Progress In Electromagnetics Research, **PIER 57, 285–310, 2006**

MIXTURE EFFECTIVE PERMITTIVITY SIMULATIONS USING IMLMQRF METHOD ON PRECONDITIONED EFIE

H. G. Wang, C. H. Chan, L. Tsang, and K. F. Chan

Wireless Communications Research Center
City University of Hong Kong
83 Tat Chee Ave
Kowloon, Hong Kong SAR, China

Abstract—Effective permittivity of mixtures of lossy dielectric are important quantities to be studied in microwave remote sensing of soil moisture, sea ice, dry and wet snow, in geophysical probing of properties of porous rocks and in composite materials. In this paper, these quantities are studied with large-scale numerical solutions of Maxwell equations in the electroquasistatic limit using fast electromagnetic algorithms. The preconditioned EFIE method instead of scalar potential scattering method for the simulation of the relative permittivity of the mixture with conducting particles in quasi-static environment is introduced. Furthermore, Algorithm IMLMQRF, which is kernel independent for quasi-static problems with a complexity of $O(N \log N)$, is implemented to accelerate the matrix-vector multiply when CG iteration is applying on this preconditioned EFIE system. Subsequently, numerical examples, viz., the permittivity extractions from two lattice structures and one random distribution structure with the unknowns from about 1,000 to 50,000 are efficiently simulated by our method in this paper. The numerical results demonstrate the efficiency of this hybrid method.

1. INTRODUCTION

Effective permittivity of mixtures of lossy dielectric are important quantities to be studied in microwave remote sensing of soil moisture, sea ice, dry and wet snow, in geophysical probing of properties of porous rocks. In microwave remote sensing of soil moisture, it is important to determine the permittivity of soil as a function of soil moisture and soil types [1, 2]. Sea ice is a mixture of ice and saline water and the effective permittivity determination is a critical issue [3–5]. The permittivities of dry and wet snow continue to draw interest [6–8]. In geophysical exploration, the permittivity of fluid filled porous rocks has to be studied [9]. The effective permittivity extraction of the composite materials becomes more and more important for material design. Analytic formulas such as Maxwell-Garnett mixing formula, the Polder Van Santern mixing formula, the quasi-static approximation and the quasi-static approximation with coherent approximation have been used with limitations on the domain of validity [10–12]. However, if the material is very complex, for instance, the particle distribution in the material is not periodic and the particles are very complex in shape, these analytic methods don't work. With the advent of modern computers, the use of numerical solutions of Maxwell equation has become an attractive method. The problem is inherently large-scale numerical simulations because a large number of particles need to be used to provide a valid statistical sample. Thus, fast numerical algorithms become necessary. A number of numerical methods, such as FDTD or FDM [13, 14], scalar-potential scattering method [15–17], etc., have been reported in the literature for solving this kind of problems. As FDTD is a time domain method, it can obtain a wide frequency response of the material effective permittivity in a single computer run. However, it requires a volumetric discretization, thus, the number of unknowns grows rapidly when the size of the problem increases.

If only the low frequency permittivity is needed, the scalar potential method based on the following equation

$$\phi^{inc}(\boldsymbol{r}) + \phi^{sca}(\boldsymbol{r}) = \phi^{total}(\boldsymbol{r}) \tag{1}$$

solving by the T-matrix method is a valid solution [15–16]. However, it is only applicable to a certain class of geometries such as the sphere and cylinder. An alternate method to solve (1) is the MoM [17]. The scalar potential can be written as

$$\phi^{sca} = \frac{1}{4\pi\varepsilon_b} \int_\Sigma dS' G(\boldsymbol{r}, \boldsymbol{r}')\rho(\boldsymbol{r}'), \tag{2}$$

where $G(\boldsymbol{r}, \boldsymbol{r}')$ is the Green's function for the specific problem, $\rho(\boldsymbol{r}')$ is the charge density on the surfaces of the particles, while Σ denotes the particle surfaces of all the particles. Assume that there are N_t particles. By meshing the surface of particle j ($j = 1, \cdots, N_t$) into n_j quadrilateral patches, one can obtain $N_p = \Sigma_{j=1}^{N_t} n_j$ patches for this particle system. Choosing the pulse base $P_{j,i}(\boldsymbol{r}') = \frac{1}{\Im_{j,i}(\boldsymbol{r}')}$, $i = 1, \cdots, n_j$, where $\Im_{j,i}(\boldsymbol{r}')$ is the Jacobian factor on patch i of particle

j, $\rho(\boldsymbol{r}')$ can be expanded as $\rho(\boldsymbol{r}') = \Sigma_{j=1}^{N_t}\Sigma_{i=1}^{n_i}P_{j,i}(\boldsymbol{r}')\mathcal{Q}_{j,i}$. For convenience, we denote the global index set of the supports of these pulse bases as $\{S_k\}_{k=1}^{N_p}$. Hence, the corresponding pulse base set can be written as $\{p_k\}_{k=1}^{N_p}$, where $p_k = P_{j,i}(\boldsymbol{r}')$, while the global index of patch i on particle j is k. To make all of the particles to be charge neutral, the restriction condition $\int_{\Omega_j} dS\rho(\boldsymbol{r}) = 0$ is used, where Ω_j is the surface of particle j.

$$\sum_{i=1}^{n_j}\mathcal{Q}_{j,i} = 0, \qquad j = 1, \cdots, N_t. \tag{3}$$

The matrix form of (3) is

$$[L][\mathcal{Q}] = [0] \tag{4}$$

The $k\,th$ entry q_k in N_p — by — 1 vector $[\mathcal{Q}]$ is expressed as

$$q_k = \mathcal{Q}_{j,i}, \quad \text{if the global index of patch } i \text{ on particle } j \text{ is } k. \tag{5}$$

The entries in N_t — by — N_p matrix $[L]$ is expressed as

$$L_{lk} = \begin{cases} 1, & \text{if } S_k \text{ is on particle } l, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Applying the Galerkin's method on (1), one obtains

$$\left[\Phi^{inc}\right] = -\left[Z\right]\left[\mathcal{Q}\right] + \left[L\right]^T\left[\Phi\right], \tag{7}$$

where N_p — by — N_p matrix $[Z]$ has entries

$$\begin{aligned} Z_{mk} &= \frac{1}{4\pi\varepsilon_b}\int_{S_m} dSp_m(\boldsymbol{r})\int_{S_k} dS'p_k(\boldsymbol{r}')\frac{1}{R} \\ &\approx \frac{1}{4\pi\varepsilon_b}\left(\frac{1}{R}\right)\Bigg|\begin{array}{ll} \boldsymbol{r} & = \text{centre of } S_m \\ \boldsymbol{r}' & = \text{centre of } S_k \end{array}, \end{aligned} \tag{8}$$

where $R = |\boldsymbol{r} - \boldsymbol{r}'|$. Combining (7) with (4), we can obtain a constrained linear system

$$\begin{bmatrix} -[Z] & [L]^T \\ [L] & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} [\mathcal{Q}] \\ [\Phi] \end{bmatrix} = \begin{bmatrix} [\Phi^{inc}] \\ 0 \end{bmatrix}, \tag{9}$$

By solving this equation we can obtain the charge distribution of arbitrary conducting particle set.

