Iterative Block Near-Field Preconditioners for Surface Integral-Equation Formulations of Dielectric Problems

Tahir Malas[1,2] and Levent Gürel[1,2]
[1]Department of Electrical and Electronics Engineering
[2]Computational Electromagnetics Research Center (BiLCEM)
Bilkent University, TR-06800, Bilkent, Ankara, Turkey
E-mail: tmalas@ee.bilkent.edu.tr, lgurel@bilkent.edu.tr

Abstract — We improve the convergence behaviour of the two commonly used integral-equation formulations of dielectric problems, namely, the combined tangential formulation and the electric and magnetic current combined-field integral equation, using iterative block preconditioners, which are obtained from approximate block solutions of the near-field matrix system. The effectiveness of the proposed preconditioners is demonstrated on large dielectric problems.

1. INTRODUCTION

Many real-life problems in computational electromagnetics necessitate the use of integral-equation formulations of dielectric problems, such as simulations of photonic crystals [1], development of effective lenses [2], and optical analysis of blood for blood-related diseases [3]. Some recently proposed formulations for dielectrics that are suitable for iterative solutions include the combined tangential formulation (CTF) and the electric and magnetic current combined-field integral equation (JMCFIE). These two formulations are of utmost interest since CTF yields more accurate scattering results and JMCFIE yields better-conditioned systems than other formulations [9].

Integral-equation formulations of dielectric problems are obtained by simultaneous discretization of the electric and magnetic surface currents and result in block-partitioned linear systems in the form

$$\begin{bmatrix} \overline{\boldsymbol{Z}}_{11} & \overline{\boldsymbol{Z}}_{12} \\ \overline{\boldsymbol{Z}}_{21} & \overline{\boldsymbol{Z}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{a}_J \\ \boldsymbol{a}_M \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} \tag{1}$$

or

$$\overline{\boldsymbol{Z}} \cdot \boldsymbol{a} = \boldsymbol{v}, \tag{2}$$

where $\boldsymbol{a}_J$ and $\boldsymbol{a}_M$ are the coefficient vectors of the basis functions expanding the electric and magnetic currents, respectively, and $\boldsymbol{v}_{1,2}$ represent excitation vectors obtained by testing the incident fields. Iterative solutions of the resulting dense systems become feasible with the multilevel fast multipole algorithm (MLFMA) [4], which performs a matrix-vector multiplication of each block in (1) in $\mathcal{O}(n \log n)$ complexity for a block of size n. However, iterative solutions of such block-partitioned matrices often suffer from slow convergence, due to highly indefinite nature of resulting matrices. In Fig. 1, we depict the spectra of the $\overline{\boldsymbol{Z}}$ matrix and its dense matrix blocks in the complex plane, for a sphere problem with 0.5λ radius and a dielectric constant of 4.0. The matrices of both formulations are highly indefinite (the eigenvalues are distributed in the left half-plane), hence it is difficult to achieve convergence without preconditioning [10]. For CTF, there are many eigenvalues close to the origin, which makes the convergence of CTF more difficult than JMCFIE [5]. Hence, effective preconditioners should be applied to these systems in order to increase robustness and efficiency.

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 105E065, 105E172, and 107E136, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM.

