Progress In Electromagnetics Research, PIER 93, 369–384, 2009

AN EFFICIENT TWOFOLD ITERATIVE ALGORITHM OF FE-BI-MLFMA USING MULTILEVEL INVERSE-BASED ILU PRECONDITIONING

Z. Peng, X.-Q. Sheng, and F. Yin

Center for Electromagnetic Simulation
School of Information and Electronics
Beijing Institute of Technology
China

Abstract—It is known that the conventional algorithm (CA) of hybrid finite element-boundary integral-multilevel fast multipole algorithm (FE-BI-MLFMA) usually suffers the problem of slow convergence, and the decomposition algorithm (DA) is limited by large memory requirement. An efficient twofold iterative algorithm (TIA) of FE-BI-MLFMA is presented using the multilevel inverse-based incomplete LU (MIB-ILU) preconditioning in this paper. It is shown that this TIA can offer a good balance of efficiency between CPU time and memory requirement. The tree-cotree splitting technique is then employed in the TIA to further improve its efficiency and robustness. A variety of numerical experiments are performed in this paper, demonstrating that the TIA exhibits superior efficiency in memory and CPU time to DA and CA, and greatly improves the computing capability of FE-BI-MLFMA.

1. INTRODUCTION

It has been shown that the hybrid finite element-boundary integral-multilevel fast multipole algorithm (FE-BI-MLFMA) is a general, accurate, and efficient computing technology for open region problems such as scattering/radiation problems (e.g., [1–14]). The conventional algorithm (CA) of FE-BI-MLFMA presented in [10] is to directly apply the iterative solver to the whole hybrid FE-BI matrix equation. Since the whole matrix is not well-conditioned due to the ill-conditioned FEM matrix part, the convergence of CA is usually very slow, which makes CA less attractive [12, 13]. The decomposition algorithm

Corresponding author: X.-Q. Sheng (xsheng@bit.edu.cn).

(DA) is to first employ fast direct sparse solver to factorize the FEM matrix, and then solve the reduced matrix with iterative solver [12]. Numerical experiments show that the decomposition algorithm (DA) converges very fast, but is greatly limited by large additional memory requirement due to the LU factorization for the FEM matrix. Even using advanced sparse solver such as the fast multifrontal LU factorization by [15], the expenditure of the memory is still enormous for large electrical-size problems, and largely limits the application of DA.

A natural idea to overcome the disadvantage of DA is to employ the iterative solver for the FEM part instead of the LU factorization, which leads to our twofold iterative algorithm (TIA). Our TIA is formed by two iterative steps viz., the inner iterative step for the FEM part and the outer iterative step similar to that in DA. Since the FEM matrix usually is not well-conditioned, the inner iterative step will take a very slow convergence. Therefore, the TIA would not be success if no other means could treat this problem. The important fact is that the great achievement of iterative solvers with preconditioning can be suitably employed into the TIA. Hence, the key point of the TIA is how to choose a suitable iterative solver with special preconditioning technique to efficiently complete the inner iterative step of the TIA.

In the past few decades, the preconditioned iterative methods have made great progress for solving large sparse linear systems. The preconditioning technique is essential to the success of the iterative solvers [16]. It has been recognized that among various preconditioning techniques, the incomplete LU (ILU) preconditioning is a good one, which can offer a good compromise between robustness and efficiency. There are many variants of the ILU preconditioning (e.g., [17–22]). The recently proposed multilevel inverse-based ILU (MIB-ILU) shows a great potential for solving large sparse linear systems (see also [21, 22]). In this paper, we employ this MIB-ILU preconditioning into our TIA, and investigate its numerical performance in computing scattering by large complex targets.

It has been pointed out by [23] that the edge elements used in the previous FE-BI-MLFMA suffer from the problem of instability at low frequencies or fine meshes. A tree-cotree splitting technique was proposed to solve this problem [23], and applied to solve 2-D and 3-D waveguiding structures problems in [24, 25]. This paper also tries to apply this tree-cotree splitting technique to the TIA for scattering problems and investigate its numerical performance in this application.

In sum, this paper presents an efficient TIA of FE-BI-MLFMA using the MIB-ILU preconditioning technique. An alternative formulation of the tree-cotree splitting technique is given for another

interpretation and understanding, and is applied to the TIA. A variety of numerical experiments are performed to investigate the numerical performance of the TIA of FE-BI-MLFMA for scattering by large complex targets.

2. TWOFOLD ITERATIVE ALGORITHM OF FE-BI-MLFMA

This section first presents the outline of the TIA. Then the MIB-ILU preconditioning is discussed and applied to the TIA. Finally, the tree-cotree splitting technique is reformulated and applied to the TIA.

