Augmented MLFMM for Analysis of Scattering from PEC Object with Fine Structures

Ming Chen[1], Ru Shan Chen[1, 2], and Xiao Qing Hu[1]

[1]Department of Communication Engineering
Nanjing University of Science and Technology, China

[2]Science and Technology on Space Microwave Technology, China
eechenrs@mail.njust.edu.cn

Abstract ― In this paper, a multilevel Green's function interpolation method (MLGFIM) combined with multilevel fast multipole method (MLFMM) is presented for solving the electromagnetic scattering from the objects with fine structures. In the conventional MLFMM, the size of the finest cube must be larger than a definite value, which is typically 0.2 λ; it often generates a large number of unknowns in each finest cube especially for objects with fine structures. Accordingly, it requires a lot of memory to store the near-field impedance matrix in MLFMM. In order to decrease the memory requirement of the near-field matrix in the MLFMM, the MLGFIM is introduced to calculate the near-field interactions. The number of unknowns in each cube can be less than a required number regardless of the size of the cube in the MLGFIM. To further reduce the computational complexity, many recompressed techniques, such as the adaptive cross approximation (ACA), QR factorization, and singular value decomposition (SVD), are applied to compress the low rank Green's function matrix for speeding up the matrix-vector multiplication. Numerical results are given to demonstrate the accuracy and efficiency of the proposed method.

Index Terms **-** Multilevel fast multipole method (MLFMM), multilevel Green's function interpolation method(MLGFIM), QR factorization.

I. INTRODUCTION

The method of moment (MoM) [1-2] has found wide-spread application in a variety of electromagnetic radiation and scattering problems, When the number of unknowns is small, the resultant matrix equations in MoM can be solved directly with computational complexity of $O(N^3)$, where N is the number of unknowns. For moderate scale problems, the matrix equations are often solved by iterative solvers, such as the conjugate gradient method (CG), and the biconjugate gradient method (BiCG), with $O(N^2)$ operation for each matrix-vector product (MVP). The memory requirement is $O(N^2)$ for both the direct and the iterative solvers. The complexity of the direct or iterative solvers mentioned above blocks their application to the analysis of scattering from electrically large objects, the MoM can only be used for small scale problems. In recent years, the fast multipole method (FMM) [3-5] has been developed to accelerate the MVP with complexity of $O(N^{1.5})$. With the multilevel fast multipole algorithm (MLFMM) [6-10], the complexity is further reduced to $O(N\log N)$; this represents an impressive improvement as compared with conventional $O(N^3)$ or $O(N^2)$ techniques. By using the MLFMM, a common PC can solve problems which only can be solved by supercomputer in the past.

The MLFMM can be applied to almost all electromagnetic problems, such as microwave circuits, antennas, scattering targets, etc. However, it is still very challenging to apply the method to objects with fine structures. Accurate

discretization produces a large number of unknowns in each of the finest cubes. Accordingly, the near-field interaction matrices grow rapidly with the surface discretization density. The time used to calculate the near-field impedance matrix is also very long. Therefore, for the objects with fine structures, the MLFMM still challenged by the CPU time and memory requirement. It is necessary to further improve the efficiency of the MLFMM.

In order to overcome the bottleneck of conventional MLFMM in the near-field for the objects with fine structure, a multilevel Green's function interpolation method (MLGFIM) [11-16] combined with the MLFMM (MLGFIM-MLFMM) is proposed in this paper. The MLGFIM enables a highly compact representation and efficient numerical computation of the dense matrices when the source and observation cubes are well separated. The complexity of storage requirements and the MVP of the MLGFIM is approximately $O(C_1 N)$ as shown in [16] while the complexity is $O(C_2 N \log(N))$ for the MLFMM as shown in [6]. By comparing the numerical results of [6] and [16], the coefficient of C_1 is much larger than C_2. This is because when applying the MLGFIM into a full wave electromagnetic problem, the number of interpolation points must be enlarged to keep the accuracy of the Green's function when the cube size increases. A large number of interpolation points drastically reduce the efficiency of the MLGFIM. However, the MLGFIM has its own merit. Compared with MLFMM, the octree in MLGFIM can be split until the number of unknowns in each cube is less than a required number regardless of the cube size. The octree structure is the same as in the low-frequency fast multipole method (LF-FMM) [17]. In this paper, the MLGFIM is used to calculate part of the near-field interaction for reducing the memory requirement of the near-field while the MLFMM is used to calculate the far-field interaction. In contrast to the conventional MLFMM, the augmented MLFMM make the near-field memory requirement reduce greatly, this idea makes the objects with fine structure problems solvable by MLGFIM-MLFMM.

The remainder of this paper is organized as follows. Section II gives a brief introduction to the electric field integral equation (EFIE) and the Lagrange interpolation in MLGFIM. The ACA technique, QR factorization, and SVD factorization is employed to compress the low rank Green's function matrix to accelerate the MVP. Section III presents the numerical results that demonstrate the accuracy and efficiency of the proposed method. Finally, some conclusions are given in section IV.

II. THEORY

Consider a three-dimensional electromagnetic problem; the object is illuminated by an incident wave $\vec{E_i}$ that induces current $\vec{J}_s$ on the conducting surface. The current satisfies the following electric-field integral equation:

$$\vec{E_i}|_{\tan} = \iint_s [j\omega\mu \vec{J_s}(\vec{r'}) g(\vec{r},\vec{r'}) - \frac{j}{\omega\varepsilon}(\nabla' \vec{J_s}(\vec{r'}))\nabla' g(\vec{r},\vec{r'})]_{\tan} ds' \quad (1)$$

In which Green's function $g(\vec{r},\vec{r'}) = \frac{e^{-jk|\vec{r}-\vec{r'}|}}{4\pi|\vec{r}-\vec{r'}|}$, ω is the angular frequency, and k is the wave number which is $\omega\sqrt{\mu\varepsilon}$. μ, ε are the free space permeability and permittivity, respectively. The second "tan" denotes the component that is tangential to the conducting surface S. By expanding the unknown surface current density $\vec{J_s}$ using Rao-Wilton-Glisson (RWG) basis functions and applying Galerkin's method on (1) gives a MoM equation:

$$\bar{\bar{Z}}\bar{x} = \bar{V} \quad (2)$$

where

$$Z_{mn} = \iint_{s_m} ds \iint_{s_n} ds' [j\omega\mu \vec{J_m}(\vec{r}) \vec{J_n}(\vec{r'}) - \frac{j}{\omega\mu}(\nabla \cdot \vec{J_m}(\vec{r}))(\nabla' \cdot \vec{J_m}(\vec{r'}))] g(\vec{r},\vec{r'}) \quad (3)$$

and

$$V_m = \iint_{s_m} \vec{J_m}(\vec{r}) \vec{E_i}(\vec{r}) ds \quad (4)$$

Here, $\bar{\bar{Z}}$ is the impedance matrix, $\bar{x}$ are the coefficients of the induced current expanded in RWG basis functions, and $\bar{V}$ is the vector of incident field. The dimension of $\bar{\bar{Z}}$ is often as high as millions for electrically large EM scattering problems. This blocks the MoM application to the analysis of scattering from electrically large objects. The FMM and its multilevel version, MLFMM has been developed to accelerate the