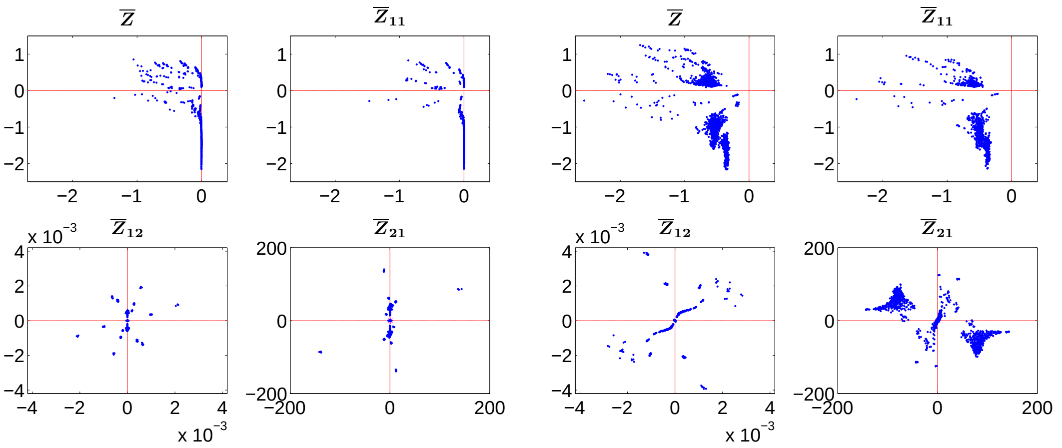

Fig. 1. The spectra of the system matrix and its blocks for (a) CTF and (b) JMCFIE on a sphere problem with 1,860 unknowns. The inside of the sphere has a dielectric constant of 4.0. Note that $\overline{\boldsymbol{Z}}_{11} = \overline{\boldsymbol{Z}}_{22}$ for these two formulations.

2. BLOCK PRECONDITIONERS FOR DIELECTRIC FORMULATIONS

Preconditioning refers to finding a suitable matrix $\overline{\boldsymbol{M}}$ that approximates the system matrix $\overline{\boldsymbol{Z}}$, for which the solution of the system

$$\overline{\boldsymbol{M}} \cdot \boldsymbol{u} = \boldsymbol{b} \tag{3}$$

is cheaper compared to the solution of the original system (2). Given the input vector $\boldsymbol{b}$, the solution vector $\boldsymbol{u}$ is required in each step of the iterative solver. In this way, instead of the original system, one of the two preconditioned systems

$$\overline{\boldsymbol{M}}^{-1} \cdot \overline{\boldsymbol{Z}} \cdot \boldsymbol{a} = \overline{\boldsymbol{M}}^{-1} \cdot \boldsymbol{v} \tag{4}$$

or

$$\left(\overline{\boldsymbol{Z}} \cdot \overline{\boldsymbol{M}}^{-1}\right) \cdot \left(\overline{\boldsymbol{M}} \cdot \boldsymbol{a}\right) = \boldsymbol{v} \tag{5}$$

can be solved, for left or right preconditioning, respectively. The better the preconditioner $\overline{\boldsymbol{M}}$ approximates the matrix $\overline{\boldsymbol{Z}}$, the faster the convergence is. However, better approximation comes with higher construction and application costs. Hence, a balance should be struck between the approximation level and the efficiency, so that the matrix system can be solved rapidly and reliably.

MLFMA decomposes the system matrix into its far-field and near-field components as

$$\begin{bmatrix} \overline{\boldsymbol{Z}}_{11} & \overline{\boldsymbol{Z}}_{12} \\ \overline{\boldsymbol{Z}}_{21} & \overline{\boldsymbol{Z}}_{22} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{Z}}_{11}^{NF} & \overline{\boldsymbol{Z}}_{12}^{NF} \\ \overline{\boldsymbol{Z}}_{21}^{NF} & \overline{\boldsymbol{Z}}_{22}^{NF} \end{bmatrix} + \begin{bmatrix} \overline{\boldsymbol{Z}}_{11}^{FF} & \overline{\boldsymbol{Z}}_{12}^{FF} \\ \overline{\boldsymbol{Z}}_{21}^{FF} & \overline{\boldsymbol{Z}}_{22}^{FF} \end{bmatrix} \tag{6}$$

or

$$\overline{\boldsymbol{Z}} = \overline{\boldsymbol{Z}}^{NF} + \overline{\boldsymbol{Z}}^{FF}, \tag{7}$$