2.1. Outline of TIA

It is known that the problem of scattering by a complex target can be discretized by the hybrid FE-BI method [10] as

$$\begin{bmatrix} K_{II} & K_{IS} & 0 \\ K_{SI} & K_{SS} & B \\ 0 & P & Q \end{bmatrix} \begin{Bmatrix} E_I \\ E_S \\ H_S \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ b \end{Bmatrix} \tag{1}$$

where $[K_{II}]$, $[K_{SS}]$, $[K_{IS}]$, $[K_{SI}]$, $[B]$ are sparse FEM matrices, $[P]$ and $[Q]$ are dense BI matrices. Also, $[K_{II}]$, $[K_{SS}]$ are symmetric, $[B]$ is skew symmetric. The iterative solvers such as GMRES are usually chosen for solving (1) instead of direct solvers in order to take full advantage of sparsity of the FEM matrixes and MLFMA to speed up the main step of matrix-vector multiplication in iterative solvers. However, since the coefficient matrix in (1) is not well-conditioned, the convergence speed will be very slow if the iterative solvers are directly applied to the whole equation of (1). It has been verified in [12] that a good approach of DA is to decompose (1) into the FEM and BI equations, and employ the sparse direct solver for the FEM equation and the iterative solver for the BI equation. However, the large memory and CPU time required in the direct solver becomes a bottleneck of DA when the FEM domain is large. To overcome this bottleneck, a natural idea is to employ preconditioned iterative solvers for the FEM equation instead of the direct solvers, which leads to our TIA. To be more specific, Equation (1) is decomposed into the following FEM equation

$$\begin{bmatrix} K_{II} & K_{IS} \\ K_{SI} & K_{SS} \end{bmatrix} \begin{Bmatrix} E_I \\ E_S \end{Bmatrix} = - \begin{bmatrix} 0 \\ B \end{bmatrix} \{H_S\}, \tag{2}$$

and the BI equation

$$[Q]\{H_S\} = \{b\} - [P]\{E_S\}. \tag{3}$$

Our TIA is as follows: (1) Given an initial guess $\{H_S\}$, $\{E_S\}$ is solved from (2) using the preconditioned GMRES iterative solver. This can be called as the inner iterative step; (2) With the obtained $\{E_S\}$ in the inner step and the initial guess $\{H_S\}$, $\{H_S\}$ is updated from (3) using the GMRES iterative solver. This can be called as the outer iterative step. These inner and outer iterative steps form a complete step of the TIA, which repeats till convergence.

2.2. The MIB-ILU Preconditioning

The key step of the TIA is to choose a suitable preconditioning technique for speeding up the convergence of GMRES for (2) in the inner step. Here we choose the recently proposed MIB-ILU preconditioning, which has shown to be a great potential for large sparse matrix equation.

This MIB-ILU preconditioning is established on the Crout LU factorization shown in Figure 1. The dark area shows the parts of the entries being computed at the k-th step. The shaded areas show the parts of the entries being accessed at the k-th step. It can be seen that the Crout LU factorization computes the k-th column of L and k-th row of U at step k. In our implementation, we store L by columns and U by rows, thus the entries can be searched and accessed efficiently during the factorization.

The ILU factorization is obtained by applying dropping rule to the entries of the k-th column of L and k-th row of U for each step k. The entries usually are dropped either by their location (positional dropping) or magnitude (threshold dropping). The positional dropping works well for regularly structured problems, but it is not designed for

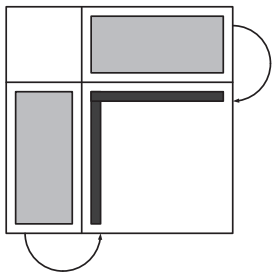

Figure 1. The computational pattern for the Crout factorization. The dark area shows the parts of the entries being computed at the k-th step. The shaded areas show the parts of the entries being accessed at the k-th step.

general cases. The threshold dropping is often used, but it is not robust. Since the threshold dropping cannot guarantee that the norms of the inverse of L or U is small, this may have some unpredictable effects to the ILU preconditioner.

To overcome this problem, a better dropping rule is presented by [21], which is to drop an entry l_{jk} at step k when it satisfies $|l_{jk}|\|e_k^T L^{-1}\| \leq \mathcal{E}$, where e_k^T denotes the k-th unit vector, and $\mathcal{E}$ is the ILU drop tolerance. Since this dropping rule is based on the inverse of L or U, this ILU is called as the inverse-based ILU.

