Performance of Preconditioned Krylov Iterative Methods for Solving Hybrid Integral Equations in Electromagnetics *

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

In solving systems of linear equations arising from practical engineering models such as the electromagnetic wave scattering problems, it is critical to choose a fast and robust solver. Due to the large scale of those problems, preconditioned Krylov iterative methods are most suitable. The Krylov iterative methods require the computation of matrix-vector product operations at each iteration, which account for the major computational cost of this class of methods. We use the multilevel fast multipole algorithm (MLFMA) to reduce the computational complexity of the matrix-vector product operations. We conduct an experimental study on the behavior of three Krylov iterative methods, BiCG, BiCGSTAB, and TFQMR, and of two preconditioners, the ILUT preconditioner, and the sparse approximate inverse (SAI) preconditioner. The preconditioners are constructed by using the near part matrix numerically generated in the MLFMA. Our experimental results indicate that a well chosen preconditioned Krylov iterative method maintains the computational complexity of the MLFMA and effectively reduces the overall simulation time.

1 Introduction

The hybrid integral equation approach combines the volume integral equation and the surface integral equation to model the scattering and radiation by mixed dielectric and conducting structures [12]. For example, when a radome is applied to an antenna, the combined system consists of both dielectrics and conductors. Hence, the hybrid 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 hybrid surface-volume integral equations for electromagnetic scattering problems can be formally written as follows,

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

where E^{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, and $L_\Omega, (\Omega = S, V)$, is an integral operator that maps the source J_Ω to electric field $E(r)$ and it is defined as

$$L_\Omega(r,r')\cdot J_\Omega(r') =$$

*This research work was supported in part by NSF under grants CCR-9902022, CCR-9988165, CCR-0092532, ACI-0202934, and ECS-0093692, by DOE under grant DE-FG02-02ER45961, by ONR under grant N00014-00-1-0605, by RIST (Japan), and by the University of Kentucky Research Committee.

[†] leejh@engr.uky.edu, http://www.csr.uky.edu/∼leejh.

[‡] jzhang@cs.uky.edu, http://www.cs.uky.edu/∼jzhang.

[§] cclu@engr.uky.edu, http://www.engr.uky.edu/∼cclu.

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

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 $J_V = i\omega(\epsilon_b - \epsilon)E$.

We follow the general steps of the method of moments (MoM) [17] to discretize the hybrid surface-volume integral equations, and solve the resultant matrix equation by a multilevel fast multipole algorithm (MLFMA), which is a multilevel implementation of the fast multipole method (FMM). A matrix equation is given as the form

$$\begin{bmatrix} Z^{SS} & Z^{SV} \\ Z^{VS} & Z^{VV} \end{bmatrix} \cdot \begin{bmatrix} a^S \\ a^V \end{bmatrix} = \begin{bmatrix} U^S \\ U^V \end{bmatrix}, \qquad (1)$$

where a^S and a^V stand for the vectors of the expansion coefficients for the surface current and the volume function, respectively [3, 12], 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 hybrid integral equations is nonsymmetric. Once the matrix equation (1) 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 (1), $x = [a^S, a^V]^T$, and $b = [U^S, U^V]^T$ for simplicity.

The basic idea of the FMM is to convert the interaction of element-to-element to the interaction of group-to-group. Using the addition theorem for the free-space scalar Green's function, the matrix-vector product Ax can be written as

$$Ax = (A_D + A_N)x + V_f \Lambda V_s x,$$

where V_f, Λ, and V_s are sparse matrices. We denote $B_N = (A_D + A_N)$ for simplicity. 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]. As the level of the MLFMA decreases, we find that the number of nonzeros in the near part matrix B_N increases significantly. The accuracy of the computed solution is strongly related to the number of levels of the MLFMA [11]. That is, as the number of the MLFMA levels decreases, the computed solution is close to the exact solution, but the near part matrix B_N becomes denser. It is well-known that the MLFMA is an approximation method.