where the far-field matrix $\overline{\boldsymbol{Z}}^{FF}$ is not stored in the memory and the application of $\overline{\boldsymbol{Z}}^{FF}$ to a vector is computed on the fly. Hence, we use the near-field matrix for preconditioning, *i.e.*, $\overline{\boldsymbol{M}} = \overline{\boldsymbol{Z}}^{NF}$. For the solution of the system (3), the Schur complement reduction method is used [10]. This method reduces the solution of the block-partitioned system

$$\begin{bmatrix} \overline{\boldsymbol{Z}}_{11}^{NF} & \overline{\boldsymbol{Z}}_{12}^{NF} \\ \overline{\boldsymbol{Z}}_{21}^{NF} & \overline{\boldsymbol{Z}}_{22}^{NF} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix} \tag{8}$$

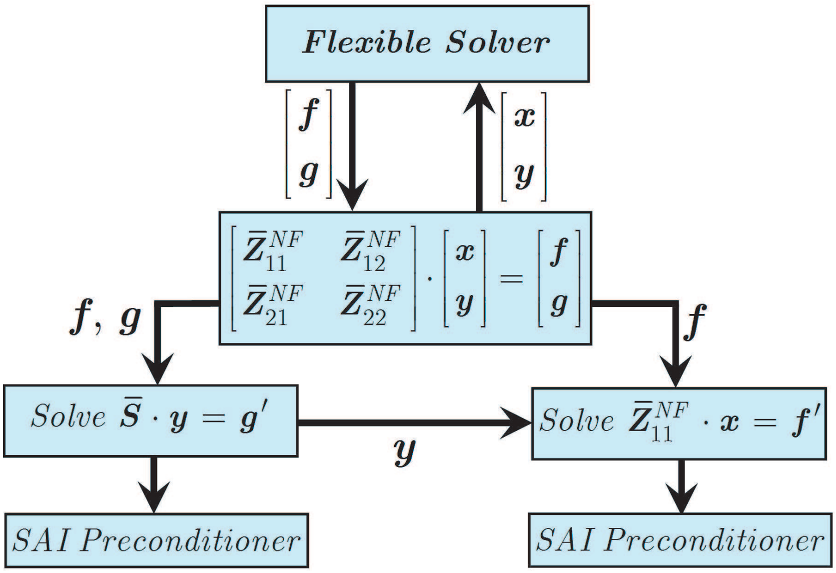

Fig. 2. Illustration of the application of IBP in a step of an iterative solver.

into the solutions of

$$\overline{Z}_{11}^{NF} \cdot \boldsymbol{x} = \boldsymbol{f}' \tag{9}$$

and

$$\overline{S} \cdot \boldsymbol{y} = \boldsymbol{g}', \tag{10}$$

where

$$\boldsymbol{f}' = \boldsymbol{f} - \overline{Z}_{12}^{NF} \cdot \boldsymbol{y}, \tag{11}$$

$$\overline{S} = \overline{Z}_{22}^{NF} - \overline{Z}_{21}^{NF} \cdot \left(\overline{Z}_{11}^{NF}\right)^{-1} \cdot \overline{Z}_{12}^{NF} \tag{12}$$

is the Schur complement matrix, and

$$\boldsymbol{g}' = \boldsymbol{g} - \overline{Z}_{21}^{NF} \cdot \left(\overline{Z}_{11}^{NF}\right)^{-1} \cdot \boldsymbol{f}. \tag{13}$$

Since the inversion of the sparse matrix $\overline{Z}_{11}^{NF}$ is unfeasible, we approximate the inverse of $\overline{Z}_{11}^{NF}$ with a sparse approximate inverse (SAI) of $\overline{Z}_{11}^{NF}$ [8] in Eqs. (12) and (13). Then, solutions of (9) and (10) are approximated by a few iterations of the generalized minimal residual method (GMRES) solver. We call these preconditioning solutions inner solutions and the preconditioning scheme the iterative block preconditioner (IBP). Note that we do not need to compute and store the Schur complement matrix $\overline{S}$; we only have to provide the application of $\overline{S}$ to a vector in each step of the inner iterative solver of (10).

