Progress In Electromagnetics Research M, Vol. 98, 67–75, 2020

An Effective Sparse Approximate Inverse Preconditioner for Multilevel Fast Multipole Algorithm

Ping Yang[1], Jinbo Liu[2], and Zengrui Li[2, *]

Abstract—In the iterative solution of the matrix equation arising from the multilevel fast multipole algorithm (MLFMA), sparse approximate inverse (SAI) preconditioner is widely employed to improve convergence property. In this paper, based on the geometric information of nearby basis functions pairs and finer octree grouping scheme, a new sparse pattern selecting strategy for SAI is proposed to enhance robustness and efficiency. Compared to the conventional selecting strategies, the proposed strategy has only one variable parameter instructing the constructing time and memory usage, which is more user friendly. Numerical results show that the proposed strategy can make use of the non-zero entries of near-field matrix in MLFMA more effectively and elaborately without compromising the numerical accuracy and the natural parallelization of SAI.

1. INTRODUCTION

As one of the most powerful fast solvers, the multilevel fast multipole algorithm (MLFMA), which drastically reduces the overall computational complexity from the order of $O(N^2)$ to $O(N\text{log}N)$, can make the method of moments (MoM) applied to the analysis of objects with electrically large size [1, 2]. In the MLFMA, based on the addition theorem and diagonalization of the translation operator, the matrix-vector product in traditional MoM is decomposed into two parts as

$$ZI = Z_N I + Z_F I = V \tag{1}$$

where Z_N and Z_F are $N \times N$ impedance matrices, denoting the reactions computing among the basis functions in near and far boxes, and I and V are $N \times 1$ unknown current coefficient and generalized voltage vectors, respectively, while N is the number of unknowns.

During the iterative solution, it is natural to apply preconditioning techniques to accelerate the convergence rate. For this purpose, the matrix Equation (1) can be transformed into one of the two kinds of equivalent equations

$$MZI = MV \quad or \quad ZM\left(M^{-1}I\right) = V \tag{2}$$

for left or right preconditioning, respectively. M is a nonsingular matrix of order N, which is referred as the preconditioner matrix for Z. The purpose of these transformations is to make the product matrix MZ or ZM better conditioned than the original matrix Z. As a consequence, the number of iterations will be greatly reduced. A suitable preconditioner is needed to retain the computational complexity of MLFMA. Algebraic preconditioners, such as incomplete LU (ILU) [3–5] and sparse approximate inverse (SAI) [6–11], have been commonly applied. With some improvements for increasing the robustness, ILU preconditioners had been widely used in the MLFMA. However, with the popularity of parallel computers, the drawback of ILU which it is difficult to be parallelized becomes more and more

Received 11 September 2020, Accepted 26 October 2020, Scheduled 4 November 2020

[*] Corresponding author: Zengrui Li (zrli@cuc.edu.cn).

[1] Beijing Polytechnic, Beijing 100176, China. [2] State Key Laboratory of Media Convergence and Communication and the School of Information and Communication Engineering, Communication University of China, Beijing 100024, China.

unacceptable. On the contrary, due to the natural parallel characteristic, SAI preconditioner, which is based on approximating the inverse of the matrix directly, won the preference. There are several types of SAI preconditioner, while the one depended on Frobenius norm minimization is usually chosen since it allows the decoupling of the constrained minimization problem into N independent linear least-square (LS) problems for each row during the constructing process, which can be parallelized naturally. Furthermore, benefited from the octree grouping schedule in MLFMA, the construction efficiency of preconditioning matrix has been greatly improved. In this way, the LS problems of the whole basis functions belonging to a same finest box need to be solved only once. Apparently, constructing time for SAI is reduced substantially. On the other hand, however, the SAI preconditioner is very sensitive to the distribution density of the basis functions. As the number of basis functions belonging to a certain box increases, the constructing time of SAI increases sharply. To solve this problem, several sparse pattern selecting strategies have been proposed. Different strategies based on both algebraic and geometric information are compared in [8] through a large set of numerical experiments, and it is concluded that the latter ones are more effective.

