

Communication

Tridiagonal and Block Tridiagonal Computed Sparse Preconditioners for Large Electrodynamic Electric Field Integral Equation (EFIE) Solution

Yoginder Kumar Negi^{1b} and N. Balakrishnan^{1b}

Abstract—In this work, we propose simple and efficient tridiagonal computed sparse preconditioners for improving the condition number for large compressed electric field integral equation (EFIE) method of moment (MoM) matrix. The preconditioner computation is based on the triangle and block triangle interaction and filled tridiagonally. The computed preconditioner is highly sparse and retains the $O(N)$ complexity of computation and preconditioner matrix solution time. Numerical results show the efficiency and accuracy of the proposed preconditioner.

Index Terms—Hierarchical matrices (H-Matrix), integral equation, method of moment (MoM), preconditioner.

I. INTRODUCTION

In the last few decades, computational electromagnetic (CEM) methods have gained popularity for various electromagnetic analyses due to their accuracy and efficiency. The frequency domain electric field integral equation (EFIE) based method of moment (MoM) [1] is one of the popular methods in CEM for solving complex electromagnetic radiation/scattering problems [2]. MoM leads to a dense matrix with $O(N^2)$ matrix fill time and memory requirement for $N \times N$ size matrix. Solving MoM system of equations requires $O(N^3)$ time with direct solver and $N_{itr} O(N^2)$ time with a conventional iterative solver for N_{itr} iterations. The real-world electromagnetics problems are geometrically large and complex; the solution of large-scale problems with MoM is limited due to the high matrix storage, computation, and solve cost. Direct solvers have the advantage of one-time factorization cost for a fixed time and memory. Solving large problems with a direct solver may be time-consuming and memory-intensive; even a few of the proposed fast direct solvers [3], [4] scale poorly for 3-D large complex problems. At the same time, the iterative solver needs less memory and fewer matrix operations than the direct solver. In the iterative solver, high storage and computation cost can be mitigated by incorporating matrix compression based fast solver methods like multilevel fast multipole algorithm (MLFMA) [5], precorrected fast Fourier transform (FFT) [6], adaptive cross approximation (ACA) [7], [8], Hierarchical matrices (H-Matrix) [9], [10], and IE-QR [11]. The matrix storage, fill-time and matrix-vector product time can be reduced to $O(N \log N)$ with a reduced solution time of $N_{itr} O(N \log N)$ for N_{itr} iterations. EFIE being a Fredholm integral equation of the first kind, the eigenvalue tends to cluster at zero and infinity leading to the poor condition number of the matrix. As the number of unknowns increases, the number of large and

small eigenvalues also increases, which leads to an increase in the ill-conditioning of the matrix. Ill-conditioned matrices are highly sensitive to perturbation in the system, which may jeopardize the accuracy of the solution and leads to a high iterative solution iteration count. Preconditioning [12]–[14] of a matrix helps to improve the condition number of the matrix by clustering the eigenvalues around 1 and reducing the solution iteration count. Preconditioning is a way to convert the coefficient matrix from a system of the equation to the desired property system before the solution.

Broadly, preconditioners can be classified as analytic and algebraic. Analytic preconditioners like Calderon preconditioner [15] are kernel-dependent and sensitive to characteristics of the operator, thus applicable to a narrow class of problems. In comparison, algebraic preconditioners are more versatile and applicable for a broad range of problems. Incomplete LU (ILU) [16]–[18] and sparse approximation inverse (SPAI) [19], [20] are the few popular algebraic preconditioners for accelerating the iterative solution process. For significant size problems, ILU is limited due to the serial nature of LU factorization and selection of drop tolerance (τ) and fill-in (p) parameters. On the contrary, SPAI is limited by the quadric cost of computation and is applicable for parallel process-based matrix solutions. The near-field matrix [21] of a fast solver can also be used as a preconditioner, but the high factorization cost limits the application as a preconditioner. A scaled near-field block-diagonal preconditioner is presented in [22]–[25], but the diagonalization process is complex. The preconditioner should be simple and low-cost in computation, and effective in improving the condition number of the matrix.