However, there is no guarantee that the solutions of these systems will be acquired fast enough. An efficient way to accelerate the convergence of these solutions is to use the available SAI of $\overline{Z}_{11}^{NF}$ as a preconditioner for (9) and (10). We note that for CTF and JMCFIE, $\overline{Z}_{11}^{NF} = \overline{Z}_{22}^{NF}$, hence SAI of $\overline{Z}_{11}^{NF}$ serves as a useful preconditioner for (10), assuming that $\overline{Z}_{22}^{NF}$ is the dominant term in the Schur complement matrix. The application of IBP in a step of an iterative solver is illutrated in Fig. 2. Since IBP requires the solution of two systems for each iterative step, we need to use a flexible solver for the solution of (1) [6].

In Table 1, we evaluate the performance of the SAI preconditioner for the solutions of (9) and (10), where we compare the number of iterations obtained with the SAI-preconditioner and the

Table 1. Number of iterations of the systems in (9) and (10) for the sphere problem.

Number of Unknowns	CTF				JMCFIE			
	$\overline{Z}_{11}^{NF} \cdot x = f'$		$\overline{S} \cdot y = g'$		$\overline{Z}_{11}^{NF} \cdot x = f'$		$\overline{S} \cdot y = g'$	
	No PC	SAI	No PC	SAI	No PC	SAI	No PC	SAI
1,860	167	9	166	10	38	7	40	10
7,446	195	10	193	10	37	6	40	10
29,742	217	10	213	10	38	6	43	9
65,724	243	10	238	9	39	6	44	9
264,006	294	10	282	9	41	6	45	9

no-preconditioner (No PC) cases for 10^{-6} residual error. For both formulations, we observe that SAI is very successful and decreases the iteration counts drastically. Furthermore, contrary to the the no-preconditioner case, number of iterations does not increase for the SAI preconditioner as the number of unknowns increase. Hence, the use of SAI in this context significantly increases the performance of IBP.

3. RESULTS

The numerical experiments are carried out in a server with two Intel Xeon 5355 CPU and 16 GB of RAM. We use flexible GMRES with no restart as the solver. We note that the solutions of dielectric problems require many more matrix-vector multiplications with other nonsymmetric solvers, such as the biconjugate gradient stabilized (BiCGStab) method [9]. Iterations are performed until the norm of the initial residual is reduced by 10^{-3}. This error level is practical and in accordance with the error performed in MLFMA. Zero initial guess and right preconditioning are used in all solutions. RHSs are determined by plane wave excitations.

We demonstrate the performance of IBP on a sphere and a lens [2] with inner dielectric constants of, 4.0 and 12.0, respectively, as shown in Fig. 3.

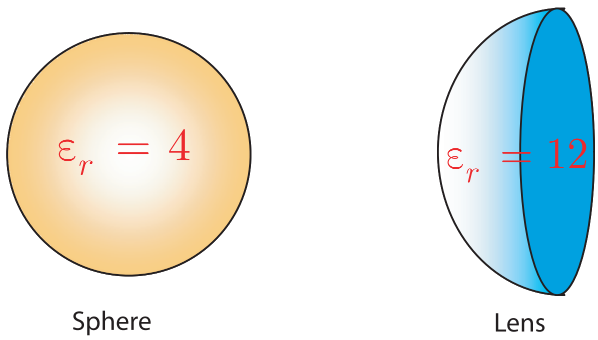

Fig. 3. Sphere and lens problems that are used in the experiments.

3.1 Selection of the Inner Stopping Criteria

A critical issue for the performance of IBP is the selection of the stopping tolerances for (9) and (10). The accuracy of these inner solves should be optimized to minimize the overall solution time. For this purpose, in Fig. 4, we analyze the convergence behaviour of SAI-preconditioned solutions for the sphere problem with 264,006 unknowns and for the lens problem with 158,286 unknowns. Note that for both formulations, only two iterations suffice to obtain a 0.1 residual error. However, for the lens problem, which has a higher dielectric constant than that of sphere, the solution of the Schur system (10) requires more iterations to reduce the norm of the residual to 10^{-3} . Hence, we set our inner stopping criteria as one order residual drop with a maximum of three iterations. The results of the experiments show that with such a relaxed stopping criteria we achieve very strong preconditioners.

