Two-level Preconditioning for Electromagnetic Scattering from Composite Conducting and Dielectric Objects *

Jeonghwa Lee[1†] , Jun Zhang[1‡] , and Cai-Cheng Lu[2§]

[1]Laboratory for High Performance Scientific Computing and Computer Simulation,
Department of Computer Science, University of Kentucky,
Lexington, KY 40506–0046, USA

[2] Department of Electrical and Computer Engineering,
University of Kentucky,
Lexington, KY 40506-0046, USA

Abstract

A coupled surface-volume integral equation approach is used for the calculation of electromagnetic scattering with composite conducting and dielectric materials. This approach has high accuracy in solutions and can be accelerated by the multilevel fast multipole algorithm (MLFMA) to reduce the CPU time and memory. By using the method of moments (MoM), the coupled surface-volume integral equations are converted into discrete matrix equations. The near part of the resulting matrix consists of four block components. Due to this specific data pattern, we can construct two-level (TL) preconditioners to treat the conducting and dielectric parts more efficiently. The main purpose of this study is to show that the TL preconditioning is effective with the MLFMA and can reduce the number of Krylov iterations substantially by using subdivided domain of the near part matrix for each level. Our experimental results indicate that the TL preconditioning maintains the computational complexity of the MLFMA, but converges a lot faster, thus effectively reduces the overall simulation time for the coated material objects.

Key words: electromagnetic scattering, method of moments, multilevel fast multipole algorithm, Krylov methods, two-level preconditioning

Mathematics Subject Classification: 65F10, 65R20, 65F30, 77C10

1 Introduction

In electromagnetic scattering, general radar targets consist of conductors and materials. The electromagnetic wave scattering by three dimensional arbitrarily shaped dielectric and conducting

*This research work was supported in part by NSF under grants CCR-9988165, CCR-0092532, ACR-0202934, ACR-0234270, and ECS-0093692, by DOE under grant DE-FG02-02ER45961, by ONR under grant N00014-00-1-0605, and by the University of Kentucky Research Committee.

†E-mail: leejh@engr.uky.edu, URL: http://www.csr.uky.edu/∼leejh.

‡E-mail: jzhang@cs.uky.edu, URL: http://www.cs.uky.edu/∼jzhang.

§E-mail: cclu@engr.uky.edu, URL: http://www.engr.uky.edu/∼cclu.

objects can be obtained by finding the solution of an integral equation. The solution to electromagnetic wave interaction with material coated objects has applications in radar cross section (RCS) prediction for coated targets, printed circuit, and microstrip antenna analysis. In particular, the computation of RCS is used to identify unknown targets as well as to develop stealth technology. The integral equations are discretized into a matrix equation by the method of moments (MoM). With this procedure, we obtain a linear system of the form

$$Ax = b, \tag{1}$$

where the coefficient matrix A is a large scale, dense, and complex valued matrix for electrically large targets. The matrix equation (1) can be solved by numerical matrix equation solvers. Our goal is to solve this matrix equation fast by using only limited computer resources (CPU time and memory).

A matrix problem involving N unknowns (columns of the matrix) may be solved in $cN^{\text{iter}}N^{Ax}$ floating point operations, where c is a constant depending on the implementation of a particular (Krylov) iterative method, N^{iter} is the number of iterations needed to reach the prespecified convergence criterion, and N^{Ax} is the floating point operations needed for each matrix-vector multiplication. By using the multilevel fast multipole algorithm (MLFMA), the complexity of the matrix-vector multiplication operations are reduced to $O(N \log N)$, from the $O(N^2)$ complexity of naive implementations. For many realistic problems, N^{iter} depends on both the iterative solver and the target properties (shape and material). With other parts of the MLFMA optimized to some degree, reducing the number of iterations, N^{iter}, has the potential to reduce the CPU time substantially. Hence preconditioning techniques, which transform the original problem into an equivalent (hopefully easier) problem, may speed up the convergence rate of the Krylov methods (iterative solvers), are needed in this application.