MVP, lower the memory requirement to $O(N^{1.5})$ and $O(N\log N)$. The process of the MVP in MLFMM is splitted in two parts as

$$\overline{\overline{Z}}\bar{x} = \overline{\overline{Z}}_{NF}\bar{x} + \overline{\overline{Z}}_{FF}\bar{x}. \tag{5}$$

Here, the first term $\overline{\overline{Z}}_{NF}$ is the interactions from the nearby cubes, and is calculated directly by MoM. While the second term $\overline{\overline{Z}}_{FF}$ is the interactions from the well-separated cubes which are computed in a group-by-group manner by MLFMM. The computation of $\overline{\overline{Z}}_{NF}\bar{x}$ is done directly, while the computation of $\overline{\overline{Z}}_{FF}\bar{x}$ is done in three stages called the aggregation phase, the translation phase, and the disaggregation phase which are contributions from far-field interaction computation. These steps are now well documented and we refer the reader to consult the literature [6-8] for more details.

For the objects with fine structures, a straightforward MoM for computing the near-field $\overline{\overline{Z}}_{NF}$ is very expensive. Our approach is to approximate part of $\overline{\overline{Z}}_{NF}$ by a matrix which can be stored in a data-sparse format; The MLGFIM is introduced in MLFMM to descript part of the near-field matrix by a sparse matrix format leading to a significant reduction in the near-field memory requirement.

A. Data-sparse representation of the low-rank matrix

Here, the free space Green function $g(\vec{r},\vec{r}') = e^{-jk|\vec{r}-\vec{r}'|}/4\pi|\vec{r}-\vec{r}'|$ is considered. $\vec{r}$ means the field point located in cube m and $\vec{r}'$ means the source point located in cube n. If cube m and cube n are two well-separated cubes, the Green function $G(\vec{r},\vec{r}') = e^{-jk|\vec{r}-\vec{r}'|}/|\vec{r}-\vec{r}'|$ can be interpolated using Lagrange interpolation technique, it can be written as

$$G(\vec{r},\vec{r}') = \sum_{p=1}^{K}\sum_{q=1}^{K}\omega_{m,p}(\vec{r})\omega_{n,q}(\vec{r}')G(r_{G_m,p}, r'_{G_n,q}) \tag{6}$$

where $\omega_{m,p}(\vec{r})$ and $\omega_{n,q}(\vec{r}')$ are the *pth* Lagrange interpolation points in cube m and *qth* Lagrange interpolation points in cube n, $r_{m,p}$ denotes the Lagrange interpolation in cube m. K is the number of interpolation points in cube m or n, $G(r_{m,p}, r'_{n,q})$ is the Green's function matrix generated from interpolation point $r'_{n,q}$ in cube n to interpolation point $r_{m,p}$ in cube m, substituting (6) into (3) gives :

$$Z_{mn} = \frac{j\omega\mu}{4\pi}\sum_{p=1}^{k}\sum_{q=1}^{k}G(r_{m,p}, r'_{n,q})\iint_{s_m}\overrightarrow{J_m}(\vec{r})\omega_{m,p}(\vec{r})ds\iint_{s_n}\overrightarrow{J_n}(\vec{r}')\omega_{n,q}(\vec{r}')ds' - \frac{j}{4\pi\omega\mu}\sum_{p=1}^{k}\sum_{q=1}^{k}G(r_{m,p}, r'_{n,q})\iint_{s_m}\nabla\cdot\overrightarrow{J_m}(\vec{r})\omega_{m,p}(\vec{r})ds\iint_{s_n}\nabla'\overrightarrow{J_n}(\vec{r}')\omega_{n,q}(\vec{r}')ds' \tag{7}$$

The submatrix Z_{mn} can be represented in a factorized form

$$\overline{\overline{Z}}^{t,s} = W^t G^{t,s} W^{s^T} \tag{8}$$

where $W^t = \left[W_1^t \;\; W_2^t\right]$, $W^s = \left[W_1^s \;\; W_2^s\right]$

$$G^{t,s} = \begin{bmatrix} \overline{\overline{G}}_1^{t,s} & 0 \\ 0 & \overline{\overline{G}}_2^{t,s} \end{bmatrix} \tag{9}$$

$W_1^t, W_2^t \in \mathbb{C}^{M_m\times K}$ $\quad W_1^s \;\; W_2^s \in \mathbb{C}^{M_n\times K}$

$\overline{\overline{G}}_1^{t,s}, \overline{\overline{G}}_2^{t,s} \in \mathbb{C}^{K\times K}$

And

$$\overline{\overline{W}}_{1mp}^{t} = \iint_{s_m}\overrightarrow{J_m}(\vec{r})\omega_{m,p}(\vec{r})ds$$

$$\overline{\overline{W}}_{2mp}^{t} = \iint_{s_m}\nabla\cdot\overrightarrow{J_m}(\vec{r})\omega_{m,p}(\vec{r})ds$$

$$\overline{\overline{W}}_{1nq}^{s} = \iint_{s_n}\overrightarrow{J_n}(\vec{r}')\omega_{n,q}(\vec{r}')ds'$$

$$\overline{\overline{W}}_{2nq}^{s} = \iint_{s_n}\nabla'\overrightarrow{J_n}(\vec{r}')\omega_{n,q}(\vec{r}')ds'$$

$$\overline{\overline{G}}_{1pq}^{t,s} = \frac{j\omega\mu}{4\pi}G(r_{m,p}, r'_{n,q})$$

$$\overline{\overline{G}}_{2pq}^{t,s} = \frac{-j}{4\pi\omega\mu}G(r_{m,p}, r'_{n,q}) \tag{10}$$

where M_m and M_n denote the number of unknowns in cube m and n. p is the number of interpolation points along each direction, in which $d = 1,2,3$, for 1-, 2-, and 3-D problems, respectively. Clearly, the rank of matrix $\overline{\overline{Z}}^{t,s}$ is at most $2\times$K regardless of the cardinality of t and s, it is obvious that if $K << \min(M_m, M_n)$, the computing of $\overline{\overline{W}}_m$, $\overline{\overline{W}}_n$ and $\overline{\overline{G}}_{mn}$ will be significantly faster than that of $\overline{\overline{Z}}_{mn}$. Both the memory requirement and the matrix filling time are greatly reduced.

