POLITECNICO DI TORINO
Repository ISTITUZIONALE

Fast Direct Solution with Mixed Skeletons and Equivalences Approximation for Electromagnetic Scattering from Conducting Objects

Original
Fast Direct Solution with Mixed Skeletons and Equivalences Approximation for Electromagnetic Scattering from Conducting Objects / Zuo, Y., Li, M., Vipiana, F., Ding, D.. - In: IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION. - ISSN 0018-926X. - (2026). [10.1109/tap.2026.3685110]

Availability:
This version is available at: 11583/3011177 since: 2026-05-21T12:06:13Z

Publisher:
IEEE

Published
DOI:10.1109/tap.2026.3685110

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

25 June 2026

Fast Direct Solution with Mixed Skeletons and Equivalences Approximation for Electromagnetic Scattering from Conducting Objects

Yuhan Zuo, Mengmeng Li, *Senior Member, IEEE,* Francesca Vipiana, *Senior Member, IEEE*, and Dazhi Ding, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a kernel-independent fast direct solver with nested equivalent source approximation (NESA) for electromagnetic scattering from conducting objects at low to moderate frequencies. With strong admissibility condition in this work, to achieve a high efficiency computation, NESA with skeletons is employed to compress the far field interactions via the adaptive cross approximation (ACA) at the leaf level. NESA with equivalent sources is employed at higher levels. Subsequently, the system matrix is factorized by two kinds of elimination matrices corresponding to far and near field interactions respectively. During the factorization, the radiation and receiving matrices are updated from the leaf level up to the top level with prescribed accuracy to account for the contributions of fill-in blocks generated from the near interactions elimination. The complexities of both computation and memory scale as *O*(*N*) for constant-rank problems. Several numerical results are presented to verify the accuracy and complexity of our method.**

***Index Terms*—Electromagnetic scattering, kernel-independent, fast direct solver, equivalence.**

I. Introduction

The integral equations are widely used in the analysis of electromagnetic (EM) scattering problems due to its high accuracy and efficiency. The method of moments (MoM) is a mathematical method that discretizes an integral equation to form a matrix equation and expands the solution into a linear combination of a series of basis functions [1].

Solving the system matrix is a crucial point in analyzing electromagnetic scattering problems and has been an important topic in computational electromagnetics. The matrix equations can either be solved by iterative solutions or direct solutions. For iterative solutions, they are commonly accelerated by fast algorithms such as the fast multipole method (FMM) [2] and its multilevel form the multilevel fast multipole algorithm (MLFMA) [3], FFT-based methods [4], [5], the ACA [6], [7], the multilevel matrix decomposition algorithm (MLMDA) [8], [9], etc. However, in case of ill-conditioned system matrices, the iterative solutions suffer from convergence problems essentially. In addition, the iterative solutions are inefficient in terms of multiple right-hand side (MRHS) problems.

On the other hand, fast direct solutions provide a new and more efficient solution for analyzing electromagnetics problems and have gained considerable interests in recent years due to their significant advantages over iterative solutions over the aforementioned issues [10]–[23]. This is because, different from the iterative solution, the system matrix is expressed in an explicit form in the direct solution scheme and the inverse of the system matrix is also expressed explicitly. For MRHS problems, the solutions are obtained by matrix–matrix multiplication instead of restarting the solving required in iterative solutions. However, conventional direct solutions like LU decomposition, Gaussian elimination, QR decomposition, etc., have high computational complexities and are only applicable to small scale problems. Subsequently, numerous direct solvers with specific matrix structures are reported in the open literature to reduce the complexities. These methods can be generally divided by different admissibility conditions. Admissibility condition is applied to determine whether interactions of different groups need to be compressed. If all the off-diagonal matrices are decomposed in the system matrix, this is referred to as the weak admissibility condition where both the far interactions and neighboring interactions are compressed. These direct methods are with matrix structures such as hierarchically semiseparable matrix (HSS) [24]–[26] and hierarchically off-diagonal low-rank (HODLR) matrix [36]. Although those representative direct solvers accelerated the computation compared to the conventional ones, they are usually less efficient than solvers with the strong admissibility condition because the neighboring interactions corresponding to some off-diagonal blocks that have rather high ranks have to be compressed. The strong admissibility condition, on the other hand, introduces a distance limitation to determine the near and far interactions. The matrix structures with the strong admissibility condition are more efficient but more complicated, as inadmissible

This work was supported in part by Natural Science Foundation of China (NSFC) under Grant 32261133623, 62222108; in part by the "INSIGHT-FOOD, An innovative microwave sensing system for the evaluation and monitoring of food quality and safety" (CUP E13C230001800005), a joint research project within the Executive Program of Scientific and Technological Cooperation between Italy and China (period 2023-2025), funded by NSFC and MAECI.

Y. Zuo, M. Li and D. Ding are with the School of Microelectronics (School of Integrated Circuits), Nanjing University of Science and Technology, Nanjing, China (e-mail: limengmeng@njust.edu.cn).

F. Vipiana is with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy (e-mail: francesca. vipiana@polito.it)

blocks may appear at any position in the system matrix and only far enough groups are considered as compressible. The strong admissibility significantly improved the compression efficiency for low-rank blocks. Some direct solvers are designed under this admissibility condition with matrix structures of butterfly structured matrix [14], [15], hierarchically matrix ($\mathcal{H}$-matrix) [27]–[29] and its improved form, the $\mathcal{H}^2$-matrix [30]. Due to fact that the connections between adjacent levels are introduced during the compression, the computational complexity of the $\mathcal{H}^2$-matrix has been further reduced compared to the $\mathcal{H}$-matrix. In some early works with $\mathcal{H}^2$-matrices, the inverse of the system matrix is obtained in a recursive framework where formatted multiplications and additions are performed [31]. These methods, however, lack direct control of accuracy. Recently, an error controllable fast direct solver of an $\mathcal{H}^2$-matrix is proposed. The direct solver factorizes the system matrix based on the expansion of the original kernel function into a series of Lagrange polynomials [12]. Later, a kernel-independent fast direct solver is proposed with skeletonization factorization [10] –[11], [32]–[33]. Interpolative decomposition (ID) is applied to represent interactions between the original full set basis functions via selected dominant basis functions (i.e., skeletons). The direct solver is a one-way traversal method from the bottom leaf level to the top level, and the skeletons are combined and regrouped into higher levels to form the full set basis functions of parent level groups.

Inspired by the direct solutions mentioned above, in this paper we propose a fast direct solver with a low complexity at low to moderate frequencies. It is worth noting that here the low frequencies refer to constant rank problems without the low-frequency breakdown. The direct solver should avoid such low-frequency regions with respect to serious ill-conditioned problems for accurate solutions. Taking both advantages of less memory consumptions with equivalences and faster computations with skeletons at the leaf level, we further improve the computational efficiency compared with pure skeletonization factorization. In terms of low-rank compression, the mixed skeletons and equivalences nested approximation method can be seen as the mixed-form NESA [34], [35]. Moreover, the strong admissibility condition is applied in our direct solver and we introduce connections between equivalences in adjacent levels to make the matrix nested. Therefore, we obtained a linear complexity for constant rank problems. It can be seen that our method is purely algebraic and does not involve the expansion of the original kernel function. The direct solver can be extended to different kinds of kernels efficiently due to kernel independence of the algorithm.

Briefly speaking, each far interaction is decomposed into three matrices in an algebraic way, e.g., the radiation matrix, the receiving matrix and the coupling matrix (i.e., translation matrices). The receiving matrix and the radiation matrix are also termed as the cluster basis in an $\mathcal{H}^2$-matrix structure. Transfer matrix is introduced to introduce the nested property between the child level and the parent level. At non-leaf levels, equivalences instead of skeletons are constructed to compress the far interactions. Specifically, for direct solution with only skeletonization factorization, the projection matrices are computed for each group, respectively, and the coupling matrices (i.e., skeletonized matrices) are computed for each interaction among all the groups. This is not the same for mixed-form NESA, as the constructed equivalences are located at the same relative positions for each group and are of equal quantity. Thus, for interactions with the same relative position relationship of two groups, we only need to compute the coupling matrices once. In addition, for non-leaf levels, the transfer matrices are required for all the groups with only skeletonization factorization while with mixed-form NESA, at most eight transfer matrices corresponding to eight relative positions from child equivalences to parent equivalences are required for each level in an octree tree. Therefore, the equivalence strategy exhibits higher computational efficiency compared to skeletonization factorization. Specifically, we construct equivalences at non-leaf levels for reasons mentioned above while still retain the skeletonization scheme at the leaf level. This is due to the fact that there are fewer basis functions in groups at the leaf level. Therefore, the matrix ranks of these groups are generally lower than those at the non-leaf levels. Although one can also use the equivalences approximation at the leaf level, the algorithm cannot gain a compression at this level. In contrast, the skeletonization scheme is more efficient at the leaf level. Comprehensively considering the above analysis, we propose a mixed algorithm with both skeletons and equivalences scheme to adapt to the compression of matrices at different levels. After the compressions for all the far interactions are obtained, in the direct solution, we construct two kinds of elimination matrices for factorizing the system matrix. We first eliminate the far field submatrices by conducting multiplications of the elimination matrix to those far field submatrices, then a partial LU decomposition is conducted to eliminate the near field submatrices hierarchically in a bottom-up traversal form. During the factorization, the Schur complement is computed with fill-in blocks, which may destroy the data-sparse representation of the algorithm. We modify the original radiation and receiving matrices to account for the contributions of these fill-in blocks while still preserve the data-sparse representation and nested property of our proposed direct solver.

The rest of the article is organized as follows. In Sect. II, the background and related works of this paper are introduced. In Sect. III, details of the factorization process with mixed-form NESA are given and the algorithm complexity is also analyzed theoretically. In Sect. IV, numerical results are presented to validate the accuracy and the complexity of the proposed method. Finally, a brief conclusion is given in Sect. V.

II. Background

With the strong admissibility condition, all the self-interactions and near interactions are computed as they are by MoM, while the far interactions of non-adjacent groups with fairly lower ranks are approximately represented by low-rank

compression methods. In the mixed-form NESA, we select skeletons from the original basis functions with the adaptive cross approximation (ACA) at the leaf level and construct equivalences at non-leaf levels. Supposing the mixed-form NESA is divided into L levels and the Lth level is the leaf level, the low-rank compression can be expressed by the following equation:

$$Z_{O,S} = Z_{O,\sigma_O} Z^{\dagger}{}_{\tau_O,\sigma_O} Z_{\tau_O,\tau_S} Z^{\dagger}{}_{\sigma_S,\tau_S} Z_{\sigma_S,S} = U_{O,\tau_O} D_{\tau_O,\tau_S} V_{\tau_S,S} \begin{cases} \tau = \text{skeletons, for } l = L; \\ \tau = \text{equivalences, for } l \neq L. \end{cases} \quad (1)$$

where σ is the test points set, τ is the skeleton set for the leaf level and equivalence set for non-leaf levels; O denotes the *observation* group and S denotes the *source* group. U_{O,τ_O} is the receiving matrix, $V_{\tau_S,S}$ is the radiation matrix and they are transposed to each other in electric field integral equations (EFIE). D_{τ_O,τ_S} is the translation matrix between skeletons or equivalences. l denotes the present level ranging from 1 to L.