In this paper, based on the geometric information of nearby basis functions pairs and finer octree grouping scheme than the scheme of original MLFMA, a new sparse pattern selecting strategy for SAI preconditioner is proposed. In contrast to the conventional selecting strategies such as [6–11], the proposed strategy is more effective and user friendly with only one variable parameter instructing the constructing time and memory usage of SAI preconditioner. Meanwhile, the nonzero entries of near-field matrix in MLFMA can be utilized more effectively and elaborately by the proposed strategy, while the numerical accuracy and natural parallelization of SAI are preserved.

2. FORMULATION

In the implementation of MLFMA, owing to the rapid decay of Green function, the general trend of magnitude of the matrix entries obeys physical proximity, i.e., basis functions that are close to each other are expected to have strong electromagnetic coupling, resulting in relatively larger magnitudes of matrix entries. Therefore, the sparse near-field matrix Z_N is likely to retain the most relevant contributions of the dense matrix Z. Hence, effective utilization of Z_N can provide strong preconditioners for Eq. (1). For the convenience of following expression, we use $\bar{Z}$ to denote the matrix extracting from Z_N, which is prepared for constructing M.

In the case of right preconditioning, the approximate inverse of the near-field matrix is computed by minimizing $\left\|E - \bar{Z}M\right\|_F$, where E is the identity matrix, and $\|\cdot\|_F$ denotes the Frobenius norm. The Frobenius norm minimization problem can be decoupled into the sum of the squares of the 2-norms of the individual columns of the residual matrix as

$$\min\left\|E - \bar{Z}M\right\|_F^2 = \min\sum_{k=1}^{N}\left\|e_k - \bar{Z}m_k\right\|_2^2 \tag{3}$$

where e_k and m_k are the kth rows of the identity matrix E and the preconditioner matrix M, respectively. Thus, the computation of M is reduced to solving N independent linear LS problems. In Eq. (3), M is constrained by a certain sparse pattern S. Various selecting strategies for S have been developed, while the pattern of $\bar{Z}$ is often chosen as the optimal one. If S is decided, the nonzero entries position of m_k is accordingly determined. Let J_k be the nonzero structure of m_k as

$$J_k = \{j\,|M(j,k) \neq 0, 1 \leq j \leq N\} \quad J_k \in S \tag{4}$$

and I_k be the set of indices corresponding to the nonzero rows of $\bar{Z}(:, J_k)$ as

$$I_k = \left\{i\left|\bar{Z}(i,j) \neq 0, j \in J_k, 1 \leq i \leq N\right.\right\} \quad I_k \in S\ , \tag{5}$$

then the kth minimization problem is equivalent to

$$\min\left\|\widehat{e}_k - \widehat{Z}_k\widehat{m}_k\right\|_2^2 \tag{6}$$

where $\widehat{m}_k = m_k(J_k)$, $\widehat{e}_k = e_k(I_k)$, $\widehat{Z}_k = Z(I_k, J_k)$. Usually, $\widehat{Z}_k$ is a small full-rank rectangular matrix. The null vectors in $\bar{Z}(:, J_k)$ which do not affect the solution of LS problems are eliminated. If n_2

denotes the number of nonzero entries in m_k and n_1 nonzero rows in $\bar{Z}\left(:, J_k\right)$, the order of independent minimization problem is reduced to $n_1 \times n_2$. Thus, the computational complexity of Eq. (3) is about $O\left(Nn_1n_2^2\right)$. Furthermore, after carefully investigating the MLFMA implementation, we can find that each basis function in a certain box on the finest level couples with the basis functions in the same and the adjacent boxes on the same level via the near-field matrix. Therefore, each basis function in the box couples with the same set of basis functions in the near-field matrix, i.e., for the whole basis functions belonging to a given box, they have an identical $\widehat{Z}_k$. Based on this information, the computational complexity of Eq. (3) can be further reduced into $O\left(Nn_1n_2^2\right)$, where N_b is the number of non-empty boxes in the finest level of MLFMA. This improvement is called modified SAI (MSAI) in [6], which greatly reduces the time cost during construction of the preconditioner matrix without any loss of accuracy.