The diagonal and block-diagonal preconditioners are the simplest but are not effective in improving the condition number of the large size matrix [26]. In this communication, we propose novel sparse preconditioners based on the tri-diagonal and block tridiagonal interaction. The preconditioner is highly sparse and has a very low solution time. Tridiagonal matrix preconditioner is presented in [27] and [28] and is applied for solving sparse matrices arising from Navier–Stokes equation. Our proposed sparse preconditioner is based on the triangle interaction and triangle cluster interaction at the lowest level of the binary-tree/oct-tree. The sparse preconditioners scale the columns of the coefficient matrix and improve the spectral property of the matrix, which further improves the iteration count during the solution process. The numerical results show the accuracy and efficiency of the proposed preconditioner method.

The communication is organized as follows: in Section II, a brief description of EFIE H-Matrix is presented. In Section III, the proposed tridiagonal and block tridiagonal preconditioner is presented. In Section IV, the efficiency and accuracy of the proposed H-Matrix are presented. Section V concludes the communication.

II. EFIE H-MATRIX

EFIE-based MoM is a popular method for solving open and closed conductor body problems in electromagnetics. For a 3-D arbitrary

Manuscript received May 7, 2021; revised November 2, 2021; accepted November 15, 2021. Date of publication December 29, 2021; date of current version June 13, 2022. This work was supported by the Department of Science and Technology (DST) Government of India through the National Supercomputer Mission (NSM) Project under Grant SP/DSTO-20-0130. (Corresponding author: Yoginder Kumar Negi.)

The authors are with the Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India (e-mail: yknegi@gmail.com; balki@iisc.ac.in).

Color versions of one or more figures in this communication are available at <https://doi.org/10.1109/TAP.2021.3137406>.

Digital Object Identifier 10.1109/TAP.2021.3137406

0018-926X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

shape conducting body, the EFIE boundary condition for an object illuminated with an incident field on the surface S is given as

$$|\mathbf{E}_s(\mathbf{J}) + \mathbf{E}_i|_{tan} = 0 \quad (1)$$

where \mathbf{E}_s is the scattered electric field due to the induced surface current \mathbf{J} on the object illuminated by an incident electric field \mathbf{E}_i and tan in (1) is a tangential component of the electric field. The scattered electric field can be further written as

$$\mathbf{E}_s(\mathbf{J}) = -j\omega\mathbf{A} - \nabla\phi \quad (2)$$

where \mathbf{A} and ϕ represent the vector and scalar potentials, and ω is the angular frequency. Expanding scalar and vector potential and using the Galerkin testing method with RWG basis function [29], the resultant MoM system of (2)

$$[\mathbf{Z}][\mathbf{x}] = [\mathbf{b}]. \quad (3)$$

In (3), for a given unknown N , $[\mathbf{Z}]$ is a dense MoM matrix of size $N \times N$, \mathbf{b} is an incident vector, and \mathbf{x} is a solution vector of size $N \times 1$. Dense MoM matrix leads to $O(N^2)$ matrix storage and filling time. The matrix storage and fill time can be reduced by incorporating fast algorithms for matrix filling and solutions. These methods work on the principle of analytic and algebraic matrix compressibility of far-field interaction blocks. MLFMA and FFT are analytic matrix compression methods. Algebraic matrix compression methods are ACA and H-Matrix, and IE-QR. These methods are kernel-independent and easy to implement compared to an analytic method like MLFMA. In this work, we used half H-Matrix [30] with recompressed ACA [31] to take advantage of algebraic compression and reduce the overall matrix storage requirement. For the H-Matrix construction, the compression scheme can be applied on a binary-tree based 3-D geometry decomposition, where the matrix compression is applied for block interaction satisfying the admissibility condition

$$\eta dis(\Omega_t, \Omega_s) \geq \min(dia(\Omega_t), dia(\Omega_s)). \quad (4)$$

The admissibility condition of (4) states that for matrix compression admissibility constant (η) times the distance between the test (Ω_t) and source blocks (Ω_s) should be greater than or equal to the minimum of the block diameter of the test block and source block. The binary-tree partition of the geometry is carried out until the number of elements in the block is less than or equal to 30 basic elements. At the leaf level, the block interaction not satisfying the admissibility condition is considered as a near-field interaction. In the case of the multilevel binary tree, the far-field block satisfying admissibility condition interacted at a higher level does not interact at the lower level. As the number of unknowns grows up, the solution time grows up due to the increase in iteration count. The condition number of the matrix deteriorates as the matrix size increases; furthermore, mesh inconsistencies and geometry type may also lead to matrix ill-conditioning. An ill-conditioned matrix leads to a high iteration count and solution time. Preconditioners can improve the condition number of the matrix and accelerate the solution time. In Section III, we propose a simple and efficient preconditioner based on the tridiagonal and block tridiagonal matrix computation.