Different from a typical H^2-matrix structure built on a binary tree, the mixed-form NESA is constructed on an octree and the corresponding matrix compression process of the leaf level is illustrated in the 2-dimension form in Fig. 1(a).

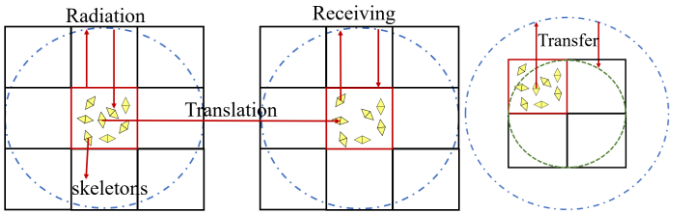


Fig. 1. (a) Mixed-form NESA low-rank representation at the leaf level. The red boxes denote the original RWG basis functions in groups and yellow RWG basis functions denote the selected skeletons. The blue dashed circles denote the test points which are used to represent the far-field electric field from the group in terms of the radiation process. (b) Transfer process where the child skeletons at the leaf level are projected onto the parent equivalences denoted by the green dashed circle.

Here, we only give the leaf level compression process and non-leaf levels are with similar process except that skeletons are replaced by equivalences. In other words, the algorithm is smoothly transferred from skeletons to equivalences at higher levels as shown in Fig. 1(b). Here, the proxy surface is applied to represent all the far field (including current level and all parents' levels) for skeleton selection to avoid inaccurate and redundant degree of freedoms. These skeletons are effective for all the far region and we can set a predetermined threshold of ACA to control the compression accuracy. The transfer process is performed where the child skeletons are projected onto the parent equivalences placed on a sphere surface to reconstruct the field radiated by the skeletons. The test points for all levels are distributed around each group on an outer sphere surface, representing the far field region for each group. The interactions of groups at a level with the same positional relationship share the same compressions. The motivation of equivalences can be explained by the fact that the memory for far interactions can be reduced as well as to preserve the multiscale property of the solver. Alternatively, the use of ID and proxy surface, as done in [38], could be a good choice for skeletons selections which would construct an almost optimal set of skeletons.

To demonstrate the improvement with the mixed skeleton and equivalence algorithm and explain the reason for choosing the leaf level as the skeleton level, we present the numerical results with respect to the memory for far interactions below in Table I.

TABLE I.
COMPUTATION STATISTICS WITH MEMORY AND TIME FOR FAR INTERACTIONS FOR THE SIMULATION OF A SPHERE WITH DIFFERENT SKELETON AND EQUIVALENCE LEVELS.

Skeleton/Equivalence levels	Memory (MB)	Compression time (s)	Matrix multiplication time (s)
0/3	560	56	0.92
1/2	1504	220	0.72
2/1	1781	251	0.68
3/0	1908	258	0.67

It can be seen from the table that with pure equivalence, the memory and time for constructing low rank approximation is significantly reduced compared with those involving skeleton levels. However, the matrix multiplication time of the equivalence-only algorithm is higher. Furthermore, the memory for the compression also increases with the number of skeleton levels. It is worth noting that, different from the iterative methods, in the direct solver the system matrix is represented explicitly and apart from the matrix compression, we also need to merge sub-blocks into higher levels and conduct the elimination process. Therefore, smaller matrix dimensions at higher levels can bring higher computational efficiency. The reason for setting the leaf level as the skeleton level is that generally coupling matrices which will be merged to the next level are smaller compared with using equivalences as can be seen from the faster matrix multiplication. As a tradeoff, this is beneficial for a faster factorization and inversion, while also preserve a relatively low memory consumption.

In the direct solution scheme, the system matrix is expressed in an explicit way. For submatrices in a system matrix, the inadmissible blocks are formed by self and near interactions while admissible blocks are formed by far interactions. To give a simple illustration, a system matrix divided into two levels with eight groups of a rectangular metal strip is shown in Fig. 2(a) where admissible and inadmissible blocks are marked in green and red respectively. The group indexes for each level is shown in Fig. 2(b). It is worth noting that indexes are not arranged in order and they correspond to the actual physical positions along the metal strip, for the consideration that all the possible situations arising during the matrix factorization process can be illustrated.

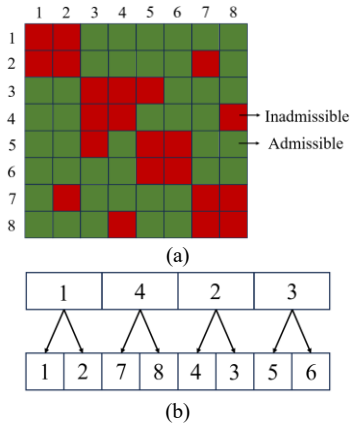


Fig. 2. (a) Illustration of the system matrix at the leaf level. (b) Index arrangement in an octree structure with eight groups for constructing a system matrix in (a). The numbers at different positions denote the indexes of each physical group of the metal strip.

III. Factorization with Mixed-form NESA

After the compressions of all the admissible blocks at each level are obtained, we can apply the following factorization algorithm hierarchically through a one-way bottom-up traversal. Details of the factorization will be elaborated in the following. We take the matrix in Fig. 2(a) as an example to illustrate the overall direct solution procedure. As receiving and radiation matrices are transposes of each other in EFIEs, we only take the receiving matrix for illustration.

A. Leaf Level Factorization

The factorization starts from the leaf level where skeletons are selected for the low-rank compression. All the G (G being the number of groups) groups are factorized following two main steps for admissible and inadmissible blocks sequentially.

The first step is to factorize the admissible blocks. According to the threshold we preset to the SVD, we truncate the left and right singular vectors as well as the singular values and the matrix to be compressed can be represented by the multiplication of the singular vectors and corresponding singular values. Taking group i=1 (i=1,2,…,G) as an example, we first perform an singular value decomposition (SVD) via LAPACK routines for the receiving matrix U_1 whose dimension is $m_1 \times k_1$, m_1 and k_1 being the number of unknowns and skeletons respectively to obtain a unitary matrix U_1^{left}. Thus, U_1 is orthogonalized and can be approximately represented by the first several singular vectors of U_1^{left}, whose singular values, normalized by the maximum one, are no less than the prescribed tolerance ε_1, so we have

$$(U_1)_{m_1 \times k_1} \approx (\tilde{U}_1)_{m_1 \times \tilde{k}_1} (R_1)_{\tilde{k}_1 \times k_1} \tag{2}$$

where $\tilde{U}_1$ is termed as the new receiving matrix for group 1 whose dimension is $m_1 \times \tilde{k}_1$, $\tilde{k}_1$ being the number of selected singular vectors of U_1. R_1 is the remaining factor resulting from SVD and it will be multiplied to the original coupling matrices with group 1 as

$$(\tilde{D}_{1,j})_{\tilde{k}_1 \times \tilde{k}_j} = (R_1)_{\tilde{k}_1 \times k_1} (D_{1,j})_{k_1 \times k_j} (R_j^T)_{k_j \times \tilde{k}_j} \tag{3}$$

where $D_{1,j}$ is the original coupling matrix between group 1 and j and R_j is the remaining factor to be generated resulting from SVD of U_j. The SVD conducted here can be used to generate the elimination matrices for admissible blocks as well as to reduce the basis dimension, which is beneficial for improving the matrix-matrix multiplication efficiency [37]. We then construct the block diagonal elimination matrix $Q_{1,sys}$ with system matrix size whose diagonal blocks are all identity matrices except for block 1 being Q_1. Q_1 is obtained directly from U_1^{left} by permuting $\tilde{U}_1$ and other singular vectors $\tilde{U}_1^{\perp}$ in U_1^{left} which are orthogonal to $\tilde{U}_1$ (i.e., $Q_1 = \left(\tilde{U}_1^{\perp} \quad \tilde{U}_1\right)$). It is worth noting that Q_1 is also a unitary matrix. We then multiply the system matrix with $Q_{1,sys}^H$ from the left and $\bar{Q}_{1,sys}$ from the right:

$$Z'_{sys} = Q_{1,sys}^H Z_{sys} \bar{Q}_{1,sys} \tag{4}$$

where $Q_{1,sys}^H$ and $\bar{Q}_{1,sys}$ denote the conjugate transpose and conjugate of $Q_{1,sys}$. This multiplication thus eliminates the first $m_1 - \tilde{k}_1$ rows and columns of admissible blocks associated with group 1 due to the orthogonality of $\tilde{U}_1^{\perp}$ and $\tilde{U}_1$. The elimination process of admissible blocks for the rest of the groups is similar and will be elaborated in the following context. The resulting system matrix is shown in Fig. 3.

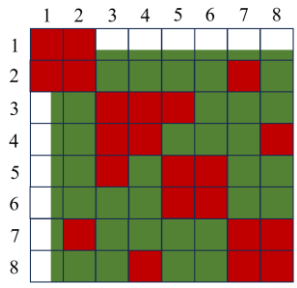


Fig. 3. System matrix after the first elimination for admissible blocks of group 1. All the admissible blocks associated with group 1 are reduced due to their low-rank property while only the inadmissible block with group 2 and the diagonal block of group 1 remain to be factorized.

In addition, this operation further reduces the receiving matrix dimension so that we only keep the minimal rank required by the prescribed accuracy tolerance.