On the other hand, however, as the density of basis functions increases, $\bar{Z}$ becomes denser. As a response, n_1 and n_2 become larger. Since the computational complexity of Eq. (3) rises with $n_1n_2^2$, the high density of basis functions causes a high constructing time. Especially when the simulated object contains refined structures or thick dielectric bodies, the meshes will be subdivided finely. Thus, each finest box may contain plenty of basis functions, which leads to very large n_1 and n_2. As a result, the cost of SAI preconditioner increases dramatically. The possible high construction cost mentioned above calls for the sparse pattern selecting strategy, that is, obtaining a sparser $\bar{Z}$ by pre-filtering Z_N, which yields a sparser S. There are two main strategies to achieve this purpose as follows:

1) Dropping entries with small magnitude from the original matrix, as done in [7]. However, improper setting may even deteriorate the condition of original matrix equation. Thus, this strategy is not popularized.

2) Dropping entries according to the distance between the centers of boxes and the basis functions. Assume that the kth basis function X_k belongs to the finest box G, then there are two filtrations, τ_1 and τ_2, to filter the set of indices J_k and I_k, respectively, as

$$\left\{ \begin{array}{l} J_G = \left\{ j \left| j \in J_k \ and \ dist\left(k, j\right) \le \tau_1 \right. \right\} \\ I_G = \left\{ i \left| i \in I_k, j \in J_k \ and \ dist\left(i, j\right) \le \tau_2 \right. \right\} \end{array} \right. \tag{7}$$

where $dist(k,\ j)$ denotes the distance between the center point of G and the edge j, and $dist(i,\ j)$ denotes the distance between the center point of groups containing all edges $j \in J$ and the edge i. In other words, when we select nonzero entries of the columns of M correlated with box G such as the kth column, firstly we need to draw a sphere of radius τ_1 centered at the center of G. All of the interactions of the basis functions contained by this sphere will be included by $\bar{Z}$, while the interactions of the others out of the sphere will be eliminated. This strategy is discussed detailed in [6]. However, it has disadvantage. Suppose that the locations of X_k and other two basis functions X_{j1} and X_{j2} in the finest boxes of MLFMA are as shown in Figure 1. For the basis function X_k belonging to G, actually, the distance between X_k and X_{j1} is smaller than τ_1, leading to a strong coupling between the two correlative basis functions, which is eliminated however. On the contrary, the distance between X_k and X_{j2} is larger than τ_1, resulting in a weak coupling, which is included. Because of the existence of this kind of situation, this strategy cannot utilize the nonzero entries of near-field matrix effectively.

To solve this problem, we produce a new octree (TREE), whose length of finest boxes is L_{SAI}, to assemble the SAI preconditioner. In other words, the matrix $\bar{Z}$ preparing for constructing M is extracted from Z_N through the so-called near-field matrix of the new octree. As shown in Figure 1, in the TREE, the box $\mathrm{B_{II}}$ containing X_{j1} is one of the neighbor boxes of the box $\mathrm{B_I}$ containing X_k, thus, the interaction between X_{j1} and X_k will be included in $\bar{Z}$ through the near interaction between $\mathrm{B_I}$ and $\mathrm{B_{II}}$ — actually, for all of the interactions of any two basis functions, if the distance between the two basis functions is smaller than L_{SAI}, the interaction between them will definitely be included. On the contrary, since X_{j2} does not belong to any neighbor box of $\mathrm{B_I}$, the interaction between X_k and X_{j2} will be eliminated when generating $\bar{Z}$.

