# Fast Low-Frequency Surface Integral Equation Solver Based on Hierarchical Matrix Algorithm

**Ting Wan[1, *], Qi I. Dai[2], and Weng Cho Chew[3]**

**Abstract**—A fast low-frequency surface integral equation solver based on hierarchical matrix algorithm is proposed. First, the augmented electric field integral equation (A-EFIE) formulation is introduced to eliminate the low-frequency breakdown of traditional EFIE. To deal with large-scale problems, the low-frequency multilevel fast multipole algorithm (LF-MLFMA) is employed to construct a hierarchical ($\mathcal{H}$-) matrix representation of the A-EFIE system matrix. Moreover, a recompression method is developed to further compress the $\mathcal{H}$-matrix generated by LF-MLFMA. The $\mathcal{H}$-matrix-based triangular factorization algorithm can be performed with almost linear computational complexity and memory requirement, which produces a fast direct solver for multiple right-hand-side (RHS) problems, and a good preconditioner to accelerate the convergence rate of an iterative solver. Numerical examples demonstrate the effectiveness of the proposed method for the analysis of various low-frequency problems.

## 1. INTRODUCTION

Electromagnetic modeling of micro-electronic structures becomes increasingly attractive because quasi-static analysis tools are inadequate as the frequency increases. The surface integral equation (SIE) method is a powerful tool for their electrodynamic analysis [1, 2]. However, traditional SIE methods have suffered from a well-known low-frequency breakdown problem [3]. This problem prevents the wide frequency band analysis in circuit simulation. Many research efforts have been carried out to tackle this problem, such as quasi-Helmholtz decomposition of basis functions [4–9] and Calderon preconditioner [10–13].

Among them, the augmented electric field integral equation (A-EFIE) method is one of the most effective methods to eliminate the low-frequency breakdown, which is also independent of basis functions [14–18]. Here, A-EFIE augments the EFIE with the current continuity equation, which offers an elegant way to renormalize the EFIE. Although the A-EFIE separates vector potential and scalar potential, the resulting system matrix is highly dense. For the solution of A-EFIE matrix equation, an iterative solver is often used because the underlying matrix-vector product (MVP) can be accelerated by fast algorithms such as the fast multipole algorithm (FMA) [15, 18–21]. Actually, the computational complexity of the iterative solver can be estimated to be $O(N_{rhs}N_{iter}N^2)$. Here, $N$ is the number of unknowns, $N_{rhs}$ denotes the number of right-hand-sides (RHSs), and $N_{iter}$ is the number of iterative steps needed for convergence. Obviously, an iterative solver becomes inefficient for the cases of ill-conditioned system matrices or multiple RHSs. Preconditioner can be used to improve the convergence rate, such as the popular constraint preconditioner [15]. In contrast, a direct solver does not suffer from the issue of slow convergence, and avoids redundant computations for multiple RHSs. However, a direct solver is generally infeasible for large-scale problems due to its extremely expensive computational costs.

[1] Department of Communication Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. [2] KLA-Tencor, Milpitas, California 95035, USA. [3] Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, Urbana, IL 61801, USA.

In this paper, an A-EFIE-based hierarchical ($\mathcal{H}$-) matrix factorization algorithm is developed to yield a fast direct solver as well as a good preconditioner of iterative solvers. $\mathcal{H}$-matrices provide a data-sparse way to approximate dense matrices arising from the discretisation of integral operators. The key idea is to represent certain sub-blocks of these dense matrices by low-rank approximations [22]. $\mathcal{H}$-matrix formatted arithmetics can significantly reduce the computational complexity of conventional matrix arithmetics [23–25]. $\mathcal{H}$-matrix-based solvers have been applied to the SIE method for electrodynamic analysis [26–29]. These solvers encounter low-frequency breakdown when dealing with low-frequency problems. Hence, we introduce the A-EFIE to perform a novel $\mathcal{H}$-matrix-based solver free of low-frequency breakdown. Based on the A-EFIE, a new hybrid hierarchical tree structure based on a current tree and a charge tree is first constructed. Then, the low-frequency multilevel fast multipole algorithm (LF-MLFMA) [30, 31] is employed to produce an $\mathcal{H}$-matrix representation of the A-EFIE system matrix. A recompression method is developed to further compress the resulting $\mathcal{H}$-matrix [25]. $\mathcal{H}$-matrix-based lower and upper (LU) triangular factorization algorithm can be implemented with almost linear computational complexity and memory requirement, which provides a direct solver as well as a preconditioner of iterative solvers for solving the low-frequency problems.

This paper is organized as follows: Section 2 summarizes the A-EFIE formulations. Section 3 introduces the $\mathcal{H}$-matrix construction and LU factorization algorithm based on the A-EFIE and LF-MLFMA. The $\mathcal{H}$-LU-based direct solver and preconditioner are further demonstrated with numerical results in Section 3. Section 4 concludes this paper.

## 2. AUGMENTED EFIE WITH DEFLATION

The EFIE formulation for an arbitrarily shaped PEC surface $S$ can be written as

$$i\omega\mu \int_S g(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')dS' - \frac{1}{i\omega\varepsilon}\nabla\int_S g(\mathbf{r},\mathbf{r}')\nabla'\cdot\mathbf{J}(\mathbf{r}')dS' = -\mathbf{E}^{inc}(\mathbf{r}) \tag{1}$$

However, the EFIE formulation breaks down for low-frequency problems due to the imbalanced spectrum branches [14]. A-EFIE is an effective remedy for this by introducing the current continuity equation into the EFIE formulation. The current continuity equation expresses the relationship between current and charge as

$$\nabla\cdot\mathbf{J}(\mathbf{r}) = i\omega q(\mathbf{r}) \tag{2}$$

Then, we discretize the surface of objects by a triangle mesh with $e$ inner edges and $p$ patches. After Galerkin's testing, the A-EFIE matrix equation with both current and charge can be constructed as

$$\begin{bmatrix} \bar{\mathbf{V}} & \bar{\mathbf{D}}^{\mathbf{T}} \\ \bar{\mathbf{P}}\cdot\bar{\mathbf{D}} & k_0{}^2\cdot\bar{\mathbf{I}} \end{bmatrix}\cdot\begin{bmatrix} ik_0\cdot j \\ c_0\cdot\bar{\mathbf{P}}\cdot\rho \end{bmatrix} = \begin{bmatrix} \eta_0{}^{-1}\cdot\mathbf{b} \\ 0 \end{bmatrix} \tag{3}$$

where $\bar{\mathbf{I}}$ is the identity matrix, and $j$ and $\rho$ denote the unknown current density coefficient and charge density coefficient, respectively. The entries of matrices $\bar{\mathbf{V}}$, $\bar{\mathbf{P}}$ and vector $\bar{\mathbf{b}}$ are defined as

$$[\bar{\mathbf{V}}]_{m,n} = \mu_r\cdot\int_S\mathbf{\Lambda}_m(\mathbf{r})\cdot\int_{S'}g(\mathbf{r},\mathbf{r}')\cdot\mathbf{\Lambda}_n(\mathbf{r}')dS'dS \tag{4}$$

$$[\bar{\mathbf{P}}]_{m,n} = \varepsilon_r^{-1}\int_S h_m(\mathbf{r})\int_{S'}g(\mathbf{r},\mathbf{r}')h_n(\mathbf{r}')dS'dS \tag{5}$$

$$\mathbf{b}_m = \int_S\mathbf{\Lambda}_m(\mathbf{r})\cdot\mathbf{E}^{inc}(\mathbf{r})dS \tag{6}$$

where $\mathbf{\Lambda}$ is the Rao-Wilton-Glisson (RWG) basis function for expanding the surface current, and $h$ is the pulse basis function for expanding the surface charge. $\bar{\mathbf{D}}$ is a sparse incidence matrix with

$$[\bar{\mathbf{D}}]_{m,n} = \begin{cases} 0, & \text{Patch } m\text{does not belong to edge } n \\ +1, & \text{Patch } m\text{is the positive part of edge } n \\ -1, & \text{Patch } m\text{is the negative part of edge } n. \end{cases} \tag{7}$$

The system matrix of Eq. (3) is denoted as $\bar{\mathbf{Z}}_0$ for short. $\bar{\mathbf{Z}}_0$ is singular at very low frequencies due to the charge neutrality. This singularity can be removed by reducing the degrees of freedom of

charge [15], but it destroys the simple pattern of matrix $\bar{\mathbf{P}}$. To preserve the appealing pattern, we remove the singularity by the deflation method, which modifies $\bar{\mathbf{Z}}_0$ to be

$$\bar{\mathbf{Z}} = \bar{\mathbf{Z}}_0 - \gamma \mathbf{a}\mathbf{a}^T \tag{8}$$

where $\gamma = \text{trace}(\bar{\mathbf{Z}}_0)/(e+p)$. Here, $\mathbf{a} \in \mathbb{R}^{(e+p)\times 1}$ is a normalized vector of

$$\mathbf{a} = \frac{1}{\sqrt{p}}(0 \ldots 0 \ 1 \ldots 1)^T \tag{9}$$

where the first $e$ elements are zero and the other $p$ elements are $1/\sqrt{p}$. Then, the A-EFIE system with deflation can be written as

$$\bar{\mathbf{Z}} \cdot \begin{bmatrix} ik_0 \cdot j \\ c_0 \cdot \bar{\mathbf{P}} \cdot \rho \end{bmatrix} = \begin{bmatrix} \eta_0^{-1} \cdot \mathbf{b} \\ 0 \end{bmatrix} \tag{10}$$

Although $\bar{\mathbf{D}}$ and $\bar{\mathbf{I}}$ are sparse, $\bar{\mathbf{Z}}$ is highly dense since $\bar{\mathbf{V}}$, $\bar{\mathbf{P}}$ and $\gamma \mathbf{a}\mathbf{a}^T$ are dense. Next, a fast solver based on $\mathcal{H}$-matrix algorithm will be introduced for the fast solution of (10).