Obviously, this system is very ill conditioned. As a result, if the conjugate gradient (CG) method is applied to solve this system, it will converge slowly. Matrix inversion method can be used to avoid this problem. However, if the system size increases, the computation time will increase drastically as the matrix inversion scheme has a complexity of $O(N^3)$ where N is the number of unknowns. In order to attack this problem, here we present a method based on the preconditioned EFIE with and improved matrix QR factorization (IMLMQRF). The low frequency currents in EFIE are expanded by the tree bases [18] and the preconditioning technique is then applied. As a result, the spectrum property of the corresponding MoM matrix is improved and, hence, the convergence is very fast when the CG method is used. When the system of equation becomes very large, the traditional MoM method is replaced by the IMLMQRF [19] with the complexity of $O(N \log N)$ for accelerating matrix-vector multiplications in the CG iterations. The organization of this paper is as follows. In Section 2 and Section 3, the preconditioned EFIE with tree base expansion are introduced and a detailed method for solving it is addressed. In Section 4, an improved kernel independent multilevel matrix QR factorization method (IMLMQRF) is implemented to accelerate the full matrix-vector multiplication arising from (26) with a complexity of $O(N \log N)$. Section 5 is the numerical validation of the preconditioned EFIE by the examples of permittivity extractions from the cubic and spherical lattices, and the numerical application of the IMLMQRF on the preconditioned EFIE in large-scale simulation of mixture with random distributed spheres. A conclusion is given in Section 6. In the Appendix, a new algorithm is presented for efficiently solving the two sparse matrix equations $[K][x] = [y]$ and $[K]^T[x] = [y]$ encountered on Section 3, where $[K]$ is a sparse matrix used to precondition the full matrix $[B]^T[Z][B]$ in the L. H. S. of (26).

2. EFIE AND ITS TREE BASE EXPANSION IN LOW FREQUENCIES

The tangential E fields on the surfaces of these particles satisfy

$$-\boldsymbol{E}^{sca} \cdot \hat{\boldsymbol{t}} = \boldsymbol{E}^{inc} \cdot \hat{\boldsymbol{t}}, \tag{10a}$$

where $\hat{\boldsymbol{t}}$ is any an unit tangential vector on these conducting surfaces. The researched problems in this paper are the low frequency near field problems. In very low frequencies, the vector potential part of the Mixed Potential Integral Equation (MPIE) can be ignored when calculating the near scattered E field, thus $\boldsymbol{E}^{sca}$ can be written as

$$\boldsymbol{E}^{sca} = -\nabla\phi = -\frac{1}{4\pi\varepsilon_b}\nabla\int_{\Sigma} dS'\nabla'\cdot\boldsymbol{J}(\boldsymbol{r}')/(i\omega R) \tag{10b}$$

where ε_b is the background media, $\boldsymbol{J}(\boldsymbol{r}')$ is the electric currents, R is the distance between source point and observation point, and ω the angle frequency. Combining (10a) and (10b), we get the low frequency Electric Field Integral Equation (EFIE).

It is known that the current vector is the combination of the solenoidal part and the non-solenoidal part [18]. However, from (10b), we see that the influence of the solenoidal part to the near E field is ignored. Thus only the non-solenoidal part of the current should be taken into account. It is known that the tree bases can be chosen as a set of non-divergence-free bases, although any of them contains a small bit of solenoidal component. Hence, in this paper, we chose tree bases as the basis functions to expand the EFIE.

The tree base has the same form as the conventional rooftop base. However, the number of the tree bases is different from that of the conventional rooftop bases. According to the graph theory, the N_p patches on the N_t particles can be viewed as the N_p nodes of a graph, while each two adjacent patch nodes are connected by a branch. Because the support of each rooftop base consists of two adjacent patch nodes, thus the number of rooftop bases N_b' is the same as the number of branches of this graph. On the other hand, each particle can be viewed as a sub-graph of this graph. Thus, there are N_t sub-graphs. For these sub-graphs, we can find their corresponding N_t spanning trees by the Depth First Search Algorithm [20]. Because in a spanning tree, no loop exists, thus any branch on this tree can be viewed as a non-divergence-free base. Hence the number of tree bases N_b equals the numbers of the branches of these N_t spanning trees. From the graph theory, the number of branches of spanning tree j equals the number of patch nodes on this tree subtracting one, viz., $n_j - 1$, hence, $N_b = \sum_{j=1}^{N_t}(n_j - 1) = N_p - N_t$. Obviously, $N_b < N_b'$. Fig. 1 shows two meshed particles and the corresponding graph.

Let the tree base set and its corresponding support set be $\{\boldsymbol{j}_k(\boldsymbol{r}')\}_{k=1}^{N_b}$ and $\{(S_{k_1}, S_{k_2})\}_{k=1}^{N_b}$, respectively. Fig. 2 is the illustration of a tree base where S_{k_1} and S_{k_2} denote the global indices of the first patch node and the second patch node of $\boldsymbol{j}_{\boldsymbol{k}}(\boldsymbol{r}')$, respectively. The mathematical form of $\boldsymbol{j}_{\boldsymbol{k}}(\boldsymbol{r}')$ can be found in [21]. Here a brief

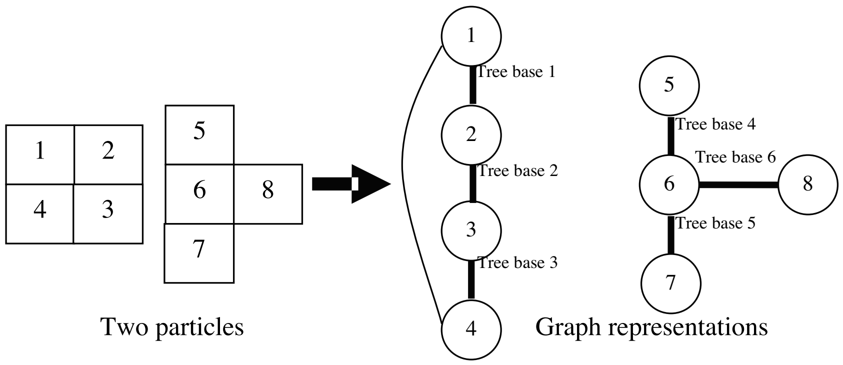

Figure 1. Two meshed particles and their graph presentations. The tree bases and the patches are globally indexed.

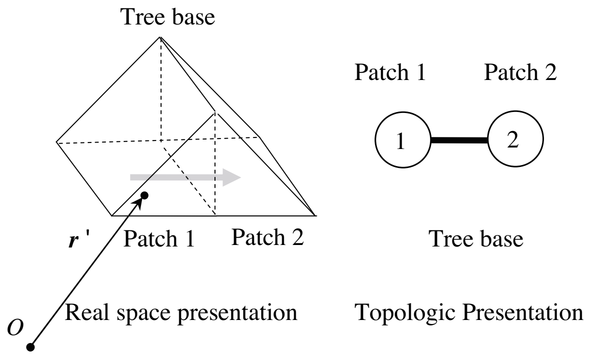

Figure 2. The real space and topologic presentation of the support of a tree base.

introduction is given.

$$\boldsymbol{j}_{\boldsymbol{k}}(\boldsymbol{r}') = \begin{cases} \dfrac{\aleph_{k,1}}{\Im_{k_1}} \boldsymbol{T}_{k,1}, & \boldsymbol{r}' \in S_{k_1} \\ \dfrac{\aleph_{k,2}}{\Im_{k_2}} \boldsymbol{T}_{k,2}, & \boldsymbol{r}' \in S_{k_2} \end{cases} . \tag{11}$$

$T_{k,1}$ and $T_{k,2}$ are two polynomials having the forms as $\xi_i \frac{\partial \boldsymbol{r}'}{\partial \xi_j}$ or $(1 - \xi_i)\frac{\partial \boldsymbol{r}'}{\partial \xi_j}$, ($i$=1 or 2, j=1 or 2), where the pair (ξ_1, ξ_2) is the parameter coordinate of point $\boldsymbol{r}'$. One should note that different tree base has different choices of the values of i and j. $\aleph_{k,i}$ (i=1 or 2) is the normalization factor used to ensure the normal current continuity on the common edge of support patch 1 and support patch 2 of $\boldsymbol{j}_k$.

