Multilevel Green’s Function Interpolation Method: An Interpolation-based Solver for Efficient Analysis of Large-scale Electromagnetic Problems

Peng Zhao\(^{(1,2)}\), Da Qing Liu\(^{(2)}\) and Chi Hou Chan\(^{(2)}\)

\(^{(1)}\)Electronics and Information College, Hangzhou Dianzi University, Hangzhou, China
\(^{(2)}\)State Key Laboratory of Millimeter Waves, City University of Hong Kong, Hong Kong SAR, China

(Email: pzhao frank@gmail.com)

Abstract—An efficient algorithm for fast field evaluation, namely multilevel Green’s function interpolation method (MLGFIM), has been successfully applied to solve electromagnetic problems for both low frequency and full-wave. Further refinements, including new interpolation scheme, QR factorization, and FFT technique, have been applied to accelerate this algorithm and extend the problem scale that MLGFIM can solve. In this paper, we present a review of different techniques in MLGFIM for the fast evaluations of large-scale electromagnetic problems. The applications of simulating periodic structures and layered media have been accelerated by MLGFIM as well. Some numerical examples are given in this paper to demonstrate the accuracy and efficiency of this algorithm.

Index Terms—Method of Moments (MoM), Green’s Function Interpolation, Fast Solver, Parallel Computing, FFT.

I. INTRODUCTION

Calculation of antenna radiation pattern, analysis of radar cross section (RCS), and extraction of circuit parameters are often performed using integral equation-based computational techniques, e.g., the method of moment (MoM) [1]. However, the computational requirements for MoM are very high, and various fast numerical methods have been proposed to accelerate the computation, for instance, multilevel fast multipole algorithm (MLFMA) [2]-[4], pre-corrected fast Fourier transform (FFT) [5],[6], adaptive integral method (AIM) [7],[8], and sparse matrix canonical grid (SMCG) [9],[10]. The first method uses multipole expansion to approximate the far field interactions, while the later three methods apply FFT for fast field evaluation. Although these methods calculate the matrix-vector multiplication indirectly and thus significantly reduce the computational complexity, each of the methods has its own weaknesses. The FFT-based methods are most suitable for densely-packed structures, in which the unknowns are associated with most of the grid points. The PFFT approach employs the bi-linear interpolation method, and it is not suitable for analysis of full-wave electromagnetic problems as bi-linear interpolation is not accurate for the rapidly-varying kernels in full-wave applications. On the other hand, the multipole-based algorithms depend on the type of Green’s function used for the specific problems and they become cumbersome when dealing with structures embedded in a multilayered medium. When the simulated structure changes, for instance, insert an additional layer of dielectric substrate, the problem has to be reformulated.

Some interpolation-based fast algorithms have the potential to address many of the aforementioned weaknesses. In [11], Brandt developed a multilevel interpolation approach, in which a softened kernel and phase compensation is employed when the kernel is oscillatory. Non-uniform grid (NG) [12], [13] and multilevel NG [14],[15] algorithms, as kernel independent algorithms, are proposed for fast field evaluation. With the technique of phase and amplitude compensations, NG-based algorithms sample the compensated field using a non-uniform polar grid. The NG and multilevel NG methods achieve computational complexities of \(O(N^{1/2})\) and \(O(N \log N)\), respectively. Adaptive cross approximation (ACA) [16],[17] employs interpolation approach directly on low-rank sub-matrices denoting the far field interaction, and thus this method is not difficult to implement. The \(H^2\) matrix-based method is proposed in [18],[19]. It applies multilevel interpolation with Lagrange interpolation function and Tartan grid.

Another kernel-independent interpolation algorithm, viz. multilevel Green’s function interpolation method (MLGFIM), has been developed to solve quasi-static [20]-[22] and full-wave electromagnetic problems [23],[24]. The application of an octa-tree structure and non-requisite of a softened kernel distinguish the MLGFIM with the algorithm in [11]. For full-wave problems, instead of using Lagrange...
interpolation as in the quasi-static problems, radial basis function (RBF) interpolation is employed. Lagrange interpolation method can not be directly applied for the approximation of the full-wave Green’s function, unless phase compensation technique is incorporated [25]. The RBF interpolation has excellent approximation properties [26], and we find that if the object is not extremely large, the full-wave Green’s function can be interpolated without the compensation technique, which is different from the NG-based algorithm. Although ACA is flexible for solving problems with a complex medium, it is difficult to enhance the efficiency by using the translational invariance characteristics of the Green’s functions in some problems, e.g. the multilayered planar problem. In addition, the multilevel structure of MLGFIM also differs from ACA. Compared with the $H^2$ matrix-based method, the applications of RBF interpolation and staggered Tartan grid (STG) make the number of interpolation points in MLGFIM increases more slowly. This is because the interpolation points at the coarser levels do not coincide with those at the finer levels in the MLGFIM, viz., the volumetric density of interpolation point at the coarser levels is lower than that at the finer levels. Moreover, some novel techniques used in MLGFIM, e.g. QR factorization [27] and hybrid quasi-2D/3D multilevel partitioning approach [24], distinguish itself from other interpolation-based methods. In addition, the applications of peer-level and lower-to-upper-level interpolations make the MLGFIM very efficient for the analysis of large-scale problems, and the computational complexity of 3-D full-wave MLGFIM is $O(N \log N)$ [23].