III. TRIDIAGONAL AND BLOCK TRIDIAGONAL PRECONDITIONING

Iterative solution of large size matrix with Krylov subspace depends on the condition number of the matrix and matrix-vector product cost. Matrix-vector product cost can be reduced using fast solver methods, whereas iteration count in the matrix is condition

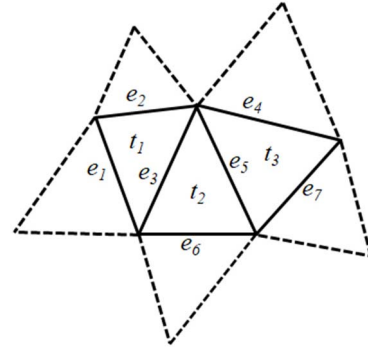


Fig. 1. Triangle interaction for sparse matrix computation.

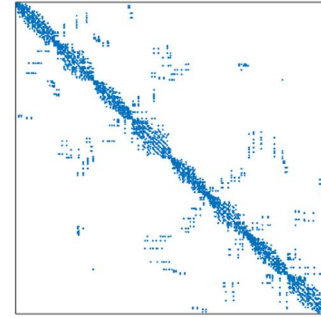


Fig. 2. Sparse tridiagonal computed preconditioner for $1\lambda \times 1\lambda$ metallic plate.

number dependent, and as the number of unknown grows, the condition number of matrix deteriorates. Preconditioning is one of the efficient methods to improve the condition number of matrix and expedite the solution process. The preconditioned system can be either as a left (5) or right (6) and is given as

$$[\mathbf{P}^{-1}][\mathbf{Z}][\mathbf{x}] = [\mathbf{P}^{-1}][\mathbf{b}] \quad (5)$$

$$[\mathbf{Z}][\mathbf{P}^{-1}][\tilde{\mathbf{x}}] = [\mathbf{b}] \quad (6)$$

where \mathbf{P} is the preconditioner matrix, \mathbf{Z} is the EFIE MoM full matrix or compressed matrix, \mathbf{b} is excitation vector, \mathbf{x} and $\tilde{\mathbf{x}}$ are the solution vectors, where $\tilde{\mathbf{x}}$ is $[\mathbf{P}]\mathbf{x}$. To keep the cost of the iterative solution low, the preconditioner matrix should be highly sparse in nature and effective in improving the condition number of the matrix. In this section, we propose a new sparse preconditioner used as a left preconditioner for solving a large compressed matrix. The inverse of the MoM matrix is the ideal preconditioner for solving the MoM matrix with an iterative method, but the cost of the MoM inverse is memory and compute-intensive. Most of the algebraic preconditioners try to depict the inverse of the actual solution matrix. Therefore, most of the proposed preconditioners in literature are derived from the actual matrix.

The proposed first tridiagonally computed sparse preconditioner is highly sparse and is derived from the MoM matrix. The sparse preconditioner is computed by considering only the mesh triangle to triangle interaction in the MoM matrix. For an illustrative purpose, Fig. 1 shows three triangles that fill the MoM sparse preconditioner matrix. Here the triangle t_2 interacts with t_1 and t_3 to fill the seven edges (e_1, e_2, \dots, e_7) indices in the MoM matrix. If the triangle t_1 is boundary triangle it interacts with t_2 else it as in the case of nonboundary, it interacts with t_2 and t_0 . Dotted lines in the figure are the contributing triangles for the RWG edges. Fig. 2 shows the sparse tridiagonal preconditioner computed for $1\lambda \times 1\lambda$ metallic plate. The

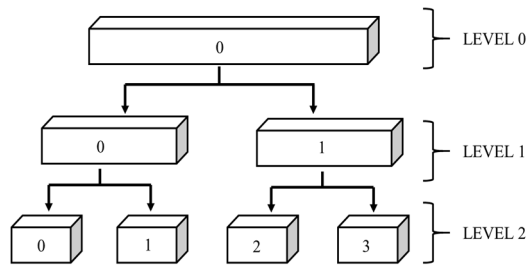


Fig. 3. Triangle block binary tree division of geometry for level 2.