A matrix problem involving N unknowns may be solved in $CN^{\text{iter}}N^{Ax}$ floating point operations, where C is a constant depending on the implementation of a particular iterative method [2, 4, 5, 15], and N^{Ax} is the floating point operations needed for each matrix-vector multiplication. For many realistic problems, N^{iter} depends on both the iterative solver and the target properties (shape and material). For example, a problem with an open-ended cavity needs much more iterations than that with a solid conducting box of the same size. Since N^{iter} is a proportional factor in the CPU counter, to further reduce the total CPU time, it is necessary to reduce the number of iterations of the iterative solvers. Hence preconditioning techniques, which may speed up the convergence rate of the Krylov iterative methods, are needed in this application.

We iteratively solve the linear system of the form $Ax = b$, where the coefficient matrix A is a large scale, dense, and complex valued matrix for electrically large targets. The biconjugate gradient (BiCG) method [9], the biconjugate gradient stabilized method (BiCGSTAB) [16], and the transpose-free variant of the quasi-minimum residual method (TFQMR) [6] are some of the well-known Krylov iterative methods which are applicable to non-Hermitian matrices. In our experimental tests, we use these three methods as the iterative solver, coupled with different preconditioning strategies to solve a few study cases of representative electromagnetic scattering problems. We propose to use an incomplete lower-upper

(ILU) triangular factorization with a dual dropping strategy [11, 14] and a sparse approximate inverse (SAI) technique [8, 10, 19] to construct a preconditioner from the near part matrix (B_N) in the MLFMA implementation. We mainly compare the performance difference of different Krylov iterative methods combined with different preconditioners.

2 Preconditioned Krylov Iterative Methods

Krylov iterative methods are considered to be the most effective iterative solution methods currently available [1, 7, 15]. The complexity of these methods is on the order of $O(N^{\text{iter}}N^2)$ if the convergence is achieved in N^{iter} iterations. The Krylov iterative methods such as BiCG require the computation of some matrix-vector product operations at each iteration, which account for the major computational cost of this class of methods.

In our experiments, we observe that the convergence behavior of BiCG is irregular. Few theoretical results are known about the convergence of BiCG [2]. BiCG requires two matrix-vector products (one with A and one with A^H, the complex conjugate transpose of A) at each step of iteration. BiCGSTAB is one of variant of BiCG to avoid the irregular convergence patterns of BiCG. A residual vector is minimized locally and it has substantially smoother convergence behavior. Each iteration step of BiCGSTAB also requires two matrix-vector products (both with A). TFQMR is also chosen to get a smoother convergence behavior. TFQMR requires two matrix-vector products (both with A) at each iteration. All of three solvers designed to solve non-Hermitian linear systems. Every related algorithm which is implemented in our program originally comes from [2, 6, 15].

The convergence behavior of the Krylov methods depends on the distribution of the eigenvalues and on the condition number of the coefficient matrix. By applying a good preconditioner we may achieve better spectrum and smaller condition number compared to those of the original coefficient matrix. Therefore, the convergence behavior of the Krylov iterative method can be improved by a good preconditioner. In our previous papers [10, 11] we show that the ILUT and the SAI preconditioned BiCG method has tight spectrum around 1 and small condition numbers with little extra CPU time to construct the preconditioners and small extra memory to store the preconditioner matrix. For some problems, iterative methods without a preconditioner might not converge. Hence, preconditioning techniques should be used in practical applications with the Krylov iterative method to reduce the number of iterations. Most preconditioning techniques, such as the ILU(0), rely on a fixed sparsity pattern, obtained from the sparsified coefficient matrix by dropping small magnitude entries. Some SAI techniques need access to the full coefficient matrix (to construct a sparsified matrix), which is not available in the FMM.