The size of the smallest boxes in TREE, L_{SAI}, is a user-defined parameter that can be tuned to control the number of nonzero entries of M, that is, the sparse pattern. If we still use N_b to donate the number of non-empty boxes in the TREE, apparently, a smaller L_{SAI} leads to a large N_b. However, according to the computational complexity $O\left(N_bn_1n_2^2\right)$ during constructing the SAI preconditioner, the number of non-empty boxes N_b affects the constructing time far less than n_1 and n_2. Besides, the

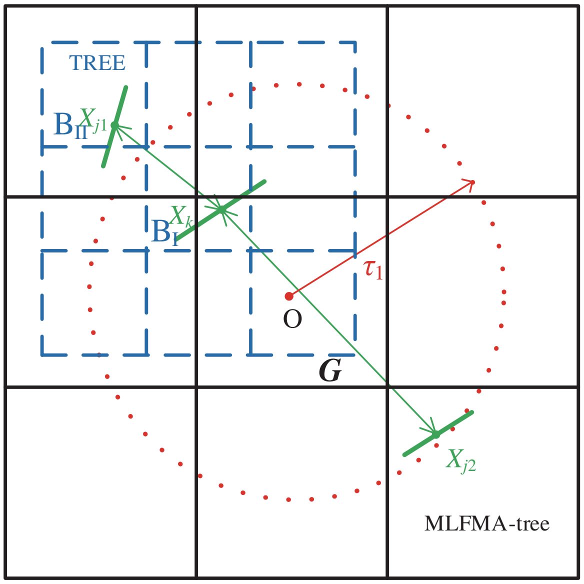


Figure 1. The locations of X_k, X_{j1} and X_{j2} in the octree of MLFMA and TREE.

proposed strategy exhibits some advantages over the conventional strategies as follows:

1) User friendly. It has only one variable parameter, i.e., L_{SAI}, which greatly simplifies the process of tuning an appropriate parameter for a certain problem. Furthermore, the value range of L_{SAI} is relatively narrow. In general, it is from 0.125λ (λ is the wavelength in free space) to the size of the finest boxes in MLFMA. According to a large number of numerical experiments, L_{SAI} is recommended to be two times of the average mesh size.
2) Better approximation of Z_N. All of the strong coupling of any two basis functions, the distance between which is smaller than L_{SAI}, will definitely be included by $\bar{Z}$.
3) Simple implementation. The proposed strategy keeps the basic characteristics of the structure of original SAI preconditioner. Thus, the existing code does not need to make major change.
4) Slight setup time for constructing M. Compared to the conventional strategies, $\bar{Z}$ can be extracted from Z_N directly without any comparison about the magnitude or position of nonzero entries. The proposed strategy needs slightly more setup time to generate the TREE. However, this time increase is very limited because of the utilization of the fast binary-tree searching algorithm.

3. NUMERICAL RESULTS

To investigate the effectiveness of the proposed strategy, several perfectly electric conductor (PEC) objects involving scattering or radiation problems are simulated respectively. In the implementation, RWG basis functions [12] are used to model the equivalent surface currents, and the restarted GMRES [13] is used as the iterative solver to reach the 0.001 convergence criterion, while the restarted number is fixed to 100, and maximum iterations is limited to 1000. The finest box size of MLFMA is fixed to 0.25λ. All experiments are performed on a workstation with 3.2 GHz CPU in single precision.

3.1. Monopoles System

This PEC object consists of two $\lambda/4$ monopoles mounted over a complex platform as shown in Figure 2. The ill-conditioned electric field integral equation (EFIE) is applied to model PEC surfaces, discretizing into 363,198 triangles resulting in 544,500 unknowns at 300 MHz while the discretized mesh size is 0.08λ. The radiation patterns of two monopoles in *xoz* and *yoz* planes are simulated. For comparison, the proposed selection strategy (*prop*) and the strategy in [3] (*conv*) for SAI with different variable parameters are studies. Table 1 shows the detailed performance, where $Tset$, TSAI, and SAI-Mem