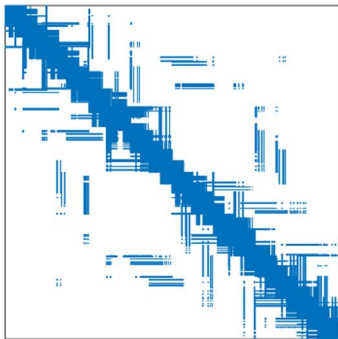


Fig. 4. Sparse block tridiagonal computed preconditioner for $2\lambda \times 2\lambda$ metallic plate for binary tree level 5.

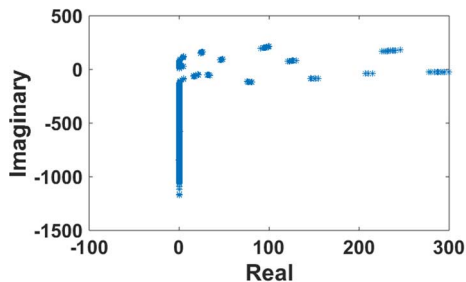


Fig. 5. Eigenvalue distribution of 1λ radius sphere MoM matrix with 5334 RWG edges.

preconditioner is computed for 280 unknowns with 2688 numbers of nonzeros (NNZs).

Similarly, we can compute block sparse preconditioner with triangle block interaction. Fast solvers like MLFMA and H-Matrix rely on the oct-tree or binary-tree (Fig. 3) geometry division. For the divided geometry, the block matrix interaction is compressed at different levels with satisfying the far-field criteria. The non-far-field block interaction at the lowest level is considered near-field. Taking advantage of the geometric block partition for fast solvers, the preconditioner is computed for tridiagonal block interaction. The geometric block partition is done for triangles up to the desired level, with an average of 30 triangles in a group. For the block preconditioner computation as shown in Fig. 3, at the lowest level, triangles in block 1 interact with block 0 and 1, and in the case of boundary block 0, it interacts with block 1 triangles.

Fig. 4 shows the sparse block tridiagonal preconditioner computed for $2\lambda \times 2\lambda$ metallic plate. The preconditioner is computed for 1160 unknowns at binary-tree level 5 with 156 644 NNZ's.

Fig. 5 above shows the eigenvalue distribution of 1λ radius sphere MoM matrix of size 5334×5334 .

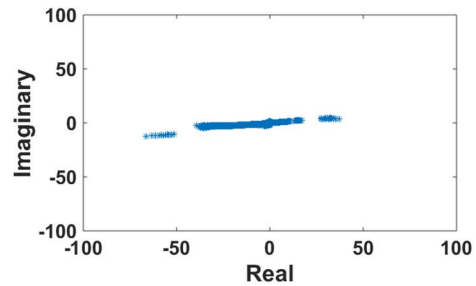


Fig. 6. Eigenvalue distribution after tridiagonal computed sparse preconditioned 1λ radius sphere matrix.

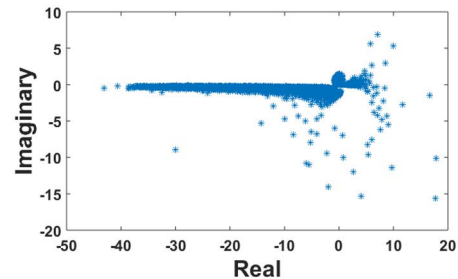


Fig. 7. Eigenvalue distribution after block tridiagonal computed sparse preconditioned 1λ radius sphere matrix.