The purpose of the preconditioning is to make the preconditioned matrix MA as close to the identity matrix I as possible. To this end, we try to construct a matrix M that approximates the matrix A^{-1}. It is difficult to make the matrix M sparse, since in most cases the inverse of a matrix A is dense even if A is sparse. We evaluate three different Krylov iterative methods with two preconditioners, the ILU preconditioner with a dual dropping strategy (ILUT) (with a fill-in parameter p and a drop tolerance τ) [11, 14] and the SAI preconditioner [10, 19], using the non-preconditioning case as comparison. In the MLFMA implementation, the global matrix A is not numerically available, but the near part matrix B_N is. We thus construct the ILUT and the SAI preconditioners with respect to the matrix B_N. The total storage of the ILUT preconditioner is bounded by $2pN$. Here the parameter τ controls the computational cost, and the parameter p controls the memory cost. We use a static sparsity pattern strategy for constructing the SAI preconditioner M. For SAI, we construct a sparsified matrix $\tilde{B}_N$ from the near part matrix B_N. Here the matrix $\tilde{B}_N$ is obtained from B_N by removing some small magnitude entries of B_N with respect to a threshold parameter ϵ_1. The computational procedure is given in [10], in which ϵ_1, ϵ_2, and ϵ_3 are three user provided threshold drop tolerance parameters chosen by a heuristic process. By judiciously choosing those parameters, we are able to construct both preconditioners that are effective and do not use much memory space. We use the notation of ILUT(τ, p) and SAI(ϵ_1, ϵ_2, ϵ_3) for simplicity, see [10, 11].

3 Numerical Experiments

The major cost of the preconditioned iterative solvers is the matrix-vector product with both the coefficient matrix and the preconditioner [18]. There are two matrix-vector products at each iteration of BiCG, BiCGSTAB, and TFQMR. A number of numerical examples are presented to demonstrate the performance of the preconditioned Krylov iterative methods.[1] Note that the examples are different from the ones in our previous two papers [10, 11] with different incident angles.

We examine the convergence behavior based on the number of preconditioned iterations. We compare three different Krylov iterative methods (BiCG, BiCGSTAB, and TFQMR) with the ILUT preconditioner, the SAI preconditioner, and no preconditioner one and another. The efficiency of the ILU preconditioner with a dual dropping strategy (ILUT) and the SAI preconditioner (with BiCG) is reported in [10, 11]. Since both ILUT and SAI preconditioners have been shown to be efficient for solving the dense complex linear systems from electromagnetic wave scattering problems, we mainly compare the performance difference of three Krylov iterative methods, BiCG, BiCGSTAB, and TFQMR.

We calculate the radar cross section (RCS) to demonstrate the performance of our preconditioned Krylov iterative solvers for different conducting geometries with and without coating. The geometries considered include plates, antenna arrays, and cavities (see Table 1). The mesh sizes for all the test structures are about one tenth of a wavelength.

The test problems are described in Table 1 and some numerical results are listed in Table 2. In Tables 1 and 2, "level" indicates the number of levels in the multilevel fast multipole algorithm, "setup" the setup CPU time in seconds for constructing a preconditioner, "#it" the number of the (preconditioned) Krylov iterations, and "total" the CPU time in seconds for both the setup and the iteration phase. The notations used for "case" are 0=BiCG, 1=BiCGSTAB, 2=TFQMR, N=NONE, I=ILUT, and S=SAI. Thus, "P3C2S" means that the P3C case is solved by using TFQMR with the SAI preconditioner.

Due to space limit, we report one set of parameters for the ILUT and the SAI preconditioners. In our experiments, we use ILUT(10^{-3}, 30) for the P1C and P3C cases, ILUT(10^{-3}, 130) for the C1C case, and SAI$(0.03, 0.04, 0.05)$ for all test cases.

According to the total CPU time for solving a problem, BiCGSTAB with the SAI and the ILUT preconditioners seems to converge very fast for most cases (see Table 2). Without a preconditioner, the results are various depending on cases. For the problem which has a small number of unknowns (say, less than 1000), the iterative solver with the SAI preconditioner takes more time than with the ILUT preconditioner, mainly due to the higher cost in constructing the SAI preconditioner.

