance sorting-based partitioning [62] in the \mathcal{H} matrix, HOD-LR, and HOD-BF

accelerated solvers. For the low-rank compression algorithms, we used a blocked variant of adaptive cross approximation algorithm [63] for better stability and parallel performance. The \mathcal{H} matrix solver and HOD-LR solvers are available in the open-source software ButterflyPACK. In the following tests, $\chi_{con} = 10^{-3}$, $\chi_{fact} = 10^{-1}$, and $\chi_{sol} = 10^{-3}$.

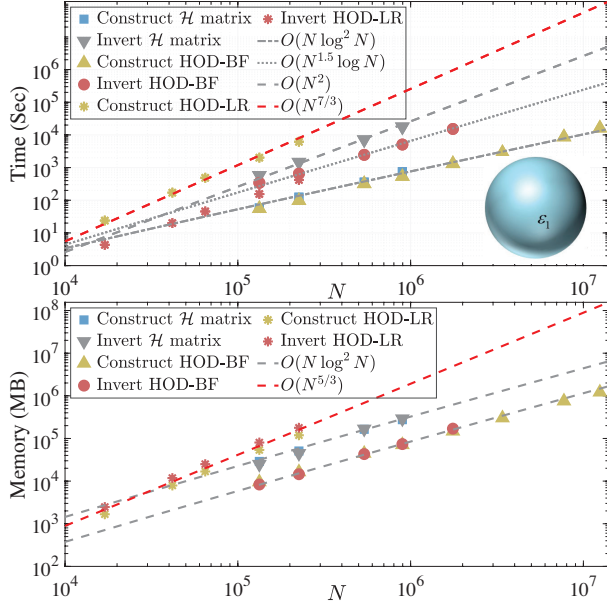


Fig. 6. (a) The CPU and (b) memory scaling of the HOD-BF-accelerated solver, \mathcal{H} matrix-accelerated solver, and HOD-LR-accelerated solver when applied to the EM analysis of a sphere with permittivity $\varepsilon(\mathbf{r}) = (4 - 0.0001j)\varepsilon_0$.

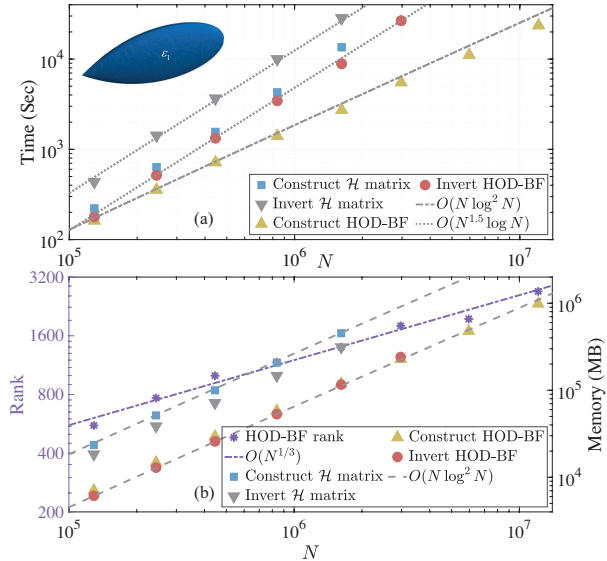


Fig. 7. (a) CPU and (b) memory scaling of the HOD-BF-accelerated solver and \mathcal{H} matrix-accelerated solver when applied to the EM analysis of a NASA almond with permittivity $\varepsilon(\mathbf{r}) = (4 - 0.0001j)\varepsilon_0$

1) *Positive permittivity*: The computational complexity of the proposed scheme is studied as a function of electrical size by varying the frequency and fixing object size, or, by keeping the frequency fixed and varying the object size. In the first example, EM analysis of an homogeneous sphere