Since the inverse of L is usually not available, the next problem is how to estimate $\|e_k^T L^{-1}\|$. To this end, the estimation of $\|e_k^T L^{-1}\|$ is approximated as $\|e_k^T L^{-1} b\|_\infty$ for any testing vector b satisfying $\|b\|_\infty = 1$. Since e_k^T denotes the k-th unit vector, the problem to estimate $\|e_k^T L^{-1} b\|_\infty$ at step k is reduced to determine the largest k-th component of the solution to the linear system $Lx = b$. The implementation adopted in this paper uses the simplest rule which amounts to selecting $b_k = \pm 1$ at each step k, in such a way as to maximize the k-th component of $L^{-1}b$.

To take full advantage of the idea of the inverse-based ILU, we extend it to a multilevel version. Suppose the k-th column of L and k-th row of U have been calculated by the Crout factorization, we then estimate $\|L^{-1}\|$ and $\|U^{-1}\|$. If $\|L^{-1}\| < \kappa$ and $\|U^{-1}\| < \kappa$ (κ is the set bound), then the inverse-based ILU is performed, otherwise, the k-th column and row will be postponed to the last column and row. This procedure is repeated through all columns and rows. Finally, all columns and rows of matrix K are categorized into two types: factorized and postponed. To be matched with the concepts of algebraic multilevel method, the factorized columns and rows can be considered corresponding to the fine grid, whereas the postponed corresponding to the coarse grid. For clear description, the original matrix can be rearranged as

$$K = \begin{pmatrix} K_{FF} & K_{FC} \\ K_{CF} & K_{CC} \end{pmatrix}. \tag{4}$$

Since the inverse-based ILU on K_{FF} has been performed, the preconditioner $\tilde{K}_{FF}^{-1}$ of K_{FF} can be easily constructed. Thus, based on the following multiplicative Schwarz approximation

$$K^{-1} \approx \begin{bmatrix} I & -K_{FF}^{-1}K_{FC} \\ 0 & I \end{bmatrix} \begin{bmatrix} K_{FF}^{-1} & 0 \\ 0 & \left(K_{CC} - K_{CF}K_{FF}^{-1}K_{FC}\right)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K_{CF}K_{FF}^{-1} & I \end{bmatrix}, \tag{5}$$

The construction of the preconditioner of K is reduced to the construction of preconditioner of S_{CC}, where $S_{CC} = K_{CC} - K_{CF}K_{FF}^{-1}K_{FC}$. In other words, the remained problem is to perform the inverse-based ILU in the coarse level. This procedure can recursively proceed to obtain a multilevel inverse-based ILU precondioner.

To improve the numerical performance of the above presented MIB-ILU, here we also employ other two advanced preprocessing techniques before performing MIB-ILU. The first one is the symmetric weighted matching technique proposed in [15]. This technique tries to permute the large entries close to the diagonal to enhance the stability of the LU factorization. The second one is the nested dissection technique, which is a fill-in reducing reordering technique [27]. This paper directly employs the version of the nested dissection in METIS [28].

2.3. The Tree-cotree Splitting Technique

It is known that the tetrahedron edge element is generally employed in the FEM part of FE-BI-MLFMA. However, it has been found in [23] that this edge element usually suffers from the problem of instability at low frequencies or fine meshes. A tree-cotree splitting technique was proposed to handle this problem [23–25]. To be more specific, the electric field $\mathbf{E}(\mathbf{r})$ is modeled by

$$\mathbf{E}(\mathbf{r}) = \sum_{i=1}^{4} E_g^i \nabla\varphi_i + \sum_{j=1}^{6} E_r^j \mathbf{N}_j, \tag{6}$$

where φ_i is the same nodal basis function as that in the conventional node element, and $\mathbf{N}_j$ is the same edge basis function as that in the conventional edge element. Since the interpolating parameters in (6) include those on edges and nodes, this element can be formally called as the node-edge element. It is obvious that some interpolating parameters in the node-edge element are redundant, and should be removed in the practical application [24]. It is worth to note that the interpolating parameter E_r^j still truly stands for the tangential field at tetrahedron edge, but E_g^i does not represent for the gradient value at tetrahedron node in (6), since $\nabla\varphi_i$ is a linear combination of $\mathbf{N}_j$. Thus the node-edge element still enforces the continuity of tangential field between neighbor tetrahedrons, and gives the flexibility of normal field between neighbor tetrahedrons.