Figures 1 – 3 show the number of iterations of (a) BiCG with three different preconditioners, (b) BiCGSTAB with three different preconditioners, (c) TFQMR with three different preconditioners, (d) NONE-preconditioned three different solvers, (e) ILUT preconditioned three different solvers, and (f) SAI preconditioned three different solvers.

In the P1C case, we observe that all three iterative solvers with the SAI preconditioner converge faster than the other two, and without a preconditioner are the slowest. Without a preconditioner, BiCG is the fastest one and TFQMR is the slowest. With the ILUT preconditioner, BiCGSTAB is the fastest one and BiCG is the slowest. With the SAI preconditioner, BiCGSTAB is the fastest one and the BiCG is the slowest.

In the C1C case, we see that BiCG and BiCGSTAB iterative solvers with the SAI preconditioner converge faster than the other two, and without a preconditioner are the slowest. TFQMR with the ILUT preconditioner is the fastest one and that without preconditioner are the slowest. Without a preconditioner, TFQMR is the fastest one and BiCG is the slowest. With

[1] All cases are tested on one processor of an HP Superdome cluster at the University of Kentucky. The processor has 2 GB local memory and runs at 750 MHz. The code is written in Fortran 77 and is run in single precision. The computation terminates when the 2-norm residual is reduced by 10^{-3}, or the number of iterations exceeds $2,000$.

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
P1C	4	1,416	A	2,005,056	Dielectric plate over conducting plate
			A_D	66,384	2.98824 × 2 × 0.1
			B_N	155,616	frequency = 200MHz
C1C	5	20,176	A	407,070,976	Cavity
			A_D	1,565,032	3 × 1 × 0.5
			B_N	3,728,842	frequency = 1GHz
P3C	7	100,800	A	10,160,640,000	Patch array (30 × 30)
			A_D	3,571,808	Array size: 22.25 × 22.25
			B_N	7,211,632	frequency = 300MHz

Table 2: Numerical results with different test cases.

case	setup	#it	total	case	setup	#it	total	case	setup	#it	total
P1C0N	–	973	112.3	C1C0N	–	812	1431.1	P3C0N	–	347	2499.9
P1C1N		1044	118.4	C1C1N		751	1310.3	P3C1N		201	1450.9
P1C2N		2000	225.9	C1C2N		509	893.2	P3C2N		216	1529.1
P1C0I	0.3	40	5.4	C1C0I	67.5	367	1028.4	P3C0I	111.5	37	414.5
P1C1I		24	3.3	C1C1I		112	327.6	P3C1I		12	205.7
P1C2I		30	4.3	C1C2I		179	577.5	P3C2I		20	275.9
P1C0S	8.0	29	11.5	C1C0S	110.9	322	714.9	P3C0S	64.2	41	376.0
P1C1S		15	9.7	C1C1S		120	326.2	P3C1S		17	189.6
P1C2S		17	10.1	C1C2S		199	476.4	P3C2S		26	254.7

the ILUT preconditioner, BiCGSTAB is the fastest one and BiCG is the slowest. With the SAI preconditioner, BiCGSTAB is the fastest one and the BiCG is the slowest.

In the P3C case, we find that all three iterative solvers with the ILUT preconditioner converge faster than the other two, and without a preconditioner are the slowest. With all three different preconditioners, BiCGSTAB is the fastest one and BiCG is the slowest.

Although a general iterative solver for solving some categories of problems efficiently might not exist [13], according to the results from our numerical experiments, we can see that BiCGSTAB with the SAI or the ILUT preconditioners is robust and converges very fast for solving three dimensional model cases from electromagnetic scattering simulations. In all cases, these Krylov iterative methods without a preconditioner are much less efficient.