Subsequently, in the second step, we need to factorize the inadmissible blocks associated with group 1. We perform an LU factorization where we use the upper left part with dimension $(m_1 - \tilde{k}_1) \times (m_1 - \tilde{k}_1)$ of the diagonal block (1,1) as the pivot block to eliminate the other blocks with group 1. During the factorization, the Schur complement is generated and added to different parts in the system matrix. Here we use a general form of Schur complement computation for illustration and this process can be explained by

$$\begin{pmatrix} E & F & G \\ H & A & B \\ M & C & D \end{pmatrix} = L \begin{pmatrix} E' & 0 & G' \\ 0 & I & 0 \\ M' & 0 & D' \end{pmatrix} U \tag{5}$$

where I is the identity matrix and

$$L = \begin{pmatrix} I & FU_A^{-1} & 0 \\ 0 & L_A & 0 \\ 0 & CU_A^{-1} & I \end{pmatrix}, \; U = \begin{pmatrix} I & 0 & 0 \\ L_A^{-1}H & U_A & L_A^{-1}B \\ 0 & 0 & I \end{pmatrix} \tag{6}$$

where A is viewed as the pivot block $A = L_A U_A$, in addition,

$$E' = E - FU_A^{-1}L_A^{-1}H \tag{7}$$

$$G' = G - FU_A^{-1}L_A^{-1}B \tag{8}$$

$$M' = M - CU_A^{-1}L_A^{-1}H \tag{9}$$

$$D' = D - CU_A^{-1}L_A^{-1}B \tag{10}$$

are the Schur complements and the second terms in (7) to (10) on the right are the dense fill-in blocks generated during the factorization. Therefore, two elimination matrices can be defined and we have

$$L^{-1} \begin{pmatrix} E & F & G \\ H & A & B \\ M & C & D \end{pmatrix} U^{-1} = \begin{pmatrix} E' & 0 & G' \\ 0 & I & 0 \\ M' & 0 & D' \end{pmatrix} \tag{11}$$

where

$$L^{-1} = \begin{pmatrix} I & -FU_A^{-1}L_A^{-1} & 0 \\ 0 & L_A^{-1} & 0 \\ 0 & -CU_A^{-1}L_A^{-1} & I \end{pmatrix}, \; U^{-1} = \begin{pmatrix} I & 0 & 0 \\ -U_A^{-1}L_A^{-1}H & U_A^{-1} & -U_A^{-1}L_A^{-1}B \\ 0 & 0 & I \end{pmatrix} \tag{12}$$

It is worth noting that there are no E, F, G, H and M matrices for the first group, where the system matrix is divided into a two by two structure. They only appear starting from the second group. We again multiply the two matrices from the left and right to the system matrix as illustrated in (11). This will zero out the first $m_1 - \tilde{k}_1$ rows and columns of inadmissible blocks with group 1 as shown in Fig. 4

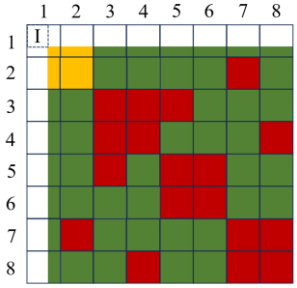


Fig. 4. System matrix after the second elimination for inadmissible blocks of group 1. All the inadmissible blocks with group 1 are reduced by LU factorization. The sub-block at the pivot position becomes an identity matrix and inadmissible blocks added by the fill-in blocks are colored in yellow.

For group 1, fill-in blocks are only added on the inadmissible blocks, i.e., the yellow parts in Fig. 4, so we perform the additions on the blocks directly. However, in the subsequent factorizations of other groups, fill-in blocks may arise in admissible blocks that are represented as low-rank approximations. They cannot be added directly in order to preserve the data-sparse representation. Therefore, we keep the fill-in blocks and store them with corresponding blocks until the elimination process reaches to the group where the fill-in blocks are located. Then, the fill-in blocks associated with that group will be compressed to a low-rank approximation form. This situation occurs when the inadmissible blocks are eliminated for group 3.

The above factorization process is carried out sequentially for each group. Assuming that the factorization reaches to group 3 and now the inadmissible blocks with group 3 are eliminated. The system matrix after the elimination is shown in Fig. 5. Four red boxes are marked in the figure to indicate different parts of Schur complement corresponding to (7) to (10). It is worth noting that actually in this step fill-in blocks only appear in the matrix block D as presented in (5), as there are no inadmissible blocks formed before group 3.

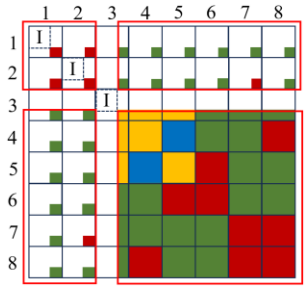


Fig. 5. System matrix after second elimination for inadmissible blocks of group 3. All the blocks with group 1 and 2 are reduced by the factorization. The admissible blocks added by the fill-in blocks are colored in blue. Four red boxes are marked to indicate different parts of matrix in Schur complement, the top left and top right denote E and G, the bottom left and bottom right denote M and D.

It is observed that the fill-in blocks (i.e., $F_{4,5}$ and $F_{5,4}$) are added to the admissible blocks colored in blue formed by elimination of $Z_{4,3}$, $Z_{5,3}$, $Z_{3,4}$ and $Z_{3,5}$. So, the receiving matrix of group 4 and 5 will be updated as $\hat{U}_4$ and $\hat{U}_5$ to take into account the contribution of the fill-in blocks. In other words, what we expected is to obtain the following expression:

$$\hat{U}_4 \hat{D}_{4,5} \hat{V}_5 \approx \tilde{U}_4 \tilde{D}_{4,5} \tilde{V}_5 + F_{4,5} \tag{13}$$

where $\hat{U}_4$, $\hat{D}_{4,5}$ and $\hat{V}_5$ are the updated forms of $\tilde{U}_4$, $\tilde{D}_{4,5}$ and $\tilde{V}_5$. Details of the updating procedure will be elaborated in Sect. III.B. Equation (12) can be viewed as the low-rank compression form of $Z_{4,5}$ which is added by the fill-in block $F_{4,5}$. Then, for group 4, $\hat{U}_4$ instead of $\tilde{U}_4$ is used for construction of elimination matrix for the corresponding admissible blocks (i.e., $Q_4 = \begin{pmatrix} \hat{U}_4^{\perp} & \hat{U}_4 \end{pmatrix}$). It can be inferred that all the Q_i (i=1,2,…,n) are unitary matrices. Therefore, the elimination accuracy for this step is actually related to the accuracy of (13).

From the above two steps of factorization, it is observed that the elimination for admissible blocks involves approximation errors while the elimination for inadmissible blocks is an exact computation without any approximation. The approximation errors of admissible blocks will be

discussed in Sect. III.B and presented with the error controllability of our algorithm in the accuracy analysis in Sect. IV.A.

The same elimination operation is sequentially performed on the remaining groups until all the groups at the leaf level are factorized.

B. Compression of Fill-in Blocks

As mentioned above, fill-in blocks are the dense matrices generated during the computation of the Schur complement. If we add them directly to admissible blocks and treat the resultant blocks as inadmissible blocks, the number of inadmissible blocks will continue to increase during the factorization. This is intolerable and will lead to redundant computations at higher levels. Therefore, we need to compress the fill-in blocks added to admissible blocks to form an updated low-rank representation of the resultant matrices.

Taking group i as an example, in order to keep the nested representation in mixed-form NESA, we retain the original receiving matrix $\tilde{U}_i$ as part of the updated receiving matrix to be constructed. However, representing the resultant blocks with $\tilde{U}_i$ accurately is not guaranteed as the column space of the fill-in blocks may not be completely covered by that of $\tilde{U}_i$. Note that $\tilde{U}_i^{\perp}$ and $\tilde{U}_i$ are orthogonal and complementary, in other words, they together form a complete space. So, we only need to find the column space of the fill-in blocks in $\tilde{U}_i^{\perp}$ which is not in $\tilde{U}_i$. To do so, we add up all the Gram matrices of the fill-ins associated with group i in a row and multiply the resultant sum by the Gram matrix of $\tilde{U}_i^{\perp}$ from the left and its complex conjugate transpose from the right. The resultant matrix multiplication is then conducted by an SVD truncated with another prescribed tolerance $\varepsilon_{fill-in}$ as:

$$\tilde{U}_i^{\perp}(\tilde{U}_i^{\perp})^H(\sum_{k=1}^{n_i}F_{i,j_k}F_{i,j_k}^H)\tilde{U}_i^{\perp}(\tilde{U}_i^{\perp})^H \overset{SVD}{\approx} U_i^{aug}S(U_i^{aug})^H \quad (14)$$

where n_i is the number of fill-in blocks associated with group i in a row, j is the column index of the corresponding fill-in block and the left singular vectors U_i^{aug} is selected as the augmented part of $\tilde{U}_i$. It is worth noting that for the previous parts, we take the changed fill-in blocks instead of original ones into (14) under the consideration that the previous eliminations have already been carried out accurately and there is no need to eliminate these previous parts in the fill-in blocks repeatedly. This strategy should enable a better capture for the column space of the remaining fill-in blocks, as it may reduce the influence brought by the already eliminated parts of the fill-in blocks. The prescribed tolerance $\varepsilon_{fill-in}$ here is referred to the direct solution algorithm tolerance. It is related to the solution error compared with MoM. We can control the error to the expected level by reducing $\varepsilon_{fill-in}$, which will be elaborated later in the numerical results. Finally, we obtain $\hat{U}_i=\begin{pmatrix}\tilde{U}_i & U_i^{aug}\end{pmatrix}$ as the updated receiving matrix for group i. It can be inferred that U_i^{aug} and $\tilde{U}_i$ are also orthogonal, therefore the rank of $\hat{U}_i$ is the minimal one required by accuracy for representing all the fill-in blocks associated with group i.

It is worth noting that if multiple fill-in blocks arising from previous eliminations are added to a same admissible block, they are summed up and stored in a single block with the admissible block. This matrix addition process actually corresponds the superposition of the space contained in each fill-in block.

C. Non-leaf Level Factorization

Before the algorithm proceed to the next level, we merge and permute the remaining sub-blocks to form a new system matrix which is also represented in a nested structure. This process is shown in Fig. 6.

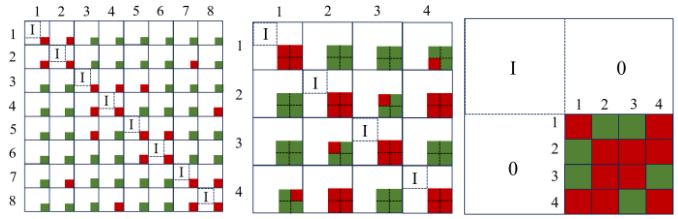


Fig. 6. The system matrix after all the groups at the leaf level are factorized. (a) The system matrix before merging and permuting. (b) The system matrix after merging the dense remaining inadmissible blocks and coupling matrices at the leaf level to form matrices to be factorized at the next level. (c) The system matrix after permuting all the sub-blocks to be gathered together, forming a new nested structure at the parent level.

In this merging and permuting process, the skeletons at the leaf level are regrouped into the coarser non-leaf level for further compression. At the leaf level, receiving matrices are all eliminated. The factorization at non-leaf levels is similar to that at the leaf level, except that the receiving matrices are all replaced by the modified transfer matrices as can be seen in the following.