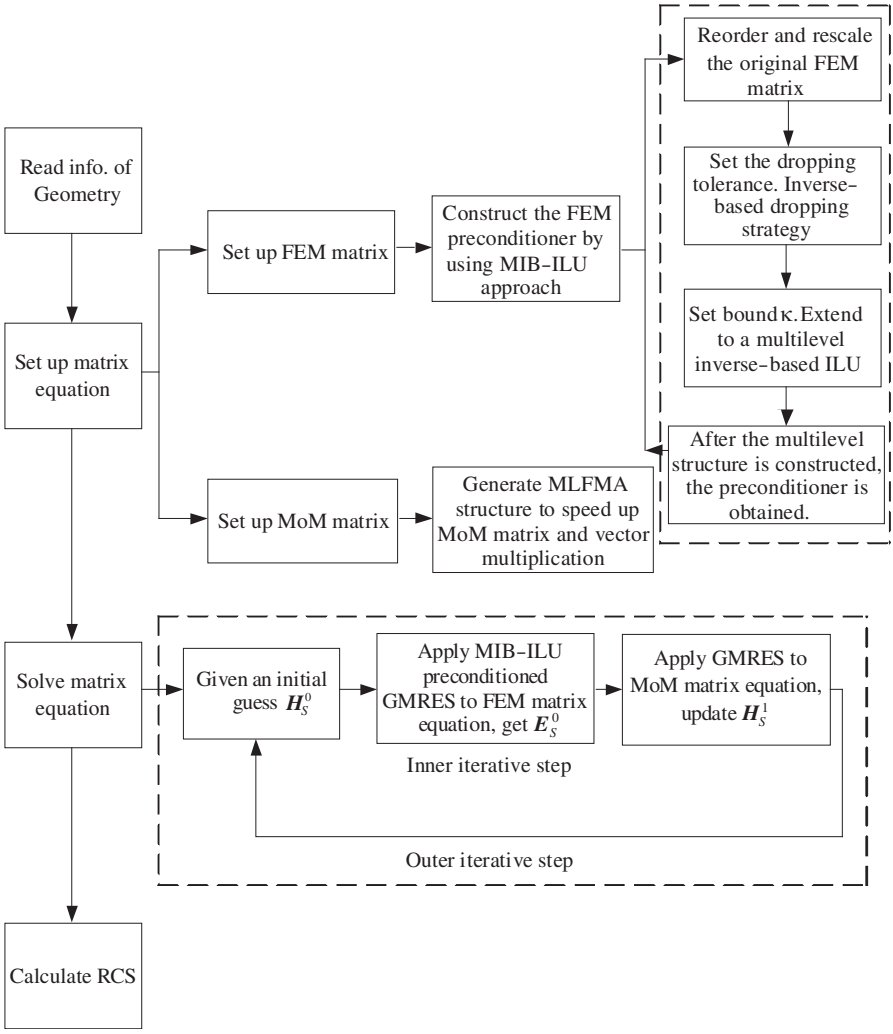

Figure 2. The flow chart of TIA algorithm.

3. NUMERICAL EXAMPLES

To demonstrate the accuracy, efficiency, and versatility of TIA described above, we present several numerical examples in this section. The program is developed according to the flow chart in Figure 2, and all the computations are performed on an IBM server xSeries 366, Xeon MP 3.66 GHz, 16 GB memory.

3.1. Investigation of the MIB-ILU Preconditioning

It has been analyzed that the numerical performance of the MIB-ILU preconditioning is the key to the success of the TIA. Thus, this section will focus on investigating the numerical performance of the MIB-ILU preconditioning. The first numerical example is an open cavity as shown in Figure 3.

The cross section of the open cavity is $2\lambda \times 2\lambda$. The depth of the cavity h increases from 1λ to 10λ. In this experiment, we use the incomplete LU with threshold (ILUT) preconditioning technique as the benchmark to exhibit the performance of our proposed MIB-ILU preconditioning method. Before comparing the two preconditioning approaches, it is better to perform some tests for the ILUT preconditioning with different parameters and pick the best one to use. The dropping tolerance is the parameter which determines the performance and fill-in of the ILUT preconditioner. We will first examine the convergence of using ILUT preconditioning GMRES with different dropping tolerance. Numerical results are shown in Table 1, where N is the restart number of GMRES, which varies from 20 to 200. From the results we can see that although the smaller dropping tolerance will result in a better convergence, but the fill-in factor and

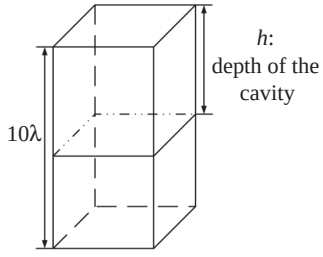

Figure 3. An open cavity object.

Table 1. Computation information of the ILUT-GMRES by varying the dropping tolerance and restart number N.

Dropping tolerance	Filling factor	Factorization time	Number of Iteration			
			N=20	N=50	N=100	N=200
0.05	2.95	3.2	272	137	65	65
0.01	5.85	12.0	180	35	35	35
0.005	10.98	25.5	42	23	23	23
0.001	37.25	76.2	6	6	6	6

factorization time significantly increase. Hence, we choose dropping tolerance of ILU as 0.005 and the restart number of GMRES as 100 in the following experiments.