## 3. $\mathcal{H}$-MATRIX FAST SOLVER BASED ON A-EFIE

The main process of the A-EFIE-based $\mathcal{H}$-matrix algorithm can be described as four steps: 1) Construct a hierarchical tree structure for the A-EFIE system matrix $\bar{\mathbf{Z}}$. 2) Generate an $\mathcal{H}$-matrix representation $\bar{\mathbf{Z}}_{\mathcal{H}}$ of $\bar{\mathbf{Z}}$ by the LF-MLFMA. 3) Perform the $\mathcal{H}$-matrix formatted LU factorization algorithm of $\bar{\mathbf{Z}}_{\mathcal{H}} = \bar{\mathbf{L}}_{\mathcal{H}} \bar{\mathbf{U}}_{\mathcal{H}}$. 4) Develop a direct solver or a preconditioned iterative solver by the $\mathcal{H}$-matrix formatted forward and backward substitution (FBS).
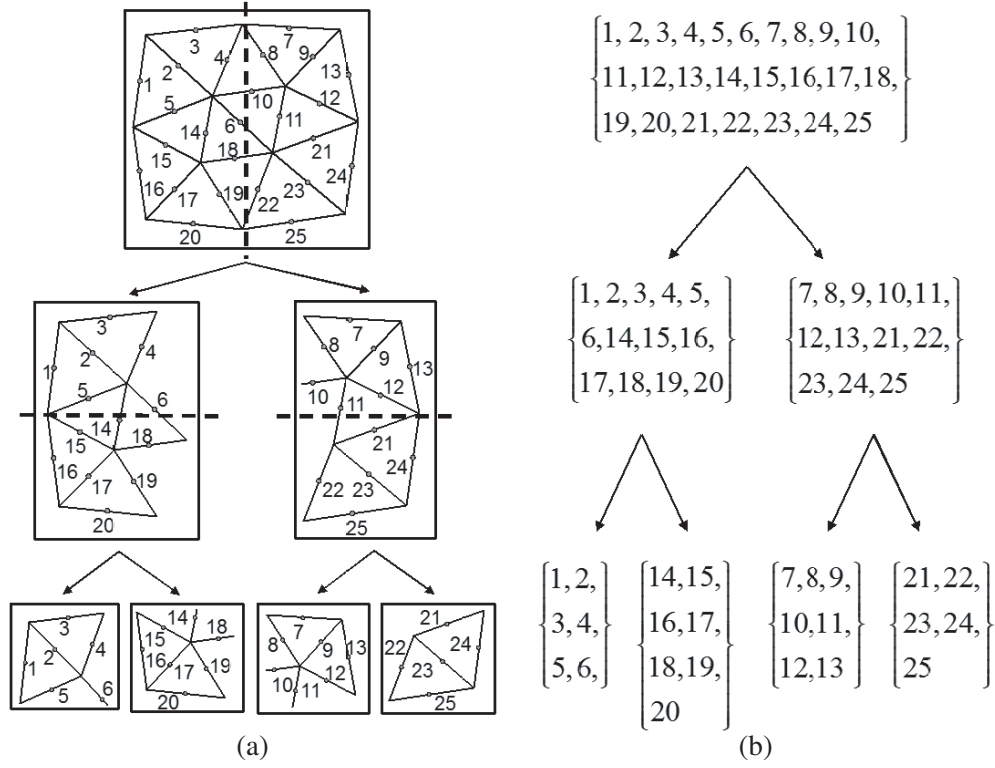
### 3.1. Construct a Hierarchical Tree

Hierarchical tree is the skeleton of an $\mathcal{H}$-matrix. A hierarchical tree is essentially a block cluster tree. To obtain a block cluster tree, one needs to construct a cluster tree first. A *cluster tree* $T_I$ is a tree of a finite index set $I$ of all degrees of freedom (DoFs). A cluster tree $T_I$ is usually generated by recursive subdivision of the index set $I$. In practice, bi-section based on a bounding box method is often employed to generate a binary tree. In A-EFIE, $I$ can be divided into two types: the current index set $I_j$ defined on edge DoFs and the charge index set $I_\rho$ defined on patch DoFs. Hence, two cluster trees, $T_{I_j}$ and $T_{I_\rho}$ should be constructed here. Take the construction of $T_{I_j}$ as an example, a simplified two-dimension case is given in Fig. 1(a). Here, $I_j$ within the corresponding bounding box is divided into two subsets recursively, and this process proceeds until the number of DoFs in the bounding box is smaller than a predetermined threshold $n_{\min}$. The resulting cluster tree $T_{I_j}$ is shown in Fig. 1(b). Similarly, $T_{I_\rho}$ can be constructed by a recursive subdivision of the charge index set $I_\rho$.

Then, a block cluster tree can be constructed based on the resultant cluster tree. A *block cluster tree* $T_{I \times J}$ arises from the interaction of two cluster trees $T_I$ and $T_J$. In Galerkin's method, $T_I$ and $T_J$ are mostly the same, which can be seen as the cluster tree of the original basis function set and that of the testing basis function set, respectively. $T_{I \times J}$ is constructed by recursive subdivision of $I \times J$, which terminates at blocks $t \times s \in T_{I \times J}$ ($t \in T_I$ and $s \in T_J$) satisfying the admissibility condition of

$$\min\{\text{diam}(B_t), \text{diam}(B_s)\} \le \eta \text{dist}(B_t, B_s) \tag{11}$$

where $B_t$ denotes the minimal bounding box for a cluster $t$, diam and dist denote the Euclidean diameter and distance of the bounding boxes, respectively, as shown in Fig. 2, and $\eta > 0$ controls the trade-off.

Since there are two cluster trees $T_{I_j}$ and $T_{I_\rho}$ in A-EFIE, we need to construct four block cluster trees, which are $T_{I_j \times I_j}$, $T_{I_j \times I_\rho}$, $T_{I_\rho \times I_j}$ and $T_{I_\rho \times I_\rho}$. To construct the block cluster tree $T_{I_j \times I_j}$, we let the cluster trees of the original basis function set $T_{I_j}$ and that of the testing basis function set $T_{I_j}$ interact level-by-level, as shown in Fig. 3(a), producing the block cluster tree $T_{I_j \times I_j}$ in Fig. 3(b). Similarly, $T_{I_j \times I_\rho}$, $T_{I_\rho \times I_j}$ and $T_{I_\rho \times I_\rho}$ can be constructed. It should be noted that both $T_{I_j \times I_j}$ and $T_{I_\rho \times I_\rho}$ are square trees, while $T_{I_j \times I_\rho}$ and $T_{I_\rho \times I_j}$ are rectangular ones due to the difference between $I_j$ and $I_\rho$.

$$\begin{bmatrix} 1,2,3,4,5,6,7,8,9,10, \\ 11,12,13,14,15,16,17,18, \\ 19,20,21,22,23,24,25 \end{bmatrix}$$

$$\begin{bmatrix} 1,2,3,4,5, \\ 6,14,15,16, \\ 17,18,19,20 \end{bmatrix} \quad \begin{bmatrix} 7,8,9,10,11, \\ 12,13,21,22, \\ 23,24,25 \end{bmatrix}$$

$$\begin{bmatrix} 1,2, \\ 3,4, \\ 5,6, \\ 20 \end{bmatrix} \begin{bmatrix} 14,15, \\ 16,17, \\ 18,19, \\ 20 \end{bmatrix} \quad \begin{bmatrix} 7,8,9, \\ 10,11, \\ 12,13 \end{bmatrix} \begin{bmatrix} 21,22, \\ 23,24, \\ 25 \end{bmatrix}$$

(a)                                             (b)

**Figure 1.** Construction of a cluster tree $T_{I_j}$. (a) Recursive subdivision of the current index set $I_j$. (b) The resulting cluster tree $T_{I_j}$.
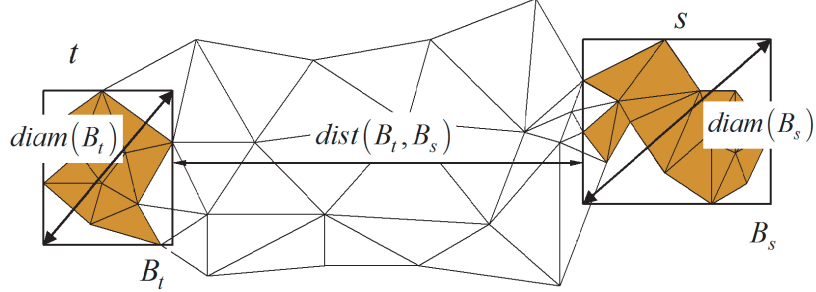