In the MLFMA implementation, the near part of the coefficient matrix A is explicitly available. Specially, the near part matrix which comes from the material coated objects consists of four explicit block components. Four block components are the conducting part, the dielectric part, and conducting-dielectric interface parts. Due to this data pattern and our successful experience in building the incomplete LU factorization (ILUT) [5] and the sparse approximate inverse (SAI) [7] preconditioners for the MLFMA, we propose to construct a domain specific two-level (TL) preconditioner from the readily available near part submatrix B (Figure 1(a)) of the coefficient matrix A in the MLFMA. Several two-level preconditioning techniques for the *sparse matrices* are dealt with and analyzed in [1, 2, 10, 14].

We also present some computational eigenvalue analysis on the preconditioned matrix to justify the use of TL preconditioning techniques. One important factor in TL preconditioning is that the conducting and dielectric parts are considered and factored separately. It is possible to selectively treat these two parts by allowing more fill-ins in the computation of one part than the other. Thus better preconditioners may be computed with only a limited amount of resources (computing cycles and memory).

This paper is organized as follows. Section 2 gives a concise introduction to the coupled surface-volume integral equation approach in electromagnetic scattering and the MLFMA. In Section 3, we outline the details to construct the TL preconditioner from the near part matrix. Numerical experiments with a few electromagnetic scattering problems are used to show the efficiency of the TL preconditioning techniques in Section 4. Some concluding remarks are given in Section 5.

2 Coupled Surface-Volume Integral Equations and MLFMA

A coupled integral equation approach combines the volume integral equation (VIE) and the surface integral equation (SIE) to model the scattering and radiation by mixed dielectric and conducting structures [8, 11]. For example, when a radome is applied to an antenna, the combined system consists of both dielectrics and conductors. Hence, the coupled surface-volume integral equation is ideal for this problem [3]. The volume integral equation is applied to the material region (V) and the surface integral equation is enforced over the conducting surface (S). The integral equations can be formally written as follows,

$$\begin{aligned} \{L_S(r,r') \cdot J_S(r') + L_V(r,r') \cdot J_V(r')\}_{\tan} &= -E^{\rm inc}_{\tan}(r), \qquad r \in S, \\ -E + L_S(r,r') \cdot J_S(r') + L_V(r,r') \cdot J_V(r') &= -E^{\rm inc}(r), \qquad r \in V, \end{aligned}$$

where $E^{\rm inc}$ stands for the excitation field produced by an instant radar, the subscript “tan” stands for taking the tangent component from the vector it applies to. $L_\Omega, (\Omega = S, V)$, is an integral operator that maps the source J_Ω to the electric field $E(r)$ and it is defined as

$$L_\Omega(r,r') \cdot J_\Omega(r') = i\omega\mu_b \int_{\Omega'} \left(I + k_b^{-2}\nabla\nabla\right) G(r,r') \cdot J_\Omega(r')\, d\Omega'.$$

Here $G(r,r') = e^{ik_b|r-r'|}/(4\pi|r-r'|)$ is the 3D scalar Green's function for the background media, and $i = \sqrt{-1}$. It should be pointed out that E is related to J_V in the above integral equations by $J_V = i\omega(\epsilon_b - \epsilon)E$. This results in a very general model as all the volume and surface regions are modeled properly. The advantage of this approach is that in the coated object scattering problems, the coating material can be inhomogeneous, and in the printed circuit and microstrip antenna simulation problems the substrate can be of finite size. The simplicity of the Green's function in both the volume integral equation and the surface integral equation has an important impact on the implementation of the fast solvers. However, the additional cost here is the increase in the number of unknowns since the volume that is occupied by the dielectric material is meshed. This results in larger memory requirement and longer solution time in solving the corresponding matrix equation. But this deficiency can be overcome by applying fast integral equation solvers such as the multilevel fast multipole algorithm (MLFMA) [3].