As the system matrix is represented in a nested form with mixed-form NESA, at non-leaf levels transfer matrices play the same role as receiving matrices at the leaf level. The receiving matrix of group i at the l-1th level can be represented by its children at the lth level expressed as:

$$U_i^{l-1}=\begin{pmatrix}U_{i1}^l & & \\ & \ddots & \\ & & U_{i8}^l\end{pmatrix}\begin{pmatrix}T_{i1}^l \\ \vdots \\ T_{i8}^l\end{pmatrix}=\begin{pmatrix}\hat{U}_{i1}^l & & \\ & \ddots & \\ & & \hat{U}_{i8}^l\end{pmatrix}\begin{pmatrix}\begin{pmatrix}R_{i1}T_{i1}^l \\ 0\end{pmatrix} \\ \vdots \\ \begin{pmatrix}R_{i8}T_{i8}^l \\ 0\end{pmatrix}\end{pmatrix} \quad (15)$$

The numbering up to 8 in (15) corresponds to an octree structure where a parent group is divided into 8 child groups. Due to the augment of the receiving matrices at the lth level, the transfer matrices are also augmented by appending zeros to ensure the consistency between the dimension of transfer

matrices and the corresponding receiving matrices. Therefore, the transfer matrix of group i is modified as

$$\hat{T}_i = \begin{pmatrix} \begin{pmatrix} R_{i1}T_{i1}^l \\ 0 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} R_{i8}T_{i8}^l \\ 0 \end{pmatrix} \end{pmatrix} \tag{16}$$

Coupling matrices are also updated due to the augment of the receiving matrices as well as the addition of fill-in blocks. Before this, they are multiplied by the factors resulting from the SVD as mentioned in (3).

For admissible without fill-in blocks, their coupling matrices are augmented by readily appending zeros from $\tilde{D}_{i,j}$ to $\hat{D}_{i,j} = \begin{pmatrix} \tilde{D}_{i,j} & 0 \\ 0 & 0 \end{pmatrix}$ similar to what is done for transfer matrices. For admissible blocks with fill-in blocks, the corresponding coupling matrices are augmented and further updated by adding the fill-in part. Taking block (i, j) as an example, as we have

$$\hat{U}_i \hat{D}_{i,j} \hat{V}_j + F_{i,j} = \hat{U}_i D'_{i,j} \hat{V}_j \tag{17}$$

so the coupling matrix of block (i, j) is finally updated by

$$D'_{i,j} = \hat{D}_{i,j} + (\hat{U}_i)^H F_{i,j} (\hat{V}_j)^H \tag{18}$$

This expression can also be explained by the following matrix multiplication diagram shown in Fig. 7, where $(\hat{V}_j)^H = \bar{\hat{U}}_j$ due to the transpose relationship between the receiving and radiation matrices. The column and row space of $F_{i,j}$ are included in corresponding updated receiving matrix $\hat{U}_i$ and $\hat{U}_j^T$, so the multiplication of $F_{i,j}$ with the complementary part in the elimination matrices marked in blue becomes zero.

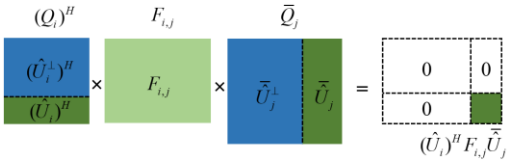


Fig. 7. Fill-in block multiplication with elimination matrices. The multiplication between the light green fill-in block and the dark green updated receiving matrix results in the sub-block remained after the elimination while the multiplication with the blue complementary part becomes zero.

So far, the matrices required for further factorization at the next level are prepared. All the inadmissible blocks are formed by merging the dense remaining blocks and coupling matrices at the child level while for admissible blocks elimination, same operations are performed on the updated transfer matrices. Accordingly, at higher non-leaf levels, the equivalences will be regrouped into their parent levels until the factorization reaches to the root level.

D. Aggregation and Augment Overflow Treatment

From the aforementioned factorization process, we can notice that there may be exceptional circumstances between the leaf level and its parent level. As we set the number of equivalences to be equal for all non-leaf levels, the quantitative relationship between equivalences and skeletons at the leaf level is indeterminate. So, when skeletons are aggregated to the next level, the total number of skeletons in a group at the next level may be less than the equivalences. This is termed as the aggregation overflow in our proposed method. Consequently, the elimination process for the group cannot proceed because the row dimension (corresponding to equivalences) is larger than the column dimension (corresponding to skeletons) for the matrices in the row associated with the group. For this situation, we do not perform any of the elimination or updating, but only keep the matrices with the group as they are. They are not processed until the factorization reaches a level that satisfies the condition for the elimination process where the column dimension is larger than the row dimension. This treatment is actually reasonable as this exceptional circumstance occurs only when there are very few basis functions in a group, so there is no need to eliminate the group to accelerate the computation. Once the elimination of a group is conducted, all its parent groups at all parent levels will be eliminated due to the fact that the total number of equivalences of the child groups are necessarily more than that of their parent group.

Similar to the aggregation overflow, the augment overflow occurs when an augmented receiving matrix has an equal column dimension to the row dimension. The different point is that the corresponding group has already been eliminated by the first step for admissible blocks and cannot proceed to the second step for inadmissible blocks, as the dimension of the diagonal block to be factorized is zero. Therefore, we skip the second elimination step and start factorizing the next group directly.

The occurrence of the above overflow circumstances also shows the difference between the skeletonization scheme and the combined skeletons and equivalences scheme of mixed-form NESA. For the skeletonization scheme, the skeletons are selected from the original basis functions, thus are necessarily no more than the total number of original basis functions for all group at all levels. The aforementioned treatment can be explained more clearly in the following pseudocode in Sect. III.E.

E. Multilevel Factorization and Inversion

The factorization mentioned above continues in a way of bottom-up traversal until we reach the top level, i.e., level 1. Finally, we apply the LU factorization to the matrix remained at the lower right diagonal at level 1. The overall multilevel factorization can thus be represented by

$$Z_{sys} = \left\{ \prod_{l=L}^{1} \left[\left(\prod_{i=1}^{m^l} R_i^l \right) P_l^T \right] \right\} Z_{diag}^1 \left\{ \prod_{l=1}^{L} \left[P_l \left(\prod_{i=m^l}^{1} S_i^l \right) \right] \right\} \tag{19}$$

where m^l is the number of groups at the lth level, $R_i^l = Q_{i,sys}^l L_i^l$

and $S_i^l = U_i^l (Q_{i,sys}^l)^T$ are the two elimination matrices for group i at the lth level. L_i^l and U_i^l are the L and U factors arising from (6). P_l is the permuting matrix for the lth level as explained in Fig. 6. Z_{diag}^1 is the remaining diagonal matrix at level 1 which is further factorized by $Z_{diag}^1 = L^1 U^1$.

Consequently, the inverse of the system matrix resulted from the factorization can be expressed as

$$Z_{sys}^{-1} = \left\{ \prod_{l=L}^{1} \left[\left(\prod_{i=1}^{m^l} (S_i^l)^{-1} \right) P_l^T \right] \right\} (Z_{diag}^1)^{-1} \left\{ \prod_{l=1}^{L} \left[P_l \left(\prod_{i=m^l}^{1} (R_i^l)^{-1} \right) \right] \right\} \quad (20)$$

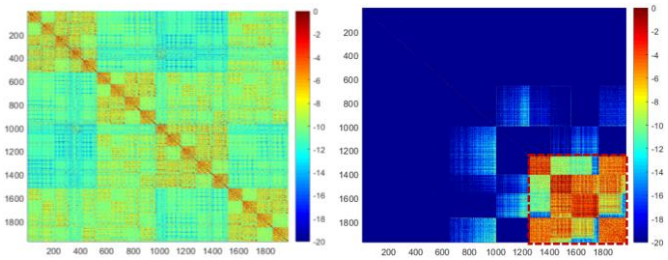
where $(R_i^l)^{-1} = (L_i^l)^{-1} (Q_{i,sys}^l)^H$ and $(S_i^l)^{-1} = \bar{Q}_{i,sys}^l (U_i^l)^{-1}$, as $Q_{i,sys}^l$ are unitary matrices for all groups. The pseudocode of the aforementioned is presented in Algorithm 1.

Algorithm 1 Proposed direct solution with mixed-form NESA

```
for level l = L to 1 do
  for group i = 1 to n do
    if aggregation overflow occurs then
      skip and cycle to the next group
    end if
    if (i==1) then
      Orthogonalize U_1^l to get Ũ_i^l
    else
      Orthogonalize and update U_i^l to get Û_i^l
      if augment overflow occurs then
        skip and cycle to the next group
      end if
    end if
    Obtain elimination matrix Q_i^l
    Perform elimination for admissible blocks
    Obtain elimination matrices L_i^l and U_i^l
    Perform elimination for inadmissible blocks
    Update remaining near and far interactions blocks
  end for
  Update coupling matrices
  Update transfer matrices
  Merge and permute sub-blocks into a coarser level
end for
Perform an LU factorization to the final diagonal block.
```

In summary, the proposed direct solution can be generally divided into three steps. The first step is to factorize the admissible blocks. The basic idea is to capture the column space occupied by the admissible blocks together with their fill-in blocks, and then construct the corresponding complementary matrix to eliminate the redundant parts. This step involves receiving matrix updating with approximation errors which are controlled by the prescribed tolerance $\varepsilon_{fill-in}$ set for the SVD. In the second step, we factorize the remaining inadmissible blocks. Schur complement is computed and fill-in blocks arise during the elimination. This step is a precise factorization without any approximations. Finally, the system matrix is decomposed into a matrix containing only an identity matrix and a diagonal matrix, and we perform an LU factorization for the diagonal matrix. Therefore, the inverse of the system matrix can be obtained by the multiplication of a series of matrices generated from the matrix factorization process.

For a more intuitive understanding of the factorization process, we construct an actual system matrix based on a rectangular metal strip, but divided into only one level with four groups instead of two levels. The groups are indexed based on the same way in the root level shown in Fig. 2(b). The system matrix before and after the elimination and permuting are shown in Fig. 8(a) and Fig. 8(b) respectively, where most parts of the system matrix is eliminated. The intermediate elimination processes for each group are shown in Fig. 9.



(a) (b)

Fig. 8. The system matrix (a) before and (b) after the elimination and permuting, where the red dashed box marks the matrix to be factorized by the final LU factorization. The matrix elements are taken as absolute values in dB.