Next we will compare the performance of the proposed MIB-ILU preconditioning GMRES with the ILUT-GMRES. The detail of computational information is listed in Table 2. ILUT-GMRES takes tens of iterations to converge to a relative residual error smaller than 10^{-4}. In contrast, MIB-ILU-GMRES converges to a relative residual error smaller than 10^{-6} with less than 10 iterations. Also we can see that the memory and factorization time required by MIB-ILU is much smaller than ILUT. In this computation, the set bound κ for $\|L^{-1}\|$ and $\|U^{-1}\|$ is 50 and the drop tolerance $\mathcal{E}$ is chosen as 0.02. It is worth to note that the optimal parameters for different objects may be different, but the numerical performances overall are acceptable. In this work, we also examine the performance of the two preconditioning technique with another Krylov iterative solver, BiCGStab. The MIB-ILU technique again shows much better efficiency than the ILUT technique. The numerical results are omitted here for the sake of simplicity.

It is interesting that the number of iterations in both ILUT-GMRES and MIB-ILU-GMRES increases slowly with the number of unknowns. This may be attributed to the geometrical specialty of this example. In this numerical experiment, we just increase the depth of the cavity and also keep the density of the mesh. Thus our numerical experiment actually is one-dimensional variation with the same mesh density. It is also easily concluded from Table 2 that the solving time of MIB-ILU-GMRES is largely dominated by the factorization part. This preconditioning technique is very suitable for our TIA, since the

Table 2. Comparisons of the ILUT-GMRES and MIB-ILU-GMRES solver for the open cavity problem.

		Memory (MB)		Number of Iteration		The CPU time for Factorization (Sec)		The CPU time for Solving FEM matrix (Sec)	
Depth (λ)	Unknowns	ILUT	MIB-ILU	ILUT	MIB-ILU	ILUT	MIB-ILU	ILUT	MIB-ILU
1	30,450	157	39	18	3	8.4	3.8	0.05	0.01
2	59,660	250	70	23	3	25.5	13.0	0.15	0.01
4	118,080	498	130	24	5	88.0	27.0	0.6	0.04
6	176,500	820	180	26	5	115.0	42.0	1.8	0.05
8	234,920	1198	240	28	5	200.0	58.0	2.5	0.08
10	293,340	1598	360	32	5	358.0	73.0	6.0	0.10

factorization is only performed once and we need to solve FEM matrix equation in each inner iterative step. The memory and CPU time required by the MIB-ILU-GMRES for solving the open cavity problem are plotted in Figure 4. It is clearly seen from Figure 4 that the memory and CPU time grow almost linearly with the number of the unknowns.

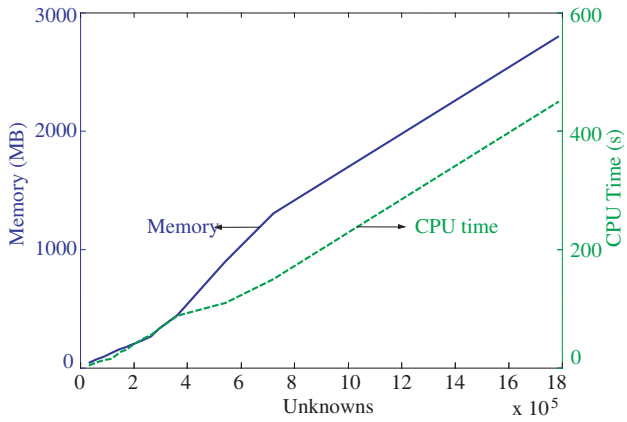

Figure 4. The memory and CPU time required by the MIB-ILU-GMRES for solving the FEM matrix of the open cavity.

Table 3. Numerical performance of the MIB-ILU-TIA versus the DA and CA for the coated spheres.

Diameter (λ)	Unknown MoM / FEM	Number of iteration			Memory (MB)			Total solving CPU time (Sec)		
		CA	DA	TIA	CA	DA MoM / FEM	TIA MoM / FEM	CA	DA	TIA
2	4800 / 16002	240	24	24	65	60 / 19	60 / 8	61	4.3	4.2
4	19200 / 64002	335	35	35	260	234 / 88	234 / 40	450	23.8	20.0
6	43200 /144002	402	43	43	444	384 / 254	384 /95	1609	128.9	114.4
8	76800 / 256002	>500	54	54	1035	927 / 433	927 / 150	>5000	202.2	180.0
10	120000 /400002	>500	65	65	1287	1027 / 687	1027 / 200	>5000	421.2	370.2

Table 4. Numerical performance of the MIB-ILU-TIA versus the DA and CA for the isotropic dielectric sphere.