From above, we know that the MLGFIM is a rank-based method; it is realized by using the Lagrange interpolation technique in Green's function when the source and observation cubes are well separated. Applying the MLGFIM to the

low-ranked impedance matrix will result in significant memory reduction and computational time savings. For this, a cube tree is first needed to construct. We first enclose the entire object in a large cube, and then the cube is partitioned into eight smaller cubes. Each subcube is recursively subdivided into eight smaller cubes until the finest cubes satisfy the termination criterion. For MLFMM, the finest cube size is about half a wavelength. After constructing a tree, numerical operations can be performed on it. Two cubes are well separated if the ratio of the cube-center-distance to the cube size is greater than or equal 2. The impedance matrix between them is low-ranked. Otherwise, they are near each other, share at least one edge point, and the impedance matrix is full-ranked. This will cost a lot of CPU time and memory for the near-field computation if the number of unknowns contained in every cube is large. By the MLGFIM technique in the near-field computation, we can continue to subdivide the cube tree until the number of unknowns in each cube is less than or equal to a given number which is a parameter to control the tree depth. Therefore, the number of basis functions contained in every cube is reduced a lot, the MoM for the near-field computation is reduced; part of the near-field in MLFMM now can be computed by MLGFIM. In order to explain the implementation of the MLGFIM in the near-field computation in the MLFMM clearly, a brief description of its workflow is presented in the following.

As a simple example, a PEC plate is considered as shown in Fig. 1; it is a 1 λ wide square plate. With MLFMM, a 2-level division is used. The finest cube size is 0.25 λ, the line-filled cubes are the neighbors of cube 5, and the interaction between them is computed by MoM and stored in memory. The other cubes are the far-field of cube 5, the interaction between them is computed by the MLFMM. With the MLGFIM in the near-field computation, the cubes at level-2 are subdivided into cubes at level-3. Therefore the number of basis functions contained in every cube at level-3 reduced to about one-fourth of the original compared with the cube at level-2. Cube 5 at level-2 is the parent of cube 6 at level-3, with the MLFMM all the line-filled and the darkened cubes at level-3 are the near-field of cube 5 at level-2. While with MLGFIM at level-3, the interaction between the Cube 6, and the darkened cubes are computed by MoM; the interaction between the Cube 6 and the line-filled cubes are computed by MLGFIM. Therefore, the total near-field computation in the MLFMM is decomposed into two parts by using MLGFIM. One part of the near-field is computed directly by the MoM, the other is computed by MLGFIM. From above, we know that the MLGFIM is a rank-based method; applying the MLGFIM to the low-ranked impedance matrix will result in significant memory reduction and saves computing time. Therefore, the total near-field memory requirement is reduced in MLFMM.

B. Lagrange interpolation technique in the Green's function

The rank-deficiency of the proposed method is realized by the interpolation of the Green's function technique. Therefore, the accuracy analysis of the Green function interpolation technique is very important. From [11], it can be seen that for static problems or problems having small electric sizes, the number of interpolation points K in every cube keeping constant at all levels can keep the accuracy across all levels of a cluster tree. However, for full-wave problems, the use of constant rank cannot keep the accuracy to the same order when the size of the cube increases. This can be analyzed as below.

In MLFMM, we known that the lower the tree is, the larger the cube size is. The phase of the Green's function oscillates rapidly when the separation between the cubes increases. Obviously to accurately compute the Green's function between two points in two well separated cubes, the number of interpolation points should be increased when the frequency increases or when the sizes of the cubes increase. Consequently, to employ the MLGFIM to the full-wave problem, K should be replaced by K_l, in which subscript l denotes the level index. K_l means the number of interpolation points in every cube at level l. Since increasing frequency is equivalent to increasing the tree depth, in order to keep the same order of the accuracy across all tree levels, the interpolation points K_l should be increased when the size of the cubes increase. From paper [12], we observe that when the electric sizes of the cubes are smaller than 0.2λ, for each variable, only 3 points are used as the interpolation points, a total of 27 interpolation points in a cube are sufficient to get a higher accuracy. While the size of the cube

increases to 2 λ, to constrain the error to 0.0135, 1000 interpolation points should be used, a large number that will drastically reduce the efficiency of the MLGFIM. Hence in this paper, the Green's function interpolation technique is just used in the near-field computation where the electric size of the cube is small. Few interpolation points can get a higher accuracy and efficiency. The accuracy and efficiency retains the same in a wide range when the electric size of the cube is smaller than 0.2 λ;

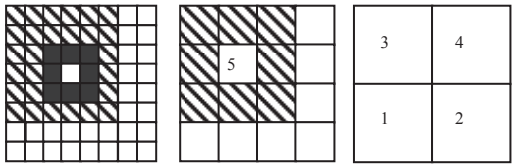

Fig. 1. A three level octree structure.

C. Lower-to-upper level interpolation

The MLGFIM is used in this paper to calculate part of the near-field defined in equation (5). Therefore, the near-field part in (5) can be written as the following form:

$$\bar{\bar{Z}}_{NF}\bar{x}=\sum_{l=L}^{1}\bar{\pi}_l+\bar{\bar{Z}}_{NF_L}\bar{x} \tag{11}$$

In (11), the term $\bar{\bar{Z}}_{NF}$ denotes the total near-field in the MLFMM which are calculated by MoM. The first term $\bar{\bar{Z}}_{NF_L}$ denotes part of the near-field in MLFMM which is calculated by MoM at the finest level of the MLFMM-MLGFIM, the second term $\sum_{l=L}^{1}\bar{\pi}_l$ is part of the near-field which is calculated by the MLGFIM. These two terms combined together form the total near-field in MLFMM. L is the number of levels which the MLGFIM technique is used for computing the near-field , The core in MLGFIM is to calculate the second term defined in equation (11). According to the tree structure used in MLFMM-MLGFIM, the near-field $\bar{\bar{Z}}_{NF_L}$ in (11) can be expressed as

$$\bar{\bar{Z}}_{NF_L}=\sum_{\substack{G_{n_L,L}\in \\ Neighbors \\ \text{of } G_{m_L,L}}}(\bar{\bar{Z}}_{m_L,L;n_L,L}\cdot\bar{x}_{n_L,L}) \tag{12-a}$$

$$\bar{\pi}_l=\sum_{\substack{G_{n_l,l}\in \\ \text{Interaction List} \\ \text{of } G_{m_l,l}}}\ \sum_{\substack{G_{n_{l+1},l+1}\subset \\ G_{n_l,l}}}\cdots\sum_{\substack{G_{n_L,L}\subset \\ G_{n_{L-1},L-1}}}\bar{\bar{A}}_{m_L,L;n_L,L}\cdot\bar{x}_{n_L,L} \tag{12-b}$$