F. Time and Memory Complexity Analysis

In this section, we analyze the theoretical time and memory complexity of the overall algorithm in details. We assume that the object being analyzed is discretized with N unknowns, which are divided into the total number of L levels. Each level has m_l non-empty groups and each group has an average of n_i near groups and f_i far groups. p_i and k_l are the average number of basis functions in the groups and selected skeletons at the leaf level or equivalences at the non-leaf levels. As for equivalences, k_l is a constant set to 50 which is independent with the number of unknowns [34]. The computational time complexity for each level can be expressed by

$$C_l = \sum_{i=1}^{m_l} n_i p_i^3 + (p_i - k_l)^3 + n_i^2 p_i^2 (p_i - k_l) + f_i k_l^3 \quad (21)$$

where the first term denotes the first elimination step for admissible blocks, as the actual computation of this step lies in the inadmissible blocks with the near groups. The second term

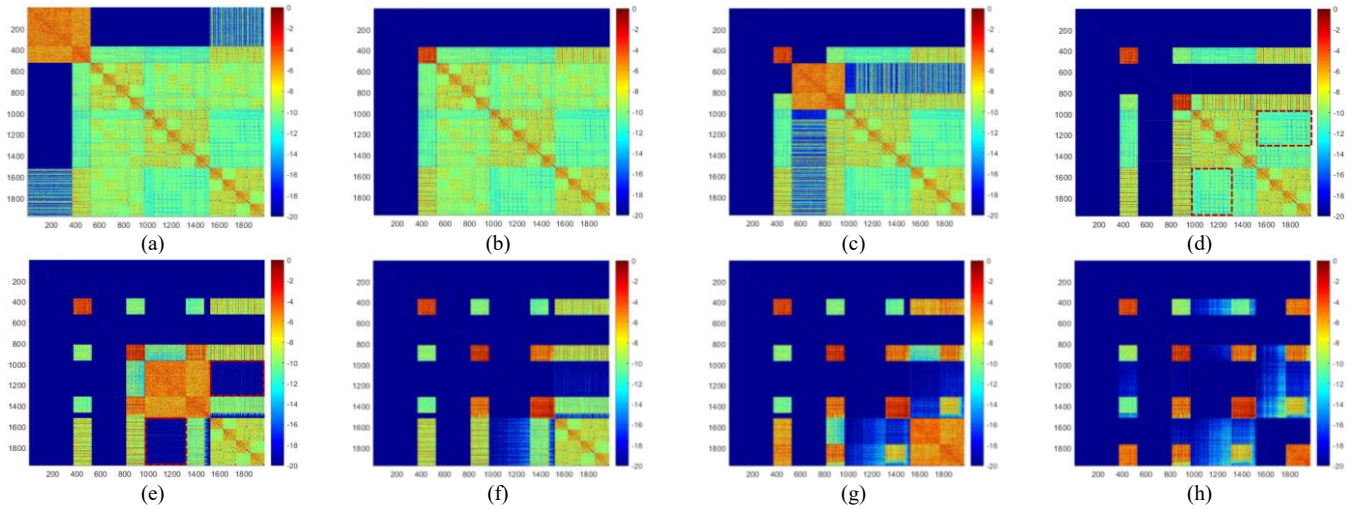


Fig. 9. The system matrix after each intermediate elimination for each group. The elimination process proceeds sequentially from (a) to (h). Eliminations for admissible blocks are presented in (a), (c), (e) and (g) while eliminations for inadmissible blocks are presented in (b), (d), (f) and (h). The fill-in blocks added to admissible blocks are generated in (d) marked by the red boxes, which are then eliminated in (e).

denotes the LU factorization for the corresponding diagonal block of size $p_i - k_l$ for the second elimination step, where Schur complement is generated after this step. The third term denotes the computation of the dense fill-in blocks and the last term is the computation of the updated coupling matrices. Therefore, we have

$$\begin{aligned} C_l &= O(m_l n_i p_i^3 + m_l (p_i - k_l)^3 + m_l n_i^2 p_i^2 (p_i - k_l) + m_l f_i k_l^3) \\ &= O(m_l n_i p_i^3) \end{aligned} \tag{22}$$

As p_i is no less than k_l, we can infer that the maximum term in (22) is the first term. Therefore, we only keep the first term $m_l n_i p_i^3$ for analyzing the complexity. The above computational complexity is then analyzed for different levels. Here we use C with different subscripts to indicate the computational complexities at the corresponding levels. For the leaf level, it is worth noting that p_i is bounded by a constant number that we preset to control the number of levels, and n_i is also bounded by a maximum value in an octree. Thus, the computational complexity at this level is

$$C_L = O(m_L n_i p_i^3) = O(m_L) O(n_i p_i^3) = C \cdot O(m_L) = O(N) \tag{23}$$

where N refers to the number of unknowns. As p_i and n_i are both bounded by constants, then the term $n_i p_i^3$ is bounded by a constant. Hence, the complexity is only related to m_L which is linear to N. Subsequently at non-leaf levels except for the root level, skeletons are regrouped into coarser levels and the computational complexity for these are

$$C_{l=L-1\sim 2} = \sum_{l=L-1}^{2} [m_l n_l (8k_{l+1})^3] = O(N) \tag{24}$$

where $p_l = 8k_{l+1}$ and k_{l+1} is less than or equal to the constant value of equivalences. It is worth noting that L scales as $O(\log N)$ and m_l scales as $O(8^l)$ in an octree. Therefore, $C_{l=L-1\sim 2}$ scales as $O(N)$. Finally, for the root level, we only computer an LU factorization for a matrix of size $8k_2$, whose complexity is

$$C_1 = O(p_1^3) = O(k_2^3) = O(C) \tag{25}$$

where C is a constant. Combining the above complexities for all levels, we conclude that the computational time complexity is

$$C = O(N) + O(N) + O(C) = O(N) \tag{26}$$

Then the memory complexity is analyzed. Obviously, the near field elements at the leaf level scales as $O(N)$. Then, we need to store the receiving matrix for each group at the leaf level while only for each pair of position relationship at non-leaf levels. The memory $M_{cluster}$ for receiving matrices is expressed as

$$\begin{aligned} M_{cluster} &= \sum_{i=1}^{m_L} p_L k_L + \sum_{l=L-1}^{1} \sum_{i=1}^{8} p_l k_l = O(m_L p_L k_L) + \sum_{l=L-1}^{1} O(8k_l^2) \\ &= O(N) \end{aligned} \tag{27}$$

In addition, because the coupling matrices are updated, we need to store them for each group across all levels as

$$M_{coupling} = \sum_{l=L}^{1} \sum_{i=1}^{m_l} k_l^2 = \sum_{l=L}^{1} O(m_l k_l^2) = O(N) \tag{28}$$

Therefore, the overall memory complexity is $O(N)$. In summary, the computational time complexity and the memory complexity are both $O(N)$ for constant rank problems. These constant rank problems are usually defined in low working frequencies. When the frequency is high, the kernel function oscillates violently with the distance and the field reconstruction is no longer accurate when test points are distributed on a sphere. This field reconstruction at high frequencies requires a directional distribution rather than an omnidirectional distribution for test points. Therefore, we limit our solution for objects with electric size below 6λ (λ being the wavelength), known as constant rank problems. The above analysis will be verified by numerical results in the following section.

IV. Numerical Results

In the following, we present several numerical examples to verify the accuracy and efficiency of the algorithm. All the computations are performed on a work station with 256 GB of RAM and Intel(R) Xeon(R) Silver 4114 CPU at 2.20 GHz or a personal computer with the 11th Gen Intel(R) Core(TM) i7-11700F at 2.50GHz. To construct a hierarchical octree, we set the threshold of the average number of unknowns of the leaf groups to control the octree levels. In fact, choosing a proper threshold is a tradeoff between the internal matrix–matrix multiplication and the external number of computation cycles for groups and levels, and both excessively large or small thresholds will reduce the computational performance. In the following examples we set the threshold to 200 under the condition that the number of equivalences is set to 50 with 150 RWG basis functions.

A. Accuracy

We first analyze the bistatic radar cross section (RCS) of a perfect electric conductor (PEC) sphere obtained by the direct solver. The radius of the sphere is 1.8 m and it is illuminated by a plane wave with a frequency of 300 MHz. The result is compared with the Mie Series analytical solution as shown in Fig. 10. The surface current distribution is also plotted in the inset in Fig. 10.

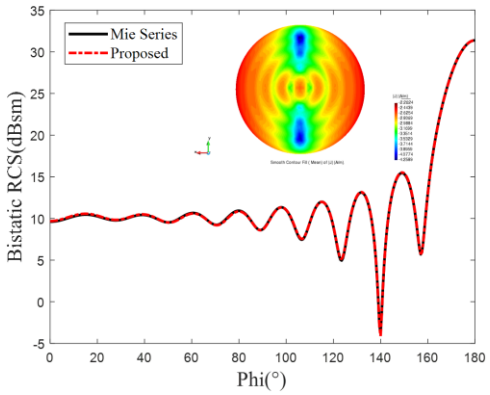


Fig. 10. Bistatic RCS of a sphere with radius of 1.8 m and working frequency of 300 MHz obtained by the proposed direct solution and Mie Series. The inset is the surface current distribution of the PEC sphere.

It is observed that the bistatic RCS obtained by our proposed method agrees well with the analytical solution. Furthermore, we study the error controllability of the direct solver. The error controllability of low rank compression for mixed-form NESA has been analyzed in detail in [34] and [35], so here we just focus on the error controllability of the direct solution algorithm. As stated in Sect. III.A, the approximation errors occur in the elimination process for admissible blocks where fill-in blocks are compressed by the SVD. In addition, the tolerance for skeleton selection with ACA also involves approximation errors. Thus, we change the tolerance $\varepsilon_{fill-in}$ of the SVD employed in the fill-in blocks compression with different tolerances of ACA for the skeleton selection and test the current of the PEC sphere mentioned above. We compare the current obtained by our method with that obtained by MoM solved with the brute force LU factorization. The current relative l_2-norm error is used and evaluated by $\|x_{NESA}-x_{LU}\|_2/\|x_{LU}\|_2$ as shown in Fig. 11(a). The condition number of this problem is 135. To test the error-controllability of our method for more ill-conditioned problems, we have added a more ill-conditioned problem of a multiscale combined cone structure with non-uniform meshes as a comparison shown in Fig. 11(b). The condition number of the multiscale combined cone problem is 44497.

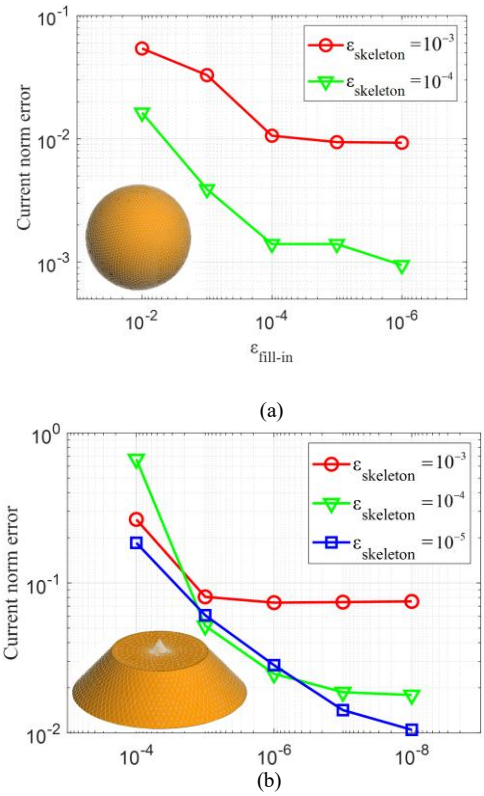


Fig. 11. The current norm error compared to the brute force LU factorization with different ACA tolerances for skeleton selection and SVD tolerances for fill-in blocks compression of (a) the sphere and (b) the multiscale combined cone structure.