Diameter (λ)	Unknown MoM / FEM	Number of iteration			Memory (MB)			Total solving CPU time (sec)		
		CA	DA	TIA	CA	DA MoM / FEM	TIA MoM / FEM	CA	DA	TIA
1	1200 / 9960	859	14	14	20	18 / 56	18 / 15	33	3.3	5.0
2	4800 / 77120	1689	17	16	93	80 / 550	80 / 100	442	67.9	65.0
4	10800 / 257480	>5000	20	20	/	187 / 2829	187 / 400	>5000	987.2	506.2
6	24300 / 862770	>5000	/	25	/	400 / 15565	400 / 2200	>5000	/	2000.0
8	43200 / 2037760	>5000	/	27	/	1002 / 31500*	1002 / 4500	>5000	/	5000.2

/: Fail in computation
*: Estimated.

3.2. Comparison of TIA and DA As Well As CA Algorithm

In this section, the numerical performances of the conventional algorithm (CA) [10], the decomposition algorithm (DA) [12] and the twofold iterative algorithm (TIA) are investigated and compared by typical experiments. The fast multifrontal LU factorization is used to factorize the FEM matrix in DA. We computed two types of objects: coated spheres and dielectric spheres. Table 3 presents the iteration number, total CPU time and memory requirement of the three algorithms for computing scattering by coated spheres having different diameter ($D = 1\lambda, 2\lambda, \ldots, 10\lambda$) of conducting core, and 0.05λ thickness and $\varepsilon_r = 4.0 - j1.0$, $\mu_r = 1.0$ of the coating material. Table 4 presents the computation information of the three algorithms for dielectric sphere with different diameter ($D = 1\lambda, 2\lambda, \ldots, 8\lambda$). It can be seen that the convergence speed of the TIA is almost the same as that of DA, and much faster than that of CA, and the memory requirement of TIA is much less than that of DA. In sum, numerical results demonstrate that the TIA offers a good compromise between robustness and efficiency, and greatly improves the capability of FE-BI-MLFMA.

3.3. Investigation of the Node-edge Element

It has been demonstrated in the work of [25] that the application of the tree-cotree splitting can greatly improve the stability of the h-p adaptive refinement in FEM especially when the size of the element is very small. In this work, this section will investigate the numerical

Table 5. The comparision of the fill-in factor in the MIB-ILU preconditioning for solving the FEM matrixes generated by node-edge (NE) element and edge element (EE) for a coated sphere with 0.05 λ-thick coating.

Diameter (λ)	Unknowns	Lossy dielectric $\varepsilon_r = 4 - j$		Uniaxial dielectric $\varepsilon_r = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$		Anisotropic dielectric $\varepsilon_r = \begin{bmatrix} 3 & 2j & 0 \\ -2j & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$	
		NE	EE	NE	EE	NE	EE
2	32004	2.6	4.7	3.0	6.9	4.3	7.6
4	108804	3.0	5.6	3.4	7.2	4.4	9.3
6	244804	3.2	5.7	3.6	7.5	4.6	9.7
8	435204	3.4	5.8	3.7	8.2	4.6	10.8
10	680004	3.5	6.0	3.8	/	4.7	/

performance of utilizing this technique in the TIA. We use the MIB-ILU approach to construct preconditioners for the matrixes generated by the node-edge element and the edge element. It is known that the fill-in factor is an important estimation parameter of a preconditioner, which determines the memory requirement. Table 5 lists the fill-in factors in the construction of the two preconditioners for matrices generated by the node-edge element and edge element for different media. From Table 5 we can see the MIB-ILU approach even fails in constructing a successful preconditioner for matrix generated by edge element when the FEM region is electrically large and anisotropic. However, the node-edge element is always stable.

3.4. Capability of TIA

To further demonstrate the capability of the TIA, the scattering by an electrically large dielectric sphere is computed, whose diameter is 10λ, and relative permittivity is $3.0 - j1.0$. In this computation, the unknowns of the FEM part and MoM part are 3, 971, 350 and 67, 500. The number of nonzero in the FEM matrix is 34, 413, 951. The outer iteration for the MoM equation converges in 31 iterations. The memory requirement of FEM part and MoM part are 7.1 GB and 1.4 GB respectively. The total CPU time is 7,500 seconds. The comparison of calculated bistatic RCS in the E-plane and Mie series is shown in Figure 5. The second example is a coated sphere, whose

conducting core has a diameter of 25λ. The conducting sphere is coated with two 0.05λ-thick lossy dielectric layers. The relative permittivity is $2.5 - j0.5$ for the inner layer and $1.5 - j0.5$ for the outer layer. The unknown of the MoM part is 480,000 and the unknown of the FEM