Although so many fast algorithms have been developed, there are still many challenges in the electromagnetic simulations. For instance, electromagnetic enhancement of surface enhanced Raman scattering (SERS) attributed to the enhanced E-fields in the vicinity of plasmonic nanostructures is the major approach of SERS enhancement [28],29]. When the exciting electromagnetic radiation approaches the resonance frequency, the enhancement factor which is proportional to the fourth power of the gain in the electromagnetic field becomes very difficult to evaluate. To accurately calculate the electromagnetic field, the nanoparticles must use deep sub-wavelength discretization. Due to interpolation-based algorithm, MLGFIM is capable of dealing with the low-frequency and multibody structure. In addition, some thin strip with high dielectric constant has been used to promote the magnetic emission in quantum emitters [30],[31], and we found when the emitter is close to the scatters, the near-field is difficult to be accurately computed as well. The high-contrast material makes the whole structure extremely in-homogenous, and thus deep sub-wavelength mesh is also required. Moreover, the analysis of metasurfaces, e.g. holographic metasurfaces [32],[33] which excites surface waves with the desirable field distributions, is challenging. For many of the holographic structures, the size and shape of each sub-scatter which stuck on a substrate may be different, so the computational burden is very large. This problem can be released by adopting multilayered medium Green’s function, and MLGFIM is able to be implemented based on this complex Green’s function.

In this paper, various techniques for the MLGFIM to solve electrically large electromagnetic problems are reviewed. The MLGFIM used for accelerating the matrix-vector multiplication is introduced first. QR factorization technique is employed to compress the matrix storage. In order to make the MLGFIM more efficient, different radial basis functions (RBFs) are compared and interpolation grid patterns are designed so that the MLGFIM could be accelerated by applying newly proposed interpolation schemes. Moreover, the technique of fast Fourier transform (FFT), which has been combined with MLFMA [34],[35], could also be applied to reduce the translation time in MLGFIM. The parallelized MLGFIM versions based on OpenMP and message passing interface (MPI) techniques have been realized to make this fast solve more efficient for the large-scale electromagnetic problems. Since the MLGFIM is a kernel-independent method, it is easy to use MLGFIM for the analysis of periodic structures and multilayer problems. Numerical results demonstrate the good accuracy and efficiency of the above implementations of the MLGFIM.

II. DESCRIPTION OF MLGFIM

For the 3-D free-space aperiodic problems, it is always described by electric-field integral equation (EFIE), magnetic-field integral equation (MFIE) and combined-field integral equation (CFIE). With boundary conditions imposed on the surface of the body, another integral equation, i.e. Poggio-Miller-Chang-Harrington-Wu-Tsai (PMCHWT) integral equations, could be used to analyze 3-D composite metallic and dielectric bodies with surface discretization. Using Galerkin method, the aforementioned integral equations are converted to $N \times N$ dense matrix equations, viz.,

$$\tilde{A} \tilde{x} = \tilde{b},$$

where $*$ can represent EFIE, MFIE, CFIE or PMCHWT. For simplicity and clarity, we use EFIE as an example. The entries in $N \times N$ matrix $\tilde{A}^{\text{EFIE}}$ and $N \times 1$ vector $\tilde{b}^{\text{EFIE}}$ are written as:

$$A_{ij} = \int_{s_i} \int_{s_j} ds_1 \left( j \left( \mathbf{r} \right) \cdot \mathbf{j} \left( \mathbf{r}' \right) \right)$$

$$- \frac{1}{\omega^2 \mu_0 \epsilon_0} \mathbf{V} \cdot \mathbf{j} \left( \mathbf{r} \right) \mathbf{V}' \cdot \mathbf{j} \left( \mathbf{r}' \right) G \left( \mathbf{r}, \mathbf{r}' \right)$$

and

$$b_i = \frac{4 \pi i}{\alpha \mu_0} \int_{s_i} ds \mathbf{j} \left( \mathbf{r} \right) \cdot \mathbf{E}^{inc} \left( \mathbf{r} \right)$$

where $\mathbf{r}, \mathbf{r}'$, $\omega$, $\epsilon_0$, $\mu_0$, $\mathbf{j} \left( \mathbf{r} \right)$, $G \left( \mathbf{r}, \mathbf{r}' \right)$ and $\mathbf{E}^{inc} \left( \mathbf{r} \right)$ are the observation point, source point, angular frequency, free space permittivity and permeability, basis function, interaction function, and incident electric field, respectively.

To apply MLGFIM, (2) is decomposed into scalar form as:

$$A_{ij} = A_{ij}^x + A_{ij}^y + A_{ij}^z - A_{ij}^d$$

where

$$A_{ij}^x = \int_{s_i} \int_{s_j} ds_1 \mathbf{j} \left( \mathbf{r} \right) \cdot \mathbf{E}^{inc} \left( \mathbf{r} \right)$$

$# = x, y$ or $z$
\[ A_j = \frac{1}{\omega^2} \mu_0 \int_{V_j} ds \int_{V_j} ds' \nabla \cdot j(r) \nabla \cdot j(r') G(r, r') \] \tag{6}

Each scalar part \( A_{ij} \) in (5) and (6) has the same form as:
\[ A_{ij} = \int_{V_j} ds \int_{V_j} ds' \tau_i(r) \zeta_j(r') G(r, r') \] \tag{7}

where \( \tau_i(r) \) and \( \zeta_j(r') \) are related to the weighting function and the basis function, respectively.

![Multilevel tree](image)

**Fig. 1.** (a) 2-D pictorial representation of 3-D multilevel division in which the interaction list of box \( n_2 \) at level 2 is marked with \( m_2 \), and the interaction list of box \( n_1 \) at level 3 is marked with \( m_3 \). (b) 2-D pictorial representation of 3-D locations of the interpolation points (white circles) and interpolated points (black circles) are in the boxes \( m_2 \) and \( n_2 \).