**Figure 2.** The admissibiliy condition of two clusters.

In a block cluster tree, there are only two types of blocks: admissible blocks and inadmissible blocks. Blocks satisfying Eq. (11) are called admissible blocks, which can be approximated by a low-rank representation in the following R$k$-matrix

$$\bar{\mathbf{F}} = \bar{\mathbf{A}}\bar{\mathbf{B}}^{\mathbf{T}}, \quad \bar{\mathbf{F}} \in \mathbb{C}^{m \times n}, \quad \bar{\mathbf{A}} \in \mathbb{C}^{m \times k}, \quad \bar{\mathbf{B}} \in \mathbb{C}^{n \times k} \tag{12}$$

with $\mathbf{A}$, $\mathbf{B}$ being rank-$k$ matrices, and $k$ is much smaller than $m$ and $n$. Besides, the inadmissible blocks are stored as full matrices. Fig. 3(b) presents one of the admissible partition cases, where the admissible blocks are grey and the inadmissible blocks are green. It should be noted that the practical block cluster tree $T_{I_j \times I_j}$ can be different from the case of Fig. 3(b), because a stronger admissibility condition based on disjoint bounding boxes is often adopted in A-EFIE.
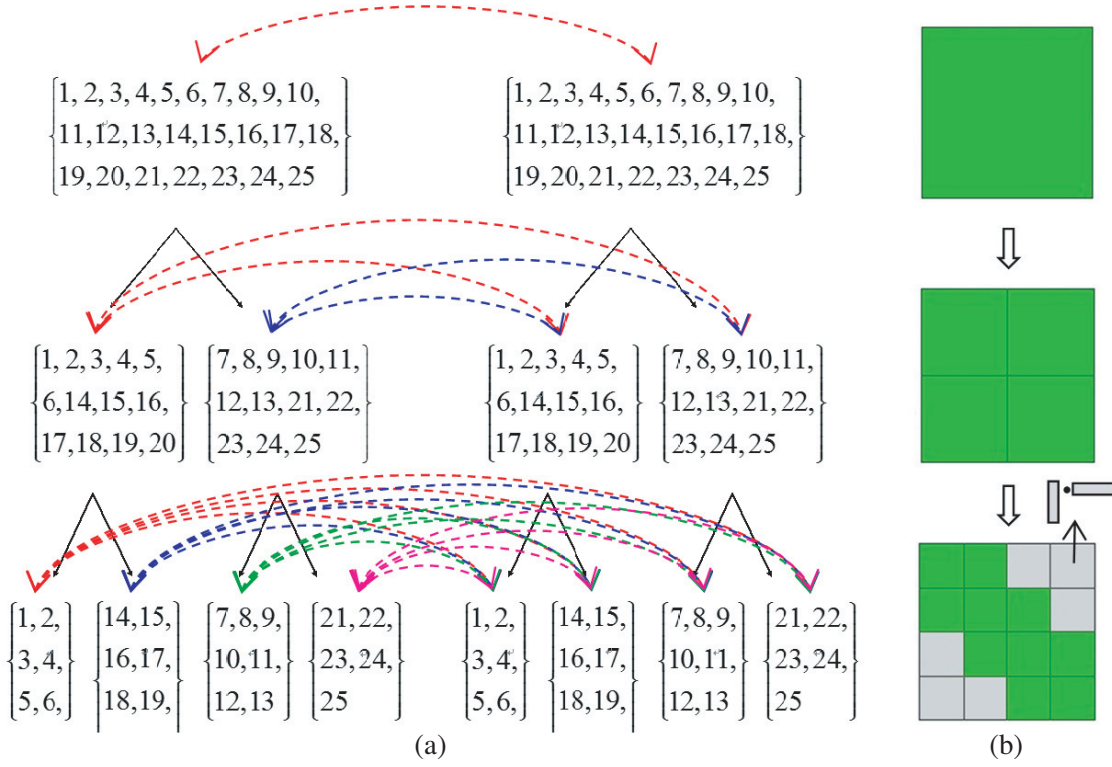
**Figure 3.** Construction of a block cluster tree $T_{I_j \times I_j}$. (a) Interaction of two cluster trees $T_{I_j}$. (b) The resulting block cluster tree $T_{I_j \times I_j}$.

## 3.2. Generate an $\mathcal{H}$-Matrix by LF-MLFMA

Based on the block cluster tree, an $\mathcal{H}$-matrix representation $\bar{\mathbf{Z}}_{\mathcal{H}}$ of the A-EFIE system matrix $\bar{\mathbf{Z}}$ can be generated by filling all entries of $\bar{\mathbf{Z}}$ into admissible or inadmissible blocks of respective block cluster tree. $\bar{\mathbf{Z}}$ includes five sub-matrices, i.e., $\bar{\mathbf{V}}$, $\bar{\mathbf{P}}$, $\bar{\mathbf{D}}$, $\bar{\mathbf{I}}$ and $\gamma \mathbf{a}\mathbf{a}^T$. Hence, we need to construct five $\mathcal{H}$-matrices, i.e., $\bar{\mathbf{V}}_{\mathcal{H}}$, $\bar{\mathbf{P}}_{\mathcal{H}}$, $\bar{\mathbf{D}}_{\mathcal{H}}$, $\bar{\mathbf{I}}_{\mathcal{H}}$ and $(\gamma \mathbf{a}\mathbf{a}^T)_{\mathcal{H}}$ to get $\bar{\mathbf{Z}}_{\mathcal{H}}$.

To construct $\bar{\mathbf{V}}_{\mathcal{H}}$ and $\bar{\mathbf{P}}_{\mathcal{H}}$, we need to fill all the entries of $\bar{\mathbf{V}}$ and $\bar{\mathbf{P}}$ based on the block cluster tree $T_{I_j \times I_j}$ and $T_{I_\rho \times I_\rho}$, respectively. The inadmissible blocks with full-matrix representation can be generated by the Method of Moments (MoM), as shown in Eqs. (4) and (5). Here, LF-MLFMA combined with a recompression method is employed to generate the admissible blocks with R$k$-matrix representations. LF-MLFMA is based on an octal tree, but the $\mathcal{H}$-matrix algorithm is based on a binary tree. Therefore, these two cluster trees should be unified first. Since one level of an octal tree can be divided into three levels of a binary tree, once an octal tree of $n$ levels is built, a binary tree of $3n$ levels is also obtained. Moreover, their block cluster trees should be unified, so that each far-field block of the LF-MLFMA is an admissible block of the $\mathcal{H}$-matrix. This can be readily achieved by setting $\eta = 1$ in Eq. (11). Thus, LF-MLFMA can be implemented to generate an $\mathcal{H}$-matrix.

The core equation of three-dimension LF-MLFMA can be written as

$$\boldsymbol{\alpha}_{LL'}^N(\mathbf{r}_{ji}) = \sum_{L_1} \sum_{L_2} \boldsymbol{\beta}_{LL_1}^N(\mathbf{r}_{jJ}) \boldsymbol{\alpha}_{L_1 L_2}^N(\mathbf{r}_{JI}) \boldsymbol{\beta}_{L_2 L'}^N(\mathbf{r}_{Ii}) \tag{13}$$

where $L_i = (l_i, m_i)$ and $\sum_{L_i} = \sum_{l_i=0}^{P} \sum_{m_i=-l_i}^{l_i}$ . Here, $P$ is the number of multipoles that controls the expanded form and the accuracy of the translation operator. One can refer to [30] for the detailed expression of

$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Equation (13) can be represented in the following matrix form

$$\bar{\boldsymbol{\alpha}}_{ji} = \bar{\boldsymbol{\beta}}_{jJ} \cdot \bar{\boldsymbol{\alpha}}_{JI} \cdot \bar{\boldsymbol{\beta}}_{Ii} \tag{14}$$

Considering the multilevel cases, as shown in Fig. 4, we have

Level $l$: $\bar{\boldsymbol{\alpha}}_{ji}^{(l)} = \bar{\boldsymbol{\beta}}_{jJ}^{(l)} \cdot \bar{\boldsymbol{\alpha}}_{JI}^{(l)} \cdot \bar{\boldsymbol{\beta}}_{Ii}^{(l)}$

Level $l-1$: $\bar{\boldsymbol{\alpha}}_{ji}^{(l-1)} = \bar{\boldsymbol{\beta}}_{jJ_1}^{(l)} \cdot \bar{\boldsymbol{\beta}}_{J_1J}^{(l-1)} \cdot \bar{\boldsymbol{\alpha}}_{JI}^{(l-1)} \cdot \bar{\boldsymbol{\beta}}_{II_1}^{(l-1)} \cdot \bar{\boldsymbol{\beta}}_{I_1i}^{(l)} = \bar{\boldsymbol{\beta}}_{jJ}^{(l-1)} \cdot \bar{\boldsymbol{\alpha}}_{JI}^{(l-1)} \cdot \bar{\boldsymbol{\beta}}_{Ii}^{(l-1)}$

and others.
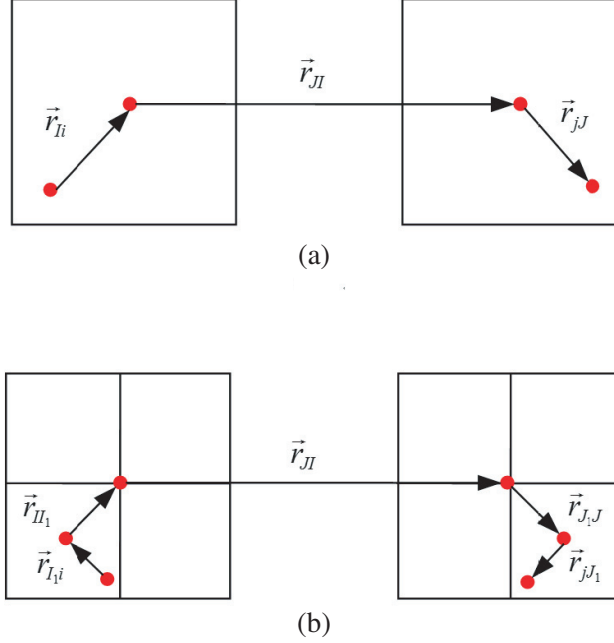


**Figure 4.** The translation process of the LF-MLFMA. (a) One-level case. (b) Two-level case.

Hence, for two clusters $t$ and $s$ that satisfy the admissibility condition in Eq. (11), by assembling all the indexes $j \in t$ and $i \in s$, the resulting admissible block $\bar{\mathbf{F}}_{t \times s}$ can be written as a matrix form of

$$\bar{\mathbf{F}}_{t \times s} = \bar{\boldsymbol{\beta}}_t \big|_{\#t \times r} \cdot \bar{\boldsymbol{\alpha}}_{t \times s} \big|_{r \times r} \cdot \bar{\boldsymbol{\beta}}_s \big|_{r \times \#s} \tag{15}$$

where the subscripts $\#t$, $\#s$ and $r$ denote the matrix dimensions. Here, $\#$ denotes the number of indices in a cluster. For the admissible blocks of $\bar{\mathbf{V}}_{\mathcal{H}}$, $\#t$ is the number of RWG basis functions in an observer cluster $t$, $\#s$ the number of RWG basis functions in a source cluster $s$, and $r = 3 \times (P+1)^2$, where $(P+1)^2$ is the number of $L_i$, and the constant 3 means three directions of a current vector. However, for the admissible blocks of $\bar{\mathbf{P}}_{\mathcal{H}}$, $\#t$ and $\#s$ denote the number of pulse basis functions in $t$ and $s$, respectively, while $r = (P+1)^2$ relates to the charge scalar.