For the sphere structure, the tolerance of ACA for skeleton selection is set to 10^{-3} and 10^{-4}, and $\varepsilon_{fill-in}$ is set from 10^{-2} to 10^{-6}. We observe that the current error keeps decreasing when we reduce the algorithm tolerance $\varepsilon_{fill-in}$. In addition, we obtain lower errors with the lower skeleton selection tolerance. It can be inferred that the proposed direct solution is error controllable. For the multiscale combined cone structure, the error curve indicates that the proposed method is also error controllable for such ill-conditioned matrix. Nevertheless, different from the sphere example in Fig. 11(a) which is well-conditioned, the accuracy is deteriorated and this multiscale structure requires smaller threshold for both skeleton selection via ACA and fill-in blocks compression via SVD to obtain accurate results. This is reasonable, as in the direct solver there are strict inversions conducted for diagonal block

matrices via LU factorizations. Therefore, the condition number is also a key factor when selecting thresholds to ensure the reliability of the computational results. As long as the condition number meets the requirement for accurate inversions via LU factorizations, the solution obtained by our method can be trusted.

Therefore, from the error curves it is observed that 10^{-3} is suitable for most cases and smaller values as 10^{-6} may be required for ill-conditioned problems (e.g., multiscale models with non-uniformly meshes). It should be noted that more precise tolerances for both skeleton selection and fill-in blocks compression will cause a tiny increase of the computation time and memory, but have no impact on the computational complexity.

B. Computational Time and Memory Complexity

Subsequently, we test the computational time and memory complexity of the proposed direct solver with the aforementioned PEC sphere. We fix the mesh size to be 0.1 λ and set the frequency to 300 MHz, 600 MHz, 1.2 GHz and 2.4 GHz corresponding to 12735, 51432, 206742 and 828153 unknowns. The computational time and memory are shown in Fig. 12.

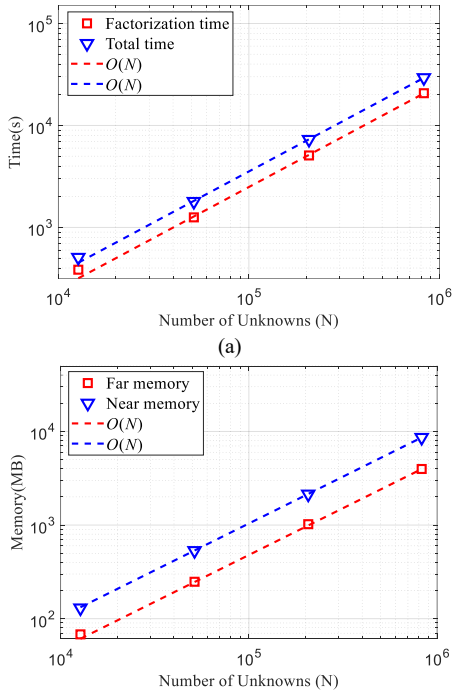


Fig. 12. The computational time and memory complexity. (a) Factorization and total time (including factorization and inversion) with the number of unknowns. (b) Memory for storing the near and far interactions with the number of unknowns.

It is observed that both time and memory scale linearly with the number of unknowns. In addition, we find that the factorization time occupies the majority of the total time. This indicates one of the advantages of direct solvers over iterative solvers in terms of MRHS problems because whether for MRHS problems or single RHS problem, the system matrix and its factorization remain unchanged. Once the matrix factorization is finished, the inversion involving matrix-matrix multiplication or matrix-vector multiplication is very fast.

C. Applications

Finally, we give some RCS and current results of realistic models compared with existing methods. We first investigate the electromagnetic scattering from a tank model which is 4.00 m in length, 1.94 m in width and 1.12 m in height. The working frequency is 300 MHz and the incident wave direction is $\theta_i = 0°$, $\varphi_i = 0°$. The number of RWG unknowns is 14604 with the average mesh size of 0.1λ. Three levels of mixed-form NESA are employed with the leaf level group size of 0.25λ. The bistatic RCS with the range of $\varphi = 0°$, $\theta = 0° \sim 180°$ obtained by MoM solved with the brute force LU factorization and by the proposed method with fill-in blocks compression tolerance $\varepsilon_{fill-in}$ of 10^{-3} is shown in Fig. 13.

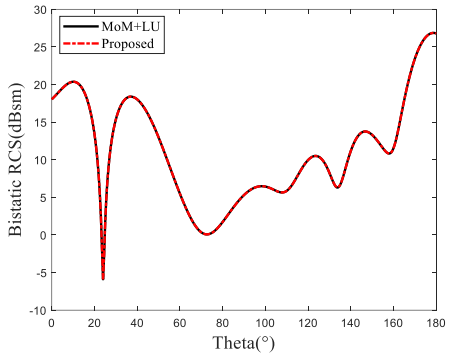
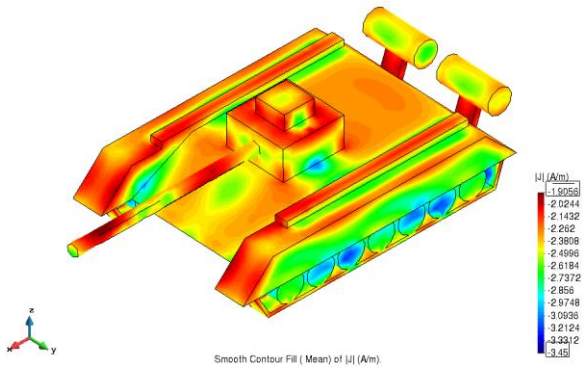
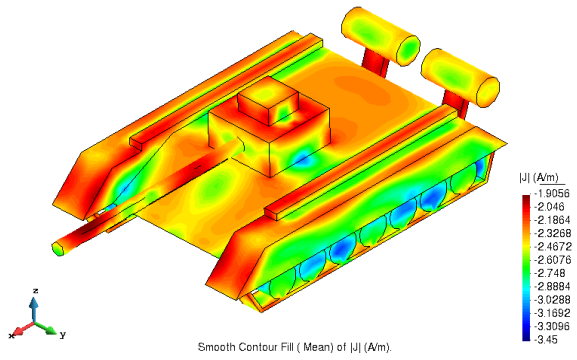


Fig. 13. Bistatic RCS of a tank obtained by the MoM with an LU factorization inversion and by the direct solution with mixed-form NESA.

A good agreement is observed for the two methods. The surface current distribution obtained by the methods mentioned above is plotted in Fig. 14, where the corresponding current error measured by the relative l_2-norm error is 0.015.



(a)



(b)

Fig. 14. Surface current of the tank model obtained by (a) MoM + LU factorization and (b) mixed-form NESA direct solution. The current measured in dB is plotted. The color bars for the two figures are unified.

Then a simulation of a fighter jet model is conducted. The working frequency is 800 MHz and the approximate dimension of the model in wavelength is $5.6\lambda\times1.5\lambda\times4.0\lambda$. The EM wave is illuminating from $\theta_i=90°$, $\varphi_i=180°$. The bistatic RCS ranging in $\theta=0°\sim180°$, $\varphi=180°$ obtained by the proposed direct solution with different algorithm tolerances $\varepsilon_{fill-in}$ and by the MoM is presented in Fig. 15.

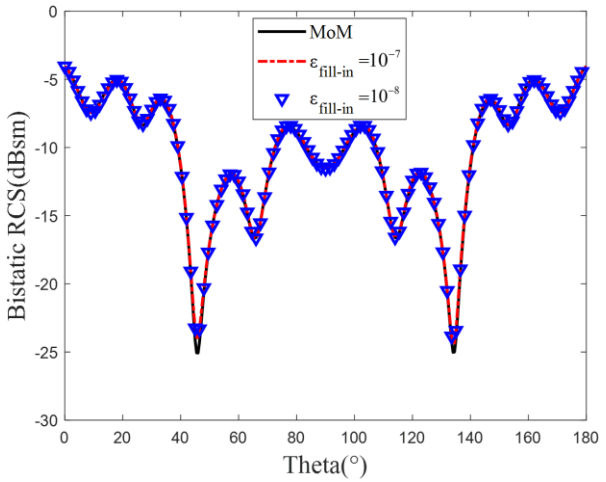
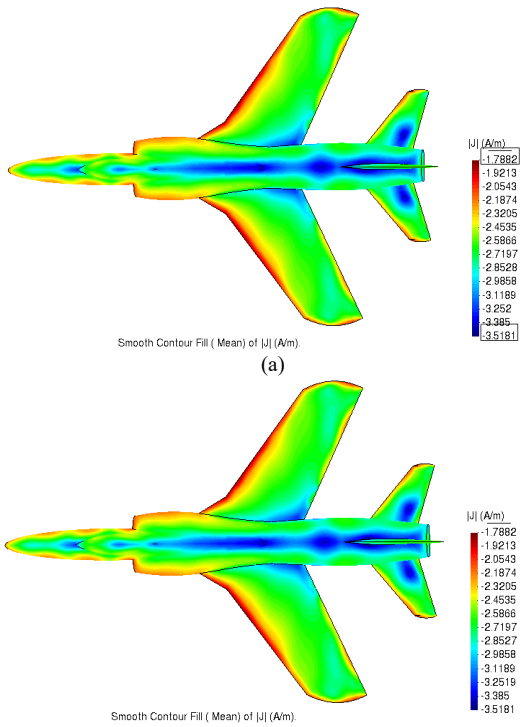


Fig. 15. Bistatic RCS of the fighter model obtained by the MoM and by the direct solution with mixed-form NESA with different algorithm tolerances.

We also plot the surface current corresponding to different $\varepsilon_{fill-in}$ in Fig. 16, where a good agreement can be observed and the relative l_2-norm current error is 0.038.



(a)

(b)

Fig. 16. Surface current of the fighter model obtained by the proposed direct solution with algorithm tolerance of (a) 10^{-7} and (b) 10^{-8}. The color bars for the two figures are unified.

Furthermore, we computed the monostatic RCS of the fighter jet model within the range of $\theta_i=90°, \varphi_i=0°\sim360°$ as shown in Fig. 17 obtained by the MoM, the proposed direct solver and the work proposed in [35]. We compared the computational statistics of our proposed method with the method in [35] as presented in Table II. The relative residual for the iterative solution is 3×10^{-3} and both methods are performed on 8 cores for a better comparison. The average number of iterations for all directions is 61. It is worth noting that for the time comparison we only present the solution time for the system matrix while the matrix filling time for the near interaction and low rank approximation time for the far interaction is the same and not included.