In [24] and [36], it is proven that PMCHWT and CFIE which are associated with both EFIE and MFIE can also be decomposed into the same form as (7). The interaction functions \( G(r, r') \) corresponding to the EFIE and MFIE can be expressed as:
\[ G(r, r') = \begin{cases} 
\frac{e^{i(\mathbf{kr} - \mathbf{kr}')}}{|\mathbf{r} - \mathbf{r}'|} & \text{(for EFIE)} \\
\frac{e^{i(\mathbf{kr} - \mathbf{kr}')}}{|\mathbf{r} - \mathbf{r}'|} - \frac{1}{i|\mathbf{r} - \mathbf{r}'|} & \text{(for MFIE)}
\end{cases} \] \tag{8}

For the far-field interactions, instead of directly calculating the function \( G(r, r') \), MLGFIM use interpolation method in a multilevel tree to quickly approximate this function. The multilevel tree is constructed by enclosing the object into a box (level 0) and recursively dividing the large box into sub-boxes. Box \( m_i \) is the neighbor/interaction list of box \( n_l \) at level \( i \) if \( m_i \) is near/well separated from \( n_l \) and their parents are neighbors of each other, as shown in Fig. 1(a). If box \( m_l \) is the interaction list of box \( n_l \) at level \( l \), the interaction function between these two boxes could be obtained by interpolation method, as shown in Fig. 1(b), viz.:
\[ G(r, r') = \sum_{i,j} \omega_{i,i'} \omega_{j,j'} (r) G(r_{i,j}, r_{i,j'}) \] \tag{9}

where \( r_{i,j} \) and \( r_{i,j'} \) are the \( i \)-th and \( j \)-th interpolation points in field box \( m_i \) and source box \( n_j \). \( \omega_{i,i'}(r) \) and \( \omega_{j,j'}(r) \) are the \( i \)-th and \( j \)-th interpolation functions, respectively. \( K_l \) is the number of interpolation points at level \( l \).

Moreover, using the interpolation method, \( \overline{G} \) and \( \overline{G} \) can also be interpolated using \( \overline{G} \) and \( \overline{G} \) [20], viz.:
\[ \begin{cases} 
G_{m_i,n_j} = C_{m_i,n_j} G_{m_i,n_j} \\
\overline{G}_{m_i,n_j} = \overline{G}_{m_i,n_j}
\end{cases} \] \tag{10}

where \( m_{i+1} \subset m_i \), \( n_{i+1} \subset n_i \), and matrix \( \overline{G} \) is named as lower-to-upper-level interpolation matrix. If the box \( m_1 \) is interaction list of box \( n_L \) at level \( L \), \( G(r, r') \) at the finest level \( L \) can be obtained using (9) and (10), viz.:
\[ G(r, r') = \overline{G}_{m_1,n_L} (r) \cdot \overline{G}_{m_1,n_L} \\
\] \tag{11}

Substituting (11) into (7) gives:

\[ \bar{A}_j = \left[ \int ds \tau_i(r) \overline{G}_{m_i,n_j} \right] C_{m_i,n_j} \cdot \overline{G}_{m_i,n_j} \\
\] \tag{12}

It should be mentioned that if box \( m_L \) is the interaction list of box \( n_L \) at level \( L \), there is no lower-to-upper-level interpolation matrix in (12), and it becomes the conventional peer-level interpolation. Subsequently, the sub-matrix \( \overline{A}_{m,n} \) can be rewritten as:
\[ \overline{A}_{m,n} = \begin{bmatrix} 
\overline{T}_{m,n_L} (r') \\
\vdots \\
\overline{T}_{m,n_L} (r') \\
\end{bmatrix} \] \tag{13}

where \( N_m \) and \( N_n \) denote the number of unknowns in boxes \( m_L \) and \( n_L \), respectively. From (13), it is observed that only the
interpolation matrices $\widehat{T}_{m_2}$ and $\widehat{T}_{n_2}$ at level $L$ should be calculated. Assuming there are $P_L$ boxes at level $L$, the MLGFIM algorithm can be finally shown as Fig. 2.

Fig. 2. Flow chart of the implementation of MLGFIM.

Considering the interaction function matrix $\overline{G}_{m,n}$ represents interactions between the interpolation points of two well separated boxes, it is in low rank and it could be factorized with QR factorization technique as:

$$\overline{G}_{m,n} = \overline{Q}_{m,n} \cdot \overline{R}_{m,n}$$

where $\overline{Q}_{m,n}$ and $\overline{R}_{m,n}$ are the $K_l \times r_i$ unitary matrix and the $r_i \times K_l$ upper triangular matrix. $r_i$ is the numerical rank of $\overline{G}_{m,n}$, and the condition $r_i \ll K_l$ always holds. Instead of directly store the Green’s function matrix $\overline{G}_{m,n}$ which is a $K_l \times K_l$ full-matrix, we use (14) to compress the Green’s function matrix and it also brings a great advantages in the computation of step 3 and 4 of Fig. 2 [23].

### III. THE INTERPOLATION SCHEME FOR MLGFIM

The accuracy and efficiency of MLGFIM tightly depend on the interpolation scheme for the approximation of interaction function. RBF interpolation method has been proven to be an appropriate approach for the interpolation of interaction function [23]. A lot of efforts have been made on searching a better RBF interpolation scheme to improve the efficiency of MLGFIM. Generally, an RBF $\phi_i (R) = \phi(|r_i - r|)$ is a continuous function that depends only on the distance between the data (interpolated) point $r$ and the center (interpolation) point $r_i$. Considering an approximation function $f(r)$ in an influence domain that has a set of $K$ arbitrarily distributed nodes with corresponding values $\{f(r_i)\}_{i=1}^K$, the RBF interpolation reads:

$$f(r) = [\phi_1 (R) \phi_2 (R) ... \phi_K (R)] \cdot \Phi^T$$

where the entries of matrix $\Phi$ are expressed as $\Phi_i = \phi(|r_i - r|)$.