The term $\bar{\bar{A}}_{m_L,L;n_L,L}$ in (12-b) can be written as the following for

$$\bar{\bar{A}}_{m_L,L;n_L,L}=\bar{\bar{W}}_{m_L,L}\cdot\bar{\bar{G}}_{m_L,L;n_L,L}\cdot\bar{\bar{W}}^{T}_{n_L,L} \tag{13}$$

For two well-separated cubes $G_{m_{L-1},L-1}$ and $G_{n_{L-1},L-1}$ at level $L-1$, cube $G_{m_L,L}$ is the child of cube $G_{m_{L-1},L-1}$ and cube $G_{n_L,L}$ is the child of cube $G_{n_{L-1},L-1}$, the Green's function matrix $\bar{\bar{G}}_{m_L,L;n_L,L}$ can be interpolated using the interpolation matrix $\bar{\bar{G}}_{m_{L-1},L-1;n_{L-1},L-1}$.

$$\bar{\bar{G}}_{m_L,L;n_L,L}=\bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L}\cdot\bar{\bar{G}}_{m_{L-1},L-1;n_{L-1},L-1}\cdot\bar{\bar{C}}_{n_{L-1},L-1;n_L,L} \tag{14}$$

$\bar{\bar{C}}_{n_{L-1},L-1;n_L,L}$ is the lower-to-upper interpolation matrix defined in [11], performing the Green's function interpolation recursively, (14) becomes

$$\bar{\bar{G}}_{m_L,L;n_L,L}=\bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L}\cdots\bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1}\cdot\bar{\bar{G}}_{m_l,l;n_l,l}\cdot\bar{\bar{C}}_{n_l,l;n_{l+1},l}\cdots\bar{\bar{C}}_{n_{L-1},L-1;n_L,L} \tag{15}$$

Substituting (15) and (13) into (12-b) gives (16)
Let

$$\bar{S}_{n_L,L}=\bar{\bar{W}}^{T}_{n_L,L}\cdot\bar{x}_{n_L,L}\quad\text{and}$$

$$\bar{S}_{n_l,l}=\sum_{G_{n_{l+1},l+1}\subset G_{n_l,l}}\bar{\bar{C}}_{n_l,l;n_{l+1},l}\cdot\bar{S}_{n_{l+1},l+1} \tag{17}$$

$\bar{S}_{n_l,l}$ here is just a symbol for recurrence without any means, Hence (16) can be rewritten as

$$\bar{\pi}_l=\bar{\bar{W}}_{m_L,L}\cdot\bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L}\cdots\bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1}\cdot\sum_{\substack{G_{n_l,l}\in\text{Interaction} \\ \text{List of } G_{m_l,l}}}\bar{\bar{G}}_{m_l,l;n_l,l}\cdot\bar{S}_{n_l,l} \tag{18}$$

Let

$$\bar{\alpha}_{m_l,l}=\sum_{\substack{G_{n_l,l}\in\text{Interaction} \\ \text{List of } G_{m_l,l}}}\bar{\bar{G}}_{m_l,l;n_l,l}\cdot\bar{S}_{n_l,l} \tag{19}$$

Hence

$$\bar{\pi}_l=\bar{\bar{W}}_{m_L,L}\cdot\bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L}\cdots\bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1}\cdot\bar{\alpha}_{m_l,l} \tag{20}$$

Substituting (20) into (11) and let

$$\bar{B}_{m_l,l} = \bar{\alpha}_{m_l,l} + \bar{\bar{C}}^{T}_{m_{l-1},l-1;m_l,l} \cdot \bar{B}_{m_{l-1},l-1} \ , \ \bar{B}_{m_1,1} = \bar{\alpha}_{m_1,1} \quad (21)$$

Substituting (21) into (20) recursively gives (22).

Hence the formula for MLGFIM algorithm is derived, what we want is to compute $\bar{B}_{m_L,L}$, which can be obtained using recurrence (22) from the top level to the finest level of the cluster tree. At the top level of the MLGFIM $\bar{B}_{m_1,1} = \bar{\alpha}_{m_1,1}$ is obtained using (22) from the finest level to the top level of the tree. Thus the procedure of the MVP of MLGFIM is similar to the MVP of MLFMM. (17) Is similar to the procedure in upward pass of MLFMM, (19) is similar to the procedure of shifting phase of MLFMM and (22) is similar to the procedure in downward pass of MLFMM. The difference is that our method uses Green's function interpolation instead of multipole expansion in each step. Equation (15) indicates that the Lagrange interpolation matrix of a cube can be interpolated using the Lagrange interpolation matrix of its child. For any other non-leaf cluster, we can directly use the contribution from its eight sons to obtain the result of MVP without any additional operations. This property is an important factor that enables us to reduce the complexity of MLGFIM.

$$\begin{aligned}
&\bar{\Pi} = \bar{\bar{W}}_{m_L,L} \cdot (\bar{\alpha}_{m_L,L} + \bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L} \cdot (\bar{\alpha}_{m_{L-1},L-1} + \cdots + \bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1} \cdot \\
&(\bar{\alpha}_{m_{l+1},l+1} + \bar{\bar{C}}^{T}_{m_l,l;m_l+1,l+1} \cdot (\bar{\alpha}_{m_l,l} \cdots + \bar{\bar{C}}^{T}_{m_2,2;m_3,3} \cdot (\bar{\alpha}_{m_2,2} + \bar{\bar{C}}^{T}_{ml,1;m2,2} \cdot \bar{\alpha}_{m_1,1}))))) \\
&= \bar{\bar{W}}_{m_L,L} \cdot (\bar{\alpha}_{m_L,L} + \bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L} \cdot (\bar{\alpha}_{m_{L-1},L-1} + \cdots + \bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1} \cdot \\
&(\bar{\alpha}_{m_{l+1},l+1} + \bar{\bar{C}}^{T}_{m_l,l;m_l+1,l+1} \cdot (\bar{\alpha}_{m_l,l} \cdots + \bar{\bar{C}}^{T}_{m_2,2;m_3,3} \cdot (\bar{\alpha}_{m_2,2} + \bar{\bar{C}}^{T}_{ml,1;m2,2} \cdot \bar{B}_{m_1,1}))))) \\
&= \bar{\bar{W}}_{m_L,L} \cdot (\bar{\alpha}_{m_L,L} + \bar{\bar{C}}^{T}_{m_{L-1},L-1;m_L,L} \cdot (\bar{\alpha}_{m_{L-1},L-1} + \cdots + \bar{\bar{C}}^{T}_{m_l,l;m_{l+1},l+1} \cdot \\
&(\bar{\alpha}_{m_{l+1},l+1} + \bar{\bar{C}}^{T}_{m_l,l;m_l+1,l+1} \cdot (\bar{\alpha}_{m_l,l} \cdots + \bar{\bar{C}}^{T}_{m_2,2;m_3,3} \cdot \bar{B}_{m_2,2})))) \\
&\cdots \\
&= \bar{\bar{W}}_{m_L,L} \cdot \bar{B}_{m_L,L}
\end{aligned} \quad (22)$$