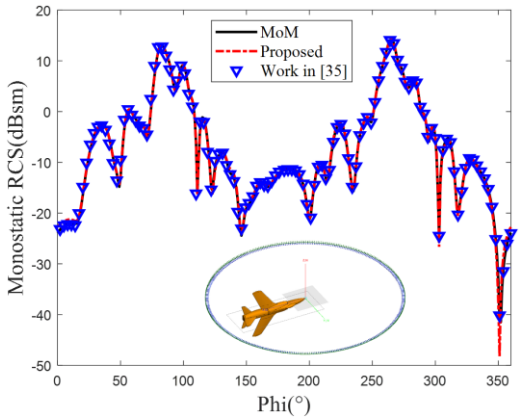
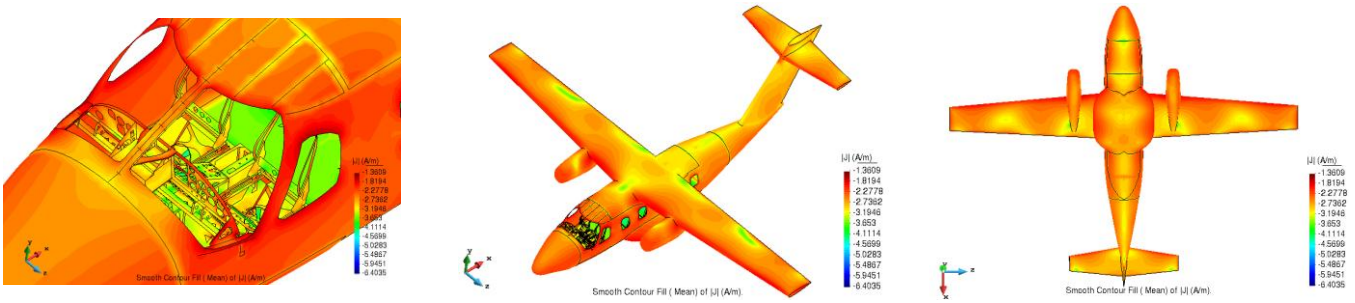


Fig. 17. Monostatic RCS of the fighter model.



(a) (b) (c)

Fig. 18. Surface current plotted in dB of the morphed EV55 aircraft. (a) Details of the cockpit including the dashboard and the seats. (b) Top view. (c) Bottom view.

TABLE II
COMPUTATIONAL STATISTICS COMPARISON FOR THE FIGHTER MODEL WITH DIFFERENT METHODS

Method	Factorization time(h)	Inversion time(h)	Total solution time(h)	Memory(MB)
Work in [35]			96.9	1536.9
Proposed	0.5	0.8	1.3	2046.6

It can be concluded that the proposed direct solver is much more efficient than the iterative methods in case of MRHS problems. Compared with the iterative solution, the memory of the proposed method has been increased due to the additional storage requirements for fill-in blocks. However, there is a limit to the additional fill-in memory which will not increase with the number of unknowns, because the number of near interactions for each group at every level has an upper limit, and once the factorization at a level is finished the fill-in memory at this level can be deallocated. In other words, only additional fill-in memory of one level is required during the whole factorization.

In addition, to test the proposed direct solver for objects with complex structures, we investigate a morphed EV55 aircraft at 100 MHz. The aircraft is 14.2 m long with a wingspan of 16.1 m, corresponding to 4.73λ and 5.37λ. The aircraft is illuminated by a plane wave from $\theta_i = 90°$, $\varphi_i = 180°$ along the positive x-axis direction impinging to the nose of the aircraft. We refine the average mesh size to 0.02λ, resulting 171 763 unknowns. Five-level mixed-form NESA is employed with an average number of 54 unknowns for each group at the leaf level. The surface current distribution obtained by the proposed method is given in Fig. 18(a) to 18(c). Then the bistatic RCS calculated by the work in [35] and by the proposed method is presented in Fig. 19.

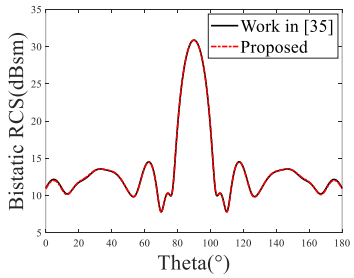


Fig. 19. Bistatic RCS of the EV55 aircraft obtained by the method in [35] and by our method.

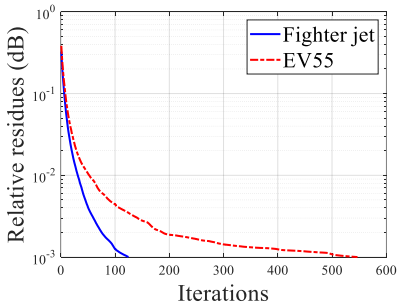


Fig. 20. Relative residues with iterations of the fighter jet example and the EV55 example.

To better show that the EV55 example is more ill-conditioned, we analyze the relative residues with iterations of the previous fighter jet example and the EV55 example as shown in Fig. 20. Under the condition that the relative residues set to 3×10^{-3} and 1×10^{-3} respectively, the numbers of iterations are 142 and 547 for EV55 while 61 and 125 for the fighter jet. This indicates that our method is also valid for more ill-conditoned problems. We further analyze the bistatic computational statistics. For the memory consumption, storing the near elements and the far compressions matrices costs 3604.6 MB and 85.3 MB with the work in [35]. With our method, the near and far memory are the same while 117.6 MB additional memory is required for the fill-in blocks. In terms of the time consumption, we estimate the statistics of the computation for the single incident angle. The total solution time is 15.0 hours for 547 iterations with the work in [35], while with our proposed direct solver the total solution time is 20.5 hours corresponding to 20.4 hours factorization time and 0.1 hour inversion time. Although for the single incident direction the proposed solver exhibits a slower speed mainly due to the factorization process, the inversion process is significantly fast. Therefore, for more incident directions the proposed solver should be far more efficient as the factorization time remains unchanged while only the inversion time will increase with the number of incident directions.

Finally, to demonstrate the computational performance of the proposed method, we simulate a sphere array model shown in Fig. 21(a) with 2,271,276 unknowns at 300MHz. The electrical size of the array is 4.3λ. The incident direction is $\theta_i = 0°$, $\varphi_i = 0°$ and we comput the bistatic RCS with the range of $\theta = 0° \sim 360°$, $\varphi = 0°$ by our method and MLFMA shown in Fig. 21(b).

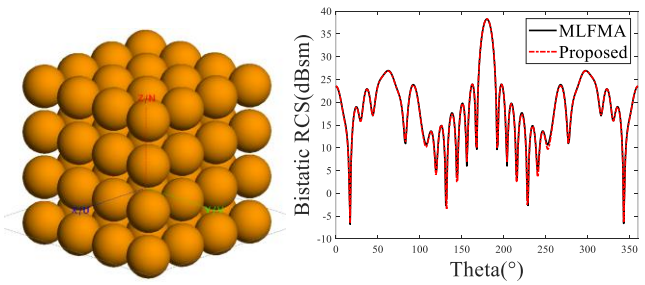


Fig. 21. (a) The geometric structure of the sphere array and (b) the bistatic RCS results by our method and MLFMA.

A good agreement is observed from the figure. The estimated peak memory of all applied processes for the MLFMA is 128.2GB. In terms of the computational time, assuming that a monostatic RCS involving 361 incident directions is calculated, the estimated time considering all the processes for the MLFMA is 200.9 hours while 104.4 hours for the proposed method. Although additional memory is required for the matrix factorization, the solution time is reduced compared with iterative methods for problems with multiple incident directions.

From the computational statistics of the simulations above, it can be concluded that with the proposed direct solver, although some limited additional memory is required due to the matrix factorization, significant time saving can be achieved for solving MRHS problems.

V. Conclusion and Perspectives

In this paper, a fast-direct solver based on mixed-form NESA is proposed to solve EM scattering from conducting objects for constant rank problems. The mixed strategy of equivalences and skeletons is employed to obtain a higher compression efficiency. In the fast-direct solution, the system matrix is represented in an explicit way. Elimination matrices for far and near interactions are introduced sequentially so that the system matrix is factorized into a series of matrix multiplications. The receiving matrices and coupling matrices are updated due to the elimination for fill-in blocks generated from the Schur complement computation. As for constant rank problems, we limit the electric size of objects been analyzed by our direct solution to be smaller than 6λ and both the computational time and memory complexities are linear. When the target size grows larger, the computational complexity remains unchanged but the accuracy deteriorates. In order to obtain accurate results at higher frequencies, wideband NESA (WNESA [39]) should be applied where we put equivalences into different directions under the consideration that the Green's function is smooth and compressible when observation is limited to a specific direction [40]. However, this will lead to a higher complexity because equivalences are constructed for each direction and the number of directions increases with levels. In addition, the skeletons only algorithm in [11] is also error controllable with linear complexity for electrical small problems. Compared with it, the proposed method requires less memory and is more efficient when at higher levels with equivalences. The kernel-independent property of the compression method enables the solver to be applicable to other integral equations Several numerical examples verify the accuracy and efficiency of the proposed direct solver.

As an extension of this paper, some follow-up works include a direct solver working in a wideband which is also valid for problems at higher frequencies.

References

[1] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," IEEE Trans. Antennas Propag., vol. 30, no. 3, pp. 409-418, May 1982.

[2] L. Greengard and V. Rokhlin, "A fast algorithm for particle simula tions," J. Comput. Phys., vol. 73, pp. 325–348, Dec. 1987.

[3] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," IEEE Trans. Antennas Propag., vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[4] J. R. Phillips and J. K. White, "A precorrected-FFT method for elec trostatic analysis of complicated 3-D structures," IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., vol. 16, no. 10, pp. 1059–1072, Oct. 1997.

[5] S. M. Seo and J.-F. Lee, "A fast IE-FFT algorithm for solving PEC scattering problems," IEEE Trans. Magn., vol. 41, no. 5, pp. 1476–1479, May 2005.

[6] M. Bebendorf, "Approximation of boundary element matrices," Numer. Math., vol. 86, no. 4, pp. 565–589, Jun. 2000.

[7] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approxi mation algorithm for accelerated method of moments computations of EMC problems," IEEE Trans. Electromagn. Compat., vol. 47, no. 4, pp. 763–773, Nov. 2005.

[8] E. Michielssen and A. Boag, "A multilevel matrix decomposition al gorithm for analyzing scattering from large structures," IEEE Trans. Antennas Propag., vol. 44, no. 8, pp. 1086–1093, Aug. 1996.

[9] J.M.Rius, J.Parrón, A.Heldring, J.M.Tamayo and E.Ubeda, "Fast iterative solution of integral equations with method of moments and matrix decomposition algorithm-singular value decomposition," IEEE Trans. Antennas Propag., vol. 56, no. 8, pp. 2314–2324, Aug. 2008.

[10] Z. Rong, M. Jiang, Y. Chen, L. Lei, Z. Nie, and J. Hu, "Fast direct surface integral equation solution for electromagnetic scattering analysis with skeletonization factorization," *IEEE Trans. Antennas Propag.*, vol. 68, no. 4, pp. 3016-3025, April 2020.

[11] Z. Rong, M. Jiang, Y. Chen, L. Lei, X. Yang, and J. Hu, "Strong admissibility skeletonization factorization for fast direct solution of electromagnetic scattering from conducting objects," *IEEE Trans. Antennas Propag.*, vol. 69, no. 10, pp. 6607-6617, Oct. 2021

[12] M. Ma and D. Jiao, "Accuracy Directly Controlled Fast Direct Solution of General $\mathcal{H}^2$ -Matrices and Its Application to Solving Electrodynamic Volume Integral Equations," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 1, pp. 35-48, Jan. 2018

[13] M. Ma and D. Jiao, "Direct Solution of General $\mathcal{H}^2$ -Matrices With Controlled Accuracy and Concurrent Change of Cluster Bases for Electromagnetic Analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 6, pp. 2114-2127, June 2019.