There are many types of RBFs that can be used as the interpolation function. Table 1 shows three types of RBFs that have been used in MLGFIM. Parameter $c$ in these expressions is named as the shape parameter, and it determines the shape of the interpolation function. In [37], it has proven that GA RBFs can obtain better interpolation accuracy than IMQ RBFs which have been used in [23] with the same number of interpolation points, because they have better convergence behavior. As a new class of oscillatory RBFs, BE RBFs have shown their good interpolation performance in [38], and have been applied to further enhance the efficiency of MLGFIM as well [39].

**Table 1. Definitions of Some Typical Radial Basis Functions**

<table>
<thead>
<tr>
<th>Functions</th>
<th>$\phi(R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infinitely smooth functions</td>
<td>$\phi(R) = \frac{1}{\sqrt{1+c^2R^2}}$</td>
</tr>
<tr>
<td>Inverse multiquadric (IMQ)</td>
<td>$\phi(R) = \frac{1}{\sqrt{(cR)^2 + a^2}}, y = 1, 2, ...$</td>
</tr>
<tr>
<td>Gaussian (GA)</td>
<td>$\phi(R) = \frac{1}{\sqrt{(cR)^2 + b^2}} \exp(-c \cdot R^2)$</td>
</tr>
<tr>
<td>Bessel (BE)</td>
<td>$\phi(R) = \frac{1}{\sqrt{1+c^2R^2}}$</td>
</tr>
</tbody>
</table>

Fig. 3. 2-D pictorial representation of (a) original STG, (b) modified STG and (c) Boundary clustered STG (square points, hollow and solid dots are volume center, surface center and vertex points in a grid element, respectively).

Besides, the distribution of interpolation grid also affects the interpolation efficiency of Green’s function. Tartan grid, as a conventional grid pattern which is always used in the Lagrange interpolation, just distributes interpolation points evenly spaced in the source/field boxes, so the total number of
interpolation points is \( K = K_{1D}^3 \). Since the RBF method allows us to use scattered point distribution, STG which outperforms Tartan grid in accuracy is used in MLGFIM. Three types of STGs have been applied as the grid patterns which are shown in Fig. 3. The original STG (Fig. 3(a)) consists of Tartan grids and the points in the volume center of grid element of which vertexes are formed by adjacent Tartan grids [23]. And the modified STG, as shown in Fig. 3(b), consists of Tartan grids and the points in the center of the sides of the grid element [37]. Hence the total number of interpolation points for the original STG and modified STG are \( K_{1D}^3+(K_{1D}-1)^3 \) and \( K_{1D}^3+3K_{1D}(K_{1D}-1)^2 \), respectively. It has been proven that with similar number of interpolation points, the results using the modified STG are more accurate than those using the original STG [37]. This is because the condition number of matrix \( \Phi \) generated by the modified STG is better than the original STG. However, it is pointed out that an accurate method on high-degree uniform grid sets leads to large errors near the boundaries [40],[41]. So if the boundary errors can be suppressed, the interpolation accuracy will be further enhanced. Boundary clustered STG is generated by making the points denser near the boundary [42]. In 1-D, the Chebyshev-Lobatto nodes in the closed interval \([-1, 1]\) are expressed as:

\[
h_s = -\cos \left( \frac{2s}{K_{1D}^3} \pi \right), s = 0, 1, \cdots, K_{1D}^3 - 1 \tag{16}
\]

where \( h_s \) is the location of \( s \)-th interpolation point. However, it has been proven that the 3-D non-uniform grid generated by simply clustering the interpolation points toward boundary using (16) in \( x, y \) and \( z \) directions cannot improve the interpolation accuracy [42]. This is because this grid helps to reduce the boundary interpolation errors but it induces large interior errors as well. In order to balance the boundary and interior errors and make interpolation points still symmetric to the center of interval, the 1-D grid pattern in the closed interval \([0, l]\) is generated as:

\[
h_s = \begin{cases} 
\frac{l}{2} \left( 1 - \cos \left( \frac{2s}{K_{1D}^3} \pi \right) \right), & s = 0, 1, \cdots, K_{1D}^3 - 1 \\
\frac{l}{2} \left( 1 + \cos \left( \frac{2s-2(K_{1D}^3+1)/2}{K_{1D}^3-1} \pi \right) \right), & s = K_{1D}^3+1, \cdots, 2K_{1D}^3-1 
\end{cases} \tag{17}
\]

In function (17), the parameter \( \delta \) is introduced to adjust the degree of boundary clustering. The interpolation points, as \( \delta \) approaches 0, are distributed at \( l/2 \). Increasing \( \delta \) will shift the interpolation points toward the boundaries, and it improves the resolution at the boundary but lowers the resolution in the interior box. By adopting function (17) in \( x, y \) and \( z \) directions, we obtain a boundary clustered Tartan grid with \( K_{1D}^3 \) interpolation points. Then, the boundary clustered STG is generated by adding \( 3K_{1D}(K_{1D}-1)^2 \) interpolation points to the center of the sides of the rectangular block (grid element), as shown in Fig. 3(c).