Using the method of moments (MoM), the coupled integral equations are discretized into a matrix equation of the form

$$\begin{bmatrix} Z^{SS} & Z^{SV} \\ Z^{VS} & Z^{VV} \end{bmatrix} \cdot \begin{bmatrix} a^S \\ a^V \end{bmatrix} = \begin{bmatrix} U^S \\ U^V \end{bmatrix}, \qquad (2)$$

where a^S and a^V stand for the vectors of the expansion coefficients for the surface current and the volume function, respectively [3, 8], and the matrix elements can be generally written as

$$Z_{jl} = i\omega\mu_b \int_\Omega d\Omega f_j^\Omega(r) \cdot \int_{\Omega'} d\Omega'(I + k_b^{-2}\nabla\nabla)G(r,r') \cdot \chi(r') f_l^{\Omega'}.$$

The material function $\chi(r') = 1$ if Ω' is a surface patch, and $\chi = (\epsilon/\epsilon_b - 1)$ if Ω' is a volume cell. It can be seen that the coefficient matrix arising from discretized coupled integral equations is nonsymmetric. Once the matrix equation (2) is solved by numerical matrix equation solvers, the expansion coefficients a^S and a^V can be used to calculate the scattered field and radar cross section. In antenna analysis problems the coefficients can be used to retrieve the antenna's input impedance and calculate the antenna's radiation pattern. In the following, we use A to denote the coefficient matrix in Equation (2), $x = [a^S, a^V]^T$, and $b = [U^S, U^V]^T$ for simplicity.

Using the addition theorem [3, 4], the matrix-vector product Ax can be written as

$$Ax = (A_D + A_N)x + V_f \Lambda V_s x, \tag{3}$$

where V_f, Λ, and V_s are sparse matrices. In fact, the dense matrix A can be structurally divided into three parts, A_D, A_N, and $A_F = V_f \Lambda V_s$. A_D is the block diagonal part of A, A_N is the block near-diagonal part of A, and A_F is the far part of A. Here the terms "near" and "far" refer to the distance between two groups of elements. We denote $(A_D + A_N)$ by B for simplicity. Figure 1(a) shows the sparse data structure of a partitioned dense matrix A from one of our examples, the P1D case in Table 1. The far part of A is scattered in the rest of the area of the Figure 1(a). The side bands shown in Figure 1(a) indicate the interactions between group-to-group in the fast multipole method (FMM) [4].

The FMM speeds up the matrix-vector product operations and reduces the computational complexity of a matrix-vector product from $O(N^2)$ to $O(N^{1.5})$, where N is the order of the matrix [4]. The computational complexity is further reduced to $O(N \log N)$ with the multilevel fast multipole algorithm (MLFMA) [3].

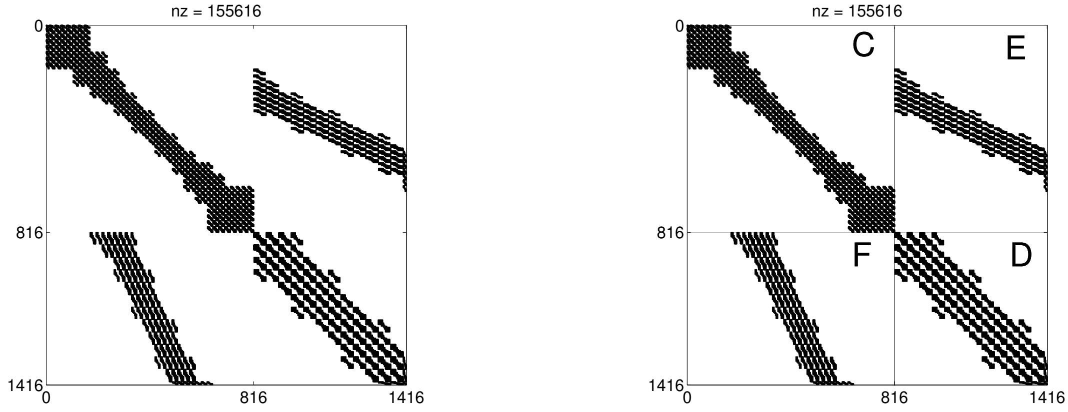
3 Two-level (TL) Preconditioning Techniques

We choose our iterative solver as one of the well-known Krylov methods BiCGSTAB [13]. It is a variant of BiCG that avoids the irregular convergence patterns, by locally minimizing the residual vector. Some experimental results with BiCGSTAB and other Krylov methods in this application can be found in [6].