The admissible block $\bar{\mathbf{F}}_{t \times s}$ generated from the LF-MLFMA has two features: 1) The admissible blocks generated from the same source cluster $s$ share an aggregation matrix $\bar{\boldsymbol{\beta}}_s$, and those generated from the same observer cluster $t$ share a disaggregation matrix $\bar{\boldsymbol{\beta}}_t$. 2) Both the aggregation matrix $\bar{\boldsymbol{\beta}}_s$ and the disaggregation matrix $\bar{\boldsymbol{\beta}}_t$ are nested. This is because an aggregation or disaggregation matrix can be represented by all the aggregation or disaggregation matrices of its child clusters multiplied by corresponding transfer matrices

$$\bar{\boldsymbol{\beta}}_s^{(l)} \Big|_{\#s \times r^{(l)}} = \sum_{i=1}^{child(s)} \bar{\boldsymbol{\beta}}_{s_i}^{(l-1)} \Big|_{\#s_i \times r^{(l-1)}} \cdot \bar{\boldsymbol{\beta}}_{s_i \to s}^{(l)} \Big|_{r^{(l-1)} \times r^{(l)}} \tag{16}$$

$$\bar{\boldsymbol{\beta}}_t^{(l)} \Big|_{\#t \times r^{(l)}} = \sum_{i=1}^{child(t)} \bar{\boldsymbol{\beta}}_{t_i}^{(l-1)} \Big|_{\#t_i \times r^{(l-1)}} \cdot \bar{\boldsymbol{\beta}}_{t_i \to t}^{(l)} \Big|_{r^{(l-1)} \times r^{(l)}} \tag{17}$$

where child($s$) and child($t$) denote the total number of child clusters of $s$ and $t$, respectively. Hence, we only need to store the aggregation and disaggregation matrices of the bottom level and the transfer matrices at other levels. Because of the aforementioned features, the admissible blocks of $\bar{\mathbf{V}}_{\mathcal{H}}$ and $\bar{\mathbf{P}}_{\mathcal{H}}$ can be produced by LF-MLFMA efficiently.

Although the admissible blocks of $\bar{\mathbf{V}}_{\mathcal{H}}$ and $\bar{\mathbf{P}}_{\mathcal{H}}$ are compressed to R$k$-matrices by LF-MLFMA, as shown in Eq. (15), they still have redundant information. Next, a recompression method is introduced to further compress the admissible blocks generated by the LF-MLFMA.

The recompression method is mainly based on QR decomposition and singular value decomposition (SVD), as shown in Fig. 5. The detailed process can be described as follows:

1. Compute a QR decomposition of $\bar{\boldsymbol{\beta}}_t = \mathbf{Q}_t \mathbf{R}_t$, where $\mathbf{Q}_t \in \mathbb{C}^{\#t \times r}$, $\mathbf{R}_t \in \mathbb{C}^{r \times r}$.

2. Compute a QR decomposition of $\bar{\boldsymbol{\beta}}_s = (\mathbf{Q}_s \mathbf{R}_s)^T$, $\mathbf{Q}_s \in \mathbb{C}^{\#s \times r}$, $\mathbf{R}_s \in \mathbb{C}^{r \times r}$.

3. Compute an SVD of $\mathbf{R}_t \cdot \bar{\boldsymbol{\alpha}}_{t \times s} \cdot \mathbf{R}_s^T = \mathbf{U} \sum \mathbf{V}^T$.

4. Extract $\tilde{\sum} = \text{diag}(\sum_{11}, \sum_{22} \ldots, \sum_{kk})$ (first largest $k$ singular values) with $\sum_{(k+1)(k+1)} \leq \varepsilon_{rec} \sum_{11} < \sum_{kk}$, where $\varepsilon_{rec}$ is a relative truncation error to control the content of recompression.

5. Extract $\tilde{\mathbf{U}} = [\mathbf{U_1 U_2} \ldots \mathbf{U}_k]$ (first $k$ columns), $\tilde{\mathbf{V}} = [\mathbf{V_1 V_2} \ldots \mathbf{V}_k]$ (first $k$ columns).

6. Set $\mathbf{A}_t = \mathbf{Q}_t \tilde{\mathbf{U}} \tilde{\sum}$ and $\mathbf{B}_s = \mathbf{Q}_s \tilde{\mathbf{V}}$.
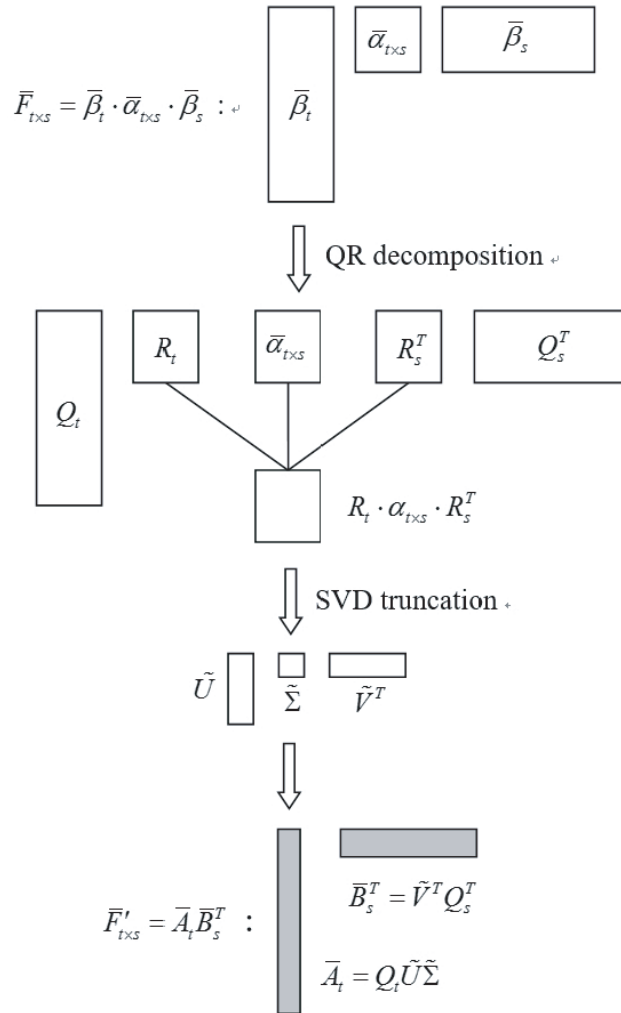


**Figure 5.** Recompression of an admissible block $\bar{\mathbf{F}}_{t \times s}$.

It should be noted that the QR decomposition of $\bar{\boldsymbol{\beta}}_t$ and $\bar{\boldsymbol{\beta}}_s$ only needs to be performed once for all admissible blocks sharing $t$ and $s$, respectively, while the SVD should be performed for each admissible block. Although we need to store $\bar{\boldsymbol{\beta}}_t$ and $\bar{\boldsymbol{\beta}}_s$ of all levels for this recompression, it can be done on-the-fly to avoid peak storage.

To construct $\bar{\mathbf{D}}_{\mathcal{H}}$, we need to fill all the non-zero entries of $\bar{\mathbf{D}}$ into the block cluster tree $T_{I_\rho \times I_j}$ since $\bar{\mathbf{D}}$ is sparse. It is easy to show that all the nonzero entries of $\bar{\mathbf{D}}$ should be filled in inadmissible blocks, while admissible blocks are empty. The reason is that two clusters $t$ and $s$ satisfying Eq. (11) must be well-separated, and the resulting entry $\bar{\mathbf{D}}_{i,j}$ ($i \in t, j \in s$) is zero according to Eq. (7). Hence, $\bar{\mathbf{D}}$ can be represented by $\bar{\mathbf{D}}_{\mathcal{H}}$ without approximation. $\bar{\mathbf{D}}_{\mathcal{H}}^T$ is nothing but the transpose of $\bar{\mathbf{D}}_{\mathcal{H}}$. Besides, $\bar{\mathbf{I}}_{\mathcal{H}}$ can be easily constructed by filling all the diagonal entries of diagonal inadmissible blocks on $T_{I_\rho \times I_\rho}$ with 1.

To construct $(\gamma \mathbf{a}\mathbf{a}^T)_{\mathcal{H}}$, we only need to fill all entries of $\gamma \mathbf{a}_\rho \mathbf{a}_\rho^T$ into the block cluster tree $T_{I_\rho \times I_\rho}$ while $T_{I_j \times I_j}$, $T_{I_j \times I_\rho}$ and $T_{I_\rho \times I_j}$ are kept empty, since the elements of $\mathbf{a}$ relating to $I_j$ are zero. Here, $\mathbf{a}_\rho \subset \mathbf{a}$ is defined as $\mathbf{a}_\rho = \mathbf{a}[(e+1):(e+p)]$. Different from the construction of $\bar{\mathbf{P}}_{\mathcal{H}}$, the admissible blocks of $(\gamma \mathbf{a}\mathbf{a}^T)_{\mathcal{H}}$ can be easily produced because $\gamma \mathbf{a}\mathbf{a}^T$ provides an R$k$-matrix representation inherently. In other words, an admissible block $\bar{\mathbf{F}}_{t \times s}$ can be directly extracted from $\gamma \mathbf{a}\mathbf{a}^T$ and represented by an R$k$-matrix with $k = 1$ as

$$\bar{\mathbf{F}}_{t \times s} = \gamma \mathbf{a}_t|_{\#t \times 1} \cdot \mathbf{a}_s|_{\#s \times 1}^T \tag{18}$$

It should be noted that $k$ equals the number of isolated objects for multiple object problems. The inadmissible blocks of $(\gamma \mathbf{a}\mathbf{a}^T)_{\mathcal{H}}$ can also be easily filled by the direct product of $\gamma \mathbf{a}_t \mathbf{a}_s^T$.