Its expression can be found in [21]. Thus the current $\boldsymbol{J}(\boldsymbol{r}')$ can be expanded as $\boldsymbol{J}(\boldsymbol{r}') = \sum_{k=1}^{N_b} \boldsymbol{j}_k(\boldsymbol{r}')I_k$, where I_k is the current coefficient. It is not difficult to obtain the identity

$$\nabla' \cdot \boldsymbol{j_k} = \begin{cases} \dfrac{\aleph_{k,1}}{\Im_{k_1}}, & \boldsymbol{r}' \in S_{k_1} \\ \dfrac{\aleph_{k,2}}{\Im_{k_2}}, & \boldsymbol{r}' \in S_{k_2} \end{cases}. \tag{12}$$

Applying Galerkin's method on (10a), one obtains the following matrix equation

$$[Z'][I] = [V], \tag{13}$$

where the entries in the N_b — by — 1 excitation vector $[V]$ can be written as

$$V_t = \sum_{i=1}^{2} \int_{S_{k_i}} dS \boldsymbol{j}_i(\boldsymbol{r}) \cdot \boldsymbol{E}^{inc}(\boldsymbol{r}). \tag{14}$$

The entry Z'_{lk} in the N_b — by — N_b matrix $[Z']$ is expressed as

$$Z'_{lk} = \frac{-1}{i\omega 4\pi\varepsilon_b} \sum_{i=1}^{2}\sum_{j=1}^{2} \int_{S_{l_1}} dS \int_{S_{k_1}} dS' \nabla \cdot \boldsymbol{j}(\boldsymbol{r}) \nabla' \cdot \boldsymbol{j}(\boldsymbol{r}') \frac{1}{R}. \tag{15}$$

One can easily observe that the computational complexity in (15) is about four times of that in (9). For the sake of reducing its complexity, we do the following transform steps. Substituting (12) into (15), one obtains

$$Z'_{lk} = \frac{i}{\omega} \sum_{i=1}^{2}\sum_{j=1}^{2} \aleph_{l,i} \aleph_{k,j} Z_{l_i k_j}. \tag{16}$$

The matrix form of (16) can be written as

$$[Z'] = \frac{i}{\omega}[B]^T[Z][B]. \tag{17}$$

The entries in this N_p — by — N_b sparse matrix $[B]$ can be written as

$$B_{mn} = \begin{cases} \aleph_{n,l}, & S_m \text{ is the first support patch of tree base } n, \\ \aleph_{n,2}, & S_m \text{ is the second support patch of tree base } n, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Thus (13) can be rewritten as

$$\frac{i}{\omega}[B]^T[Z][B][I] = [V]. \tag{19}$$

Instead of computing $[Z']$, we calculate $[Z]$, which has less computational complexity. Obviously, because matrices $[B]$ and $[B]^T$ are very sparse, when CG iteration is applied to solving (19), the additional time consumed in matrix-vector multiplications $[B][x]$ and $[B]^T[x]$ can be ignored.

However, matrix $[Z']$, viz., $[B]^T[Z][B]$ is still ill-conditioned. This means that a large number of steps are still needed to solve (19) or (13) iteratively. Fortunately, in [18], this problem has been solved by introducing a sparse matrix $[K]$.

3. THE PRECONDITION MATRIX $[K]$ AND THE NEW ALGORITHM FOR CALCULATING $[K]^{-1}[y]$

According to [18], the main reason that causes the ill-conditioned matrix $[Z']$ is the divergence of the tree base. However, it is known that the matrix equation for electrostatic problem based on pulse basis converges rapidly. Hence, the relation matrix between the current bases and the pulse bases can be used as a precondition matrix for (19) or (13). The current continuity equation is given by

$$\nabla \cdot \boldsymbol{J}(\boldsymbol{r}) = i\omega\rho(\boldsymbol{r}). \tag{20}$$

We should note that, although as mentioned in section 1, the charge density can be expanded as $\rho(\boldsymbol{r}') = \sum_{j=1}^{N_t}\sum_{i=1}^{n_i} P_{j,i}(\boldsymbol{r}')\mathcal{Q}_{j,i}$, the charge neutral property of each particle, i.e., (3), will cause the number of independent pulse bases to be $N_b = N_p - N_t$, a number that equals the number of tree bases. By combining (3) with this charge density expansion, we can obtain another form of the charge expansion

$$\rho(\boldsymbol{r}) = \sum_{j=1}^{N_t}\sum_{i=1}^{n_j} P_{j,i}\mathcal{Q}_{j,i} = \sum_{j=1}^{N_t}\sum_{i=1}^{n_j-1}(P_{j,i} - P_{j,n_j})\mathcal{Q}_{j,i} = \sum_{k=1}^{N_b} p'_k q'_k. \tag{21}$$

From (21), we can see that the support of an independent charge base consists of two patch nodes. Hence, let its globally indexed support set be $\{(S^*_{k_1}, S^*_{k_2})\}_{k=1}^{N_b}$. The second patch node is the last locally indexed patch nodes in the spanning tree, while the first patch node is corresponding to the support of a tree base. Hence a one-to-one correspondence can be found between the first support patch node of each charge bases and the support of each tree bases. For instance, in Fig. 1, the support set of the tree bases is $\{(S_{j_1}, S_{j_2})\}_{j=1}^{6} = \{(n1, n2), (n2, n3), (n3, n4), (n5, n6), (n6, n7), (n6, n8)\}$, while the corresponding support set of the independent charge bases is $\{(S^*_{j_1}, S^*_{j_2})\}_{j=1}^{6} = \{(n1, n4), (n2, n4), (n3, n4), (n5, n7), (n6, n7), (n8, n7)\}$, where nj denotes the global index of patch node j of this graph. Here the global indices of the patch nodes, the tree bases and the independent charge bases are all generated by Depth First Searching the spanning trees once. Subsequently, substituting $\boldsymbol{J}(\boldsymbol{r}') = \sum_{k=1}^{N_b} \boldsymbol{j}_k(\boldsymbol{r}')I_k$ into (20) and performing the surface integral on the first support patch node of each independent charge base, we get

$$\sum_{k=1}^{N_b} \int_{S^*_{l_1}} dS\nabla \cdot \boldsymbol{j_k}(\boldsymbol{r}) = i\omega \int_{S^*_{l_1}} dS\rho(\boldsymbol{r}) = i\omega q'_l, \qquad l = 1, \cdots, N_b. \tag{22a}$$

Its matrix form is

$$[K][I] = i\omega[\mathcal{Q}']. \tag{22b}$$

Note that in equation (22b), Matrix $[K]$ is an N_b — by — N_b sparse matrix that will be used as a precondition matrix. Its entries are given by

$$K_{lk} = \begin{cases} \aleph_{k,1}, & \text{if } S^*_{l_1} = S_{k_1}, \\ \aleph_{k,2}, & \text{if } S^*_{l_1} = S_{k_2}, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Multiplying the inverse of $[K]$ on the two sides of (22a)–(22b), one obtains

$$[I] = i\omega[K]^{-1}[\mathcal{Q}']. \tag{24}$$

Substituting (24) into (13) and (19) one obtains

$$[K]^{-T}[Z'][K]^{-1}(-[\mathcal{Q}']) = [K]^{-T}[V], \tag{25}$$

and

$$[K]^{-T}[B]^T[Z][B][K]^{-1}(-[\mathcal{Q}']) = [K]^{-T}[V], \tag{26}$$

where $[K]^{-T}$ is used to make the matrix product on the L. H. S. of (26) symmetric. It has been proved that matrix $[K]^{-T}[Z'][K]^{-1}$, viz.,

$$\begin{bmatrix} * & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}$$

Figure 3. The matrix $[K]$ of the example in Fig. 1. The dash blocks correspond to two particles. The asterisks denote the nonzero entries of the sparse matrix.

$[K]^{-T}[B]^T[Z][B][K]^{-1}$ has a good spectrum property, thus (25) and (26) converge very fast [18]. Fig. 3 is the illustration of the matrix $[K]$ of the two particles in Fig. 1.