4 Conclusions

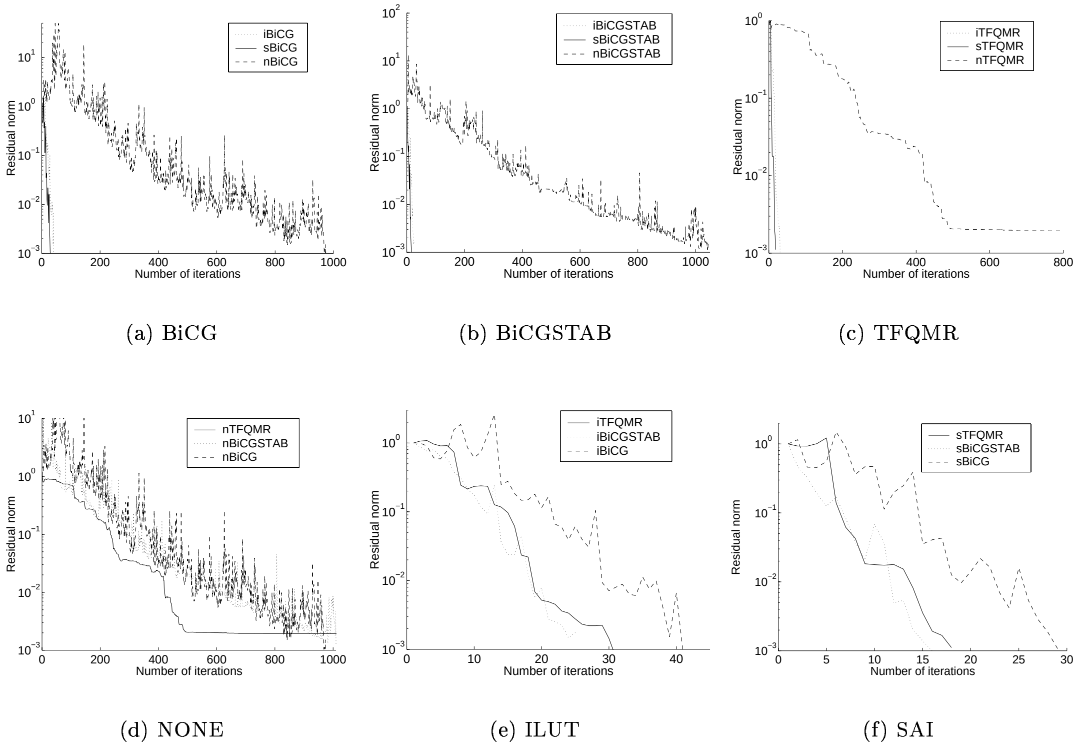
We conducted a few numerical tests to show that the Krylov iterative methods coupled with the ILUT and the SAI preconditioners are efficient to solve the problems arising from electromagnetic scattering.

Our numerical results indicate that, solving the large non-Hermitian dense linear system arising from the electromagnetic scattering by using the BiCGSTAB method with the ILUT preconditioner and the SAI preconditioner achieves faster convergence in most cases. The ILUT and the SAI preconditioned Krylov iterative solvers (BiCG, BiCGSTAB, and TFQMR) maintain the computational complexity of the MLFMA, and consequently reduces the total CPU time. Our experimental experience may help researchers and engineers choose suitable robust solvers in practical large scale electromagnetic simulations.

References

[1] O. Axelsson. *Iterative Solution Methods.* Cambridge Univ. Press, Cambridge, 1994.

[2] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst.



(a) BiCG (b) BiCGSTAB (c) TFQMR

(d) NONE (e) ILUT (f) SAI

Figure 1: Convergence history for the P1C case.

Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition. SIAM, Philadelphia, PA, 1994.

[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 Propagat. Mag.*, 35(3):7–12, 1993.

[5] E. Darve. The fast multipole method: numerical implementation. *J. Comput. Phys.*, 160:195–240, 2000.

[6] R. W. Freund. A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comput.*, 14:470–482, 1993.