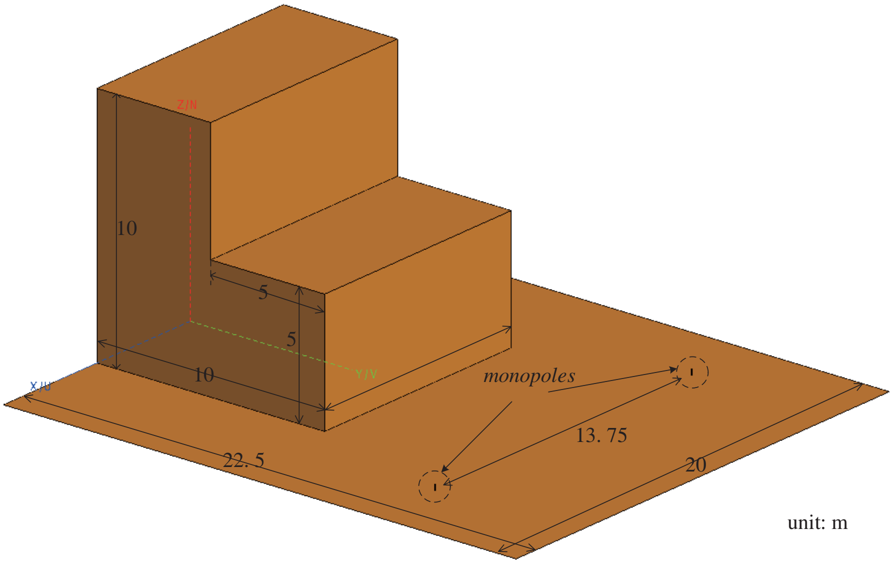
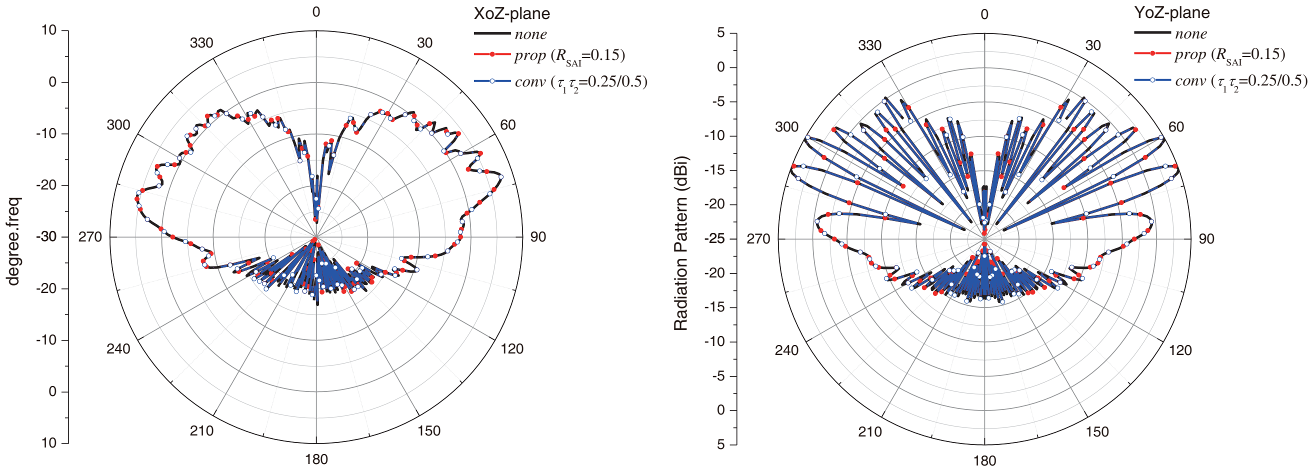


Figure 2. Two $\lambda/4$ monopoles attached to a complex PEC structure.

Table 1. Comparison of the details of different strategies and parameters on the SAI preconditioner.