However, to obtain the inverse of $[K]$ is a problem. If the number of tree bases in each spanning tree is small enough, the matrix can be viewed as a block diagonal matrix, while each block corresponds to a spanning tree, or say, a particle. Thus the inverse of matrix $[K]$ can be fast calculated by inverting each of the diagonal block. However, if the volume fraction increases, the mesh density may also increase, so does the number of tree bases. The direct inverse of $[K]$ becomes time consuming. In fact, if we want to obtain vector $[x] = [K]^{-1}[y]$, we only need to solve the equation $[K][x] = [y]$. In [18], an efficient method has been provided. The main idea is to solve the equation with the recursive elimination method with $O(N_b)$ complexity. However, in [18], the base reorder and complex matrix operations, such as matrix divisions and matrix multiplications have been used. This makes its realization very cumbersome.

Instead of using the algorithm in [18], we provide, in the Appendix, two very simple algorithms for solving $[K][x] = [y]$ and $[K]^T[x] = [y]$ with the same computational complexity, in which no complex matrix operation is needed. Here, we give an example to describe the basic idea of our algorithm for solving $[K][x] = [y]$. Let matrix $[K]$ be that shown in Fig. 3. We observe that row one of $[K]$ has only one diagonal non-zero entry. This means that the unknown x_1 can be directly obtained. Subsequently, the off-diagonal non-zero entries in column one can be eliminated. It is obvious that each row has at most two non-zeros, where one of them is the diagonal entry. Because the off-diagonal non-zeros in column one have been eliminated, thus, in each of their corresponding rows, only one non-zero entry, viz., the diagonal entry remains. Consequently, the unknown value in these rows can be directly obtained. Thus, we can obtain all the entries in $[x]$ recursively. In order to evaluate their performance, we have generated a system of 27 sphere particles with 4,077 unknowns. We use a computer with 2.53 GHz CPU to solve equation $[K][x] = [y]$ and equation $[K]^T[x] = [y]$ each for 10,000 times, with only 3 seconds and 4 seconds, respectively. This means that the time used in them can be ignored. The results have been compared with the results obtained by the direct inverse of the block diagonal matrix $[K]$, no difference has been found between them.

4. THE IMLMQRF FOR ACCELERATING THE MATRIX-VECTOR MULTIPLICATIONS

Here, we choose the popular CG method to solve (26). In CG the matrix-vector multiplications $[K]^{-T}[B]^T[Z][B][K]^{-1}[x]$ should be performed in 5 steps. They are $[y] = [K]^{-1}[x]$, $[x] = [B][y]$, $[y] = [Z][x]$, $[x] = [B]^T[y]$, and $[y] = [K]^{-T}[x]$. Just as mentioned before, the CPU time consumed in step 1, 2, 4, and 5 can be ignored. Thus, the only time consuming step is $[y] = [Z][x]$, which has a complexity of $O(N_p^2)$.

In [22] and [23], the method IES^3 based on Multi-Level Matrix QR Factorization is developed, which can rapidly solve all kinds of low-frequency problem and is kernel independent. Its complexity for matrix-vector multiplication is $O(N \log N)$.

It is known that QR factorization of an $n — by — n$ matrix $[A]$ results in one $n — by — r$ unitary matrix $[Q]$ and one $r — by — n$ upper triangular matrix $[R]$, where r is the numerical rank of $[A]$ [24]. This satisfies the identity $[A] = [Q][R]$ with certain accuracy. The storages for these two matrices Q and R are nr and rn, respectively. Usually, if the numerical rank of $[A]$ is much less than its size, i.e., $r << n$, the total storage of $[Q]$ and $[R]$ will be much less than that of matrix $[A]$ as $2rn << n^2$. Furthermore, the CPU time consumed in matrix-vector multiplications of $[Q][R][x]$ will be much less than that in $[A][x]$ due to the fact that $O(nr + rn) << O(n^2)$, despite both operations yielding the same result. This is the basic idea of IES^3.

Although the moment matrix $[Z]$ must be full rank, the ranks of its submatrices can be very low. In IES^3, the geometry is firstly divided into multilevel groups shown in Fig. 4. Let these subscatterer groups form a set denoted as $\{G_j^l\}$, where l is the level index, while j is the jth subgroup at level l. Hence the moment matrix $[Z]$ is correspondly divided into some square-like submatrices according to a multilevel matrix division scheme [22]. These matrices form a submatrix set $\{A_{ij}^l\}$, where each submatrix A_{ij}^l denotes the interaction from group

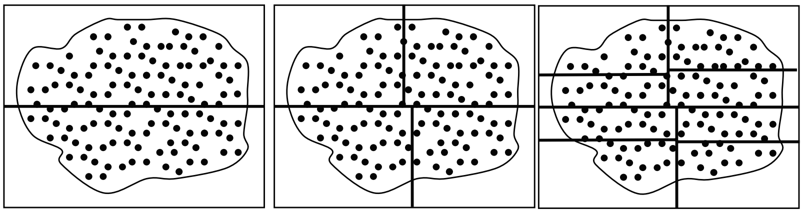

Figure 4. The illustration of complete multi-level division of an object.

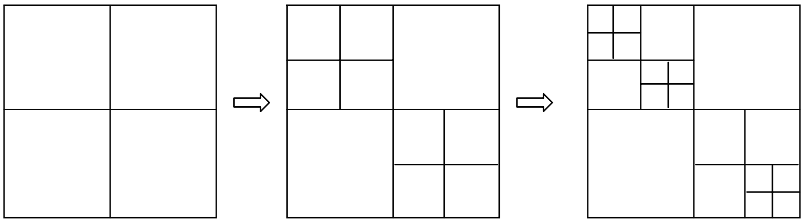

Figure 5. An example of the incomplete multi-level matrix division, which only shows an incomplete 3-level matrix division.

G_j^l to group G_i^l at level l. Fig. 5 is the illustration of the multilevel matrix division. From Fig. 5, we can see that the multilevel matrix division is incomplete.

For any submatrix A_{ij}^l, if its two corresponding subscatterer groups G_i^l and G_j^l are well separated, the numerical rank r of this submatrix will be very low. It is obvious that, for a moment matrix, the more low rank submatrices it has, the more efficient we can compute the matrix-vector multiplication. Fig. 6 shows the rank map of a 6448-by-6448 matrix of a mixture of 124 randomly distributed spheres with a fractional volume of 0.05. In this rank map, the largest square denotes the moment matrix $[Z]$, while the small rectangles denote the submatrices this moment matrix contains. The number in each frame, i.e., each submatrix denotes the numerical rank of it. We can see that there are a lot of low rank submatrices.

In IES^3, for any low rank submatrix A_{ij}^l in $[Z]$, instead of calculating all of its entries, a sampling algorithm is developed to compute only a few selected rows and columns of it and the Modified Gram-Schmidt Algorithm is used to obtain its QR representation, viz., $A_{ij}^l = Q_{ij}^l R_{ij}^l$. As a result, the computational complexity decreases to $O(N \log N)$ [22, 23]. However, the sampling scheme depicted in

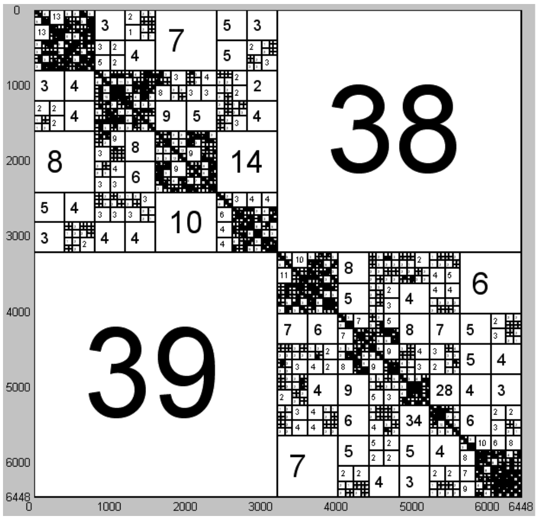

Figure 6. Rank map of the moment matrix of 124 randomly distributed sphere particles with volume fraction of 0.05.