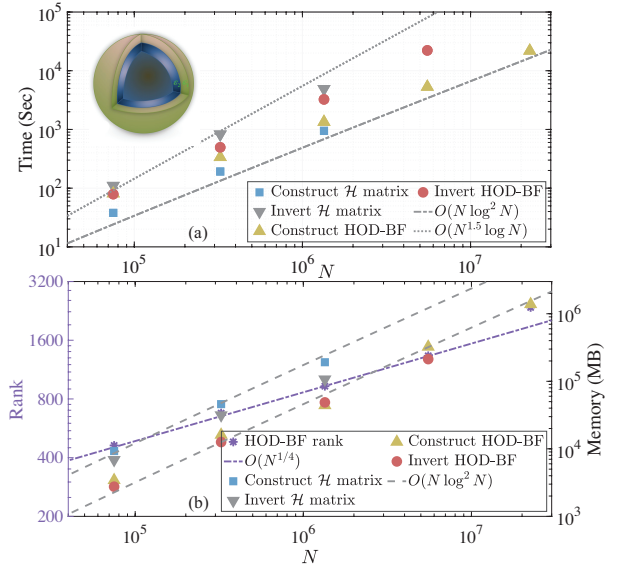


Fig. 8. (a) The CPU and (b) memory scaling of the HOD-BF-accelerated solver and \mathcal{H} matrix-accelerated solver when applied to the EM analysis of a two-layered shell with inner and outer shell permittivities of $4\varepsilon_0$ and $2\varepsilon_0$

TABLE II
TIME (IN SECONDS) AND ITERATION COUNT FOR THE SOLVE PHASE OF THE PROPOSED HOD-BF-ACCELERATED VIE SOLVER WITH ('w/') AND WITHOUT ('w/o') PRECONDITIONER. NOTE THAT EACH TFQMR ITERATION REQUIRES 6 MVPs.

N	225,746	538,782	896,001	1,757,024
Solve time w/	1.3	3.3	6.77	17
Iteration # w/	3	3	4	5
Time per MVP w/	0.07	0.185	0.28	0.56
Solve time w/o	48.9	310.1	509.2	1940
Iteration # w/o	279	712	786	1535
Time per MVP w/o	0.03	0.07	0.11	0.21

TABLE III
TECHNICAL DATA FOR THE EXAMPLES: SPHERE, ALMOND, HUMAN HEAD

Example	Sphere	Almond	Head
Max. dimension	$3.4\lambda_0$	$10\lambda_0$	$0.662\lambda_0$
N	1,757,024	2,972,024	1,759,208
Max. rank	1496	2006	1981
Memory for \mathcal{Z}	148 GB	228 GB	314 GB
Memory for \mathcal{Z}^{-1}	170 GB	241 GB	353 GB
Construction time	22 min	1.5 h	3 h
Inversion time	4.2 h	7.4 h	7.3 h
Solve time w/ precondition.	17 s	24 s	5.4 min
Iteration # w/ precondition.	5	4	90
Solve time w/o precondition.	32 min	35 min	1.8 h
Iteration # w/o precondition.	1535	962	3698

with radius 0.3 m and permittivity $\varepsilon(\mathbf{r}) = (4 - 0.0001j)\varepsilon_0$ is considered for frequencies 700 MHz, 850 MHz, 1.2 GHz, 1.3 GHz, 1.6 GHz, 2.0 GHz, 2.5 GHz, and 2.8 GHz. The analysis at 2.8 GHz requires a tetrahedral mesh with $N = 12,455,135$. Comparison of the complexities of the \mathcal{H} matrix-accelerated solver and the proposed HOD-BF-accelerated solver in Fig. 6 reveals that the CPU time for HOD-BF construction and inversion scale as $O(N \log^2 N)$ and $O(N^{1.5} \log N)$, respectively and the memory requirement scales as $O(N \log^2 N)$. These

scaling results match with the predicted ones in Section III-C. The maximum butterfly rank ranges from 745 to 2800, which scales as $O(N^{1/3})$. In contrast, the inversion and memory cost of \mathcal{H} matrix-accelerated solver scale roughly $O(N^2)$ and $O(N \log^2 N)$, respectively. In addition, its construction cost scales with $O(N^{1.5})$, although the scaling plot was not provided in Fig. 6 to avoid overcrowding. In addition, the proposed HOD-BF-accelerated solver requires four times less memory compared to \mathcal{H} matrix-accelerated solver. Moreover, the HOD-LR VIE solver scales as $O(N^{5/3})$ for memory and $O(N^{7/3})$ for CPU. Due to its prohibitively high CPU and memory requirement for 3D VIE, we do not consider HOD-LR solvers for the other experiments.