SAI style	Parameter	Peak memory (MB)	Total time (s)	iterations	SAI-Mem (MB)	T_{SAI} (s)	N_b	Δ	Non-zero entries
none	——	3260	3636.16	984	—				
prop (L_{SAI})	0.125	3818	1791	172	553.8	86.49	57604	9.5	47,358,060
	0.15	4057	1923	158	792.9	242.2	40306	13.5	68,406,112
	0.2	4660	4535	1000*	1394.5	649.9	22804	23.9	121,065,072
	0.25	5447	5187	1000*	2179.9	1257	14402	37.8	189,713,182
conv (τ_1/τ_2)	0.2/0.4	3575	3875	1000*	317.2	154.3	—		26,985,129
	0.25/0.5	3854	2376	569	589.8	201.2			50,794,601
	0.3/0.5	4195	4136	1000*	921.3	322.6			79,695,191

denote the setup time, constructing time, and memory usage of SAI, respectively; Δ denotes the average number of basis functions per non-empty box in TREE; and * denotes no convergence after maximum iterations. It is observed that both *conv* and *prop* with appropriate parameters can reduce the iteration number, while *prop* is much more effective than *conv*. Besides, under the condition of the similar SAI-Mem, e.g., *prop* with LSAI=0.125λ vs. *conv* with $\tau_1/\tau_2 = 0.25/0.5$, *prop* exhibits much better performance than *conv* in both the constructing time and iterations. This validates the high efficiency of the proposed strategy. In addition, we find that a larger LSAI even slows down the convergence. It might be because when LSAI becomes larger, $\bar{Z}$ will become denser, leading to the increase of the number of nonzero entries with tiny magnitude. This may have negative effects on the performance of SAI, which demonstrates that enlarging the number of nonzero entries in M does not always lead to better performance. Underlying reasons are expected for further research. The radiation patterns of the monopoles system are shown in Figure 3, where $L_{\mathrm{SAI}} = 0.15\lambda$ for *prop*, and $\tau_1/\tau_2 = 0.25/0.5$ for *conv*. Excellent agreement between the results from different implementations is observed, which convincingly illustrates that the application of the proposed strategy has no negative impact on the numerical stability, maintaining computational accuracy of original matrix equation.



(a) *xoz* plane

(b) *yoz* plane

Figure 3. Radiation pattern of the monopoles system.

3.2. PEC Sphere

The bistatic RCS of a PEC sphere in diameter of 10λ is computed. A suitable mesh is adopted which results in 96,702 triangles and 145,053 unknowns, respectively. EFIE with poor condition and well-conditioned combined field integral equation (CFIE) are adopted to disperse the PEC surface, respectively. The computed results are depicted in Figure 4, and the result from Mie series is also presented for comparison. Excellent agreement between the numerical results and Mie series is observed. The detailed calculated information is given in Table 2. From it, we can find that the SAI preconditioner is effective for both EFIE and CFIE. However, because CFIE is well conditioned which has an excellent convergence property, the usage of SAI even increases the total time and memory usage. This phenomenon illustrates that preconditioners are not always needed by all kinds of calculation, especially when the type of integral equation modeling the calculated object is well conditioned. However, for EFIE, the SAI preconditioner reduces the number of iterations and total cost-time sharply. On the other hand, compared to L_{SAI} =0.2λ or 0.25λ situation, despite that L_{SAI} =0.15λ shows a little more iterations number, it has the lowest total time and peak memory usage. It states that a suitable L_{SAI} is necessary for a high efficiency, while a bigger value of L_{SAI} is not always better.

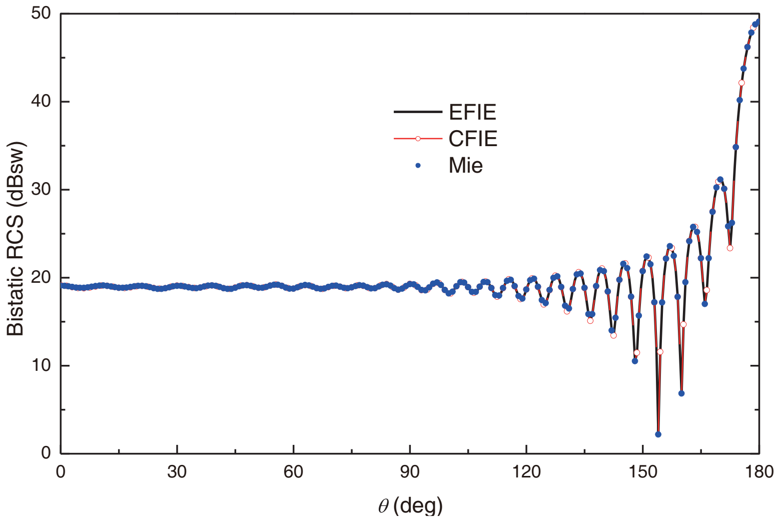


Figure 4. Bistatic RCS of a conducting sphere of diameter 10λ.