[23] is straightly mathematical and not necessarily efficient. In it, a large number of vector operations with least square method were used for the sampling, making it time consuming. In this paper, we use the alternate algorithm developed in [19], viz., the Improved Multi-Level Matrix QR Factorization (IMLMQRF). In IMLMQRF, a new sampling methodology is presented in which a physics-based column and rigorous Gram-Schmidt row selection process are introduced. This makes the sampling more stable and efficient. In [19], large-scale RFIC is simulated. Here, IMLMQRF will be applied to solve the large-scale low frequency random particles scattering problem. The sampling method of IMLMQRF is given in [19] and a brief introduction is given here.

For any submatrix A^l_{ij} in moment matrix $[Z]$, we develop two algorithms for sampling its s^l_{ij} rows and s^l_{ij} columns, and subsequently reconstruct matrix A^l_{ij} by its QR factorization representation, viz., $A^l_{ij} = Q^l_{ij} R^l_{ij}$.

Based on the interpolation principles, the reconstruction of matrix A^l_{ij} via its sampled rows and columns is just the same as the reconstruction of a two-dimensional graph with the sampled points as the interpolation points. But if we do not know the variations of colors in the graph before interpolation, the only thing that we

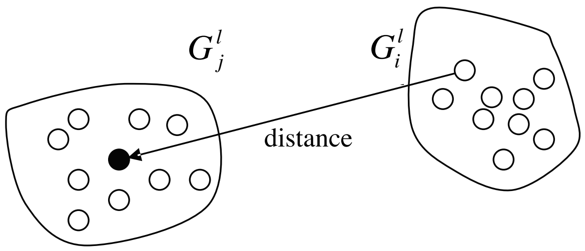

Figure 7. The illustration of the distance from a source point (the hollow circle) in G_j^l to the center point (the black dot) of observation region G_j^l.

can do is to fully sample all its points. However if we have some prior information, the number of sampled points can be drastically decreased. For instance, if we have previously known that there is a blue-sky area, the point density in this area can be reduced. On the contrary, the point density should be increased in the fast varying regions such as the hills, the grasses and the trees. Numerically, we should sample those points with the largest second order derivatives. But none of the entries of A_{ij}^l are known so as their second derivatives. Fortunately, some approximations can be used to model the average variations of entries in A_{ij}^l along the column directions. The algorithm steps are described in *Algorithm* 1.

Algorithm 1

Step 1. Find the center points of the observation point G_i^l (Fig. 7).

*Step 2. Find the approximations of Green's function values from the source points in G_j^l to this center point (see Fig. 7). For free space problem, the reciprocal of the distance shown in Fig. 7 can be viewed as the approximation of the Green's function. Hence store all these values to an array **d**, where d_k is its k^{th} element.*

*Step 3. Solve for the second order difference along the array index axis and overwrite them to the array **d**. The formula of the k^{th} element is $d_{k-1} + d_{k+1} - 2d_k$.*

*Step 4. Employ the quick sort algorithm [25] to choose the s_{ij}^l indices of, **d** for which their values are the largest.*

Step 5. Hence, the corresponding s_{ij}^l columns are sampled from A_{ij}^l.

It is obvious that the CPU time consumed in *Algorithm* 1 is negligible. Assume that we have sampled s_{ij}^l columns from submatrix A_{ij}^l and then calculated and stored them in A_c. The remaining problem is to sample some rows from A^l_{ij}. Obviously, the same manner can be used in sampling these rows. However, we had designed a rigorous algorithm described below.

Algorithm 2

Step 1. Sample s^l_{ij} columns from submatrix A^l_{ij} based on Algorithm 3 and calculate them. These sampled columns form a new temporary matrix A_c of m^l_{ij}—by—s^l_{ij}. The number of operations is $m^l_{ij} \times s^l_{ij}$.

Step 2. QR factorize matrix A_c and obtain a temporary unitary matrix $\mathcal{Q}_c$ of m^l_{ij}—by—r^l_{ij}. Hence, $\mathcal{Q}^l_{ij} = \mathcal{Q}_c$ The number of operations is $m^l_{ij} \times s^l_{ij} \times r^l_{ij}$.

Step 3. Find the r^l_{ij} rows of orthogonal bases that span the rows of matrix $\mathcal{Q}_c$ by using the modified Gram-Schmidt Algorithm described in [24]*. At the same time, record the indices of these corresponding rows of matrix $\mathcal{Q}_c$ in an index set. Subsequently, sample and calculate r^l_{ij} rows from A^l_{ij} according to the index set. All of these sampled rows are stored in matrix A_r of r^l_{ij}—by—n^l_{ij}. The number of operations is $m^l_{ij} \times r^l_{ij} \times r^l_{ij}$.*

Step 4. Sample r^l_{ij} rows from $\mathcal{Q}_c$ according to the index set. These sampled rows form a new temporary matrix $\mathcal{Q}'_c$ of r^l_{ij}—byr^l_{ij}. The number of operations can be neglected, because $\mathcal{Q}_c$ has been calculated in Step 2.

Step 5. Solve the equation $\mathcal{Q}'_c \times R = A_r$, where $\mathcal{Q}'_c$ is invertible (no least square solution is needed). Hence we had obtained $R^l_{ij} = R$. The complexity is about $O(r^l_{ij} \times r^l_{ij} \times n^l_{ij})$.

In numerical practices, $s^l_{ij} \propto r^l_{ij}$ and $m^l_{ij} \approx n^l_{ij}$. Thus, the total computational complexity of this of sampling algorithm is $O(r^l_{ij} \times r^l_{ij} \times m^l_{ij})$. Comparing the row sampling algorithm in [23], we see that no distance between two multi-dimensional vectors (columns or rows) is computed here and no least square solution is used. Furthermore, the number of sampled rows is r^l_{ij}, not s^l_{ij}. Hence, the sampling efficiency has been drastically increased and we can obtain $(\mathcal{Q}^l_{ij}, R^l_{ij})$ with much less computing time.

5. NUMERICAL RESULTS

To validate the accuracy and efficiency of our method, here we apply the preconditioned EFIE with the IMLMQRF implementation to calculate the effective permittivity of the periodic and the disordered mixture. If the macroscopic mixture sample is spheroid, the following formulation can be used to extract the effective permittivity [26]

$$\boldsymbol{P}_{\mathit{eff}} = \frac{3\varepsilon_b(\varepsilon_{\mathit{eff}} - \varepsilon_b)}{\varepsilon_{\mathit{eff}} + 2\varepsilon_b}\boldsymbol{E}^{inc}, \tag{27}$$

where the effective dipole moment is the assemble average of the dipole moment, viz.,

$$\boldsymbol{P}_{\mathit{eff}} = \frac{\int_V dV\boldsymbol{P}(\boldsymbol{r})}{V}. \tag{28}$$

For the conducting particles,

$$\int_V dV\boldsymbol{P}(\boldsymbol{r}) = \int_\Sigma dS\rho(\boldsymbol{r})\boldsymbol{r}, \tag{29}$$

where ρ is obtained by (26).

In this paper, for simplicity, all of the samples are cubic shaped. Cube sample can be viewed as an approximation of sphere sample. This will not affect the accuracy so much [17].

5.1. The Permittivity Extraction from Lattice

To validate the preconditioned EFIE devised in this paper, we had simulated the effective relative permittivity of a sphere centered cubic lattice and a cube centered cubic lattice in free space. They are shown in Fig. 8. Because the lattice sample consists of a large number of periodically arranged micro-cells, thus V can be replaced by the volume

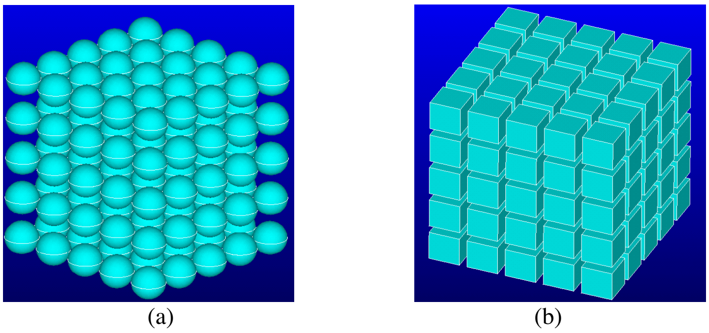

Figure 8. The illustration of cubic lattice. (a) Sphere centered. (b) Cube centered.