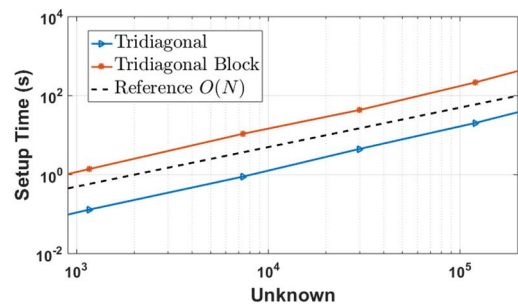


Fig. 8. Computation time for tridiagonal and block tridiagonal computed sparse preconditioner with increasing unknowns.

Figs. 6 and 7 show the eigenvalue distribution of 1λ radius sphere with 5334 unknowns MoM matrix after preconditioning. It can be observed from the figure that the proposed preconditioners efficiently scale the columns of the MoM matrix and cluster the eigenvalues around 1, thus improving the spectral property of the EFIE matrix.

IV. COMPLEXITY ANALYSIS

In this section, we show the efficiency of the proposed sparse preconditioners for set-up time, LU solve time, and memory. The complexity analyses are carried for perfect electric conductor (PEC) plates with increasing unknown and size. One of the prime properties of a preconditioner should be its linear time complexity for set-up, and Fig. 8 shows that the proposed preconditioners retain the $O(N)$ complexity for computation.

Iterative solver cost depends on the matrix-vector product time, and preconditioned iterative solver depends on the preconditioner LU solve time of the preconditioner. An efficient preconditioner should have a very less LU solve time with linear complexity. Fig. 9 below shows the linear LU solve time complexity for the proposed sparse preconditioners.

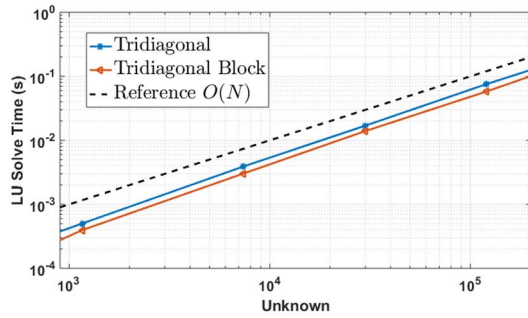


Fig. 9. LU solve time for tridiagonal and block tridiagonal computed sparse preconditioner with increasing unknowns.

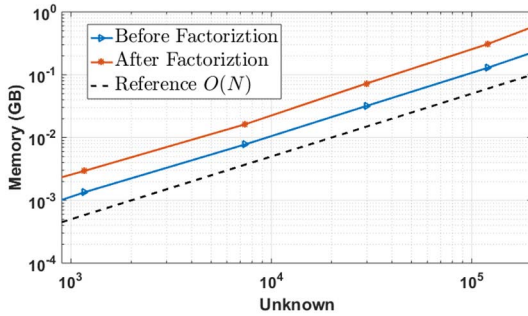


Fig. 10. Memory required for tridiagonal computed sparse preconditioner before and after factorization with increasing unknowns.

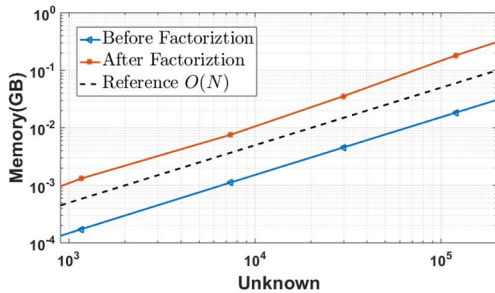


Fig. 11. Memory required for block tridiagonal computed sparse preconditioner before and after factorization with increasing unknowns.

Along with time complexity, memory requirement plays a vital role in the efficiency of the solution process. A high memory preconditioner may jeopardize the iterative solution limiting the preconditioner applicable to a small size problem. Figs. 10 and 11 show the $O(N)$ memory complexity for tridiagonal and block tridiagonal computed sparse preconditioner before and after factorization.

V. NUMERICAL RESULTS

In this section, we show the accuracy and the efficiency of the proposed preconditioners. All the simulations are done with ACA compressed H-Matrix fast solver (compression tolerance = $1e-3$) and solved with Krylov subspace-based iterative solver (GMRES) for convergence error of $1e-6$ for PEC geometry. Computation was carried out for double-precision data type on 128 GB memory and Intel (Xeon E5-2670) processor system.