The solution obtained via HOD-BF-accelerated solver without preconditioner, which only requires performing HOD-BF construction, will not converge to $\chi_{sol} = 10^{-3}$ within 3000 iterations when the electrical size of the object is large. For example, iterative solution via such solver requires 167 iterations for $N = 13,3207$, 1535 iterations for $N = 1,757,024$, and fails to converge for larger problems. In contrast, the HOD-BF-accelerated solver with preconditioner requires at most 5 iterations for the analysis at all frequencies. The data for the solve phase of HOD-BF-accelerated VIE solver with and without preconditioner is provided in Table II. As N increases, the discretized VIE system becomes increasingly ill-conditioned, yet the approximate inverse preconditioner remains very effective and allows retaining the number TFQMR iterations a few. The time for each matrix-vector product (MVP) with the preconditioner is about twice of that without the preconditioner. The technical data for $N = 1,757,024$ is provided in Table III.

Next, a NASA almond with size $0.25 \times 0.1 \times 0.04$ m and permittivity $\varepsilon(\mathbf{r}) = (4 - 0.0001j) \varepsilon_0$ is considered to study the computational complexity for frequencies 4.4, 5.4, 6.7, 8, 10, 12, 15, and 20 GHz. The analysis at 20 GHz requires a tetrahedral mesh with $N = 12,112,059$. Fig. 7 shows that the CPU and memory requirements of the HOD-BF-accelerated solver for the construction and inversion scale as those predicted in Section III-C. In contrast, the CPU scaling of the \mathcal{H} matrix-accelerated solver for the construction and inversion is at least $O(N^{1.5})$. Again, the iterative solution via HOD-BF-accelerated solver without preconditioner does not converge for the meshes with more than $N = 2,972,024$ elements. However, the HOD-BF-accelerated solver with preconditioner requires maximum 4 TFQMR iterations for the analysis at all frequencies. More details on the analysis with the mesh comprising $N = 2,972,024$ elements are provided in Table III.

Furthermore, a three-layered sphere with a hollow spherical core (i.e., two shells) illuminated by a plane wave at 600 MHz is considered for the complexity analysis. The permittivities of inner and outer shells with thickness of 0.03 m are $4\varepsilon_0$ and $2\varepsilon_0$, respectively. The radius of the sphere is increased from 0.3 m to 4.8 m; the sphere with the largest radius requires a mesh with $N = 22,491,763$. Fig. 8 shows that the CPU and memory requirements of the HOD-BF-accelerated solver for the construction and inversion scale as those predicted in Section III-C. Note that the memory requirement and

construction time behave slightly better than the predicted $O(N \log^2 N)$ scaling. For this example, the scalings for the \mathcal{H} matrix-accelerated solver are similar to those of the proposed solver, but with much larger leading constants. More details on the analysis with the mesh comprising $N = 5,530,950$ elements are provided in Table I.

2) *Negative Permittivity*: In the next example, a sphere with permittivity $\varepsilon(\mathbf{r}) = (-4 - 2j) \varepsilon_0$ is analyzed for frequencies 700 MHz, 900 MHz, 1.2 GHz, 1.4 GHz, and 1.7 GHz. The analysis at 1.7 GHz requires a mesh with $N = 1,757,024$.

Fig. 9 shows that the construction and inversion costs of the proposed HOD-BF-accelerated solver scale as $O(N \log^2 N)$ and $O(N^{1.5} \log N)$, respectively; its memory requirement scales as $O(N \log^2 N)$; the maximum butterfly rank scales as $O(N^{1/3})$. All the scaling results match the predicted ones in Section III-C. Note that the solve time is also plotted (in blue) in Fig. 9(a), which is negligible compared to the construction and inversion times, since the TFQMR iteration count is maximum 12 for the analyses at all frequencies.