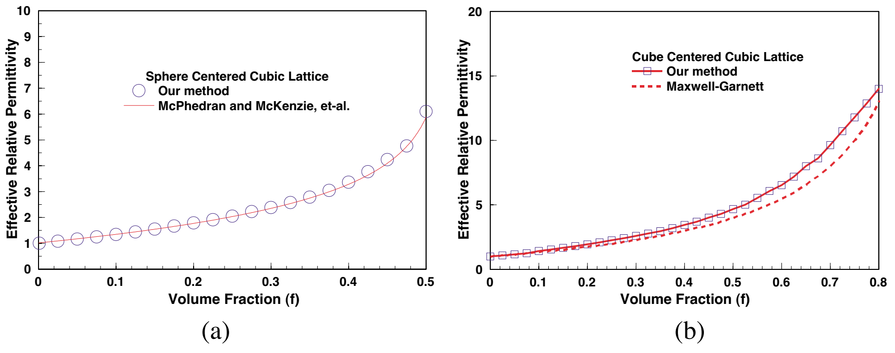

Figure 9. The effective relative permittivity of (a) the sphere centered cubic lattice and (b) the cube centered cubic lattice.

of a cell V_{cell} and the periodic Green's function can be used in (10a–10b) or (2) [17] for reducing the computational complexity.

Fig. 9(a) shows the permittivity curves of the sphere centered cubic lattice. In the Volume fraction between 0 and 0.4, 1950 patches are employed to model the surface of the sphere, while in the fractional volume between 0.4 and 0.5, 3780 patches are used. The number of spheres used to calculate the periodic Green's functions is 125 $(5 \times 5 \times 5)$ [17]. The comparison has been made between our method and Doyle's interpolation formula for simple cubic lattice $\varepsilon_{r,\,eff} = (1 - 0.5\pi \ln(1 + 6f/\pi))$ [12], a good agreement is observed between them.

Our method can be used to solve the problems with different particle shapes. The permittivity of the cube centered cubic lattice has also been calculated and is shown in Fig. 9(b). Here, 3750 patches are used to model the surface of the cube and 729 cubes are used to calculate the periodic Green's function in the fractional volume between 0 and 0.8 [17]. Maxwell-Garnet solution $\varepsilon_{r,\,eff} = 1+3f/(1-f)$ is also plotted as a comparison. The numerical results by our method are slightly larger than the Maxwell-Garnett results.

5.2. The Permittivity Extraction from Random Distributed Mixtures

In this paper, for simplicity, we only studied the randomly distributed mixture with identical sized and shaped conducting spheres, although the sizes and shapes of the particles in a mixture can be nonuniform. The surface of each sphere shown in Fig. 10 is modeled by 52 patches. The spheres are randomly distributed in a cubic volume filled with the

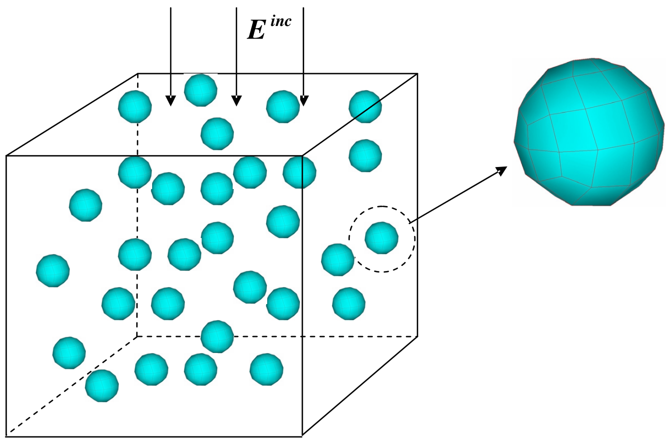

Figure 10. The illustration of the uniformly randomly distributed particles in a cubic volume filled with the background media ε_b.

background medium ε_b. Here we choose $\varepsilon_b = \varepsilon_0$ for convenience. We distribute these particles randomly in two steps. At first, the cube is divided into $10 \times 10 \times 10$ cubic grids. Subsequently, the particles will be uniformly randomly located in these grids one by one. The random location generator is a slight modification of the quasi-random generator in [27].

To check the convergence of our method, we have calculated the effective permittivity of a specific distribution of 51 spheres in the cubic volume with the fractional volume equals to 0.02. Two methods have been used. The first method is the scalar potential method, i.e., solving equation (8), while the second method is our method of solving equation (26). The numbers of unknowns for these two methods are $51 \times 52 + 51 = 2704$ and $51 \times (52 - 1) = 2601$, respectively. Fig. 11 shows the CG convergence comparison between these two methods. We can observe that equation (26) converges very fast, while equation (8) becomes stagnated after 20 iterations. This means that the matrix in (8) is very ill-conditioned. Because the problem is small enough, we can use the direct matrix inversion in solving (8). The effective relative permittivity obtained by our method is 1.06062, while the result obtained by solving (8) using matrix inversion method is 1.06596. They agree well with each other. However, when the number of particles increases, the number of unknowns increases. Subsequently, the direct matrix inversion becomes intractable and our proposed method based on preconditioned EFIE prevails.

When the fractional volume approaches 0.4, the number of

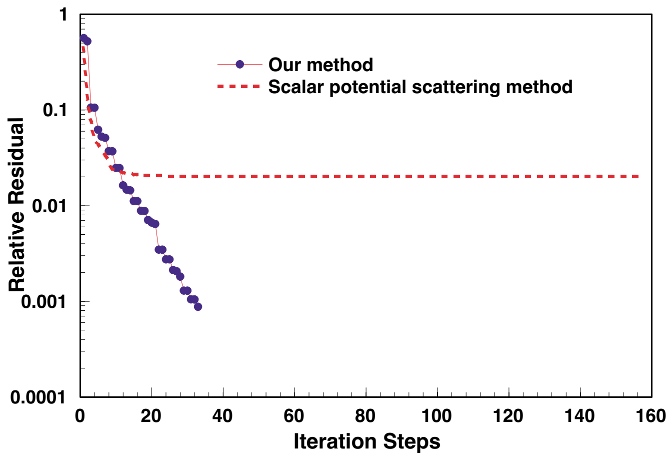

Figure 11. CG iteration convergence comparison for the system of 51 randomly distributed spheres with a fractional volume of 0.02.

randomly distributed spheres and the number of unknowns are 984 and 50,184, respectively. For this case, if the conventional MoM method is used, a huge amount of memory (20.1 GBytes) and CPU time are needed. Fortunately, the IMLMQRF with $O(N \log N)$ computational complexity only requires about 800 MBytes of memory. The CPU time for computing the reduced-rank matrix and 565 CG iterations for the normalized L-2 norm of the residual below 2.5×10^{-2} are about 107 seconds and 1689 seconds, respectively. Note that the entries in matrix $[Z]$ given in (10a) are very simple, thus the matrix filling time in IMLMQRF method is very fast. In this paper, all the mixture permittivity results are calculated by IMLMQRF.