A. Accuracy

In this section, we show the accuracy of the proposed preconditioners for open and closed geometry. Fig. 12 shows the monostatic

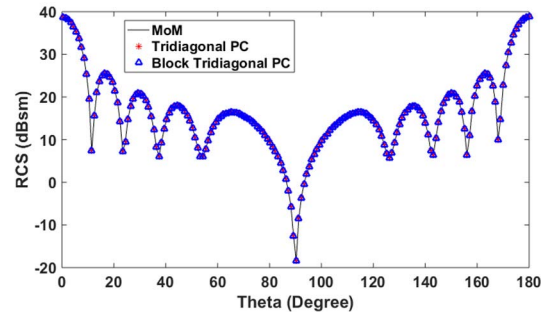


Fig. 12. Bistatic RCS of 5λ square plate with VV polarized plane wave incident at $\theta = 0^\circ$, $\phi = 0^\circ$, and observation angles $\theta = 0^\circ$ to 180° , $\phi = 0^\circ$.

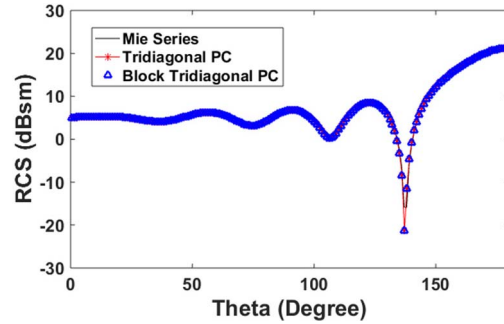


Fig. 13. Bistatic RCS of 1λ radius sphere with VV polarized plane wave incident at $\theta = 0^\circ$, $\phi = 0^\circ$ and observation angles $\theta = 0^\circ$ to 180° , $\phi = 0^\circ$.

RCS computation for 5λ square plate from MoM iterative solver, tridiagonal and block tridiagonal computed sparse preconditioned fast solvers for 7400 unknowns.

The RCS is computed for the VV polarized plane wave incident at $\theta = 0^\circ$, $\phi = 0^\circ$ and observation angles $\theta = 0^\circ$ to 180° , $\phi = 0^\circ$. It can be observed that the RCS computed from the preconditioned fast solvers agrees with the MoM computed RCS. For the MoM solution, the iterative solver takes 430 iterations, preconditioned tridiagonal preconditioner takes 45 iterations, and preconditioned block tridiagonal preconditioner takes 30 iterations to converge.

Fig. 13 shows the bistatic RCS computation for a 1λ radius PEC sphere with 5334 unknowns. The RCS is computed using the Mie series analytic method and preconditioned tridiagonal and block tridiagonal fast solver for the VV polarized plane wave incident at $\theta = 0^\circ$, $\phi = 0^\circ$ and observation angles $\theta = 0^\circ$ to 180° , $\phi = 0^\circ$. It can be observed that the RCS computed from the preconditioned fast solvers agrees with the Mie series RCS. For matrix solution, fast solver iterative solution takes 302 iterations, preconditioned tridiagonal computed sparse preconditioner takes 73 iterations, and preconditioned block tridiagonal computed sparse preconditioner takes 52 iterations to converge.

B. Efficiency

In this section, the efficiency of the proposed preconditioners is validated. As discussed in our previous works [23], [25], the preconditioner efficiency cannot be concluded with fast iteration only. Along with iteration count, the preconditioner LU factorization solve time plays a vital role in overall solution time. In Table I, we show the solve time efficiency of our proposed preconditioners. The performance of the proposed preconditioners is compared with that of ILUT, with the parameters are chosen as given in [32]. The relative efficiency of a preconditioner depends on some key parameters:

TABLE I
PRECONDITIONER EFFICIENCY FOR DIFFERENT GEOMETRY

	N	PC Type	T_{pc} (s)	N_{itr}	T_{pcsol} (s)	T_{mmv} (s)	T_{total} (H) (180 RHS)	Speed-up
20 λ Plate	119,600	TD	63.8798	80	0.063675	0.703204	3.1384	1.46
		TD Block	237.9965	51	0.067345	0.703204	2.5727	1.78
		ILUT	273.7414	89	0.309779	0.703204	4.5838	----
5 λ Sphere	130,293	TD	75.9628	762	0.221020	1.450142	63.6923	1.36
		TD Block	801.7663	711	0.079829	1.450142	54.6131	1.59
		ILUT	842.6682	843	0.611366	1.450142	87.1266	----
AC (1GHz)	412,690	TD	724.8528	7788	0.520205	9.319170	3831.65	2.0
		TD Block	6077.1354	7006	0.248953	9.319170	3353.40	2.3
		ILUT	26138.339	12191	3.370137	9.319170	7742.02	----