D. Compression of the Green's function matrices using ACA, QR factorization, SVD

For any two well-separated cubes *m* and *n* at the same level, the Green's function matrix $G(r_m, r_n)$ is a $K \times K$ full matrix. Since, the Green's function matrix represents interactions between the interpolation points of two well-separated cubes, it is low rank. In order to reduce the computational complexity of MLGFIM, the ACA [13-14], QR factorization [15-16], and SVD, are used to compress the Green's function matrix as data sparse representation, which brings a great advantage in the MVP operation $\bar{\bar{G}}_{m_l,l;n_l,l} \cdot \bar{S}_{n_l,l}$.

Let the $K \times K$ rectangular matrix $G(r_m, r_n)$ represent the interactions between the interpolation points of two well-separated cubes m and n, the ACA allows the low rank Green's function matrix $G(r_m, r_n)$ to be represented by only a few rows and columns of $G(r_m, r_n)$ to obtain the numerical representation from namely,

$$\tilde{G}^{K\times K} = A^{K\times r}(B^{K\times r})^{H} \quad (23)$$

where the number of terms r is much less than K, $A^{K\times r}$ and $B^{K\times r}$ are two dense rectangular matrices. The goal of the ACA is to achieve $\left\|R^{K\times K}\right\| = \left\|G^{K\times K} - \tilde{G}^{K\times K}\right\| \le \varepsilon \left\|G^{K\times K}\right\|$ for a given tolerance ε, where R is termed as the error matrix. $\|.\|$ refer to the matrix Frobenus norm. If $r \ll \min(m,n)$, then a significant reduction in MVP can be accomplished. For the matrix U and V, we can continue to use QR decomposition technique to compress it

$$A^{K\times r} = Q_1^{K\times r} R_1^{r\times r} \quad (24\text{-}1)$$

$$B^{K\times r} = Q_2^{K\times r} R_2^{r\times r} \quad (24\text{-}2)$$

Then, the matrix $\tilde{G}^{K\times K}$ can be expressed as the following form

$$\begin{aligned}
\tilde{G}^{K\times K} &= A^{K\times r}(B^{r\times r})^{H} = Q_1^{K\times r} R_1^{r\times r} (Q_2^{K\times r} R_2^{r\times r})^{H} \\
&= Q_1^{K\times r} R_1^{r\times r} (R_2^{r\times r})^{H} (Q_2^{K\times r})^{H}
\end{aligned} \quad (25)$$

Here, we let $W^{r\times r} = R_1^{r\times r}(R_2^{r\times r})^{H}$, using singular value decomposition (SVD),

$$W^{r\times r} = U^{r\times r} S^{r\times r} V^{r\times r} \quad (26)$$

where *U* and *V* are orthonormal matrices, and *S* is the diagonal matrix whose elements are the nonnegligible singular values of $W^{r\times r}$, it can be written as $S = diag(\sigma_1, \sigma_2, \sigma_3 \ldots, \sigma_r)$. We discard those normalized values which fall below the threshold; typically chosen threshold to be 10^{-3}, the columns of *U* and *V* corresponding to negligible singular values of *S* are discarded. Then, the matrix $\tilde{G}^{K\times K}$ can be written as the following

$$\begin{aligned}
\tilde{G}^{K\times K} &= Q_1^{K\times r} R_1^{r\times r} (R_2^{r\times r})^{H} (Q_2^{K\times r})^{H} \\
&\cong Q_1^{K\times r} U^{r\times r1} S^{r1\times r1} V^{r1\times r} (Q_2^{K\times r})^{H}
\end{aligned} \quad (27)$$

Let

$$A^{K\times r1} = Q_1^{K\times r} R_1^{r\times r1} S^{r1\times r1} \text{ and } B^{r1\times K} = V^{r1\times r}(Q_2^{K\times r})^H$$

then

$$\widetilde{G}^{K\times K} = A^{K\times r1} \times B^{r1\times K} \tag{28}$$

Usually the compressed matrices rank $r1$ is much smaller than the number of interpolation points K. This brings a great advantage in the MVP because

$$\overline{\overline{G}}^{K\times K} \cdot \overline{S}^K = A^{K\times r1} \times B^{r1\times K} \cdot \overline{S}^K \tag{29}$$

When $r1 << K$, to calculate $A^{K\times r1} \times B^{r1\times K} \cdot \overline{S}^K$ is much faster than to calculate $\overline{\overline{G}}^{K\times K} \cdot \overline{S^K}$. Table 1 lists the corresponding average numerical ranks of Green's function matrices with different sizes of the cube. We can see that the corresponding numerical ranks are very small. Thus, a high compression of the Green's function matrices is obtained.

III. NUMERICAL RESULTS

In this section, three examples are presented to demonstrate the benefits of the proposed method. All the simulations are performed on a computer with 2.8GHz CPU and 2 GB RAM. The terminating tolerances of the ACA and SVD are set as $\varepsilon = 0.001$ and $\sigma = 0.001$, respectively. The resulting linear systems are solved iteratively by the GMRES (30) solver with a relative residual of 10^{-3}.