The near part matrix from the material coated objects has explicit data block pattern (see Figure 1(a)). Due to this block pattern, we can construct a block LU factorization of the near part B. Figure 1(b) shows the four explicit data blocks in a near part matrix B, which consists of a conducting part C, a dielectric part D, and conducting-dielectric interface parts E and F. Based on these four components, we develop a domain specific TL preconditioning techniques for solving this type of matrix equations.

The matrix B can be written in a block LU factorization form as

$$B = \begin{pmatrix} C & E \\ F & D \end{pmatrix} = \begin{pmatrix} I & 0 \\ FC^{-1} & I \end{pmatrix} \begin{pmatrix} C & E \\ 0 & S \end{pmatrix},$$



(a) Near part matrix B

(b) Four block components of B

Figure 1: The sparse data pattern of the near part part of a dense matrix A.

where $S = D - FC^{-1}E$ is the Schur complement matrix. A two level preconditioner, based on this factorization, can be constructed as [2, 12, 14]

$$P = \begin{pmatrix} I & 0 \\ FM & I \end{pmatrix} \begin{pmatrix} M^{-1} & E \\ 0 & \tilde{S} \end{pmatrix}.$$

Here M is an approximate inverse of C and $\tilde{S} = D - FME$ is an approximate Schur complement matrix.

Instead of solving the Equation (1), we now solve the TL preconditioned linear system

$$P^{-1}Ax = P^{-1}b.$$

There are various ways to construct M. We can use a direct inverse or an approximate inverse of C, or an approximate LU factorization of C. Since the SAI and ILUT have been shown to be efficient for solving the dense complex linear systems from electromagnetic wave scattering problems in [5, 6], we use either the SAI or the ILUT to construct M. Other options are also tested in the numerical test section.

By a forward elimination and a backward substitution, we solve the following matrix equation in each preconditioning step:

$$\begin{pmatrix} I & 0 \\ FM & I \end{pmatrix} \begin{pmatrix} M^{-1} & E \\ 0 & \tilde{S} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

In this step, we need an efficient computing technique to find a solution for the smaller linear system:

$$\tilde{S}x_2 = b_2. \tag{4}$$

The main issues in the TL preconditioning techniques are how to construct M and how to solve the subsystem (4). To avoid the calculation of M and $\tilde{S}$ at each iteration step, we need to obtain them before the iteration starts.

The subsystem (4) can be solved by using one of the preconditioning schemes such as the ILUT preconditioners and the SAI preconditioners. We use these preconditioning techniques for a direct approximate computation without any iteration to solve the equation (4). In case that the subsystem is small enough we can use a direct exact method to get more accurate solutions. However, in real world problems, because of the expensive CPU time and high memory requirements, an exact direct method may not be a good choice.

4 Numerical Tests and Analysis

We calculate the radar cross section (RCS) to demonstrate the performance of our TL preconditioned BiCGSTAB for different conducting geometries with coating. The geometries considered include plate, microstrip antenna, and sphere (see Table 1). The mesh sizes for all the test structures are about one tenth of a wavelength.[1]

We examine the convergence behavior of the preconditioners based on the number of preconditioned iterations and some theoretical facts (such as the eigenvalue clustering and the condition numbers). We show the efficiency of the TL preconditioner by comparing the results with non-preconditioners.

Table 1: Information about the matrices used in the experiments (all length units are in λ_0, the wavelength in free-space).

case	level	unknowns	matrix	nonzeros	target size and description
P1D	4	1,416	A	2,005,056	$2.98824 \times 2 \times 0.1$, frequency=0.2GHz
			B	155,616	Dielectric plate over Conducting plate
P2D	3	1,683	A	2,832,489	$3 \times 0.266 \times 0.116 \times 0.001587$, frequency=1.157GHz
			B	773,181	Microstrip antenna
S2C	4	32,400	A	1,049,760,000	$5.2 \times 5.2 \times 5.2$, frequency=0.3GHz
			B	14,756,720	Large conducting sphere with dielectric coating

Here are the explanations of the notations shown in the tables with numerical data and in some figures.