1) T_{pc} : preconditioner set-up time; 2) N_{itr} : average number of iterations required for convergence for one right-hand side (RHS); 3) N_{rhs} : number of RHS vectors; 4) T_{pcsol} : preconditioner solve time; and 5) T_{mmv} : MoM matrix-vector product time. The total solve time is given by

$$T_{total} = T_{pc} + [N_{itr} \times N_{rhs} \times (T_{pcsol} + T_{mmv})]. \quad (7)$$

Table I shows the speed-up efficiency of the proposed preconditioners for 180 RHS total solve time for a PEC plate, sphere, and aircraft (AC). For plate and sphere, the iterations are computed for the VV polarized plane wave incident and observation angles at $\theta = 0^\circ$ to 180° and $\phi = 0^\circ$ and for 14 m length and 8 m wingspan AC meshed with $\lambda/10$ element size at 1 GHz the iterations are computed for the VV polarized plane wave incident and observation angles at $\theta = 90^\circ$ and $\phi = 0^\circ$ to 180° .

VI. CONCLUSION

The proposed preconditioners are simple to compute and effective in accelerating the iterative solution of large-size problems. The preconditioners are sparse matrices based on the tridiagonal triangle and block tridiagonal interaction. These preconditioners maintain $O(N)$ set-up time, solution time, and memory complexity. The preconditioners have a very low set-up time and can be divided into blocks and computed independently, thus making them highly efficient for parallel application.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for the insightful comments and suggestions which helped us to improve this communication.

REFERENCES

- [1] R. F. Harrington, *Field Computation by Moment Methods*. Malabar, FL, USA: Krieger Publishing, 1982.
- [2] V. P. Padhy, Y. K. Negi, and N. Balakrishnan, "RCS enhancement due to Bragg scattering," in *Proc. Int. Conf. Math. Methods Electromagn. Theory*, Aug. 2012, pp. 443–446.
- [3] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numerica*, vol. 18, pp. 243–275, May 2009.
- [4] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2306–2313, Aug. 2008.
- [5] W. C. Chew, J. M. Jin, E. Michielssen, and J. Song, *Fast Efficient Algorithms in Computational Electromagnetics*. Boston, MA, USA: Artech House, 2001.
- [6] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 10, pp. 1059–1072, Oct. 1997.
- [7] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Math.*, vol. 86, no. 4, pp. 565–589, Jun. 2000.
- [8] S. Kurz, O. Rain, and S. Rjasanow, "The adaptive cross-approximation technique for the 3D boundary–element method," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 421–424, Mar. 2002.
- [9] W. Hackbusch, "A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.
- [10] W. Hackbusch and B. N. Khoromskij, "A sparse H-matrix arithmetic. Part II: Application to multi-dimensional problems," *Computing*, vol. 64, pp. 21–47, Apr. 2000.
- [11] S. Kapur and D. E. Long, "N-body problems: IES³: Efficient electrostatic and electromagnetic simulation," *IEEE Comput. Sci. Eng.*, vol. 5, no. 4, pp. 60–67, Oct. /Dec. 1998.
- [12] M. Benzi, "Preconditioning techniques for large linear systems: A survey," *J. Comput. Phys.*, vol. 182, no. 2, pp. 418–477, 2002.
- [13] B. Carpentieri, "Preconditioning for large-scale boundary integral equations in electromagnetics [open problems in CEM]," *IEEE Antennas Propag. Mag.*, vol. 56, no. 6, pp. 338–345, Dec. 2014.
- [14] B. Carpentieri, I. S. Duff, L. Giraud, and M. Magolu Monga Made, "Sparse symmetric preconditioners for dense linear systems in electromagnetism," *Numer. Linear Algebra Appl.*, vol. 11, no. 89, pp. 753–771, Oct. 2004.
- [15] S. Christiansen and J. Nédélec, "A preconditioner for the electric field integral equation based on Calderon formulas," *SIAM J. Numer. Anal.*, vol. 40, no. 3, pp. 1100–1135, 2003.
- [16] Y. Saad, "ILUT: A dual threshold incomplete LU factorization," *Numer. Linear Algebra Appl.*, vol. 1, no. 4, pp. 387–402, 1994.
- [17] T. Malas and L. Gürel, "Incomplete LU preconditioning with the multilevel fast multipole algorithm for electromagnetic scattering," *SIAM J. Sci. Comput.*, vol. 29, no. 4, pp. 1476–1494, 2007.
- [18] B. Carpentieri and M. Bollhöfer, "Symmetric inverse-based multilevel ILU preconditioning for solving dense complex non-hermitian systems in electromagnetics," *Prog. Electromagn. Res.*, vol. 128, pp. 55–74, 2012.
- [19] M. Benzi, C. D. Meyer, and M. Tüma, "A sparse approximate inverse preconditioner for the conjugate gradient method," *SIAM J. Sci. Comput.*, vol. 17, no. 5, pp. 1135–1149, Sep. 1996.