First, the proposed method is used to analyse scattering from a PEC sphere of radius 0.5 λ, its surface is discretized with 6312, 11649, and 25944 unknowns, respectively. The finest cube size is 0.25 λ in MLFMM, two-level MLGFIM are added to calculate part of the near-field of the MLFMM. The finest cube size is 0.0625 λ in MLFMM-MLGFIM. Figure 2 shows the bistatic radar cross-section (RCS) results obtained from the MIE series and the MLFMM-MLGFIM. It can be seen from Fig. 2, that the result from the MLFMM-MLGFIM has good agreement with the MIE series. Table II lists the near-field memory requirement, the matrix filling time and MVP time of MLFMM and MLFMM-MLGFIM for different discretizations. The time and memory requirement in computing the near-field impedance matrix in MLFMM-MLGFIM includes two parts. The first part denotes the time and memory requirement in computing the part of the near-field matrix by MoM, while the second part denotes the time and memory requirement in computing part of the near-field matrix by MLGFIM. It can be seen from Table 2 that the memory requirement and filling time of the near-field matrix in MLFMM-MLGFIM is significantly reduced as the number of unknowns increases compared with MLFMM.

The second example is a PEC ogive, whose length and maximum radius is 2λ and 0.5 λ. The ogive is discretized with 11874 and 18876 unknowns, respectively. 1-level MLGFIM is added to calculate part of the near-field interaction when the number of unknowns is 11874, the finest cube size is 0.125 λ. 2-level MLGFIM is added to calculate part of the near-field interaction when the number of unknowns is 18876, the finest cube size is 0.0625 λ. Figure 3 is the bistatic RCS of the ogive computed by MLFMM and MLFMM-MLGFIM. It can be seen from Fig. 3 that the proposed method agrees well with the MLFMM results. Table 3 lists the near-field memory requirement, the near-filed impedance matrix filling time, and the MVP time needed by MLFMM-MLGFIM and MLFMM for different discretizations. It can be seen from Table 3 that the memory requirement by the MLFMM-MLGFIM can be saved by a factor of 4.1 with one-level MLGFIM when the number of unknowns is 11874. The flaw for the MLFMM-MLGFIM is that the MVP time is 1.9 while the MLFMM is 1.07s. The near-filed matrix filling time and memory requirement can be saved by a factor of 16.8 and 14.2 with two-level MLGFIM when the number of unknowns is 18876. Again, it can be seen that the MLFMM-MLGFIM can greatly reduce the near-filed memory requirement and the matrix filling time compared with MLFMM.

The last example is a VIAS structure as shown in Fig.4, the electric size of the VIAS structure is $1.2\lambda \times 1\lambda \times 1\lambda$, it is discretized with 10609 and 15305 unknowns, respectively. For MLFMM, a 2-level division is used since the finest cube size is 0.30 λ. For MLFMM-MLGFIM, a 4-level division is used. The finest cube size is 0.075 λ. A 2-level MLGFIM is added to calculate part of the near-field interaction. Good agreement is achieved as shown in Fig. 4. Table 4 lists the comparison of the near-field memory requirement, the near-filed matrix filling time, and the MVP time between the MLFMM-MLGFIM and MLFMM for the two above discretizations. It can be found that the MLGFIM can extremely decrease the near-filed

matrix filling time when compared with MLFMM, MLFMM-MLGFIM can save much of the memory requirement by a factor of 9.9 with 10609 unknowns and 10.3 with 15305 unknowns for the near-field. When the number of unknowns is 15305, the MVP time in each iteration step in the MLFMM-MLGFIM is less than that in MLFMM. This is because the number of elements in the near-field impendence matrix is greatly reduced by the MLGFIM. The result of Table 4 indicates again that the matrix filing time and memory requirement can be greatly reduced with the MLGFIM in MLFMM in the near-field computation.

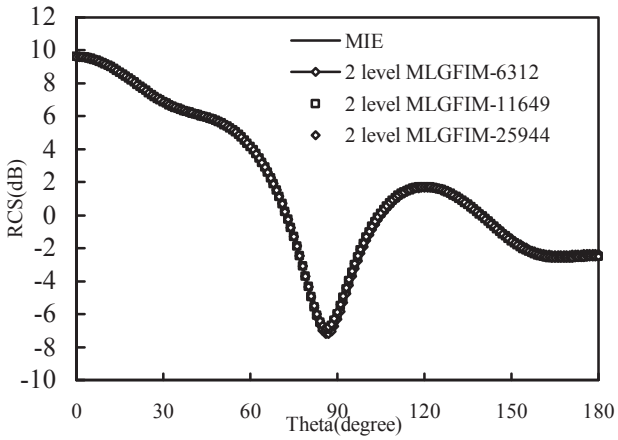

Fig. 2. RCS of a PEC sphere obtained from the MIE series and the MLFMM-MLGFIM.

IV. CONCLUSIONS

In this paper, the MLGFIM is introduced in MLFMM to solve electromagnetic scattering problems of the objects with fine structures. It is found that with MLGFIM we can continue to subdivide the cube until the number of unknowns in each cube is less or equal to a required number regardless of the cube size. Several examples have demonstrated that with MLGFIM the near-field memory requirement is greatly reduced in MLFMM-MLGFIM compared with MLFMM without compromising the accuracy. Moreover, the ACA, QR factorization, SVD are applied to compress the low rank Green's function matrix for speeding up the MVP in MLFMM-MLGFIM. Therefore, the MLGFIM is an efficient augment for MLFMM.

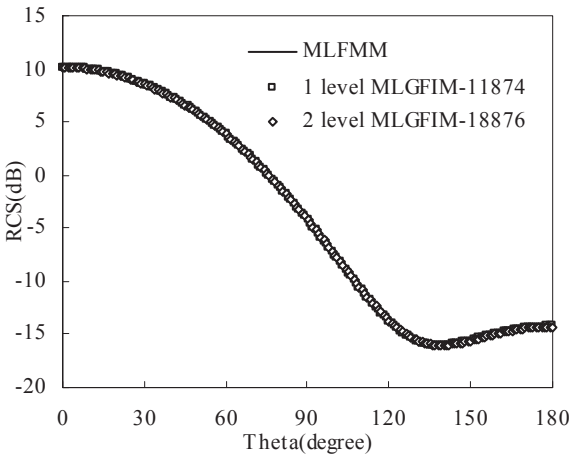

Fig. 3. The bistatic RCS from MLFMM and MLFMM-MLGFIM for the PEC ogive example.