From the analysis in this subsection, we can conclude that the proposed HOD-BF-accelerated solver requires much less CPU time and memory compared to \mathcal{H} matrix-accelerated solver when the electrical size of the scatterer is large. Indeed, the complexities of \mathcal{H} matrix-accelerated solver strongly depend on the geometry of the structure and can scale as bad as $O(N^2)$. Furthermore, it is clear that the proposed simulator can provide efficient and accurate EM analysis even for the problems necessitating the solution of ill-conditioned VIE systems.

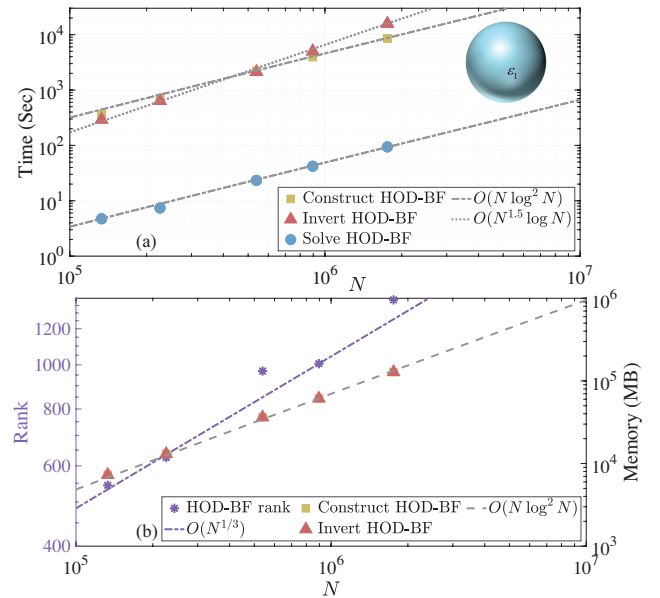


Fig. 9. (a) The CPU and (b) memory scaling of the HOD-BF-accelerated solver and \mathcal{H} matrix-accelerated solver when applied to the EM analysis of a sphere with permittivity $\varepsilon(\mathbf{r}) = (-4 - 2j) \varepsilon_0$

C. Biomedical Application

The proposed HOD-BF-accelerated VIE solver is applied to the characterization of EM fields inside a human head induced by a cellphone operated at 900 MHz. To this end, a head model

is discretized using tetrahedrons with an average volume of 1 mm^3 ; the resulting mesh has $N = 1,759,208$ faces. The head model is excited by a z -directed dipole positioned near to the ear. The relative permittivities of the tissues are plotted in Fig. 10; those range from $12.50 - j2.86$ to $68.60 - j48.134$. The electric fields induced inside the human head is computed using the proposed HOD-BF-accelerated solver [Fig. 11].

The HOD-BF-accelerated solver with $\chi_{con} = 10^{-3}$, $\chi_{fact} = 10^{-1}$ and $\chi_{sol} = 10^{-5}$ requires 45 and 2568 TFQMR iterations with and without the preconditioner, respectively. This example was run on 32 nodes and the details of the simulation are provided in Table III.

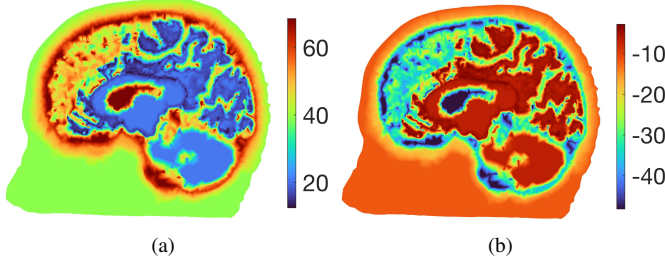


Fig. 10. The profile of the (a) real and (b) imaginary parts of the relative permittivities of tissues in the head model at 900 MHz.