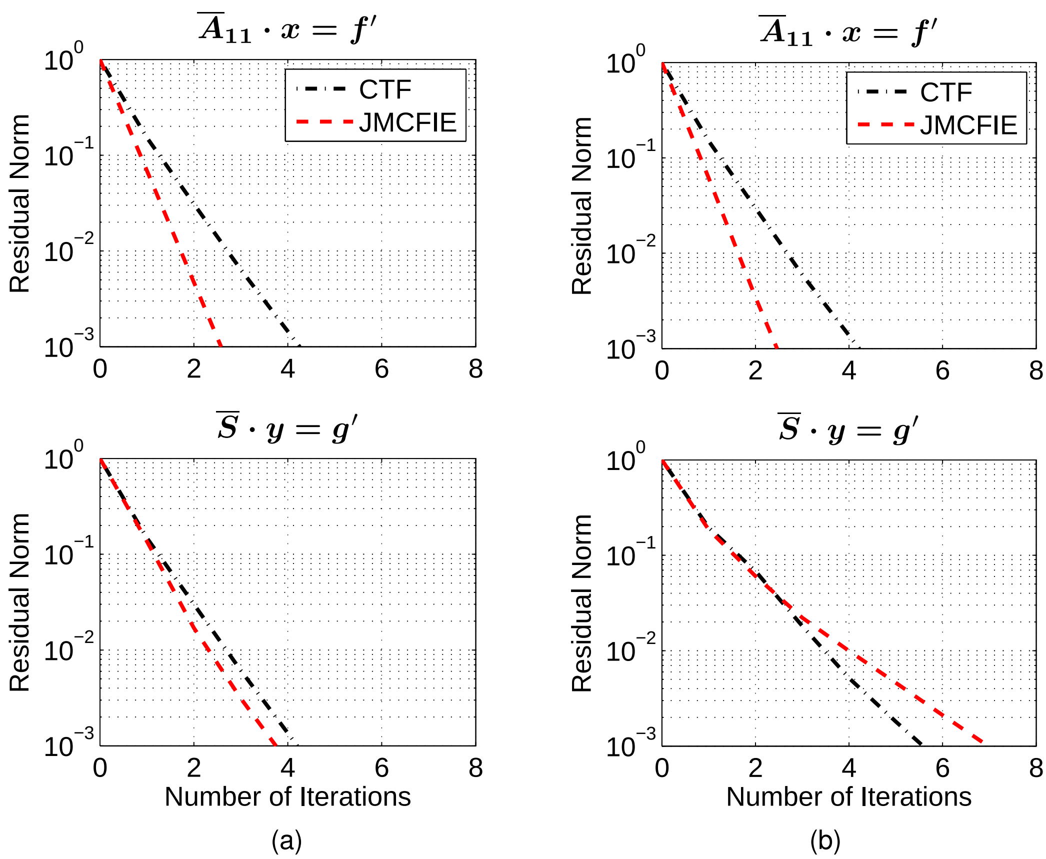

Fig. 4. SAI-preconditioned residual plots of inner solutions for (a) sphere and (b) lens problems.