As we are primarily concerned with the mean value of the permittivity, here we use the formula $\langle \varepsilon_{r,\,\mathit{eff}} \rangle = (\sum_{i=1}^{N_r} \varepsilon_{r,\,\mathit{eff}_i})/N_r$ to calculate it, where N_r is the number of realizations for a specific fractional volume. Fig. 12 shows the calculated permittivity versus fractional volume from 0.1 to 0.4. Results obtained by Maxwell-Garnett formula are also plotted in this figure for comparison. We can see that the Maxwell-Garnett curve acts as a lower bound of the mixture permittivity. Note that for each fractional volume, only 10 realizations are averaged. The total CPU time used is about 19 hours for the whole curve.

It is obvious that only 10 realizations for each fractional volume are insufficient. In order to obtain higher accuracy, more realizations should be used. Here, we have performed the ensemble average of the

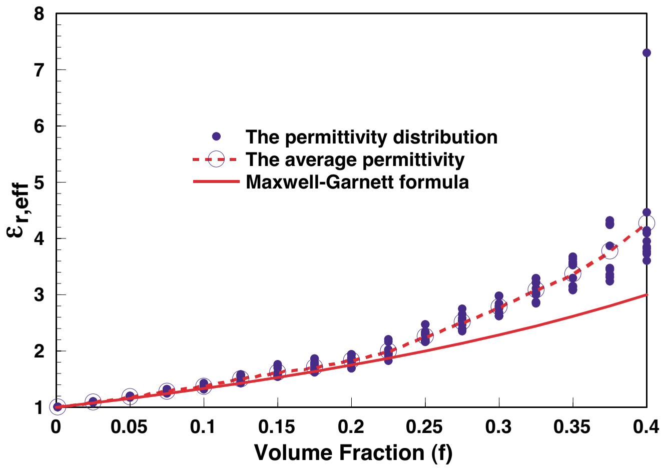

Figure 12. The calculated permittivity of the randomly distributed spherical particles versus fractional volume. The number of realization for each volume fraction is 10.

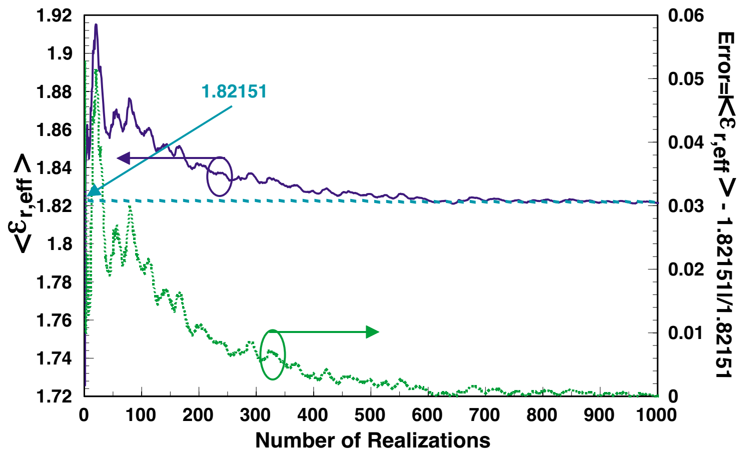

Figure 13. The mean value of the effective relative permittivity and its relative error versus the number of realizations. The fractional volume is 0.2.

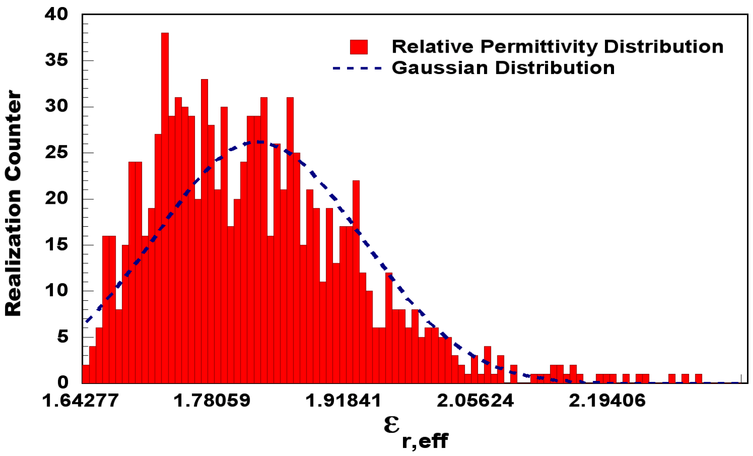

Figure 14. The distribution of the effective relative permittivity.

effective relative permittivity of the system for 1,000 realizations for the fractional volume of 0.2. The number of unknowns is 25143. The total CPU time used is about 75 hours. Fig. 13 shows the average value of the effective relative permittivity versus the number of realizations. We can see that the average value converges at 1.82151.

Fig. 14 shows the distribution of the effective relative permittivity of these 1,000 realizations. Here, the Gaussian distribution with mean value $\mu = 1.82151$, and standard deviation $\sigma = \sqrt{(\sum_{i=1}^{N_r}(\varepsilon_{r,\,eff_i} - \mu)^2)/N_r} = 0.107272$ is also plotted.

6. CONCLUSION

The aim of this paper is to find an efficient integral equation method to solve the static mixture problem. For this purpose, the preconditioned EFIE method, i.e., (26) instead of scalar potential scattering method, i.e., (8) is used to simulate the relative permittivity of the mixture with conducting particles in quasi-static environment. Hence, the matrix arising from (26) is well-conditioned, leading to a more stable solution. To accelerate the matrix-vector multiplication $[Z][x]$ arising from (26), the kernel independent algorithm IMLMQRF for quasi-static problems with the complexity of $O(N \log N)$ is applied. As a result, not only the matrix-vector multiplies in CG iteration is accelerated, but also the memory requirement is drastically reduced. Subsequently, some numerical examples for the permittivity extractions from two kinds of lattice structures and the randomly distributed spherical particles with different fractional volumes and different realizations are efficiently simulated. The numbers of unknowns for these realizations range from about 1,000 to 50,000. Numerical results demonstrating the efficiency of this hybrid method are also given.

If the fractional volume is larger than 0.4, the mesh density should be increased. Consequently, the number of unknown will increase drastically. Hence, new method such as parallelized IMLMQRF should be used to solve it. On the other hand, if the permittivity of the particle is finite, the surface integral equation contains the integro-differential operator acting on magnetic currents. Thus, a more complex treatment should be implemented. However, our method in this paper still can be efficiently used in it. Our next step of the research work is on this aspect.

ACKNOWLEDGMENT

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU1/02C].

APPENDIX A.

A.1. Solution of Equation $[K][x] = [y]$

Before solving $[K][x] = [y]$, seven temporary arrays should be defined. They are $[D]$, $[O]$, $[I]$, $[C]$, $[TC]$, $[x]$ and $[y]$ where D_i, O_i, I_i, C_i, TC_i, x_i, and y_i, $i = 1, \cdots, N_b$, are their ith entries, respectively. D_i is the diagonal entry of column i of matrix $[K]$. O_i is the off-diagonal non-zero entry of column i of matrix $[K]$. I_i is the row index of the off-diagonal non-zero entry of column i of matrix $[K]$. C_i is the number of non-zero entries of row i of matrix $[K]$. TC_i $i = 1, \cdots, N_b$ is used as a temporary counter. x_i is the ith entry of vector $[x]$. y_i is the ith entry of vector $[y]$. Note that $[D]$, $[O]$, $[I]$, $[C]$ and $[y]$ are the input arrays, while $[x]$ is the output solution and $[TC]$ is a temporary array. It is known that the off-diagonal non-zero entry of a column of $[K]$ is at most one. Hence, if the number of the off-diagonal non-zeros of row i is zero, the value of I_i is set to be -1000. The algorithm is given in Table A1.

Because in this algorithm, each non-zero entry of sparse matrix $[K]$ has been called only once, thus its complexity is $O(N_b)$. Note that Step 2.3 is the step for calculating the unknowns, while Step 2.5 is the column elimination step. The main idea of this algorithm is to eliminate the off-diagonal non-zeros recursively.