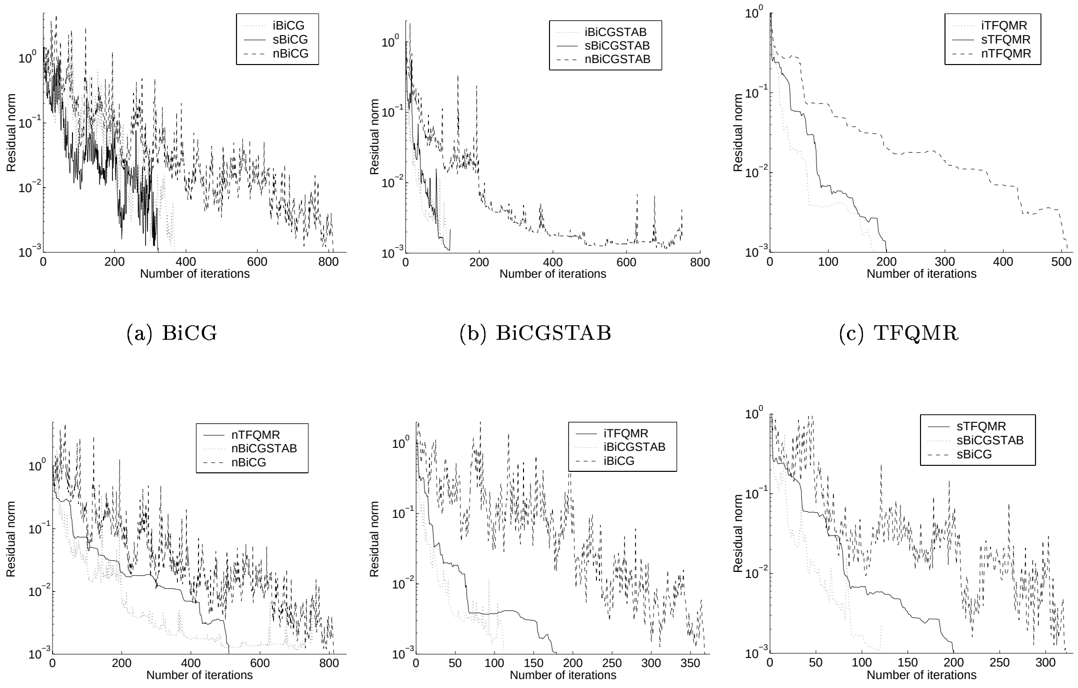
[7] A. Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, Philadelphia, 1997.

[8] L. Y. Kolotilina. Explicit preconditioning of systems of linear algebraic equations with dense matrices. *J. Sov. Math.*, 43:2566–2573, 1988.

[9] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Research National Bureau Standards*, 49:33–53, 1952.

[10] J. Lee, J. Zhang, and C. C. Lu. Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics. Technical Report No. 363-02, Department of Computer Science University of Kentucky, Lexington, KY, 2002.

[11] J. Lee, J. Zhang, and C. C. Lu. Incomplete LU preconditioning for large scale dense



(a) BiCG (b) BiCGSTAB (c) TFQMR

(d) NONE (e) ILUT (f) SAI

Figure 2: Convergence history for the C1C case.

complex linear systems from electromagnetic wave scattering problems. *J. Comput. Phys.*, 185(1):158–175, 2003.

[12] 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.

[13] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM Matrix Anal. Appl.*, 13(3):778–795, 1992.