- [20] B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand, "Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 774–792, 2005.
- [21] M. Carr and M. J. L. Bleszynski Volakis, "A near-field preconditioner and its performance in conjunction with the BiCGstab(ell) solver," *IEEE Antennas Propag. Mag.*, vol. 46, no. 2, pp. 23–30, Apr. 2004.
- [22] Y. K. Negi, N. Balakrishnan, S. M. Rao, and D. Gope, "Null field preconditioner for fast 3D full-wave MoM package-board extraction," in *Proc. IEEE Electr. Design Adv. Packag. Syst. Symp. (EDAPS)*, Dec. 2014, pp. 57–60.
- [23] Y. K. Negi, N. Balakrishnan, S. M. Rao, and D. Gope, "Null-field preconditioner with selected far-field contribution for 3-D full-wave EFIE," *IEEE Trans. Antennas Propag.*, vol. 64, no. 11, pp. 4923–4928, Nov. 2016.
- [24] Y. K. Negi, N. Balakrishnan, S. M. Rao, and D. Gope, "Schur complement preconditioner for fast 3D full-wave MoM package-board extraction," in *Proc. IEEE Electr. Design Adv. Packag. Syst. (EDAPS)*, 2016, pp. 163–165.
- [25] Y. K. Negi, N. Balakrishnan, and M. Sadasiva Rao, "Symmetric near-field Schur's complement preconditioner for hierarchal electric field integral equation solver," *IET Microw., Antennas Propag.*, vol. 14, no. 14, pp. 1846–1856, 2020.
- [26] L. Gurel, T. Malas, and O. Ergul, "Preconditioning iterative MLFMA solutions of integral equations," in *Proc. URSI Int. Symp. Electromagn. Theory*, Aug. 2010, pp. 810–813.
- [27] L. Lukšan and J. Vlček, "Efficient tridiagonal preconditioner for the matrix-free truncated Newton method," *Appl. Math. Comput.*, vol. 235, pp. 394–407, May 2014.
- [28] J.-C. Li and Y.-L. Jiang, "Generalized tridiagonal preconditioners for solving linear systems," *Int. J. Comput. Math.*, vol. 87, no. 14, pp. 3297–3310, Nov. 2010.
- [29] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.
- [30] Y. K. Negi, "Memory reduced half hierarchal matrix (H-matrix) for electrodynamic electric field integral equation," *Prog. Electromagn. Res. Lett.*, vol. 96, pp. 91–96, 2021.
- [31] Y. K. Negi, V. Prasad Padhy, and N. Balakrishnan, "Re-compressed H-matrices for fast electric field integral equation," in *Proc. IEEE Int. Conf. Comput. Electromagn. (ICCEM)*, Singapore, Aug. 2020, pp. 176–177.
- [32] X. Sherry, "SuperLU: Sparse direct solver and preconditioner," in *Proc. 13th DOE ACTS Collection Workshop*, 2004, pp. 1–55.