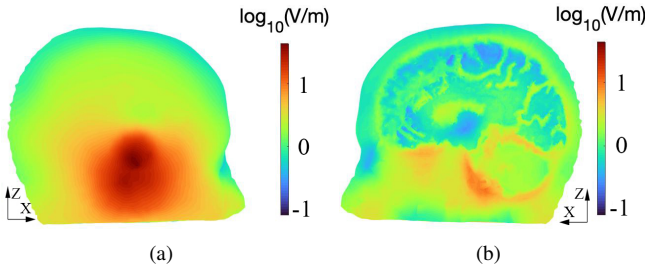


Fig. 11. (a) Right view and (b) left view of the magnitude of the electric field (in log scale) inside the human head excited by a dipole with 1 mA-m strength at 900 MHz computed by the proposed HOD-BF-accelerated solver.

V. CONCLUSION

In this paper, a butterfly-accelerated VIE solver for the EM analysis of electrically-large scatterers comprising of broad permittivity values was proposed. The solver constructs the HOD-BF compressed blocks in the VIE system and accelerates the matrix-vector multiplications using these blocks. Furthermore, it leverages a preconditioner formed by the approximate inverse of the system matrix to ensure the fast convergence of the iterative solution. The complexity analysis and numerical experiments confirmed $O(N \log^2 N)$ CPU and memory scaling of the proposed solver for the construction of HOD-BF compressed blocks. Moreover, an $O(N^{1.5} \log N)$ CPU time scaling and $O(N \log^2 N)$ memory scaling were validated for the inversion of the system matrix to form the preconditioner. The proposed HOD-BF-accelerated VIE solver was applied to EM analyses of various canonical and realistic scatterers. It was shown that the proposed solver can achieve significant CPU and memory savings compared to the low-rank-based \mathcal{H} matrix-accelerated VIE solver for the high-frequency EM analysis.

ACKNOWLEDGMENT

Research reported in this publication was supported by the National Institute Of Mental Health of the National Institutes of Health under Award Number R00MH120046. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. In addition, Y. Liu thanks the support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program through the FAST-Math Institute under Contract No. DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory.