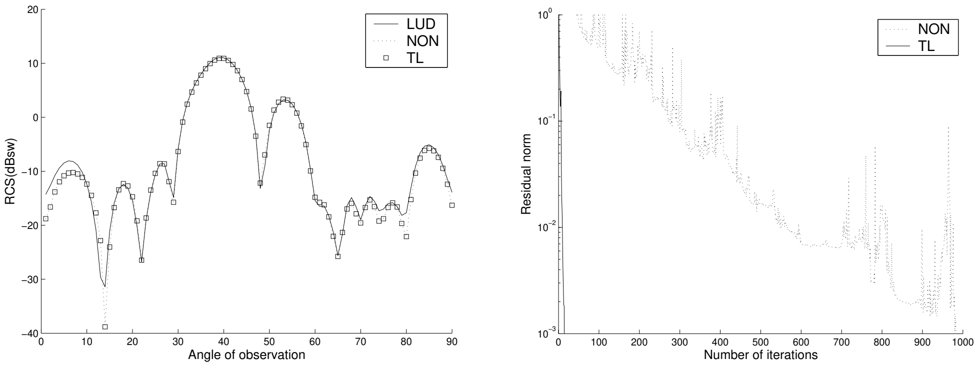
- LUD : the exact solution computed from the exact LU factorization method
- *prec* : the preconditioner used with the BiCGSTAB method
- NON : no preconditioner
- TL(parameters for constructing M ; parameters for solving the equation (4)) :
  - *direct*: the direct inverse of the matrix
  - $(\epsilon_1, \epsilon_2, \epsilon_3)$: the SAI preconditioner with three drop tolerance parameters [7]

[1] All cases are tested on one processor of an HP superdome supercluster at the University of Kentucky. The processor has 2 GB local memory and runs at 1.5 GHz. The code is written in Fortran 77 and is run in single precision. The iteration process is terminated when the 2-norm of the residual vector is reduced by 10^{-3}, or the number of iterations exceeds 2000.

– (τ, p) : the ILUT preconditioner with a dual dropping strategy [5, 9]

- level: the number of levels in the MLFMA
- *ratio* : the sparsity ratio of the number of nonzeros of a preconditioner with respect to the number of nonzeros of the near part matrix B
- *setup* : the setup CPU time in seconds for constructing a preconditioner
- it_{num} : the number of the (preconditioned) BiCGSTAB iterations
- it_{cpu} : the CPU time in seconds for the iteration phase
- $total_{cpu}$: the CPU time in seconds for both the setup and the iteration phase

Figure 2(a) shows that the solutions of the P1D case computed with the TL preconditioner and without a preconditioner are similar and approximate to the exact solution computed from the LU factorization. The solution graphs indicate the scattering pattern of the objects. Figure 2(b) shows the convergence history of the residual 2-norm of the BiCGSTAB method. TL is seen to converge sharply faster.



(a) Solution comparison

(b) Convergence history

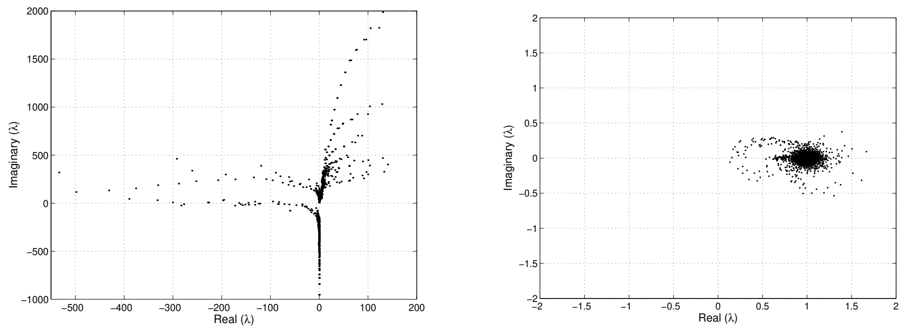
Figure 2: The P1D case: No-preconditioner (NON) and TL(0.01, 0.02, 0.03;10^{-3}, 5).