The maximum interpolation errors using three different interpolation schemes are compared in Table 2. The first two schemes employ the same interpolation grid (viz., original STG), and the threshold of the maximum error for the second scheme is set as 0.05. With the same number of interpolation points, the maximum errors using IMQ RBFs are always larger than those using BE RBFs, as shown in Table 2. And as the edge length of the box increases, the method using BE RBFs becomes more accurate than IMQ RBFs. The third scheme adopts a new interpolation grid pattern, viz., boundary clustered STG, to further enhance the accuracy and efficiency of the interpolation. With the same error threshold, the method using boundary clustered STG requires less number of interpolation points than the original STG. Moreover, the interpolation accuracy using the third scheme, even with less number of points, is better than the other two. And it is also observed that if we employ the third scheme, much more number of interpolation points is saved as the size of box increases. The efficiency of MLGFIM can be dramatically enhanced by adopting the third method.

<table>
<thead>
<tr>
<th>Interpolation scheme</th>
<th>Box length (( \lambda ))</th>
<th>Number of interpolation points</th>
<th>Optimized ( \delta ) in (17)</th>
<th>Optimized shape parameter ( c )</th>
<th>Maximum error</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMQ RBFs and original STG</td>
<td>1</td>
<td>189</td>
<td></td>
<td>2.10</td>
<td>0.00681</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>559</td>
<td></td>
<td>0.80</td>
<td>0.0731</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1241</td>
<td></td>
<td>0.65</td>
<td>0.131</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2331</td>
<td></td>
<td>0.55</td>
<td>0.296</td>
</tr>
<tr>
<td>BE RBFs and original STG</td>
<td>1</td>
<td>189</td>
<td></td>
<td>6.00</td>
<td>0.00632</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>559</td>
<td></td>
<td>3.20</td>
<td>0.0174</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1241</td>
<td></td>
<td>2.35</td>
<td>0.0295</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2331</td>
<td></td>
<td>1.705</td>
<td>0.0498</td>
</tr>
<tr>
<td>BE RBFs and boundary clustered STG</td>
<td>1</td>
<td>172</td>
<td></td>
<td>0.75</td>
<td>6.90</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>365</td>
<td></td>
<td>0.97</td>
<td>3.21</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>666</td>
<td></td>
<td>0.96</td>
<td>2.10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1099</td>
<td></td>
<td>0.83</td>
<td>1.60</td>
</tr>
</tbody>
</table>
IV. MLGFIM FOR PERIODIC STRUCTURES AND MULTILAYERED PROBLEMS

As a kernel independent algorithm, MLGFIM can also be used for the fast analysis of other complex applications, e.g., periodic structures and multilayer problems. The Floquet theorem allows limiting the analysis of periodic structures to that of a unit cell; however, the number of unknowns can still be large with there are fine features exist in the unit cell.

For periodic structures, instead of interpolating free-space Green’s function, we should interpolate periodic Green’s function. Ewald’s transformation [43],[44], which efficiently combines both spatial and spectral formulations of the periodic Green’s function, is applied to speed up the evaluation of periodic Green’s function. Besides, since the unit cells are actually connected together, the periodic boundary condition must be taken into consideration. Half basis functions at the connection boundary between two adjacent unit cells have to be introduced to ensure the boundary currents continuous.

After that, we can use MLGFIM to accelerate the analysis of periodic structures. There are some modifications in the definition of neighbor and interaction list when applying MLGFIM for these structures, and they are defined as [45]: box $m$ is the neighbor/interaction list of box $n$ when box $m$ is near/well separated from box $n$ and the images of box $n$, and parent box of box $m$ is also near parent boxes of box $n$ and images of box $n$ simultaneously. Fig. 4 shows the neighbor and interaction lists of box $n$ at level 3. In order to implement MLGFIM easier, the box of interaction list which is not around box $n$ (but around the image of box $n$), e.g. box $m_1$ in Fig. 4, should be moved to the place of box $m_2$ (image of $m_1$) which is around box $n$.

![Fig. 4. 2-D representation of a 3-D multilevel division in which the neighbor (white colored) and interaction (slash) lists of box $n$ at level 3 are shown.](image)

In order to address the problem of the value of optimum shape parameter $c$ shift during the interpolation, RBF-QR method is applied for the interpolation of periodic Green’s function [46]. As the periodicity of one structure changes, the interpolated function (viz., periodic Green’s function) changes, and consequently the optimum value of shape parameter $c$ in RBF should be retested. Because the RBF-QR method is insensitive to the value of $c$, it is not required to retest the shape parameter when the interpolated function changes. By using expansion with Chebyshev polynomials and QR-factorization to the Gaussian RBFs, a better conditioned interpolation matrix with its condition number insensitive to $c$ is generated [46],[47]. With RBF-QR interpolation, MLGFIM becomes more robust for the analysis of periodic structures.

By interpolating the multilayered Green’s function, MLGFIM has been applied to solve the problems of multilayered medium structures. An improved discrete complex image method (DCIM) which uses a new sampling path directly goes through the branch point is adopted for the evaluation of the multilayered Green’s function [48]. This method helps to capture the contribution to the lateral wave more accurately. The property of planar structures of multilayered problems demands only 2-D interpolation scheme (viz., the interpolation points are only distributed at the interfaces of every two layers), while 1-D interpolation scheme is used for the vertical via holes only at the finest level [49]. We will later show that this interpolation approach is very efficient for the analysis of multilayered medium problems.

V. OTHER APPROACHES TO COMBINE WITH MLGFIM

Given that the translation occupies most of the computation time in the interpolation procedure of lower-to-upper-level, it is possible to further enhance the efficiency of MLGFIM by reducing the translation time. Although the interpolation performance of 3-D Tartan grid is not as good as the STG-based grid patterns, it generates a 3-D Toeplitz translation matrix $\tilde{G}$ which could use FFT to accelerate the matrix-vector multiplication. Moreover, the difference in the interpolation efficiency between 2-D Tartan grid and STG-based grid patterns is smaller than that of the 3-D case, and hence the MLGFIM-FFT can be used to accelerate the solution, especially the solution of multilayer problems where 2-D interpolation scheme is adopted.