REFERENCES

- [1] D. Schaubert, D. Wilton, and A. Glisson, "A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies," *IEEE Trans. Antennas Propag.*, vol. 32, no. 1, pp. 77–85, 1984.
- [2] M. Sancer, K. Sertel, J. Volakis, and P. VanAlstine, "On volume integral equations," *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1488–1495, may 2006.
- [3] M. M. Botha, "Solving the volume integral equations of electromagnetic scattering," *J. Comput. Phys.*, vol. 218, no. 1, pp. 141–158, oct 2006.
- [4] K. Sertel and J. L. Volakis, "Multilevel fast multipole method solution of volume integral equations using parametric geometry modeling," *IEEE Trans. Antennas Propag.*, vol. 52, no. 7, pp. 1686–1692, 2004.
- [5] C.-C. Lu, "A fast algorithm based on volume integral equation for analysis of arbitrarily shaped dielectric radomes," *IEEE Trans. Antennas Propag.*, vol. 51, no. 3, pp. 606–612, 2003.
- [6] J. R. Phillips and J. K. White, "A precorrected-fft method for electrostatic analysis of complicated 3-d structures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1059–1072, 1997.
- [7] Xiao-Chun Nie, Le-Wei Li, Ning Yuan, Tat Soon Yeo, and Yeow-Beng Gan, "Precorrected-FFT solution of the volume Integral equation for 3-D inhomogeneous dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 53, no. 1, pp. 313–320, 2005.
- [8] A. C. Yucel, I. Georgakakis, A. G. Polimeridis, H. Bagci, and J. White, "VoxHenry: A fast Fourier transform - accelerated 3D inductance extraction program for voxelized geometries," *IEEE Trans. Microwave Theory Tech.*, vol. 66, no. 4, pp. 1723–1735, 2018.
- [9] N. A. Ozdemir and Jin-Fa Lee, "A low-rank IE-QR algorithm for matrix compression in volume integral equations," *IEEE Antennas Propag. Mag.*, vol. 40, no. 2, pp. 1017–1020, 2004.
- [10] W. Chai and D. Jiao, "An \mathcal{H}^2 -matrix-based integral-equation solver of reduced complexity and controlled accuracy for solving electrodynamic problems," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 3147–3159, 2009.
- [11] W. Chai and D. Jiao, " \mathcal{H} and \mathcal{H}^2 -matrix-based fast integral-equation solvers for large-scale electromagnetic analysis," *IET microwaves, antennas & propagation*, vol. 4, no. 10, pp. 1583–1596, 2010.
- [12] Z. Chen, L. J. Gomez, S. Zheng, A. C. Yucel, Z. Zhang, and V. I. Okhmatovski, "Sparsity-aware precorrected tensor train algorithm for fast solution of 2-D scattering problems and current flow modeling on unstructured meshes," *IEEE Trans. Microwave Theory Tech.*, vol. 67, no. 12, pp. 4833–4847, 2019.
- [13] M. Wang, C. Qian, J. K. White, and A. C. Yucel, "VoxCap: FFT-accelerated and Tucker-enhanced capacitance extraction simulator for voxelized structures," *arXiv preprint*, 2020.
- [14] J. Markkanen, C.-C. Lu, X. Cao, and P. Yla-Oijala, "Analysis of volume integral equation formulations for scattering by high-contrast penetrable objects," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2367–2374, 2012.
- [15] M. Costabel, E. Darrigrand, and H. Sakly, "The essential spectrum of the volume integral operator in electromagnetic scattering by a homogeneous body," *Comptes Rendus Mathematique*, vol. 350, no. 3-4, pp. 193–197, 2012.
- [16] J. Kottmann and O. Martin, "Accurate solution of the volume integral equation for high-permittivity scatterers," *IEEE Trans. Antennas Propag.*, vol. 48, no. 11, pp. 1719–1726, 2000.