## 3.3. Hierarchical LU Factorization Algorithm

According to Eq. (8), the $\mathcal{H}$-matrix representation of $\bar{\mathbf{Z}}_{\mathcal{H}}$ can be written as

$$\bar{\mathbf{Z}}_{\mathcal{H}} = \begin{bmatrix} \bar{\mathbf{V}}_{\mathcal{H}} & \bar{\mathbf{D}}_{\mathcal{H}}^T \cdot \bar{\mathbf{P}}_{\mathcal{H}} \\ \bar{\mathbf{D}}_{\mathcal{H}} & k_0^2 \cdot \bar{\mathbf{I}}_{\mathcal{H}} \end{bmatrix} - \left[ \gamma \mathbf{a}\mathbf{a}^T \right]_{\mathcal{H}} \tag{19}$$

which can be denoted as a $2 \times 2$ blockwise matrix

$$\bar{\mathbf{Z}}_{\mathcal{H}} = \begin{bmatrix} \bar{\mathbf{Z}}_{11} & \bar{\mathbf{Z}}_{12} \\ \bar{\mathbf{Z}}_{21} & \bar{\mathbf{Z}}_{22} \end{bmatrix} \tag{20}$$

where $\bar{\mathbf{Z}}_{11} = \bar{\mathbf{V}}_{\mathcal{H}}$, $\bar{\mathbf{Z}}_{21} = \bar{\mathbf{D}}_{\mathcal{H}}$, $\bar{\mathbf{Z}}_{12} = \bar{\mathbf{D}}_{\mathcal{H}}^T \cdot \bar{\mathbf{P}}_{\mathcal{H}}$ and $\bar{\mathbf{Z}}_{22} = k_0^2 \cdot \bar{\mathbf{I}}_{\mathcal{H}} - (\gamma \mathbf{a}_\rho \mathbf{a}_\rho^T)_{\mathcal{H}}$.

First, we need to compute the multiplication of $\bar{\mathbf{D}}_{\mathcal{H}}^T \cdot \bar{\mathbf{P}}_{\mathcal{H}}$ to get $\bar{\mathbf{Z}}_{12}$. The $\mathcal{H}$-matrix formatted multiplication $\otimes_{\mathcal{H}}$ is employed to compute $\bar{\mathbf{D}}_{\mathcal{H}}^T \cdot \bar{\mathbf{P}}_{\mathcal{H}}$ instead of conventional matrix multiplication. Besides, the $\mathcal{H}$-matrix formatted addition $\oplus_{\mathcal{H}}$ is required to compute $k_0^2 \cdot \bar{\mathbf{I}}_{\mathcal{H}} - (\gamma \mathbf{a}_\rho \mathbf{a}_\rho^T)_{\mathcal{H}}$ to get $\bar{\mathbf{Z}}_{22}$. The detailed process of $\otimes_{\mathcal{H}}$ and $\oplus_{\mathcal{H}}$ can be referred to in [23]. Then, the $\mathcal{H}$-matrix formatted LU factorization $\bar{\mathbf{Z}}_{\mathcal{H}} = \bar{\mathbf{L}}_{\mathcal{H}} \bar{\mathbf{U}}_{\mathcal{H}}$ can be recursively computed from

$$\begin{bmatrix} \bar{\mathbf{Z}}_{11} & \bar{\mathbf{Z}}_{12} \\ \bar{\mathbf{Z}}_{21} & \bar{\mathbf{Z}}_{22} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{L}}_{11} & 0 \\ \bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{U}}_{11} & \bar{\mathbf{U}}_{12} \\ 0 & \bar{\mathbf{U}}_{22} \end{bmatrix} \tag{21}$$

by the following steps:

1. Perform LU factorization $\bar{\mathbf{Z}}_{11} = \bar{\mathbf{L}}_{11} \bar{\mathbf{U}}_{11}$ to get $\bar{\mathbf{L}}_{11}$, $\bar{\mathbf{U}}_{11}$.
2. Solve a lower triangular equation $\bar{\mathbf{L}}_{11} \bar{\mathbf{U}}_{12} = \bar{\mathbf{Z}}_{12}$ to get $\bar{\mathbf{U}}_{12}$.
3. Solve an upper triangular equation $\bar{\mathbf{L}}_{21} \bar{\mathbf{U}}_{11} = \bar{\mathbf{Z}}_{21}$ to get $\bar{\mathbf{L}}_{21}$.
4. Perform LU factorization $\bar{\mathbf{Z}}_{22} - \bar{\mathbf{L}}_{21} \bar{\mathbf{U}}_{12} = \bar{\mathbf{L}}_{22} \bar{\mathbf{U}}_{22}$ to get $\bar{\mathbf{L}}_{22}$, $\bar{\mathbf{U}}_{22}$.

In Steps 2 and 3 above, a lower or upper triangular solver $\bar{\mathbf{L}}\bar{\mathbf{X}} = \bar{\mathbf{Q}}$ or $\bar{\mathbf{X}}\bar{\mathbf{U}} = \bar{\mathbf{Q}}$ is required for a given $\bar{\mathbf{L}}$, $\bar{\mathbf{U}}$ and RHS matrix $\bar{\mathbf{Q}}$. The lower triangular solver can also be recursively computed to find the unknown matrix $\bar{\mathbf{X}}$ from

$$\begin{bmatrix} \bar{\mathbf{L}}_{11} & 0 \\ \bar{\mathbf{L}}_{21} & \bar{\mathbf{L}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{X}}_{11} & \bar{\mathbf{X}}_{12} \\ \bar{\mathbf{X}}_{21} & \bar{\mathbf{X}}_{22} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{Q}}_{11} & \bar{\mathbf{Q}}_{12} \\ \bar{\mathbf{Q}}_{21} & \bar{\mathbf{Q}}_{22} \end{bmatrix} \tag{22}$$

by the following steps:

    1. Solve $\bar{\mathbf{L}}_{11}\bar{\mathbf{X}}_{11} = \bar{\mathbf{Q}}_{11}$ to get $\bar{\mathbf{X}}_{11}$.
    2. Solve $\bar{\mathbf{L}}_{11}\bar{\mathbf{X}}_{12} = \bar{\mathbf{Q}}_{12}$ to get $\bar{\mathbf{X}}_{12}$.
    3. Solve $\bar{\mathbf{L}}_{22}\bar{\mathbf{X}}_{21} = \bar{\mathbf{Q}}_{21} - \bar{\mathbf{L}}_{21}\bar{\mathbf{X}}_{11}$ to get $\bar{\mathbf{X}}_{21}$.
    4. Solve $\bar{\mathbf{L}}_{22}\bar{\mathbf{X}}_{22} = \bar{\mathbf{Q}}_{22} - \bar{\mathbf{L}}_{21}\bar{\mathbf{X}}_{12}$ to get $\bar{\mathbf{X}}_{22}$.

The case of upper triangular solver is similar. When $\bar{\mathbf{X}}$ and $\bar{\mathbf{Q}}$ are vectors, the process of solving $\bar{\mathbf{L}}\bar{\mathbf{X}} = \bar{\mathbf{Q}}$ and $\bar{\mathbf{X}}\bar{\mathbf{U}} = \bar{\mathbf{Q}}$ is the $\mathcal{H}$-matrix formatted FBS.

Since the $\mathcal{H}$-LU factorization is based on a recursive procedure, we should perform the above steps repeatedly on the child level until the leaf level is reached. It is important that the addition and multiplication in above steps should be replaced by the $\mathcal{H}$-matrix formatted counterparts ($\oplus_{\mathcal{H}}$ and $\otimes_{\mathcal{H}}$). A truncation operator $\mathcal{T}^{\mathcal{H}}_{k \leftarrow k'}$ based on QR-decomposition and SVD is adopted to define $\bar{\mathbf{A}} \oplus_{\mathcal{H}} \bar{\mathbf{B}} = \mathcal{T}^{\mathcal{H}}_{k \leftarrow 2k}(\bar{\mathbf{A}} + \bar{\mathbf{B}})$ and $\bar{\mathbf{A}} \otimes_{\mathcal{H}} \bar{\mathbf{B}} = \mathcal{T}^{\mathcal{H}}_{k \leftarrow k'}(\bar{\mathbf{A}} \times \bar{\mathbf{B}})$ [23]. In this paper, a relative truncation error $\varepsilon_t$ is employed to generate an adaptive truncation scheme, which determines the accuracy of the $\mathcal{H}$-LU factors. It has been proved that the $\mathcal{H}$-LU factorization can be performed with $O(k^2 N \log^2 N)$ computational complexity and $O(kN \log N)$ memory requirement, where $N$ is the number of unknowns and $k$ is the average rank of all R$k$-matrices [24]. For the electrodynamic analysis at medium and high frequency regime, $k$ will increase with the electrical size of objects. However, for the quasi-static analysis at low frequency regime, $k$ for achieving any prescribed accuracy is almost constant as $N$ increases. Hence, the computational complexity and memory requirement of $\mathcal{H}$-LU factorization can be estimated as $O(N \log^2 N)$ and $O(N \log N)$ for low-frequency problems, respectively. After the $\mathcal{H}$-LU factorization, $\mathcal{H}$-LU factors are stored and used for the solution of (10) by the $\mathcal{H}$-FBS, which can be implemented with $O(N \log N)$ computational complexity for low-frequency problems.