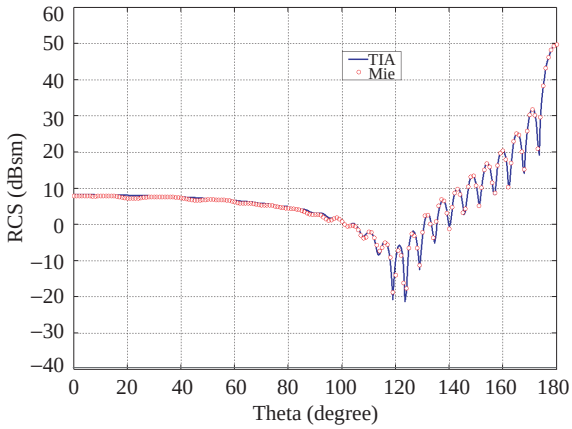

Figure 5. Bistatic RCS in the E-plane for a dielectric sphere with a diameter of 10λ and permittivity of $3 - j$.

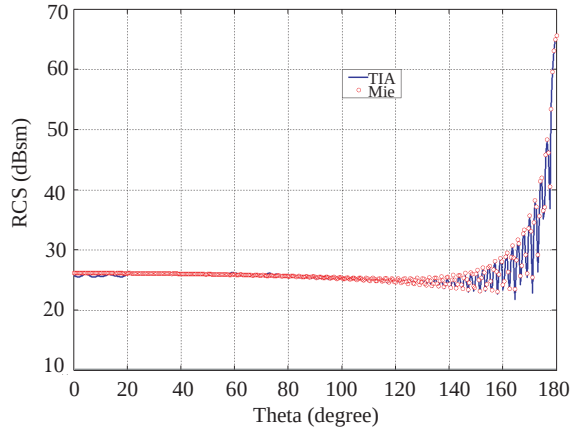

Figure 6. Bistatic RCS in the E-plane for a coated sphere with two layers. The conducting core has a diameter of 25λ, inner layer has a thickness of 0.05λ, a relative permittivity of $2.5 - j0.5$, outer layer has a thickness of 0.05λ, a relative permittivity of $1.5 - j0.5$.

part is 2,720,004 and the number of nonzero in the FEM matrix is 20,960,005. The memory requirement of the FEM and MoM part are only 2.5 GB and 4.6 GB. The total CPU time is 4,692 seconds. The calculated result is shown in Figure 6, from which we can see the high accuracy and efficiency of the proposed TIA.

4. CONCLUSIONS

An efficient TIA of FE-BI-MLFMA is presented using the MIB-ILU preconditioning and the tree-cotree splitting technique. The analysis and numerical experiments of the TIA demonstrate that: (1) the MIB-ILU preconditioning technique is highly efficient and accurate, and very stable even the computational domain is electrically large and anisotropic; (2) the tree-cotree splitting technique further enhances the robustness and improves the efficiency of TIA.

ACKNOWLEDGMENT

Here we thank professor Jin-Fa Lee for reminding us to pay attention to the works [23–25]. This work was supported by the NSFC under Grant 10832002 and the 973 Program under Grant.2005CB321702.

REFERENCES

1. Yuan, X., "Three-dimensional electromagnetic scattering from inhomogeneous objects by the hybrid moment and finite element method," *IEEE Trans. Microwave Theory Tech.*, Vol. 38, No. 8, 1053–1058, 1990.
2. Angelini, J. J., C. Soize, and P. Soudais, "Hybrid numerical method for harmonic 3-D Maxwell equations: Scattering by a mixed conducting and inhomogeneous anisotropic dielectric medium," *IEEE Trans. Antennas Propagat.*, Vol. 41, No. 1, 66–76, 1993.
3. Antilla, G. E. and N. G. Alexopoulos, "Scattering from complex three-dimensional geometries by a curvilinear hybrid finite-element-integral equation approach," *J. Opt. Soc. Amer. A*, Vol. 11, No. 4, 1445–1457, 1994.
4. Boyes, W. E. and A. A. Seidl, "A hybrid finite element method for 3-D scattering using nodal and edge elements," *IEEE Trans. Antennas Propagat.*, Vol. 42, No. 10, 1436–1442, 1994.
5. Rogier, H., F. Olyslager, and D. De Zutter, "A hybrid finite element integral equation approach for the eigenmode analysis

of complex anisotropic dielectric waveguides," *Radio Science*, Vol. 31, No. 4, 999–1010, 1996.