[14] Y. Liu, H. Guo, and E. Michielssen, "An HSS matrix-inspired butterfly based direct solver for analyzing scattering from two-dimensional objects," *IEEE Antennas Wireless Propag. Lett.*, vol. 16, pp. 1179–1183, 2017.

[15] 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, Sept. 2017.

[16] 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. Microw. Theory Techn.*, vol. 63, no. 3, pp. 897-912, March 2015.

[17] Z. Jiang, Y. Sheng and S. Shen, "Multilevel Fast Multipole Algorithm-Based Direct Solution for Analysis of Electromagnetic Problems," *IEEE Trans. Antennas Propag.*, vol. 59, no. 9, pp. 3491-3494, Sept. 2011.

[18] X. Chen, C. Gu, Z. Li and Z. Niu, "Accelerated Direct Solution of Electromagnetic Scattering via Characteristic Basis Function Method With Sherman-Morrison-Woodbury Formula-Based Algorithm," *IEEE Trans. Antennas Propag.*, vol. 64, no. 10, pp. 4482-4486, Oct. 2016.

[19] Bowei Wu, Min Hyung Cho, "Robust fast direct integral equation solver for three-dimensional doubly periodic scattering problems with a large number of layers," *J. Comput. Phys.*, vol 495, pp. 112573, Dec. 2023

[20] T. Wan and M. Xie, "H-Matrix-Based Direct Solver of JMCFIE for the Analysis of Scattering From Penetrable Objects," *IEEE Trans. Antennas Propag.*, vol. 70, no. 9, pp. 8654-8659, Sept. 2022

[21] C. Liu, Y. Chen, Y. Ren, J. Hu and Z. Nie, "Fast Direct Solution of Electromagnetic Scattering From Metallic Objects in a Planarly Layered Medium," *IEEE Trans. Antennas Propag.*, vol. 72, no. 2, pp. 1779-1791, Feb. 2024.

[22] Jiaqing Lu, JinFa Lee, "A compression scheme for domain decomposition method in solving electromagnetic problems," *J. Comput. Phys.*, vol 503, pp. 112824, Apr 2024.

[23] J. Lu, "A Low-Rank Direct Solver for Nonconformal Domain Decomposition Methods," *IEEE Trans. Antennas Propag.*, vol. 72, no. 8, pp. 6663-6677, Aug. 2024.

[24] A. B. Manić, A. P. Smull, F. -H. Rouet, X. S. Li and B. M. Notaroš, "Efficient Scalable Parallel Higher Order Direct MoM-SIE Method With Hierarchically Semiseparable Structures for 3-D Scattering," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2467-2478, May 2017.

[25] S. Wang, X. S. Li, J. Xia, Y. Situ, and M. V. de Hoop, "Efficient scalable algorithms for solving dense linear systems with hierarchically semisep arable structures," *SIAM J. Sci. Comput.*, vol. 35, no. 6, pp. C519–C544, 2013.

[26] S. Chandrasekaran et al., "Some fast algorithms for sequentially semi separable representations," *SIAM J. Sci. Comput.*, vol. 27, no. 2, pp. 341–364, 2015.

[27] W. Hackbusch, "A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[28] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, vol. 64, pp. 21–47, Apr. 2000.

[29] S. Börm, L. Grasedyck, and W. Hackbusch, "Hierarchical matrices," *Lecture Note of the Max Planck Institute for Mathematics in the Sciences*, vol. 21. 2003.

[30] S. Börm, *Efficient Numerical Methods for Non-local Operators: H2-Matrix Compression Algorithms, Analysis*. Zurich, Switzerland: Eur. Math. Soc., 2010.

[31] W. Chai and D. Jiao, "Dense matrix inversion of linear complexity for integral-equation-based large-scale 3-D capacitance extraction," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 10, pp. 2404–2421, Oct. 2011.

[32] K. L. Ho and L. Greengard, "A fast direct solver for structured linear systems by recursive skeletonization," *SIAM J. Sci. Comput.*, vol. 34, no. 5, pp. A2507–A2532, Jan. 2012.

[33] K. L. Ho and L. Ying, "Hierarchical interpolative factorization for elliptic operators: Integral equations," *Commun. Pure Appl. Math.*, vol. 69, no. 7, pp. 1314–1353, 2015.

[34] M. Li, M. A. Francavilla, F. Vipiana, G. Vecchi, and R. Chen, "Nested equivalent source approximation for the modeling of multiscale structures," *IEEE Trans. Antennas Propag.*, vol. 62, no. 7, pp. 3664-3678, July 2014

[35] M. Li, M. A. Francavilla, D. Ding, R. Chen and G. Vecchi, "Mixed-Form Nested Approximation for Wideband Multiscale Simulations," *IEEE Trans. Antennas Propag.*, vol. 66, no. 11, pp. 6128-6136, Nov. 2018.

[36] Z. Rong et al., "Fast Direct Solution of Integral Equations with Modified HODLR Structure for Analyzing Electromagnetic Scattering Problems," *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3288-3296, May 2019.

[37] C. Yang and D. Jiao, "Nested Reduction Algorithms for Generating a Rank-Minimized $\mathcal{H}^2$ -Matrix from FMM for Electrically Large Analysis," *IEEE Trans. Antennas Propag.*, vol. 69, no. 7, pp. 3945-3956, July 2021.

[38] Minden V, Ho K L, Damle A and L Ying, "A recursive skeletonization factorization based on strong admissibility" *Multiscale Modeling & Simulation*, vol. 15, pp. 768-796, 2017.

[39] M. Li, M. A. Francavilla, R. Chen and G. Vecchi, "Wideband Fast Kernel-Independent Modeling of Large Multiscale Structures Via Nested Equivalent Source Approximation," *IEEE Trans. Antennas Propag.*, vol. 63, no. 5, pp. 2122-2134, May 2015.

[40] B.Engquist and L. Ying, "Fast directional multilevel algorithms for oscillatory kernels," *SIAM J. Sci. Comput.*, vol. 29, no. 4, pp. 1710–1737, 2007.

[41] F. Vipiana, P. Pirinoli, G. Vecchi, "Spectral properties of the EFIE-MoM matrix for dense meshes with different types of bases", IEEE Trans. Antennas Propag., Vol. 55, No. 11, Nov. 2007, pp. 3229-3238.

[42] M. A. Echeverri Bautista, M. A. Francavilla, F. Vipiana, G. Vecchi, "A Hierarchical Fast Solver for EFIE-MoM Analysis of Multiscale Structures at Very Low Frequencies", IEEE Trans. Antennas Propag., Vol. 62, No. 03, March 2014, pp. 1523-1528.

Yuhan Zuo received the B.S. degree in electronic information engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2021. He is currently pursuing the double Ph.D. degree in electronic science and technology with the Nanjing University of Science and Technology and the Politecnico di Torino, Turin, Italy. His research interests include electromagnetic scattering analysis, computational electromagnetics, and fast integral equation solvers.

Mengmeng Li (M'15–SM'19) received the B.S. degree (Hons.) in physics from Huaiyin Normal College, Huai'an, China, in 2007, and the Ph.D. degree in electromagnetic field and microwave technology from the Nanjing University of Science and Technology, Nanjing, China, in 2014. From 2012 to 2014, he was a Visiting Student with the Electronics Department, Politecnico di Torino, Turin, Italy, and also with the Antenna and EMC Laboratory (LACE), Istituto Superiore Mario Boella, Turin, where he carried out fast solver for multiscale simulations. Since 2014, he has been with the Department of Communication Engineering, Nanjing University of Science and Technology, where he has been an Assistant Professor, Associate Professor, and Professor since 2020. In 2017, he was a Visiting Scholar with Pennsylvania State University, Pennsylvania, PA, USA. His current research interests include computational electromagnetics for electromagnetic fields and scattering, space-time-modulated array design, and automatic target recognition.

Dr. Li was a recipient of the National Science Fund for Excellent Young Scholars in 2022, the Young Scientist Award at the ACES-China Conference in 2019, the Doctoral Dissertation Award of Jiangsu Province in 2016, and ten

student paper/contest awards at the international conferences with the students. He is an active reviewer for many IEEE journals and conferences. He is an Associate Editor of the IEEE Antennas and Propagation Magazine, IEEE Open Journal of Antennas and Propagation (OJAP), and a Guest Editor of OJAP.

Francesca Vipiana (M'07–SM'13) received the Laurea and Ph.D. degrees in electronic engineering from the Politecnico di Torino, Torino, Italy, in 2000 and 2004, respectively, with doctoral research carried out partly at the European Space Research Technology Center, Noordwijk, The Netherlands. From 2005 to 2008, she was a Research Fellow with the Department of Electronics, Politecnico di Torino. From 2009 to 2012, she was the Head of the Antenna and EMC Laboratory, Istituto Superiore Mario Boella, Torino. Since 2012, she has been an Assistant Professor with the Department of Electronics and Telecommunications, Politecnico di Torino, where she has been an Associate Professor since 2014 and a Full Professor since 2021.

Her main research activities concern numerical techniques based on integral equations and method of moments, with a focus on multiresolution and hierarchical schemes, domain decomposition, preconditioning and fast solution methods, advanced quadrature integration schemes, and analysis of glide-periodic structures. Moreover, her research interests include the modeling, design, realization and testing of microwave imaging and sensing systems for medical and industrial applications.

She received the Lot Shafai Mid-Career Distinguished Award from the IEEE Antennas and Propagation Society (AP-S) in 2017, she was an Associate Editor of the IEEE Transactions on Antennas and Propagation (2018-2024) and the founder and responsible of the Women in Engineering Column in the IEEE Antennas and Propagation Magazine (2019-2024). She is a member of the EurAAP Board of Directors and the Vice-Chair of the IEEE AP-S Expanding Collaboration & Engagement (ECE) committee.

Dazhi Ding (SM' 21) received the B.Sc. and Ph.D. degrees in electromagnetic field and microwave technique from the Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2002 and 2007, respectively. In 2005, he was with the Center of wireless Communication, City University of Hong Kong, Hong Kong, as a Research Assistant. He joined the Department of Electrical Engineering, NJUST, where he became a Lecturer in 2007. In 2014, he was promoted to Full Professor in NJUST, where he was appointed as the Head of the Department of Communication Engineering, in September 2014. He is the author or coauthor of over 30 technical articles. He has authored or coauthored more than 80 articles. His current research interests include computational electromagnetics and electromagnetic scattering and radiation. Dr. Ding was a recipient of the National Excellent Youth Fund by the National Science Foundation of China (NSFC) in 2020.