Since the accuracy of $\mathcal{H}$-LU factorization is adjustable, $\mathcal{H}$-LU factorization with sufficient accuracy can be used as a direct solver for multiple-RHS-vector problems, while that with inadequate accuracy can be employed as a preconditioner of an iterative solver to accelerate its convergence.

For the $\mathcal{H}$-LU direct solver, the system Equation (10) can be written as

$$\bar{\mathbf{L}}_{\mathcal{H}}\bar{\mathbf{U}}_{\mathcal{H}} \cdot \mathbf{x} = \mathbf{f} \tag{23}$$

We perform the forward substitution $\bar{\mathbf{L}}_{\mathcal{H}} \cdot \mathbf{g} = \mathbf{f}$ to get a temporary vector of $\mathbf{g}$, and then do backward substitution $\bar{\mathbf{U}}_{\mathcal{H}} \cdot \mathbf{x} = \mathbf{g}$ to obtain the solution vector $\mathbf{x}$.

For the $\mathcal{H}$-LU preconditioned iterative solver, the system Equation (10) can be modified to be

$$(\bar{\tilde{\mathbf{L}}}_{\mathcal{H}}\tilde{\bar{\mathbf{U}}}_{\mathcal{H}})^{-1} \cdot \bar{\mathbf{Z}}_{\mathcal{H}} \cdot \mathbf{x} = (\bar{\tilde{\mathbf{L}}}_{\mathcal{H}}\tilde{\bar{\mathbf{U}}}_{\mathcal{H}})^{-1} \cdot \mathbf{f} \tag{24}$$

The MVP of the preconditioner matrix $(\bar{\tilde{\mathbf{L}}}_{\mathcal{H}}\tilde{\bar{\mathbf{U}}}_{\mathcal{H}})^{-1}$ can be easily performed by the $\mathcal{H}$-FBS, and then used at each iteration step to get the solution vector $\mathbf{x}$.

## 4. NUMERICAL EXAMPLES

A number of examples are simulated to validate the performance of the proposed method. First, to demonstrate the low-frequency stability of the proposed $\mathcal{H}$-LU algorithm, the condition numbers and solution accuracy are tested for a sphere. Then, the performance of the $\mathcal{H}$-LU algorithm and preconditioner are shown for the scattering from a car shell. Finally, the $\mathcal{H}$-LU direct solver and preconditioner are tested for a multi-RHS problem of a realistic package board. For all the testing, the generalized minimum residual method (GMRES) is adopted for the iterative solvers, the number of multipoles in the LF-MLFMA is chosen as $P = 5$, and the recompression accuracy is set to be $\varepsilon_{rec} = 10^{-4}$. All the computations are performed on a computer with Intel Core i5 2.8 GHz CPU.

### 4.1. Sphere

The first example considers a sphere to verify the low-frequency stability and accuracy of the proposed method. The sphere has a radius of 1 m, which is discretized into 2,672 patches and 4,008 edges. By sweeping the frequency from 3 Hz to $3 \times 10^8$ Hz, the condition numbers of the A-EFIE and EFIE system matrices are recorded in Fig. 6, and the solution errors of the A-EFIE-based $\mathcal{H}$-LU and EFIE-based $\mathcal{H}$-LU are shown in Fig. 7. Here, a two-level LF-MLFMA is used to construct $\mathcal{H}$-matrices. The relative
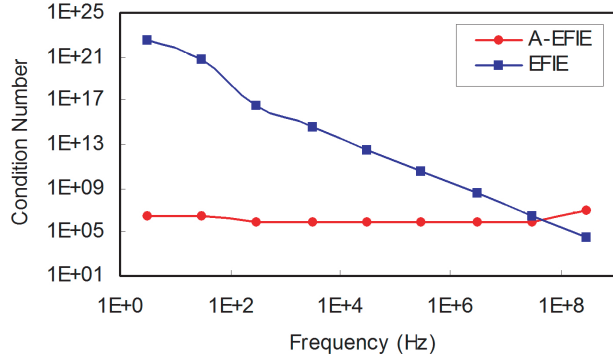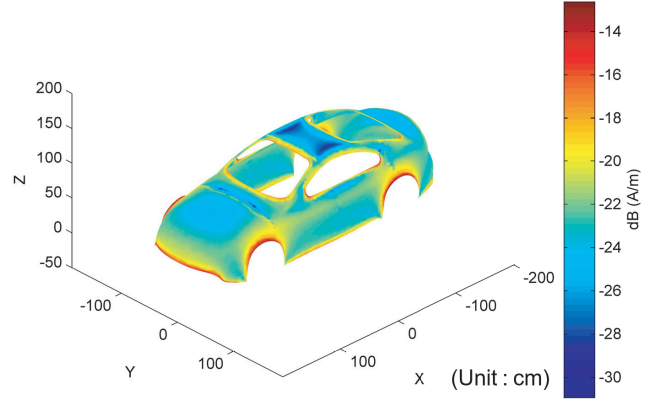
**Figure 6.** Condition numbers of the system matrices as frequency varies.
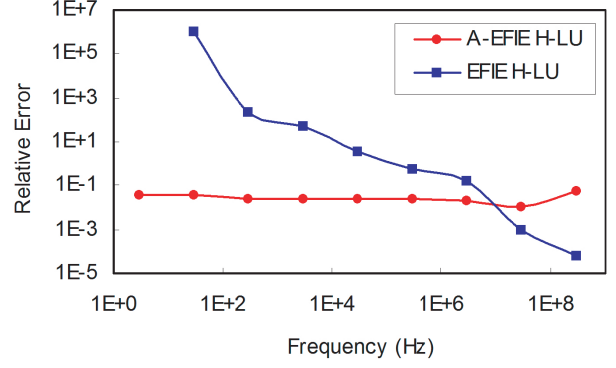


**Figure 7.** Relative errors of $\mathcal{H}$-LU factorization as frequency varies.
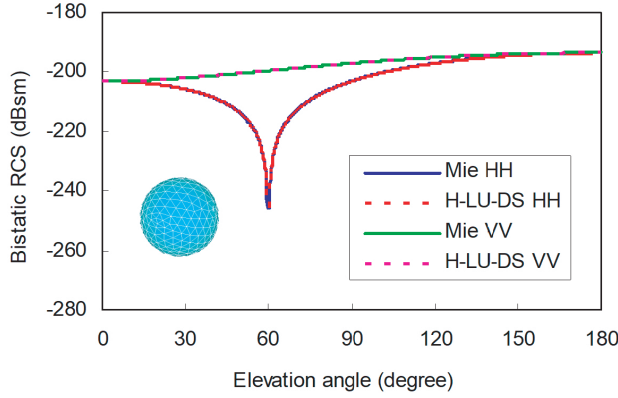


**Figure 8.** Comparison of the computed RCS and Mie series results at 30 kHz.



**Figure 9.** The surface current distribution on the car shell at 30 MHz.

truncation error is set to be $\varepsilon_t = 10^{-3}$ for the $\mathcal{H}$-LU. $\delta = \|\tilde{\bar{\mathbf{U}}}_{\mathcal{H}}^{-1} \tilde{\bar{\mathbf{L}}}_{\mathcal{H}}^{-1} \bar{\mathbf{Z}}_{\mathcal{H}} - I\|_F / \|I\|_F$ is used to define the relative error of the $\mathcal{H}$-LU factorization, where $\|\cdot\|_F$ denotes the Frobenius norm and $I$ is an identity matrix. It is obvious that the EFIE-based $\mathcal{H}$-LU undergoes low-frequency breakdown as the frequency decreases, while the A-EFIE-based $\mathcal{H}$-LU remains stable. Then, the scattering of this sphere for a plane wave excitation of 30 kHz is tested to verify the accuracy of the proposed $\mathcal{H}$-LU direct solver (DS) with $\varepsilon_t = 10^{-6}$. The radar cross section (RCS) is computed and plotted in Fig. 8. It can be seen that the results agree well with the analytical results computed by Mie series. It should be noted that a low frequency inaccuracy issue exists if the frequency approaches zero. This problem can be overcome by the perturbation method proposed in [17].

### 4.2. Car Shell

The second example analyzes the scattering from a car shell. The car shell is illuminated by plane wave at 30 MHz. It has a dimension of $0.40\lambda \times 0.16\lambda \times 0.10\lambda$ as shown in Fig. 9. The mesh contains 22,116 patches and 32,832 edges. The minimum, maximum, and average edge lengths are $0.410 \times 10^{-3}\lambda$, $0.789 \times 10^{-2}\lambda$ and $0.286 \times 10^{-2}\lambda$, respectively.

First, we test the performance of the $\mathcal{H}$-LU preconditioner. The $\mathcal{H}$-matrix representation of the A-EFIE system matrix is constructed by a 5-level LF-MLFMA. The $\mathcal{H}$-LU preconditioner and the popular constraint preconditioner [15] are compared, and the iterative convergence curves are plotted in Fig. 10. It can be seen that the $\mathcal{H}$-LU preconditioner significantly accelerates the convergence, compared with the constraint preconditioner. The $\mathcal{H}$-LU with two different truncation errors $\varepsilon_t = 10^{-2}$

and $\varepsilon_t = 10^{-3}$ are employed for the preconditioner, as presented in Fig. 10. Then, the monostatic RCS with 361 different directional excitations are computed for the scattering from this car shell. The computational costs of these three preconditioned iterative solvers are reported in Table 1. Obviously, the $\mathcal{H}$-LU preconditioner outperforms the constraint preconditioner for this multi-RHS problem because it greatly reduces the iteration time. Besides, the $\mathcal{H}$-LU preconditioner with higher accuracy has better performance but higher cost, and the $\mathcal{H}$-LU with sufficient accuracy can be used as a direct solver. The computational costs of the $\mathcal{H}$-LU direct solver with $\varepsilon_t = 10^{-6}$ are also reported in Table 1. The surface current distribution computed by the $\mathcal{H}$-LU preconditioned GMRES solver with $10^{-4}$ relative residual is also presented in Fig. 9.