3.2 Sphere Results

In Table 2, we show the solution frequencies and corresponding number of unknowns for the sphere problem. We use sphere problem since the accuracy of the solutions can be evaluated by comparing them with analytical solutions. In Fig.5, we show the iteration counts and in Fig. 6 we show the solution times for these problems. In [9], a four-partition block-diagonal preconditioner (4PBDP) has been proposed for the solution of dielectric problems, which use the self interactions of the lowest-level clusters MLFMA. However, for CTF, this preconditioner decelerates the convergence and increases the number of iterations, hence we do not include it in CTF solutions. We summarize our observations about the sphere solutions as follows:

- When a strong preconditioner is not used for CTF, even though we use the robust FGMRES solver, the iteration counts grow rapidly as the problem sizes get larger. Even though iteration counts are less for JMCFIE compared to CTF, for problems having larger than 100,000 unknowns iteration counts get larger also for JMCFIE. Hence, JMCFIE solutions can still benefit from a strong preconditioner. We also note that, without a preconditioner, it becomes even more difficult to attain convergence of these formulations with other non-optimal but less memory-hungry solvers, such as BiCGStab.

- For CTF, convergence is attained at least four times faster with IBP compared to the no-preconditioning case. For JMCFIE, IBP provides convergence three times faster compared to the no-preconditioning case, and two times faster compared to 4PBDP.
- For the sphere problem, the solution times of CTF problems become close to those of JMCFIE. Since CTF requires less memory and produces better accuracy [9], it may be preferable to JMCFIE for sphere solutions when accelerated by IBP.

Table 2. Information about sphere problems.

Problem	Frequency (GHz)	Size (λ)	MLFMA Levels	Number of Unknowns
S1	0.5	1	3	1,860
S2	1.0	2	4	7,446
S3	2.0	4	6	29,742
S4	3.0	6	6	65,724
S5	6.0	12	7	264,006
S6	7.5	15	7	412,998
S7	8.5	17	8	540,450

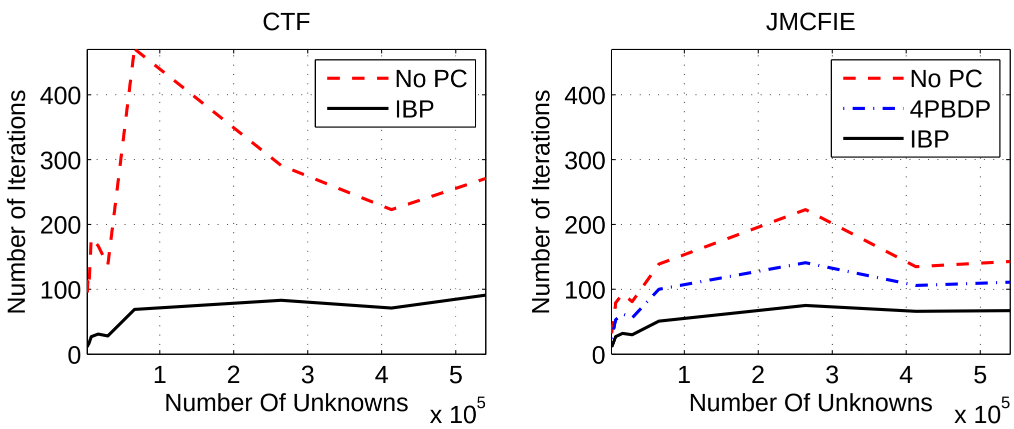

Fig. 5. Number of iterations of the sphere problem for CTF and JMCFIE.