Table 2: Extreme eigenvalues and conditioning information of the matrices in the P1D case.

Matrix	$\lambda_{\text{smallest}}$	λ_{largest}	cond($Matrix$)	cond(X)
A	$-0.002 + 6.83i$	$(0.14 + 2.01i) * 10^3$	298.90	295.34
$(TL)^{-1}A$	$0.13 - 0.05i$	$1.66 + 0.09i$	29.92	11.40

The convergence behavior of the Krylov methods depends on the distribution of the eigenvalues and on the condition number of the coefficient matrix. Table 2 lists the eigenvalue and the condition number information related to the P1D case. The condition number of the eigenvector matrix of the original dense matrix A is relatively large. The smallest magnitude eigenvalues of the original matrix is near the origin. The original dense matrix makes the spectrum spread around the origin, see Figure 3(a). (We note that these subfigures are not drawn on the same scale.) This makes the preconditioned matrix more indefinite. These features cause slow convergence of the BiCGSTAB method on the original dense matrix.

The eigenvalues of the TL preconditioned matrix (see Figure 3(b)) are shifted to the right hand side of the origin and they are away from zero. They are also clustered around 1. The smallest eigenvalue in magnitude of the TL preconditioned matrix has the real part of 0.13, see Table 2. In addition, the condition number of the eigenvector matrix X is significantly decreased (the TL preconditioned matrix is close to normal). The condition number of the TL preconditioned matrix is relatively small. These features explain the good convergence behavior of the BiCGSTAB method on this TL preconditioned matrix.



(a) Eigenvalues of A (b) Eigenvalues of $(TL)^{-1}A$

Figure 3: Clustering of the eigenvalues in the P1D case.

Table 3: Numerical data of testing models.

case	*prec*	*ratio*	*setup*	it_{num}	it_{cpu}	$total_{cpu}$
P1D	NON			982	42.35	42.35
	TL(0.01,0.02,0.03;10^{-3},5)	0.47	1.74	13	0.62	1.09
P2D	NON			2000	26.90	26.90
	TL(0.005,0.02,0.03;10^{-3},5)	0.38	8.67	127	2.23	10.90
S2C	NON			2000	4828.90	4828.90
	TL(0.02,0.03,0.04;10^{-3},5)	0.25	507.61	143	358.15	865.76

Table 3 shows that the TL preconditioner reduces the total CPU time dramatically. TL also converges in a relatively small number of iterations. In particular, we point out the small sparsity ratios of the TL preconditioner.

We can see the strategies chosen for constructing M and for solving the equation (4) have a large effect on the required CPU time and the memory in Table 4. By using the direct method for the C part, we can get the least number of iterations but the largest sparsity ratio. By using the

Table 4: Numerical data of different strategies in TL for the P1D case.

prec	*ratio*	*setup*	it_{num}	it_{cpu}	$total_{cpu}$
TL(*direct*;*direct*)	6.85	0.90	7	0.40	1.30
TL(*direct*;10^{-3},5)	4.59	0.75	7	0.37	1.12
TL(*direct*;0.03,0.04,0.05)	4.57	0.79	7	0.39	1.18
TL(10^{-3},50;*direct*)	1.14	0.99	10	0.45	1.45
TL(10^{-3},50;10^{-3},5)	1.07	0.93	11	0.51	1.58
TL(10^{-3},50;0.03,0.04,0.05)	1.05	0.85	11	0.51	1.36
TL(0.01,0.02,0.03;*direct*)	2.74	1.90	13	0.65	2.55
TL(0.01,0.02,0.03;10^{-3},5)	0.47	1.74	13	0.62	2.36
TL(0.01,0.02,0.03;0.03,0.04,0.05)	0.45	1.75	13	0.58	2.33

ILUT and the SAI preconditioners for constructing M, we get a good sparsity ratio as well as a small number of iterations. For solving the equation (4), we only need small fill-in. It is also seen that the accuracy of C^{-1} have more effect on the overall convergence rate than that of S^{-1}.