- [17] L. J. Gomez, A. C. Yucel, and E. Michielssen, "Internally combined volume-surface integral equation for a 3-d electromagnetic scattering analysis of high-contrast media," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 1691–1694, 2017.
- [18] —, "Low-frequency stable internally combined volume-surface integral equation for high-contrast scatterers," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 1423–1426, 2015.
- [19] N. V. Budko and A. B. Samokhin, "Spectrum of the volume integral operator of electromagnetic scattering," *SIAM J. Sci. Comput.*, vol. 28, no. 2, pp. 682–700, 2006.
- [20] Y.-N. Liu, X.-M. Pan, and X.-Q. Sheng, "Fast direct solution of 3-D volume integral equations by skeletonization for dynamic electromagnetic wave problems," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 33, no. 2, p. e2667, 2020.
- [21] L. J. Gomez, A. C. Yucel, and E. Michielssen, "Volume-surface combined field integral equation for plasma scatterers," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 1064–1067, 2015.
- [22] S. B. Sayed, A. C. Yucel, and L. J. Gomez, "A multi-region internally combined volume surface integral equation for em analysis of inhomogeneous negative and positive permittivity scatterers," in *2020 IEEE USNC-CNC-URSI North American Radio Science Meeting (Joint with AP-S Symposium)*, 2020, pp. 185–186.
- [23] L. J. Gomez, A. C. Yucel, and E. Michielssen, "The icvsie: A general purpose integral equation method for bio-electromagnetic analysis," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 3, pp. 565–574, 2017.
- [24] A. C. Yucel, L. J. Gomez, and E. Michielssen, "Internally combined volume-surface integral equation for em analysis of inhomogeneous negative permittivity plasma scatterers," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 4, pp. 1903–1913, 2018.
- [25] S. P. Groth, A. G. Polimeridis, A. Tambova, and J. K. White, "Circulant preconditioning in the volume integral equation method for silicon photonics," *JOSA A*, vol. 36, no. 6, pp. 1079–1088, 2019.
- [26] W. C. Chew and C.-C. Lu, "NEPAL—an algorithm for solving the volume integral equation," *Microw. Opt. Tech. Lett.*, vol. 6, no. 3, pp. 185–188, 1993.
- [27] W. C. Chew and C. . Lu, "The use of Huygens' equivalence principle for solving the volume integral equation of scattering," *IEEE Trans. Antennas Propag.*, vol. 41, no. 7, pp. 897–904, 1993.
- [28] R. Gholami, J. Mojlalage, A. Menshov, F. S. Hosseini Lori, and V. I. Okhmatovski, " \mathcal{H} -matrix arithmetic for fast direct and iterative method of moment solution of surface-volume-surface EFIE for 3-D radiation problems," *Progress In Electromagnetics Research*, vol. 82, pp. 189–210, 2018.
- [29] R. Gholami and V. Okhmatovski, "Surface-volume-surface EFIE formulation for fast direct solution of scattering problem on general 3-D composite metal-dielectric objects," *IEEE Trans. Antennas Propag.*, 2020.
- [30] S. Omar and D. Jiao, "A linear complexity direct volume integral equation solver for full-wave 3-D circuit extraction in inhomogeneous materials," *IEEE Trans. Microwave Theory Tech.*, vol. 63, no. 3, pp. 897–912, 2015.
- [31] M. Ma and D. Jiao, "Accuracy directly controlled fast direct solution of general matrices and its application to solving electrodynamic volume integral equations," *IEEE Trans. Microwave Theory Tech.*, vol. 66, no. 1, pp. 35–48, 2017.
- [32] —, "Direct solution of general -matrices with controlled accuracy and concurrent change of cluster bases for electromagnetic analysis," *IEEE Trans. Microwave Theory Tech.*, vol. 67, no. 6, pp. 2114–2127, 2019.
- [33] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numerica*, vol. 18, pp. 243–275, 2009.
- [34] E. Corona, P.-G. Martinsson, and D. Zorin, "An $O(N)$ direct solver for integral equations on the plane," *Appl. Comput. Harmon. Anal.*, vol. 38, no. 2, pp. 284 – 317, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1063520314000529>
- [35] L. Börm, Steffen Grasedyck and W. Hackbusch, "Introduction to hierarchical matrices with application," *Eng. Anal. Bound. Elem.*, vol. 27, pp. 405–422, 2003.
- [36] E. Michielssen and A. Boag, "Multilevel evaluation of electromagnetic fields for the rapid solution of scattering problems," *Microw Opt Technol Lett.*, vol. 7, no. 17, pp. 790–795, 1994.
- [37] —, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," *IEEE Trans. Antennas Propag.*, vol. 44, no. 8, pp. 1086–1093, 1996.