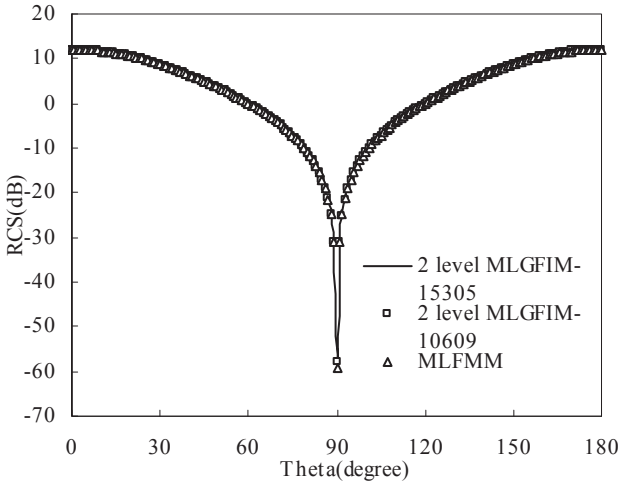

Fig. 4. RCS of a VIAS structure obtained from MLFMM and MLFMM-MLGFIM.

ACKNOWLEDGMENT

The authors would like to thank the support of National Key Laboratory Foundation of China (No: 9140C5305021006).

$$\overline{\pi}_l = \sum_{\substack{G_{n_l,l}\in \text{Interaction} \\ \text{List of } G_{m_l,l}}} \sum_{\substack{G_{n_{l+1},l+1} \\ \subset G_{n_l,l}}} \cdots \sum_{\substack{G_{n_L,L} \\ \subset G_{n_{L-1},L-1}}} \overline{\overline{W}}_{m_L,L} \cdot \overline{\overline{G}}_{m_L,L;n_L,L} \cdot \overline{\overline{W}}^T_{n_L,L} \cdot \overline{x}_{n_L,L}$$

$$= \sum_{\substack{G_{n_l,l}\in \text{Interaction} \\ \text{List of } G_{m_l,l}}} \sum_{\substack{G_{n_{l+1},l+1} \\ \subset G_{n_l,l}}} \cdots \sum_{\substack{G_{n_L,L} \\ \subset G_{n_{L-1},L-1}}} \overline{\overline{W}}_{m_L,L} \cdot \overline{\overline{C}}^T_{m_{L-1},L-1;m_L,L} \cdots \overline{\overline{C}}^T_{m_l,l;m_{l+1},l+1} \cdot \overline{\overline{G}}_{m_l,l;n_l,l} \cdot \overline{\overline{C}}_{n_l,l;n_{l+1},l} \cdots \overline{\overline{C}}_{n_{L-1},L-1;n_L,L} \cdot \overline{\overline{W}}^T_{n_L,L} \cdot \overline{x}_{n_L,L} \quad (16)$$

$$= \overline{\overline{W}}_{m_L,L} \cdot \overline{\overline{C}}^T_{m_{L-1},L-1;m_L,L} \cdots \overline{\overline{C}}^T_{m_l,l;m_{l+1},l+1} \cdot \sum_{\substack{G_{n_l,l}\in \text{Interaction} \\ \text{List of } G_{m_l,l}}} \overline{\overline{G}}_{m_l,l;n_l,l} \cdot \sum_{\substack{G_{n_{l+1},l+1} \\ \subset G_{n_l,l}}} \overline{\overline{C}}_{n_l,l;n_{l+1},l} \cdots \sum_{\substack{G_{n_L,L} \\ \subset G_{n_{L-1},L-1}}} \overline{\overline{C}}_{n_{L-1},L-1;n_L,L} \cdot \overline{\overline{W}}^T_{n_L,L} \cdot \overline{x}_{n_L,L}$$

Table 1: Corresponding rank of the Green's function with different size of the cube

Cube size		d=1.0	d=0.5	d=0.25	d=0.125
Interpolation points		8×8×8	6×6×6	4×4×4	3×3×3
# of entries in G		262144	47524	4096	729
Threshold $\varepsilon = 0.01$ $\sigma = 0.01$	Numerical rank	11	6	5	4
	# of entries in Q and R	11264	2616	640	216
Threshold $\varepsilon = 0.01$ $\sigma = 0.001$	Numerical rank	17	11	9	8
	# of entries in Q and R	17408	4752	1152	432
Threshold $\varepsilon = 0.001$ $\sigma = 0.01$	Numerical rank	11	8	5	4
	# of entries in Q and R	11264	3456	640	216
Threshold $\varepsilon = 0.001$ $\sigma = 0.001$	Numerical rank	17	12	9	8
	# of entries in Q and R	17408	5184	1152	432

Table 2: The near-field memory, the matrix filling time and one MVP time of MLFMM、MLFMM-MLGFIM for different numbers of unknowns

Unknowns	Methods	CPU time for Matrix filling	CPU time for each MVP	Memory
6312	MLFMM	35.2 s	0.22 s	70 Mb
	MLFMM-MLGFIM	2.1 s+0.57 s	3.65 s	4.1 Mb+ 2.5 Mb
11649	MLFMM	124.3 s	0.81 s	240 Mb
	MLFMM-MLGFIM	6.9 s+0.96 s	3.9 s	14 Mb+ 4.7 Mb
25944	MLFMM	641.1 s	3.1 s	1191 Mb
	MLFMM-MLGFIM	32.5s+2.0 s	4.4 s	69 Mb+ 10 Mb

Table 3: The near-field memory, the matrix filling time and one MVP time of MLFMA、MLFMM-MLGFIM for different numbers of unknowns

Unknowns	level	Methods	CPU time for Matrix filling	CPU time for each MVP	Memory
11874	3	MLFMM	206.9s	1.07s	404Mb
	4	MLFMM-MLGFIM	46.5s+0.32s	1.9s	89Mb + 9Mb
18876	3	MLFMM	533.5s	2.5s	1019 Mb
	5	MLFMM-MLGFIM	31.2s+0.56s	2.9s	56 Mb + 15.5 Mb

Table 4: The near-field memory, the matrix filling time and one MVP time of MLFMA, MLFMM-MLGFIM for different numbers of unknowns

Unknowns	level	Methods	CPU time for Matrix filling	CPU time for each MVP	Memory
10609	2	MLFMM	214.2s	1s	388Mb
	4	MLFMM-MLGFIM	20.9s+0.33s	1.48s	35Mb + 4 Mb
15305	2	MLFMM	451.4s	2.04s	805Mb
	4	MLFMM-MLGFIM	42s+0.43s	1.59s	72Mb+6Mb

REFERENCES

[1] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape," *IEEE Trans. Antennas and Propag*., vol. 30, no. 3, pp. 409-418, May. 1982.

[2] B. J. Fasenfest, F. Capolino, and D. R. Wilton, "Preconditioned GIFFT: A Fast MoM Solver for Large Arrays of Printed Antennas,"*ACES Journal*, vol.21, no.3, pp. 276-283, November. 2006.