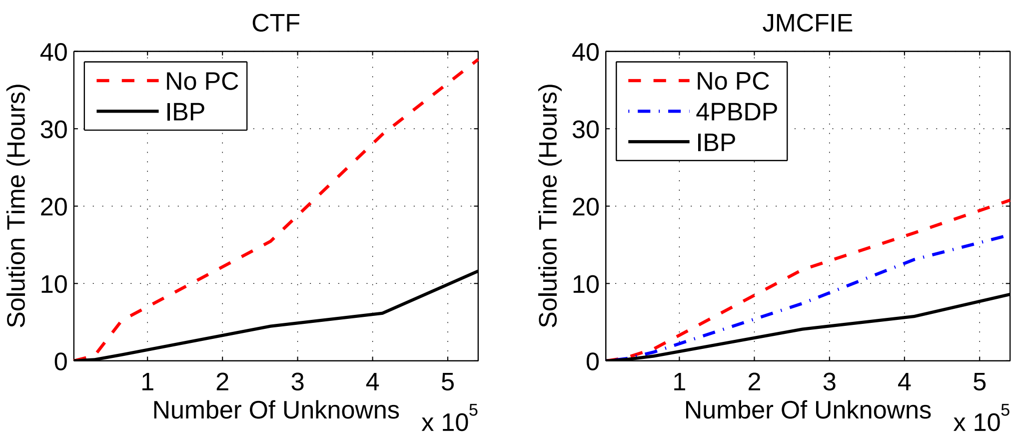

Fig. 6. Solution times of the sphere problem for CTF and JMCFIE.

3.3 Lens Results

For radiometric remote sensing applications, delicate simulations of dielectric lenses are required for a wide spectrum beginning from 30 GHz [2]. In this section, we analyze preconditioned iterative solutions of this important problem. We solve problems from 30 GHz to 120 GHz as shown in Table 3. The inner dielectric constant of the hemisphere is 12.0. Note that the solution of the Schur system becomes more difficult as the inner dielectric constant is increased, as shown in Fig. 4-(b).

Table 3. Information about lens problems.

Problem	Frequency (GHz)	Size (λ)	MLFMA Levels	Number of Unknowns
L1	30	2.5	6	38,466
L2	60	5.0	7	158,286
L3	90	7.5	7	353,646
L4	120	10.0	8	632,172

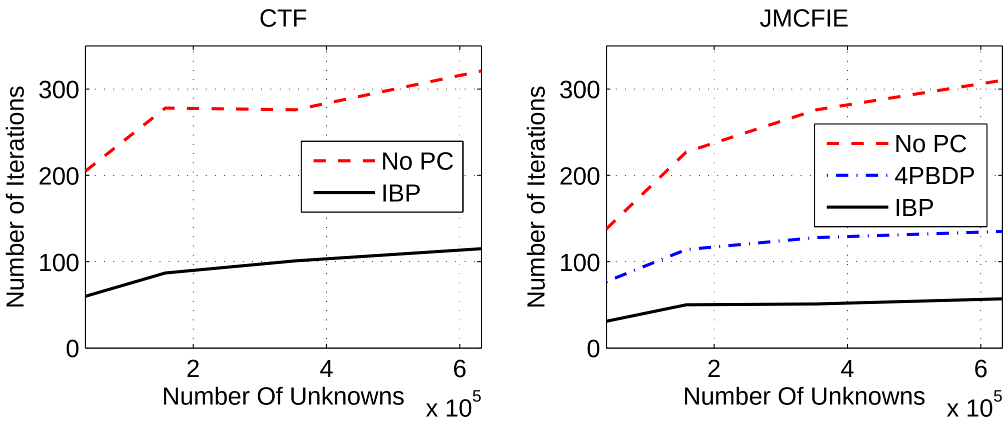

Fig. 7. Number of iterations of the lens problem for CTF and JMCFIE.

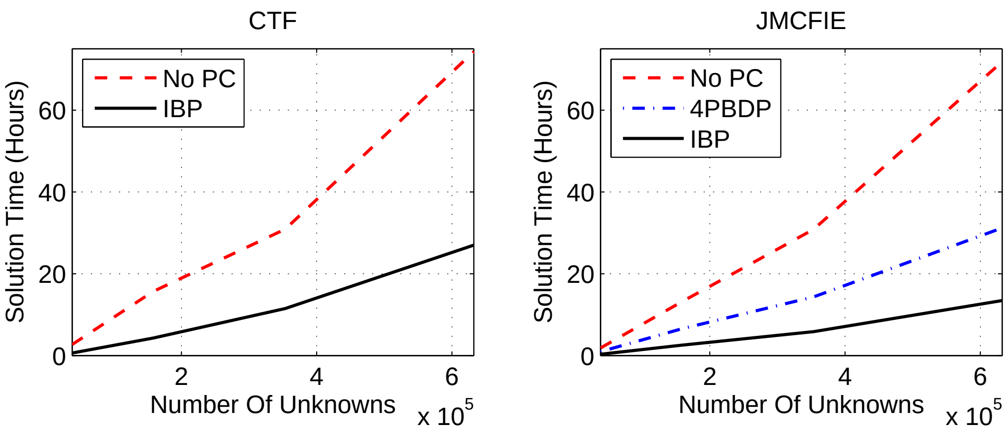

Fig. 8. Solution times of the lens problem for CTF and JMCFIE.