For the multilevel tree structure, there are two ways to implement the FFT-accelerated translation procedure. In [50],[51], the multilevel tree will be truncated at certain level, and the pre-corrected FFT approach is applied at this level. A global Toeplitz matrix is generated and the interactions among all boxes are efficiently calculated by performing only one FFT. However, a pre-corrected near field evaluation is required for this method. So the translation (also shown in Fig. 2) is rewritten as a convolution form, viz.:

$$ B_m = FFT^{-1} \left\{ FFT(\tilde{G}) \cdot FFT(\tilde{S}) \right\} - \sum_{n \text{neighbors of } m} \tilde{G}_{m,n} \tilde{S}_{n} \quad (18) $$

where $\tilde{G}$ is comprised of Green’s function matrix $\tilde{G}_{m,n}$ and $\tilde{S}$ consists of vector $\tilde{S}_n$. Below the truncated level, the lower-to-upper-level interpolation as shown in Fig. 2 could be carried out as well [50].

The second way is based on local boxes, so it generates a sub-domain Toeplitz matrix. Consequently, sub-domain FFT is applied to accelerate the translation. Pre-correction is
not required because the sub-domain Toeplitz Green’s function matrix is formed by two well separated boxes [52]. Hence, instead of (18), the translation could be expressed as:

$$\bar{B}_{m} = \sum_{n_{i}\text{interaction list of } m_{i}} \bar{G}_{m_{i}n_{i}} \bar{S}_{n_{i}}$$

$$= F F F^{-1} \left\{ \sum_{n_{i}\text{interaction list of } m_{i}} F F T \left( \bar{G}_{m_{i}n_{i}} \right) \cdot F F T \left( \bar{S}_{n_{i}} \right) \right\}$$

(19)

where $\bar{G}_{m_{i}n_{i}}$ is a vector extracted from the Toeplitz matrix $G_{m_{i}n_{i}}$. The FFT is not implemented only at one level, but at many high levels unless the number of interpolation points is too small and direct matrix-vector multiplication will be more efficient than the FFT. With the second method, no truncated level exists, so the lower-to-upper-level interpolation is implemented from the top level to the bottom level to make the MLGFIM-FFT more efficient.

In addition, parallel computing could also be applied to accelerate the solution. Parallelized MLGFIM using OpenMP [50] and MPI [36] have been developed. For achieving high efficiency, we must make workload balanced and that could be accomplished by assigning approximately the same number of boxes at each level into each processor. Different from the OpenMP parallelization on a share-memory computer system, the MPI parallelization on a distributed-memory computer must take into account of the communications among different processors as well. If the upward or downward pass shown in Fig. 2 is implemented through different processors, a large amount of data communication is required. Hence, each root box (at level 2) and all its descendants should be assigned to the same processor [36]. Besides, in order to reduce the data communication in the computation of near field, the neighbors of each finest-level box are duplicated if they are not assigned to the same processor.

Parallelized MLGFIM on a distributed-memory computer system can be developed by modifying the steps of the original MLGFIM. First, the matrix-vector multiplication $\bar{S}_{n_{i}} = \bar{Z}_{n_{i}} x_{n_{i}}$ is calculated at the finest level $l = L$ with its own sub-tasks. Second, the vectors $\bar{S}_{n_{i}} = \sum_{m_{i}=n_{i}} C_{n_{i}m_{i}} \bar{S}_{m_{i}}$ for each box assigned to this processor are computed from level $l = L$ to 2. Then the translation from the source box to the interaction list is computed. Data communication is required in this step, and we use MPI_Allreduce command to collect the vectors $\bar{S}_{n_{i}}$ from each processor, and distribute it to all processors. After that, the vectors $\bar{B}_{m_{i}} = \sum_{n_{i}\text{interaction list of } m_{i}} \bar{G}_{m_{i}n_{i}} \bar{S}_{n_{i}} + C_{m_{i}m_{i}} \bar{B}_{m_{i}}$ in the corresponding boxes for the processor from level $l = 3$ to $L$ are computed. Finally, the matrix-vector multiplication $\bar{B}_{m_{i}} = \bar{B}_{m_{i}} + T_{m_{i}} \bar{B}_{m_{i}}$ is performed and the vector $\bar{F}_{m_{i}}$ is also collected and distributed to all processors.

Certain factors should be considered when examining the efficiency of parallel computing. Intuitively, a parallel program leads to a shorter execution time for a given computation task, which can be expressed by a speedup and efficiency factors as:

$$s(n_{p}) = \frac{t_{s}}{t_{p}}$$

(20)

and

$$e(n_{p}) = \frac{t_{s}}{n_{p} \times t_{p}} \times 100\%$$

(21)

where $t_{s}$ and $t_{p}$ are serial execution time using one processor and parallel execution time using $n_{p}$ processors, respectively. If the efficiency factor approaches 100%, this indicates that the parallel computing is very efficient and the communication/computation ratio is very small.

VI. NUMERICAL EXAMPLES

In order to demonstrate good performance of the proposed method, some examples are given as follows. The generalized minimal residual (GMRES) iteration method with a relative error norm of $10^{-3}$ is employed, and the inner loop of GMRES contains 100 matrix-vector multiplications.