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

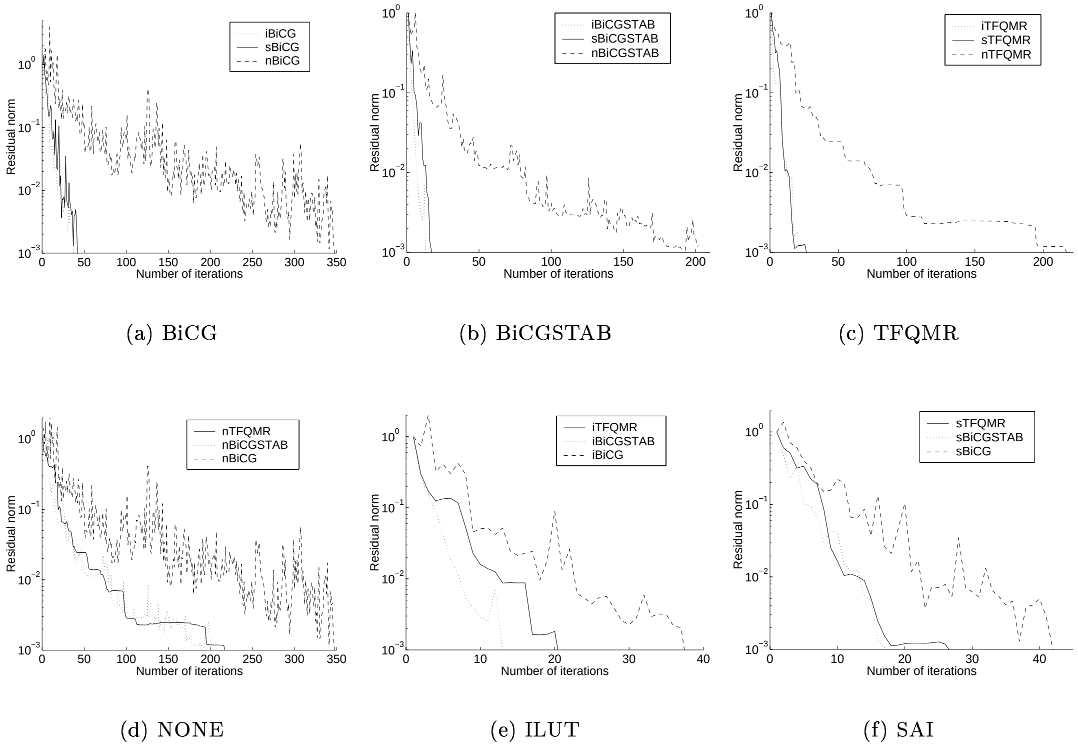
[15] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing, New York, NY, 1996.

[16] 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.

[17] Y. V. Vorobyev. *Method of Moments in Applied Mathematics.* Gordon And Breach Science Publishers, New York, 1965.

[18] J. Zhang. Preconditioned Krylov subspace methods for solving nonsymmetric matrices from CFD applications. *Comput. Methods Appl. Mech. Engrg.*, 189(3):825–840, 2000.

[19] J. Zhang. Sparse approximate inverse and multilevel block ILU preconditioning techniques for general sparse matrices. *Appl. Numer. Math.*, 35(1):67–86, 2000.



(a) BiCG (b) BiCGSTAB (c) TFQMR

(d) NONE (e) ILUT (f) SAI

Figure 3: Convergence history for the P3C case.

Short Biography

Jeonghwa Lee received an M.S. in Mathematics from Chonnam National University, South Korea. She is currently a Ph.D. student of Computer Science at the University of Kentucky. Her research interests include large scale parallel and scientific computing, iterative and preconditioning techniques for large scale matrix computation, and computational electromagnetics.

Jun Zhang received a Ph.D. from The George Washington University in 1997. He is an Associate Professor of Computer Science and Director of the Laboratory for High Performance Scientific Computing and Computer Simulation at the University of Kentucky. His research interests include large scale parallel and scientific computing, numerical simulation, iterative and preconditioning techniques for large scale matrix computation. Dr. Zhang is associate editor and on the editorial boards of three international journals in computer simulation and computational mathematics, and is on the program committees of a few international conferences. His research work is currently funded by the U.S. National Science Foundation and the Department of Energy. He is recipient of National Science Foundation CAREER Award and several other awards.

Cai-Cheng Lu got his Ph.D. degree from University of Illinois at Urbana Champaign in 1995, and now he is an assistant professor in the Department of Electrical and Computer Engineering at the University of Kentucky. Priori to join University of Kentucky, he was with Demaco, Inc. (now SAIC) where he worked on a number of new features for the Xpatch code. His research interests are in wave scattering, microwave circuit simulation, and antenna analysis. He is especially experienced in fast algorithms for computational electromagnetics and is one of the authors for a CEM code FISC. He is a recipient of the 2000 Yong Investigator Award from the Office of Naval Research, and a CAREER Award from the National Science Foundation. Dr. Lu is a senior member of IEEE.