6. Eibert, T. and V. Hansen, "Calculation of unbounded field problems in free space by a 3D FEM/BEM-hybrid approach," *Journal of Electromagnetic Waves and Applications*, Vol. 10, No. 1, 61–78, 1996.
7. Cwik, T., C. Zuffada, and V. Jamnejad, "Modeling three-dimensional scatterers using a coupled finite element — Integral equation formulation," *IEEE Trans. Antennas Propagat.*, Vol. 44, No. 4, 453–459, 1996.
8. Bindiganavale, S. S. and J. L. Volakis, "A hybrid FE-FMM Technique for electromagnetic scattering," *IEEE Trans. Antennas Propagat.*, Vol. 45, No. 1, 180–181, 1997.
9. Soudais, P., H. Steve, and F. Dubois, "Scattering from several test-objects computed by 3-D hybrid IE/PDE methods," *IEEE Trans. Antennas Propagat.*, Vol. 47, No. 4, 646–653, 1999.
10. Sheng, X. Q., J. M. Song, C. C. Lu, and W. C. Chew, "On the formulation of hybrid finite-element and boundary-integral method for 3D scattering," *IEEE Trans. Antennas Propagat.*, Vol. 46, No. 3, 303–311, 1998.
11. Rogier, H., B. Baekelandt, F. Olyslager, and D. De Zutter, "Application of the FE-BIE technique to problems relevant to electromagnetic compatibility: Optimal choice of mechanisms to take into account periodicity," *IEEE Trans. on Electromagnetic Compatibility*, Vol. 42, No. 3, 246–256, 2000.
12. Sheng, X. Q. and E. K. N. Yung, "Implementation and experiments of a hybrid algorithm of the MLFMA-enhanced FE-BI method for open-region inhomogeneous electromagnetic problems," *IEEE Trans. Antennas Propagat.*, Vol. 50, No. 2, 163–167, 2002.
13. Liu, J. and J. M. Jin, "A highly effective preconditioner for solving the finite element-boundary integral matrix equation for 3-D scattering," *IEEE Trans. Antennas Propagat.*, Vol. 50, No. 9, 1212–1221, 2002.
14. Vouvakis, M. N., S. C. Lee, K. Z. Zhao, and J. F. Lee, "A symmetric FEM-IE formulation with a single-level IE-QR algorithm for solving electromagnetic radiation and scattering problems," *IEEE Trans. Antennas Propagat.*, Vol. 52, No. 11, 3060–3070, 2004.
15. Duff, I. S. and J. K. Reid, "The multifrontal solution of indefinite sparse symmetric linear system," *ACM Trans. on Mathematical Software*, Vol. 9, No. 3, 302–325, 1983.

16. Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company Press, Boston, 1996.
17. Saad, Y. and J. Zhang, "BILUM: Block versions of multi-elimination and multi-level ILU preconditioner for general sparse linear systems," *SIAM J. Sci. Comput.*, Vol. 20, 2103–2121, 1999.
18. Zhang, J., "A grid based multilevel incomplete LU factorization preconditioning technique for general sparse matrices," *Applied Mathematics and Computation*, Vol. 124, No. 1, 95–115, 2001.
19. Bollhofer, M. and Y. Saad, "On the relations between ILUs and factored approximate inverses," *SIAM J. Matrix Anal. Appl.*, Vol. 24, 219–237, 2002.
20. Li, N., Y. Saad, and E. Chow, "Crout versions of ILU for general sparse matrices," *SIAM J. Sci. Comput.*, Vol. 25, No. 2, 716–728, 2003.
21. Bollhofer, M., "A robust and efficient ILU that incorporates the growth of the inverse triangular factors," *SIAM J. Sci. Comput.*, Vol. 25, No. 1, 86–103, 2003.
22. Bollhofer, M. and Y. Saad, "Multilevel preconditioners constructed from inverse-based ILUs," *SIAM J. Sci. Comput.*, Vol. 27, No. 5, 1627–1650, 2006.
23. Albanese, R. and G. Rubinacci, "Solution of three dimensional eddy current problems by integral and differential methods," *IEEE Trans. Magn.*, Vol. 24, No. 1, 98–101, 1988.
24. Lee, S. C., J.-F. Lee, and R. Lee, "Hierarchical vector finite elements for analyzing wave guiding structures," *IEEE Trans. Microwave Theory Tech.*, Vol. 51, No. 8, 1897–1905, 2003.
25. Lee, J. F. and D. K. Sun, "p-Type multiplicative Schwarz (pMUS) method with vector finite elements for modeling three-dimensional waveguide discontinuities," *IEEE Trans. Microwave Theory Tech.*, Vol. 52, No. 3, 864–870, 2004.
26. Duff, I. S. and S. Pralet, "Strategies for scaling and pivoting for sparse symmetric indefinite problems," *SIAM J. Matrix Anal. Appl.*, Vol. 27, No. 2, 313–340, 2005.
27. Karypis, G. and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, Vol. 20, No. 1, 359–392, 1998.
28. Karypis, G. and V. Kumar, Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, available on line at: http://www.cs.umn.edu/∼ karypis/metis/metis.html, 1998.