Table A1. Algorithm 3.

```
Algorithm 3. ($[K][x]=[y]$)
Step 1. Copy vector $[y]$ to vector $[x]$. Copy vector $[C]$ to vector $[TC]$. $counter := 0$.
Step 2. For $i := 1, \cdots, N_b$.
    Step 2.1. $k = i$. $non\_zeros := C_k$.
    Step 2.2. If $non\_zeros = 1$ do Step 2.3. Otherwise, go to Step 2.7
    Step 2.3. $x_k = x_k / D_k$. $counter := conter + 1$.
    Step 2.4. If $I_k \geq 0$ do Step 2.5. Otherwise, go to Step 2.6.
        Step 2.5. $j := I_k$. $x_j := x_j - O_k * x_k$. $TC_j := TC_j - 1$. $k := j$.
    Step 2.6. $non\_zeros := TC_k$. Go to Step 2.2.
    Step 2.7. $i := i + 1$.
    Step 2.8 If $counter = N_b$, break the loop and go to step 3.
Step 3. The solution $[x]$ is obtained.
```

A.2. Solution of Equation $[K]^T[x] = [y]$

Before performing $[K]^T[x] = [y]$, we need another three integer arrays $[H]$, $[S]$, and $[E]$, where H_i, S_i, E_i, $i = 1, \cdots, N_b$ are their entries, respectively. The usage of $[H]$, $[S]$, and $[E]$ is that the entries from H_{S_i} to H_{E_i} store the column indices of the off-diagonal entries of row i of matrix $[K]$. The Algorithm is given in Table A2.

Table A2. Algorithm 4.

```
Algorithm 4. ( $[K]^T[x]=[y]$)
Step 1. Copy $[y]$ to $[x]$. $Counter := 0$.
Step 2. For $i := 1, \cdots, N_b$, do loop.
    Step 2.1. If $I_i < 0$, do step 2.2. Otherwise, go to Step 2.4.
    Step 2.2. $x_i := x_i / D_i$. $Counter := Counter + 1$.
    Step 2.3. If $C_i > 1$, call function Eliminate. Otherwise, go to Step 2.4.
    Step 2.4. If $Counter < N_b$, break the loop and go to Step 3.
Step 3. The solution $[x]$ is obtained.
```

In Algorithm 2, each non-zero of sparse matrix $[K]$ has been called only once. Thus, its computational complexity is $O(N_b)$.

The function ***Eliminate*** is a recursive function. For simplicity, here we set all the variables to be global variables. Thus no input parameter is required for this function. This function is given in Table A3.

Table A3. Function ***Eliminate***.

```
Eliminate
Step 1. For $j = S_i, \cdots, E_i$, do loop.
    Step 1.1. $k = H_j$ . $x_k = x_k - O_k * x_i$ . $x_k = x_k / D_k$ . $Counter = Counter + 1$.
    Step 1.2. If $C_k > 1$, call function Eliminate.
Step 2. The end of function Eliminate.
```

REFERENCES

1. Wang, J. R. and T. J. Schmugge, "An empirical model for the complex dielectric permittivity of soils as a function of water content," *IEEE Trans. Geosci. Remote Sens.*, Vol. 18, 288–295, 1980.
2. Schmugge, T. J., "Effect of soil texture on the microwave emission from soils," *IEEE Trans. Geosci. Rem. Sens.*, Vol. 18, 353–361, 1980.
3. Golden, K. M., "The interaction of microwave with sea ice," *Wave Propagation in Complex Media, IMA Volumes in Mathematics and Its Applications*, G. Papanicolaou (ed.), Vol. 96, 75–94, Springer-Verlag, Berlin, Germany, 1997.
4. Nghiem, S. V. et al., "Evolution in polarimetric signatures of thin saline ice under constant growth," *Radio Sci.*, Vol. 32, No. 1, 127–151, 1997.
5. Perovich, D. K. and A. J. Gow, "A statistical description of the microstructure of young sea ice," *J. Geophys. Res.*, Vol. 96, No. 16, 943–1695, 1991.
6. Linlor, W., "Permittivity and attenuation of wet snow between 4 and 12 GHz," *J. Appl. Phys.*, Vol. 23, 2811–2816, 1980.
7. Colbeck, S. C., "The geometry and permittivity of snow at high frequencies," *J. Appl. Phys.*, Vol. 20, 45–61, 1982.
8. Hallikainen, M. T., F. T. Ulaby, and T. E. V. Deventer, "Extinction behavior of dry snow in the 18–90 GHz range," *IEEE Trans. Geosci. Rem. Sens.*, Vol. 25, 737–750, 1987.
9. Chew, W. C., J. A. Friedrich, and R. Geiger, "A multiple scattering solution for the effective permittivity of a sphere mixture," *IEEE Trans. Geosci. and Rem. Sens.*, Vol. 28, No. 2, 207–214, March 1990.
10. Tsang, L. and J. A. Kong, *Scattering of Electromagnetic Waves: Advanced Topics*, John Wiley & Sons, 2001.

11. McPhedran, R. C. and D. R. McKenzie, "The conductivity of lattices of spheres I. The simple cubic lattice," *Proceeding of the Royal Society*, 359A, 45–63, London, 1978.
12. Doyle, W. T., "The Clausius-Mossotti problem for cubic arrays of spheres," *Journal of Applied Physics*, Vol. 49, 795–797, 1978.
13. Kärkkäinen, K. K., A. H. Sihvola, and K. I. Nikoskinen, "Effective permittivity of mixtures: numerical validation by FDTD method," *IEEE Trans. Geosci. Rem. Sens.*, Vol. 38, No. 3, 1303–1308, May 2000.
14. Kärkkäinen, K. K., A. H. Sihvola, and K. I. Nikoskinen, "Analysis of a three-dimensional dielectric mixture with finite difference method," *IEEE Trans. Geosci. Rem. Sens.*, Vol. 39, No. 5, 1013–1018, May 2001.
15. Whites, K. W., "Permittivity of a multiphase and isotropic lattice of spheres at low frequency," *Journal of Applied Physics*, Vol. 88, No. 4, 1962–1970, August 2000.
16. Wu, F., and K. W. Whites, "Computation of static effective permittivity for a multiphase lattice of cylinders," *Electromagnetics*, Vol. 21. 97–114, 2001.
17. Whites, K. W., and F. Wu, "Effects of particle shape on the effective permittivity of composite materials with measurements for lattice cubes," *IEEE Trans. Microwave Theory Tech.*, Vol. 50, No. 7, 1723–1729, July 2002.
18. Zhao, J. S. and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Trans. Antennas Propagat.*, Vol. 48, No. 10, 1635–1645, October 2000.
19. Wang, H. G., C. H. Chan, L. Tsang, and V. Jandhyala, "An improved multi-level matrix QR factorization for large-scale simulations on magnetoquasistatic analysis of integrated circuits over multi-layered lossy substrates," submit to *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*
20. Even, S., *Graph Algorithms*, Computer Science Press, Rockville, 1979.
21. Graglia, R. D., D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas and Propagat.*, Vol. 45, No. 3, 329–342, March 1997.
22. Kapur, S. and D. E. Long, "IES^3: a fast integral equation solver for efficient 3-dimensional extraction," *IEEE/ACM Int. Conf. Computer-Aided Design*, 448–455, Nov. 1997.

23. Kapur, S. and D. E. Long, "IES^3: efficient electrostatic and electromagnetic simulation," *IEEE Trans. Computational Science and Engineering*, Vol. 5, 60–67, Oct.–Dec. 1998.
24. Golub, G. H. and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.
25. Fischer, A. E., D. W. Eggert, and S. M. Ross, *Applied C: An Introduction and More*, McGraw-Hill, Boston, 2001.
26. Tsang, L., J. A. Kong, K. H. Ding, and C. O. Ao, *Scattering of Electromagnetic Waves Volume II*, John Wiley & Sons, Inc., New York, 2001.
27. Press, W. H. et al., *Numerical Recipes in C++*, University Press, Cambridge, 2002.