**Table 1.** Performance of different solvers for the computation of monostatic RCS of the car shell.

| Solver | Construction Time (s) | Solution Time (s) | Total Time (s) |
|---|---|---|---|
| Constraint Pre | 3.1 | 421.7 | 153,875 |
| $\mathcal{H}$-LU Pre $10^{-3}$ | 11,921.0 | 10.5 | 16,721 |
| $\mathcal{H}$-LU Pre $10^{-2}$ | 6,880.2 | 36.4 | 21,430 |
| $\mathcal{H}$-LU direct solver | 19,313.7 | 3.3 | 20,512 |
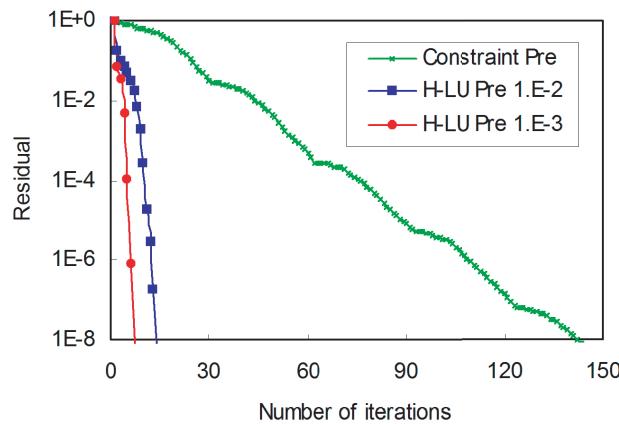


**Figure 10.** Iterative convergence history of three different preconditioners.

Then, the computational costs of the $\mathcal{H}$-LU algorithm are tested. The truncation error is fixed at $\varepsilon_t = 10^{-3}$ for the tests. We increase the number of unknowns $N$ from 4,132 to 201,156 by increasing the frequency and keeping the discretization density. The computational complexity and memory requirement are respectively observed to scale approximately as $O(N\log^2 N)$ and $O(N\log N)$ for this example in Figs. 11(a) and (b). In Fig. 11(b), the memory requirements of full matrix based LU factorization are also plotted as a comparison. The $\mathcal{H}$-FBS shows a computational complexity of $O(N\log N)$ in Fig. 11(c), and the relative error of the $\mathcal{H}$-LU factorization $\delta$ is stable over the entire range as shown in Fig. 11(d).

## 4.3. Package Board

The last example deals with a realistic package board that includes two interconnect pairs. This package board has a dimension of $0.15\lambda \times 0.14\lambda \times 0.0037\lambda$ at 3 GHz. A dense mesh is generated containing 72,898 patches and 109,347 edges, which leads to 182,245 unknowns in the computation. The minimum, maximum, and average edge lengths are $0.115 \times 10^{-4}\lambda$, $0.430 \times 10^{-2}\lambda$ and $0.137 \times 10^{-2}\lambda$, respectively.

The 5-level LF-MLFMA is employed to produce the $\mathcal{H}$-matrix. First, the $\mathcal{H}$-LU with $\varepsilon_t = 10^{-3}$ is used as a preconditioner for the iterative solver. We excite one of the 14 ports by a delta gap
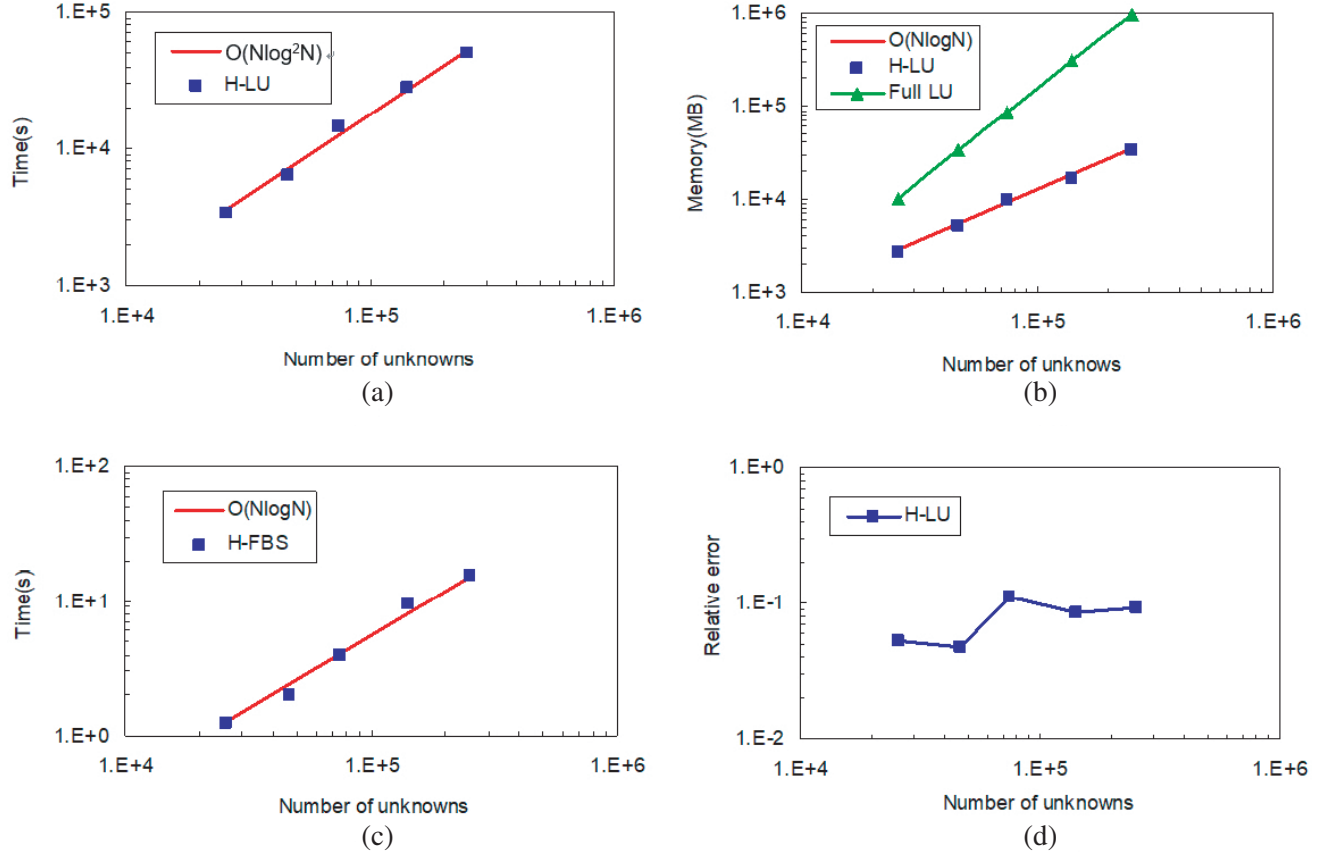
**Figure 11.** Performance of the $\mathcal{H}$-LU factorization algorithm. (a) CPU time of $\mathcal{H}$-LU factorization. (b) Memory usage of $\mathcal{H}$-LU and full matrix LU factorization. (c) CPU time of $\mathcal{H}$-FBS. (d) Relative errors of $\mathcal{H}$-LU factorization..

voltage source. Fig. 12 presents the surface current distribution computed by the $\mathcal{H}$-LU preconditioned GMRES solver with $10^{-4}$ relative residual error. The $\mathcal{H}$-LU preconditioner can further accelerate the convergence rate when compared with the constraint preconditioner for this example, as shown in Fig. 13. To show the accuracy of the final solutions, the constraint preconditioned GMRES solver with $10^{-9}$ relative residual error is employed to get the surface current coefficients $x_{ref}$ as reference. We define the accuracy of the results $\delta_x$ by calculating $\delta_x = \|\tilde{x} - x_{ref}\|_F / \|x_{ref}\|_F$. For the constraint and $\mathcal{H}$-LU preconditioned iterative solvers with $10^{-4}$ residual error, the calculated accuracy is $\delta_x = 8.3 \times 10^{-4}$ and $\delta_x = 6.5 \times 10^{-4}$, respectively.

Then, the $\mathcal{H}$-LU with $\varepsilon_t = 10^{-6}$ is tested as a direct solver. The calculated accuracy is $\delta_x = 9.2 \times 10^{-4}$, which exhibits the same order of accuracy as the iterative solvers. We compare the performances of the constraint preconditioned iterative solver, the $\mathcal{H}$-LU preconditioned iterative solver, and the $\mathcal{H}$-LU direct solver for a multi-RHS problem. In realistic circuit design, excitations imposed at different ports need to be simulated, which is a typical multi-RHS problem. Here, we test a multi-RHS case of 14 excitations at 14 ports. The results are reported in Table 2. It can be seen that the $\mathcal{H}$-LU preconditioner outperforms the constraint preconditioner, although the latter wins for a single RHS. This superiority increases as the number of RHSs increases. It should be noted that, for this example, the $\mathcal{H}$-LU preconditioned iterative solver is more efficient than the $\mathcal{H}$-LU direct solver. However, the $\mathcal{H}$-LU direct solver eventually outperforms the $\mathcal{H}$-LU preconditioned iterative solver when a large number of RHSs are present.
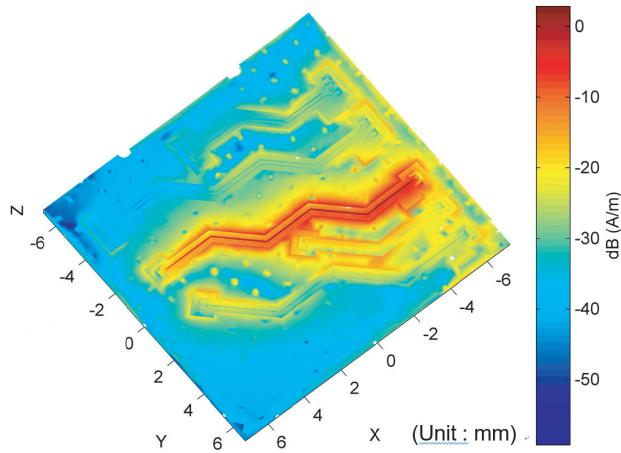
**Figure 12.** The surface current distribution on the package board at 3 GHz.
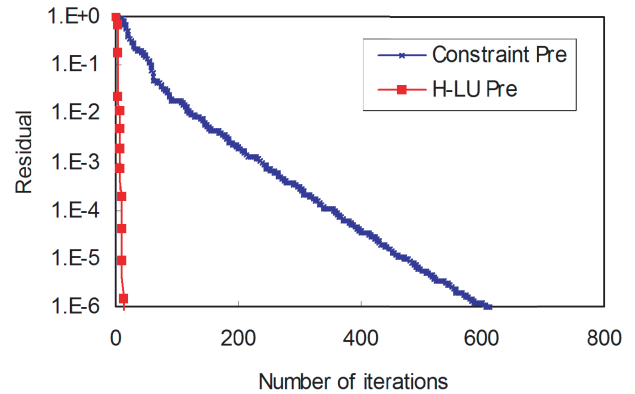