In Figs. 7 and 8, we depict the iteration counts and solution times, respectively. We summarize our comments on the solutions of the lens problem as follows:

- We see that particulary for JMCFIE, the iteration counts are larger than the sphere problem if preconditioner is not used. Hence, preconditioning of CTF and JMCFIE becomes more critical for real-life problems having high inner dielectric constants.
- CTF solves the problems three times faster with respect to the no-preconditioning case. The solutions of CTF with IBP are as fast as those of JMCFIE with 4PBDP.
- JMCFIE solution times has been significantly reduced by IBP for the lens problem. We see that solutions are obtained 5-6 times faster with respect to no preconditioner and 2.5-3 times faster with respect to 4PBDP. JMCFIE with IBP provides fastest solutions for the lens problem.

4. CONCLUSION

In this work, we propose an efficient block preconditioner generated from the near-field matrix of MLFMA to accelerate the convergence of the two common dielectric formulations. We show how to optimize the inner solutions so that maximum efficiency is obtained. Both of the formulations CTF and JMCFIE highly benefit from IBP. For the sphere problem with an inner dielectric constant of 4.0, CTF solutions can be obtained as fast as those of JMCFIE when IBP is utilized. For the lens problem, however, when accelerated by IBP, JMCFIE provides solutions faster than CTF. For both problems, solution times are significantly reduced by IBP compared to both no-preconditioner case and previously proposed 4PBDP.

REFERENCES

[1] Loschialpo P., D. W. Forester, and J. Schelleng, "Anomalous transmission through near unit index contrast dielectric photonic crystals," *J. Appl. Phys.*, Vol. 86, No. 10, 5342–5347, 1999.

[2] Pavacic A. P., D. Llorens del Rio, J. R. Mosig, and G. V. Eleftheriades, "Three-dimensional ray-tracing to model internal reflections in off-axis lens antennas," *IEEE Trans. Antennas Propagat.*, Vol. 54, 604–612, 2006.

[3] Lloyd T. W., J. M. Song, and M. Yang, "Numerical Study of Surface Integral Formulations for Low-Contrast Objects," *IEEE Antenna Wireless Propag. Lett.*, Vol. 4, 482–485, 2005.

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

[5] Trefethen, L. N., and D.,III, Bau, *Numerical Linear Algebra.* SIAM, Philadelphia, USA, 1997.

[6] Saad Y., *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, USA, second ed., 2003.

[7] Malas, T. and L. Gürel, "Sequential and Parallel Preconditioners for Large-Scale Integral-Equation Problems," in *CEM'07 Computational Electromagnetics Workshop*, Izmir, Turkey, Aug. 2007, 35–43 urkey, Aug. 2007, 35–43.

[8] Malas, T. and L. Gürel, "Accelerating the Multilevel Fast Multipole Algorithm with the Sparse-Approximate-Inverse (SAI) Preconditioning," *SIAM J. Sci. Comput.*, submitted for publication, Feb. 2008.

[9] Ergül, Ö. and L. Gürel, "Comparison of integral-equation formulations for the fast and accurate solution of scattering problems involving dielectric objects with the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propagat.*, accepted for publication, Sept. 2008.

[10] Benzi, M., G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems", *Acta Numer.*, vol. 14, 2005, 1–137.

[11] Balay S., W. D. Gropp, L. C. McInnes, and B. F. Smith, "PETSc Users Manual," Tech. Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.