![Fig. 5. Plane wave impinges normally on a fishnet-type array. (a) Geometry. (b) Unit cell. (c) Bistatic RCS with $\theta \theta$ polarization.](image-url)

In the first example, MLGFIM is applied to plane wave scattering from a fishnet-type array with 10x10 elements, as shown in Fig. 5(a). The geometrical parameters of the array are as follows: $w_{1} = 10 \mu m$, $w_{2} = 17.5 \mu m$, $d = 115 \mu m$, $t = 9 \mu m$. The patterns on the top and bottom surfaces of the substrate are made of the same shaped PEC, and the relative permittivity of the substrate is 2.25. A plane wave with electric field parallel to the $x$-axis propagates along $-z$-axis, and the center frequency is 0.5 THz. PMCHWT integral equation is used for this structure to obtain the surface current. The bistatic RCS with $\theta \theta$ polarization is calculated using the 5-level MLGFIM, and compared with MoM, as shown in Fig. 5(b). From this figure, it is observed that these two results agree very well.
Table 3. Comparison of CPU Time and Memory Requirement for Simulating the 11×11 Dielectric Sphere Array

<table>
<thead>
<tr>
<th>Method</th>
<th>Maximum electrical length($\lambda_d$)</th>
<th>The number of levels</th>
<th>The number of unknowns</th>
<th>CPU time for per matrix–vector multiplication</th>
<th>Memory requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original MLGFIM</td>
<td>16.87</td>
<td>6</td>
<td>144,020</td>
<td>96.3 s</td>
<td>8.7 GB</td>
</tr>
<tr>
<td>Improved MLGFIM</td>
<td>21.4 s</td>
<td>3.1 GB</td>
<td>21.4 s</td>
<td>3.1 GB</td>
<td></td>
</tr>
</tbody>
</table>

In the second example, a radiation problem that a dipole in front of an 11×11 dielectric sphere array ($\varepsilon_r = 2.25$) is investigated, as shown in Fig. 6(a). The radius of the sphere is 2 cm and the distance between the centers of two adjacent spheres is 5 cm. A Hertzian dipole which works at the frequency of 6.25 GHz is located 10 cm in front of the array center and placed along the $z$-axis. In this example, ten grids per wavelength $\lambda_d$ (where $\lambda_d = \lambda_0 / \sqrt{\varepsilon_r}$) are used to discretize this structure, so that the number of unknowns for equivalent electric and magnetic currents is 144,020. The polar plot of the directivity versus $\phi$ when $\theta = 90^\circ$ is shown in Fig. 6(b). Two methods, i.e., original MLGFIM (IMQ RBF and original STG) and improved MLGFIM (BE RBF and boundary clustered STG), are used for the simulation. Good agreement can be observed between these two results. The distribution of Poynting vector in the plane of $x = -20$ cm is also calculated, as shown in Fig. 6(c). The computational time and memory requirement for this example are shown in Table 3. Compared with the original method, only 22.2% CPU time and 35.6% memory are required for the improved one.
Next, an example is provided to show the CPU time and memory requirement with different number of unknowns. A dielectric prolate spheroid is considered, as shown in Fig. 7(a). The long axis of the prolate spheroid is 6 m, whereas the short axis is 1.5 m. A plane wave with a frequency of 300 MHz impinges on the prolate spheroid along $-x$-axis. The plane wave scattered by prolate spheroids composed of double positive ($\varepsilon_r = 2.25$, $\mu_r = 1.0$), single negative ($\varepsilon_r = -2.25$, $\mu_r = 1.0$) and double negative ($\varepsilon_r = -2.25$, $\mu_r = -1.0$) medium are studied. Fig. 7(b) compares the bistatic RCS with $\phi \phi$ polarization for these three cases. The forward scattering is enhanced significantly due to negative permittivity and permeability. The CPU time for performing each matrix-vector multiplication and the corresponding memory requirement are shown in Figs. 7(c) and (d). In these two figures, about 20 grids per wavelength are used to discretize the prolate spheroid, and the frequency varies from 150 MHz to 660 MHz. Three types of numbers of levels are used for the calculation of the structure, and the original MLGFIM cannot simulate the structure when the number of unknowns is larger than $5 \times 10^5$ because of the memory limitation. From Fig. 7(c) and (d), it is observed that the CPU time and memory requirement of the improved MLGFIM are always smaller than those of the original one. And when the number of unknowns increases to 1,089,224, the CPU time for performing one matrix-vector multiplication and the memory requirement for the improved MLGFIM are 128.7 seconds and 17.68 GB, respectively. In addition, it is also observed that the improved MLGFIM approximately obeys a computational complexity of $O(N \log N)$ and memory complexity of $O(N)$.

Table 4. Comparison of CPU Time and Memory Requirement

<table>
<thead>
<tr>
<th>Number of levels</th>
<th>Number of processors</th>
<th>Speed-up</th>
<th>Efficiency (%)</th>
<th>Save time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>100</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1.98</td>
<td>99.2</td>
<td>49.6</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>3.66</td>
<td>91.5</td>
<td>72.4</td>
</tr>
</tbody>
</table>

To show the effect and accuracy of parallelized MLGFIM, we consider the plane wave scattering from a complicated metallic scatterer, as shown in Fig. 8(a). This structure is composed of orthogonal square rings, and four square rings orthogonally connect with one ring to generate a unit, as shown in Fig. 8(b). The outer length, inner length and height of each ring are 3 mm, 1.8 mm and 0.6 mm, respectively. On the boundary of this structure, there exists only half rings, and hence the total structure contains 6 orthogonal rings in each direction. The structure is investigated at 300 GHz and illuminated by a plane wave propagating along the $-x$-direction, with the electric field polarized in the z-direction. After the structure is discretized, the total number of unknowns obtained is 1,134,768. The bistatic RCS result
calculated by the serial MLGFIM and parallelized MLGFIM using 8 processors are compared in Fig. 8(c). Good agreement is observed from this figure. Furthermore, the speed-up factor and efficiency factor are demonstrated, as shown in Table 4. According to this table, we observe that the computational time is reduced significantly with the increase of the number of processors. The efficiencies of the parallelized MLGFIM are larger than 80% for all cases. The CPU time is reduced by 84.6% after calculating this problem with eight processors, simultaneously.