**Figure 13.** Iterative convergence history of different preconditioners.

**Table 2.** Performance of three different solvers for a 14-port excitation problem of the package board.

| Solver | Construction Time (s) | Solution Time (s) | Total Time (s) |
|---|---|---|---|
| Constraint Pre GMRES | 8.7 | 2459.6 | $8.7 + 2459.6 \times 14 = 34443.4$ |
| $\mathcal{H}$-LU Pre GMRES | 20568.2 | 67.7 | $20,568.2 + 67.7 \times 14 = 21515.2$ |
| $\mathcal{H}$-LU Direct Solver | 34052.5 | 5.6 | $34052.5 + 5.6 \times 14 = 34130.9$ |

## 5. CONCLUSION

An efficient low-frequency surface integral equation solver is developed for the fast analysis of low-frequency electromagnetic problems. By using the A-EFIE formulation with deflation, the proposed method is free of low-frequency breakdown. For large-scale problems, the LF-MLFMA is introduced to construct a data-sparse $\mathcal{H}$-matrix of the A-EFIE system matrix. A recompression technique is employed to further compress the $\mathcal{H}$-matrix. Based on the compact $\mathcal{H}$-matrix, an $\mathcal{H}$-LU factorization algorithm is developed, which provides not only an efficient direct solver with logarithmic linear computational complexity and memory requirement, but also a good preconditioner for an iterative solver. Three typical examples are analyzed and numerical results demonstrate the proposed method is robust for the analysis of large-scale low-frequency problems.

## REFERENCES

1. Morita, N., N. Kumagai, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*, Artech House, Boston, MA, 1990.
2. Chew, W. C., M. S. Tong, and B. Hu, *Integral Equation Methods for Electromagnetic and Elastic Waves*, Morgan and Claypool, New York, NY, USA, 2007.
3. Qian, Z. G. and W. C. Chew, "A quantitative study of the low frequency breakdown of EFIE," *Microw. Opt. Tech. Lett.*, Vol. 50, No. 5, 1159–1162, May 2008.
4. Wilton, D. R. and A. W. Glisson, "On improving the stability of the electric field integral equation at low frequencies," *Proc. URSI Radio Sci. Meeting*, 24, Los Angeles, CA, Jun. 1981.
5. Wu, W., A. W. Glisson, and D. Kajfez, "Study of two numerical solution procedures for the electric field integral equation at low frequency," *Appl. Computat. Electromagn. Soc. J.*, Vol. 10, No. 3, 69–80, Nov. 1995.
6. Burton, M. and S. Kashyap, "A study of a recent, moment-method algorithm that is accurate to very low frequencies," *Appl. Computat. Electromagn. Soc. J.*, Vol. 10, No. 3, 58–68, Nov. 1995.
7. Zhao, J. S. and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Trans. Antennas Propag.*, Vol. 48, 1635–1645, Oct. 2000.
8. Vecchi, G., "Loop-star decomposition of basis functions in the discretization of EFIE," *IEEE Trans. Antennas Propag.*, Vol. 47, 339–346, Feb. 1999.
9. Lee, J. F., R. Lee, and R. J. Burkholder, "Loop star basis functions and a robust preconditioner for EFIE scattering problems," *IEEE Trans. Antennas Propag.*, Vol. 51, No. 8, 1855–1863, Aug. 2003.
10. Andriulli, F. P., K. Cools, H. Bagci, F. Olyslager, A. Buffa, S. Christiansen, and E. Michielssen, "A multiplicative Calderon preconditioner for the electric field integral equation," *IEEE Trans. Antennas Propag.*, Vol. 56, 2398–2412, Aug. 2008.
11. Stephanson, M. B. and J. F. Lee, "Preconditioner electric field integral equation using Calderón identities and dual loop/star basis functions," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 4, 1274–1279, Apr. 2009.
12. Yan, S., J. M. Jin, and Z. Nie, "EFIE analysis of low-frequency problems with loop-star decomposition and Calderón multiplicative preconditioner," *IEEE Trans. Antennas Propag.*, Vol. 58, No. 3, 857–867, Mar. 2010.
13. Sun, S., Y. G. Liu, W. C. Chew, and Z. Ma, "Calderón multiplicative preconditioned EFIE with per-turbation method," *IEEE Trans. Antennas Propag.*, Vol. 61. No. 1, 247–255, Jan. 2013.
14. Qian, Z. G. and W. C. Chew, "An augmented EFIE for high speed interconnect analysis," *Micro. Opt. Technol. Lett.*, Vol. 50, No. 10, 2658–2662, Oct. 2008.
15. Qian, Z. G. and W. C. Chew, "Fast full-wave surface integral equation solver for multiscale structure modeling," *IEEE Trans. Antennas Propag.*, Vol. 50, 3594–3601, Nov. 2009.
16. Xia, T., H. Gan, M. Wei, W. C. Chew, H. Braunisch, Z. Qian, K. Aygün, and A. Aydiner, "An integral equation modeling of lossy conductors with the enhanced augmented electric field integral equation," *IEEE Trans. Antennas Propag.*, Vol. 65, No. 8, 4181–4190, Jun. 2017.
17. Qian, Z. G. and W. C. Chew, "Enhanced A-EFIE with perturbation method," *IEEE Trans. Antennas Propag.*, Vol. 58, No. 10, 3256–3264, Oct. 2010.
18. Xia, T., H. Gan, M. Wei, W. C. Chew, H. Braunisch, Z. Qian, K. Aygün, and A. Aydiner, "An enhanced augmented electric field integral equation formulation for dielectric objects," *IEEE Trans. Antennas Propag.*, Vol. 64, No. 6, 2339–2347, Jun. 2016.
19. Meng, L. L., X. Y. Xiong, T. Xia, and L. J. Jiang, "The error control of mixed-form fast multipole algorithm based on the high-order multipole rotation," *IEEE Antenn. Wireless Propag. Lett.*, Vol. 16, 1655–1658, Jan. 2017.
20. Wu, J. W., Z. G. Qian, J. E. Schutt-Ainé, and W. C. Chew, "Fast solution of low-frequency complex problems over a frequency band using enhanced A-EFIE and FMM," *Micro. Opt. Technol. Lett.*, Vol. 56, No. 9, 2153–2158, 2014.

21. Dai, Q. I., J. W. Wu, H. Gan, Q. S. Liu, W. C. Chew, and W. E. I. Sha, "Large-scale characteristic mode analysis with fast multipole algorithms," *IEEE Trans. Antennas Propag.*, Vol. 64, No. 7, 2608–2616, Jul. 2016.

22. Hackbusch, W. and B. Khoromaskij, "A Sparse Matrix arithmetic based on $H$-matrices. Part I: Introduction to $H$-matrices," *Computing*, Vol. 62, 89–108, 1999.

23. Grasedyck, L. and W. Hackbusch, "Construction and arithmetics of $H$-matrices," *Computing*, Vol. 70, No. 4, 295–344, Aug. 2003.

24. Börm, S., L. Grasedyck, and W. Hackbusch, "Introduction to hierarchical matrices with applications," *Engineering Analysis with Boundary Elements*, No. 27, 405–422, 2003.

25. Bebendorf, M. and Leipzig, "Hierarchical LU decomposition-based preconditioners for BEM," *Computing*, Vol. 74, 225–247, 2005.

26. Chai, W. 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, 3147–3159, Oct. 2009.

27. Chai, W. and D. Jiao, "A complexity-reduced $\mathcal{H}$-matrix based direct integral equation solver with prescribed accuracy for large-scale electrodynamic anaysis," *Proc. IEEE Int. Symp. Antennas Propag.*, Jun. 2010.

28. Chai, W. and D. Jiao, "Direct matrix solution of linear complexity for surface integral-equation-based impedance extraction of complicated 3-D structures," *Proceedings of the IEEE*, Vol. 01, No. 2, 372–388, Feb. 2013.

29. Wan, T., Z. N. Jiang, and Y. J. Sheng, "Hierarchical matrix techniques based on matrix decomposition algorithm for the fast analysis of planar layered structures," *IEEE Trans. Antennas Propag.*, Vol. 59, No. 11, 4132–4141, Nov. 2011.

30. Zhao, J. S. and W. C. Chew, "Three dimensional multilevel fast multipole algorithm from static to electrodynamic," *Microw. Opt. Technol. Lett.*, Vol. 26, No. 1, 43–48, 2000.

31. Zhao, J. S. and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Trans. Antennas Propag.*, Vol. 48, No. 10, 1635–1645, Oct. 2000.