[3] J. M. Song and W. C. Chew, "Fast Multipole Method Solution using Parametric Geometry," *Microwave and Optical Technology Letters*, vol. 7, no. 16, pp. 760-765,1994.

[4] C. Guo and T. H. Hubing, "Development and Application of a Fast Multipole Method in a Hybrid FEM/MoM Field Solver," *Applied Computational Electromagnetic Society (ACES) Journal*, vol. 19, no. 3, pp. 126- 134, November 2004.

[5] R. Coifman, V. Rokhlin, and S. M. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," *IEEE Trans. Antennas and Propag., Mag*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[6] J. M. Song, C. C. Lu., and W .C. Chew, "Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects," *IEEE Trans. Antennas and Propag.*, 45(10), pp. 1488-1493, 1997.

[7] C. C. Lu and W. C. Chew, "A Multilevel Algorithm for Solving Boundary Integral Equations of Wave Scattering," *Microwave and Optical Technology Letters*, vol. 7, pp. 466-470, July 1994.

[8] J. M. Song, and W. C. Chew, "Multilevel Fast-Multipole Algorithm for Solving Combined Field Integral Equation of Electromagnetic Scattering," *Microwave and Optical Technology Letters*, vol. 10, pp. 14-19, Sept. 1995.

[9] H. Zhao, J. Hu, and Z. Nie, "Parallelization of MLFMA with Composite Load Partition Criteria and Asynchronous Communication," *Applied Computational Electromagnetic Society (ACES) Journal*, vol. 25, no. 2, pp. 167-173, February 2010.

[10] H. Fangjing, N. Zaiping, and H. Jun, "An Efficient Parallel Multilevel Fast Multipole Algorithm for Large-scale Scattering Problems," *Applied Computational Electromagnetic Society (ACES) Journal*, vol. 25, no. 4, pp. 381-387, April 2010.

[11] H. G. Wang, C. H. Chan, and L. Tsang, "A New Multilevel Green's Function Interpolation Method for Large-Scale Low-Frequency EM Simulations," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 9, pp. 1427-1443, Sept. 2005.

[12] H. G. Wang and C. H. Chan, "The Implementation of Multilevel Green's Function Interpolation Method for Full-Wave Electromagnetic Problems," *IEEE Trans. Antennas and Propag.*, vol. 55, no. 5, pp. 1348-1358, May 2007.

[13] L. Li, H. G. Wang, and C. H. Chan, "An Improved Multilevel Green's Function Interpolation Method with Adaptive Phase Compensation," *IEEE Trans. Antennas and Propag.*, vol. 56, no. 5, pp. 1381-1393, May 2008.

[14] Y. Shi, H. G. Wang, L. Li, and C. H. Chan, "Multilevel Green's Function Interpolation Method for Scattering from Composite Metallic and Dielectric Objects," *J. Opt. Soc. Am. A*, vol. 25, no. 10, pp. 2535-2548, October 2008.

[15] W. Chai and D. Jiao, "An H2-Matrix-Based Integral-Equation Solver of Linear-Complexity for Large-Scale Electromagnetic Analysis," *2008 Asia Pacific Microwave Conference*, 4 pages, Dec. 2008.

[16] W. Chai and D. Jiao, "An H2-Matrix-Based Integral-Equation Solver of Reduced Complexity and Controlled Accuracy for Solving Electrodynamic Problems," *IEEE Trans. Antennas and Propag.*, vol. 57, no. 5, pp. 3147-3159, Oct. 2009.

[17] L. J. Jiang, W. C. Chew, "A Mixed-Form Fast Multipole Algorithm," *IEEE Trans. Antennas and Propag.*, vol. 53, no. 12, pp. 4145-4156, Dec. 2005.

[18] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "Application of the Multilevel Adaptive Cross-Approximation on Ground Plane Designs," *IEEE Int. Symp. Electromagnetic Compatibility*, Santa Clara, CA, pp. 124–127, 2004.

[19] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, "Reorthogonalization and

Stable Algorithms for Updating the Gram-Schmidt QR Factorization," *Math. Comp*, vol. 30, pp. 772–795, 1976.

Ming Chen was born in Anhui, China. He received the B.S. degree in Physics from Anhui University in 2006, and is currently working toward the Ph.D. degree at Nanjing University of Science and Technology (NJUST), Nanjing, China. His current research interests include computational electromagnetics, antennas, and electromagnetic scattering and propagation.

Ru-Shan Chen was born in Jiangsu, P. R. China. He received his B.Sc. and M.Sc. degrees from the Dept. of Radio Engineering, Southeast University, in 1987 and in 1990, respectively, and his Ph.D. from the Dept. of Electronic Engineering, City University of Hong Kong in 2001. He became a Teaching Assistant in 1990 and a Lecturer in 1992. Since September 1996, he has been a Visiting Scholar with the Department of Electronic Engineering, City University of Hong Kong, first as Research Associate, then as a Senior Research Associate in July 1997, a Research Fellow in April 1998, and a Senior Research Fellow in 1999. From June to September 1999, he was also a Visiting Scholar at Montreal University, Canada. In September 1999, he was promoted to Full Professor and Associate Director of the Microwave & Communication Research Center in NJUST and in 2007, he was appointed Head of the Dept of Communication Engineering, Nanjing University of Science & Technology. His research interests include microwave and millimeter-wave systems, measurements, antenna, RF-integrated circuits, and computational electromagnetics. He is a Senior Member of the Chinese Institute of Electronics (CIE). He received the 1992 third-class science and technology advance prize given by the National Military Industry Department of China, the 1993 third class science and technology advance prize given by the National Education Committee of China, the 1996 second-class science and technology advance prize given by the National Education Committee of China, and the 1999 first-class science and technology advance prize given by JiangSu Province as well as the 2001 second-class science and technology advance prize. At NUST, he was awarded the Excellent Honor Prize for academic achievement in 1994, 1996, 1997, 1999, 2000, 2001, 2002, and 2003. He has authored or co-authored more than 200 papers, including over 140 papers in international journals. He is the recipient of the Foundation for China Distinguished Young Investigators presented by the National Science Foundation (NSF) of China in 2003. In 2008, he became a Chang-Jiang Professor under the Cheung Kong Scholar Program awarded by the Ministry of Education, China.

Xiaoqing Hu was born in Hubei, China, in June 1981. He received the B.S. degree and M.S. degree in Applied Mathematics from Jilin University, Jilin, China, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree at Nanjing University of Science and Technology, Jiangsu, China. His research interests include integral equation and its fast methods.