In the fifth example, MLGFIM is used to analyze a periodic structure, in which two centered square loops are etched on a dielectric layer ($\varepsilon_r = 3$). Fig. 9(a) shows the geometry of the frequency selective surface (FSS), and it is illuminated by a uniform plane wave incident from the normal direction. Periodic Green’s function interpolation and periodic boundary conditions are used for this structure. The results for different dimensions $d$ of the inner loop are calculated by MLGFIM and compared with the simulation results in [53] and experimental results in [54]. RBF-QR method is adopted as the interpolation approach, so that the MLGFIM becomes more robust for the analysis of periodic problems. From Fig. 9(b), it can be seen that our results are closer to the experimental results compared with the simulation results in [53]. And we also find that the second resonance shifts toward a lower frequency band with the increase of the dimension $d$.

In the sixth example, MLGFIM is applied to analyze some patch arrays with planar multilayer substrate, in which sub-domain FFT is combined to accelerate the calculation. A plane wave with the electric field parallels to the $y$ axis incident from the normal direction. The parameters of the microstrip patch array are $l = 49$ mm, $w = 41$ mm, $\rho_a = 90$ mm, $\rho_b = 82$ mm, as shown in Fig. 10(a). The thickness and dielectric constant of the substrate are 1.59 mm and 2.2, respectively. The working frequency is set as 2.2GHz, and the normalized electric fields on $xz$ plane are shown in Fig. 10(b). In Fig. 10(c), the efficiencies, using MLGFIM and MLGFIM-FFT, for analyzing these arrays with different number of unknowns are compared. For this example, 2-D interpolation method is applied, and hence the CPU time and memory requirement are both much smaller than those of the above examples which use 3-D interpolation. From Fig. 10(c), it is observed that the CPU time and memory requirement using MLGFIM-FFT are always smaller than those using MLGFIM alone. With the
increase of the array size, the reductions of time and memory become more significant.

Finally, the MLGFIM-FFT method is used to calculate the radiation pattern from a 32×32 patch array with planar multilayer substrate. The structure of this array is shown in Fig. 11(a). And this structure is composed by eight 4×4 unit in one direction which is given in Fig. 11(b). The parameters of the microstrip patch array are l = 41 mm, w = 49 mm, ρa = 100 mm, ρb = 100 mm, t1 = 5.0 mm, t2 = 1.5 mm, and the thickness and relative dielectric constant of the substrate are 1.59 mm and 2.2, respectively. The working frequency is 9.42 GHz and the number of the unknowns of the metallic layer is about 310,000. The radiation patterns in both E- and H-plane are plotted in Fig. 11(c) and (d). A very narrow beam is formed as expected. The FFT will be applied at level 2 and 3. Because the structure is electrically large and complex, the convergence of the iteration process will be a big problem. To ensure convergence, the number of the spanning vector in GMRES is set to 1,000. The CPU time per iteration is about 1.8 seconds which is still very efficient, and the memory requirement is 11 GB due to the large number of spanning vector.

VII. CONCLUSIONS

The development and improvement of the efficient fast algorithm, namely MLGFIM, is reviewed. Objects in free-space, periodic and multilayer structures are analyzed using this fast simulation algorithm. Various techniques, including: BE RBF interpolation functions, boundary clustered STG pattern, FFT technique and parallelization approaches, are used to enhance the efficiency of the MLGFIM. Seven examples are given to validate the accuracy, and demonstrate the improvement of the MLGFIM after using the above new techniques.

REFERENCES


Peng Zhao received his B.Eng degree and M.Phil degree in electronic engineering from Zhejiang University, China, in 2006 and 2008, respectively. He earns his Ph.D. degree in electronic engineering from City University of Hong Kong, China, in 2014. He is currently a faculty member in the CAD laboratory at Hangzhou Dianzi University. His research interest focuses on computational electromagnetics.

Da Qing Liu received the B.S. degree in optical science from Wuhan University of Science and Technology, Wuhan, China, in 2008 and the M.S. degrees in Electronic Engineering from Zhejiang University, Hangzhou, China, in 2011. He is currently pursuing his Ph.D. degree in Electronic Engineering at City University of Hong Kong, China. His research interests include computational electromagnetic and antenna design.

Chi Hou Chan received the Ph.D. degree in electrical engineering from the University of Illinois, Urbana, IL, USA, in 1987. From 1989 to 1998, Dr. Chan was a faculty member in the Department of Electrical Engineering at the University of Washington, Seattle, WA, USA. In 1996, he joined the Department of Electronic Engineering, City University of Hong Kong, and was promoted to Chair Professor of Electronic Engineering in 1998. From 1998 to 2009, he was first Associate Dean and then Dean of College of Science and Engineering. He also served as Acting Provost of the university from July 2009 to September 2010. He is currently the Director of State Key Laboratory of Millimeter Waves, Partner Laboratory in the City University of Hong
Kong. His current research interests include computational electromagnetics, millimeter-wave circuits and antennas, and terahertz science and technology.

Dr. Chan received the U.S. National Science Foundation Presidential Young Investigator Award in 1991 and the Joint Research Fund for Hong Kong and Macao Young Scholars, National Science Fund for Distinguished Young Scholars, China, in 2004. He received outstanding teacher awards from the Department of Electronic Engineering, CityU in 1998, 1999, 2000, and 2008. He is the General Co-Chair of ISAP 2010, iWAT2011, and iWEN 2013.