- [38] Y. Li, H. Yang, E. R. Martin, K. L. Ho, and L. Ying, "Butterfly factorization," *Multiscale Model. Sim.*, vol. 13, no. 2, pp. 714–732, 2015.
- [39] Y. Li and H. Yang, "Interpolative butterfly factorization," *SIAM J. Sci. Comput.*, vol. 39, no. 2, pp. A503–A531, 2017.
- [40] Q. Pang, K. L. Ho, and H. Yang, "Interpolative decomposition butterfly factorization," *SIAM J. Sci. Comput.*, vol. 42, no. 2, pp. A1097–A1115, 2020.
- [41] E. Candès, L. Demanet, and L. Ying, "A fast butterfly algorithm for the computation of Fourier integral operators," *Multiscale Model. Sim.*, vol. 7, no. 4, pp. 1727–1750, 2009.
- [42] M. O'Neil, F. Woolfe, and V. Rokhlin, "An algorithm for the rapid evaluation of special function transforms," *Appl. Comput. Harmon. Anal.*, vol. 28, no. 2, pp. 203 – 226, 2010, special Issue on Continuous Wavelet Transform in Memory of Jean Morlet, Part I.
- [43] J. Bremer, Z. Chen, and H. Yang, "Rapid application of the spherical harmonic transform via interpolative decomposition butterfly factorization," *arXiv preprint arXiv:2004.11346*, 2020.
- [44] H. Guo, J. Hu, and E. Michielssen, "On MLMDA/butterfly compressibility of inverse integral operators," *IEEE Antennas Wirel. Propag. Lett.*, vol. 12, pp. 31–34, 2013.
- [45] Y. Liu, H. Guo, and E. Michielssen, "An HSS matrix-inspired butterfly-based direct solver for analyzing scattering from two-dimensional objects," *IEEE Antennas Wirel. Propag. Lett.*, vol. 16, pp. 1179–1183, 2017.
- [46] Y. Liu and H. Yang, "A hierarchical butterfly LU preconditioner for two-dimensional electromagnetic scattering problems involving open surfaces," *J. Comput. Phys.*, vol. 401, p. 109014, 2020.
- [47] 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 Trans. Antennas Propag.*, vol. 65, no. 9, pp. 4742–4750, 2017.
- [48] —, "A butterfly-based direct solver using hierarchical LU factorization for Poggio-Miller-Chang-Harrington-Wu-Tsai equations," *Microw Opt Technol Lett.*, vol. 60, p. 1381–1387, 2018.
- [49] Y. Brick, "Increasing the butterfly-compressibility of moment matrix blocks: A quantitative study," *IEEE Trans. Antennas Propag.*, vol. 69, no. 1, pp. 588–593, 2021.
- [50] Y. Liu, P. Ghysels, L. Claus, and X. S. Li, "Sparse approximate multifrontal factorization with butterfly compression for high-frequency wave equations," *SIAM Journal on Scientific Computing*, no. 0, pp. S367–S391, 2021.
- [51] W. Hackbusch, B. N. Khoromskij, and R. Kriemann, "Hierarchical matrices based on a weak admissibility criterion," *Computing*, vol. 73, no. 3, pp. 207–243, 2004.
- [52] S. Ambikasaran and E. Darve, "An $O(n \log n)$ fast direct solver for partial hierarchically semi-separable matrices," *Journal of Scientific Computing*, vol. 57, no. 3, pp. 477–501, 2013.
- [53] A. Aminfar, S. Ambikasaran, and E. Darve, "A fast block low-rank dense solver with applications to finite-element matrices," *Journal of Computational Physics*, vol. 304, pp. 170–188, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999115006750>
- [54] J. Shaeffer, "Hodlr direct mom solver," in *2018 International Applied Computational Electromagnetics Society Symposium (ACES)*, 2018, pp. 1–2.
- [55] Y. Liu, X. Xing, H. Guo, E. Michielssen, P. Ghysels, and X. S. Li, "Butterfly factorization via randomized matrix-vector multiplications," *SIAM Journal on Scientific Computing*, vol. 43, no. 2, pp. A883–A907, 2021.
- [56] R. W. Freund, "A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems," *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 470–482, 1993.
- [57] H. Cheng, Z. Gimbutas, P. Martinsson, and V. Rokhlin, "On the compression of low rank matrices," *SIAM J. Sci. Comput.*, vol. 26, pp. 1389–1404, 2005.
- [58] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.
- [59] O. M. Bucci and G. Franceschetti, "On the spatial bandwidth of scattered fields," *IEEE Trans. Antennas Propag.*, vol. 35, no. 12, pp. 1445–1455, 1987.
- [60] B. Engquist and H. Zhao, "Approximate separability of the Green's function of the Helmholtz equation in the high frequency limit," *Commun Pur. Appl. Math.*, vol. 71, no. 11, pp. 2220–2274, 2018.
- [61] J. Markkanen and P. Ylä-Oijala, "Numerical comparison of spectral properties of volume-integral-equation formulations," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 178, pp. 269–275, 2016.
- [62] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 m unknowns," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2306–2313, aug 2008.

- [63] 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.