The TL preconditioner, with suitable strategies for computing C^{-1} and S^{-1}, does not need a large amount of storage space. Thus, the memory and floating point operation complexity of the MLFMA in each iteration is maintained. All cases show that the BiCGSTAB method with the TL preconditioner converges very fast with a small number of iterations, compared to that without preconditioners. Since the number of iterations of the BiCGSTAB is decreased significantly by using TL, the total simulation time is reduced substantially.

5 Conclusions

We described a two-level (TL) preconditioning strategy based on the data pattern of the near part matrix in the MLFMA. In all cases, the memory cost of the TL preconditioning is much less than the cost of storing the near part matrix. With well chosen parameters, the computational (construction) cost of the TL preconditioner is even smaller than the preconditioned iterative solution cost in some range of the sparsity ratio.

Numerical results and computational eigenvalue analysis show that the TL preconditioner constructed from the near part matrix improves the computational efficiency in terms of both the memory and the CPU time. The results show that the BiCGSTAB method with the TL preconditioner is robust for solving three dimensional model cases from electromagnetic scattering simulations.

Along with previous two preconditioning techniques ILUT and SAI, the TL preconditioning proposed in this paper is proven to be effective for simulating composite conducting and dielectric objects from electromagnetic scattering problems.

References

[1] O. Axelsson and V. Eijkhout. The nested recursive two-level factorization method for nine-point difference matrices. *SIAM J. Sci. Stat. Comput.*, 12(6):1373–1400, 1991.

[2] L. M. Carvalho, L. Giraud, and P. Le Tallec. Algebraic two-level preconditioners for the Schur complement method. *SIAM J. Sci. Comput.*, 22(6):1987–2005, 2001.

[3] W. C. Chew, J. M. Jin, E. Midielssen, and J. M. Song. *Fast and Efficient Algorithms in Computational Electromagnetics.* Artech House, Boston, MA, 2001.

[4] R. Coifman, V. Rokhlin, and S. Wandzura. The fast multipole method for the wave equation: a pedestrian prescription. *IEEE Antennas Propag. Mag.*, 35(3):7–12, 1993.

[5] J. Lee, J. Zhang, and C. C. Lu. Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems. *J. Comput. Phys.*, 185(1):158–175, 2003.

[6] J. Lee, J. Zhang, and C. C. Lu. Performance of preconditioned Krylov iterative methods for solving hybrid integral equations in electromagnetics. *J. Appl. Comput. Electromag. Society*, 18(4):54–61, 2003.

[7] J. Lee, J. Zhang, and C. C. Lu. Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics. *IEEE Trans. Antennas Propag.*, to appear.

[8] C. C. Lu and W. C. Chew. A coupled surface-volume integral equation approach for the calculation of electromagnetic scattering from composite metallic and material targets. *IEEE Trans. Antennas Propag.*, 48(12):1866–1868, 2000.

[9] Y. Saad. ILUT: a dual threshold incomplete LU preconditioner. *Numer. Linear Algebra Appl.*, 1(4):387–402, 1994.

[10] Y. Saad and J. Zhang. Diagonal threshold techniques in robust multi-level ILU preconditioners for general sparse linear systems. *Numer. Linear Algebra Appl.*, 6(4):257–280, 1999.

[11] T. K. Shark, S. M. Rao, and A. R. Djordievic. Electromagnetic scattering and radiation from finite microstrip structures. *IEEE Trans. Micro. Opt. Tech.*, 38(11):1568–1575, 1990.

[12] C. Shen and J. Zhang. Parallel two level block ILU preconditioning techniques for solving large sparse linear systems. *Paral. Comput.*, 28(10):1451–1475, 2002.

[13] H. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13:631–644, 1992.

[14] J. Zhang. On preconditioning Schur complement and Schur complement preconditioning. *Electronic Trans. Numer. Anal.*, 10:115–130, 2000.