Table 2. Details with different value of L_{SAI} for the PEC sphere of diameter 10λ.

Integral equation	SAI type	L_{SAI} (λ)	SAI-Mem (MB)	T_{SAI} (sec)	Peak memory (MB)	Total time (min)	iterations
EFIE	none	——			788.7	15.3	691
	prop	0.15	117.2	32.2	906.8	5.8	249
		0.2	208.6	61.6	997.9	6.2	221
		0.25	325.5	180	1116	8.5	203
CFIE	none	——			788.7	1.1	29
	prop	0.15	117.2	32.2	906.8	1.6	15
		0.2	208.6	61.6	997.9	2.5	14
		0.25	325.5	180	1116	5.1	13

3.3. Complex Vessel

As shown in Figure 5, a complex vessel with length of 162.2 m, which contains lots of fine and open structures, is placed across x-axis. The frequency of incident plane wave is 300 MHz. A moderate mesh size is chosen to generate totally 2,836,938 unknowns with respect to 1,891,288 triangles. EFIE is applied to model PEC surfaces. This vessel is illuminated by a plane wave across $-x$-axis, and the observation range is $0^\circ \leq\varphi\leq 360^\circ$ and $\theta = 90^\circ$. This calculation is executed using OpenMP parallelization with 16 cores. In addition, because SAI can be naturally parallelized, serial and parallel codes for constructing SAI are compared. The parallel efficiency η is defined as

$$\eta = \frac{T_{ser}}{np \times T_{par}}\% \tag{8}$$

where np is the number of cores used in the parallelization. In this experiment, $np = 16$. The detailed information is shown in Table 3, while the serially and parallelism constructing time for SAI are denoted by T_{ser} and T_{par}, respectively, and the total time means the whole solution time applying the OpenMP parallelization. It is observed that SAI shows very good performance on both accelerating convergence and parallel computation. Besides, the residual error of iterative solver cannot reach convergence criterion after 1000 iterations without any preconditioner, while the residual error is also presented in the table. On the contrary, when SAI preconditioner is applied with suitable L_{SAI}, the iterative solution can be converged. Compared to SAI with $L_{\text{SAI}} = 0.25\lambda$, $L_{\text{SAI}} = 0.2\lambda$ shows a similar number of iterations, while the constructing time and peak memory usage are much lower to varying degrees. The bistatic RCSs of the vessel with different polarization modes are shown in Figure 6.

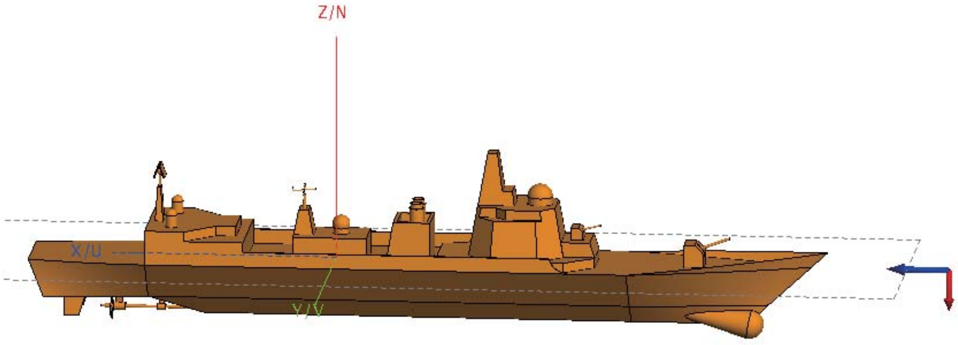


Figure 5. Model of a complex vessel.

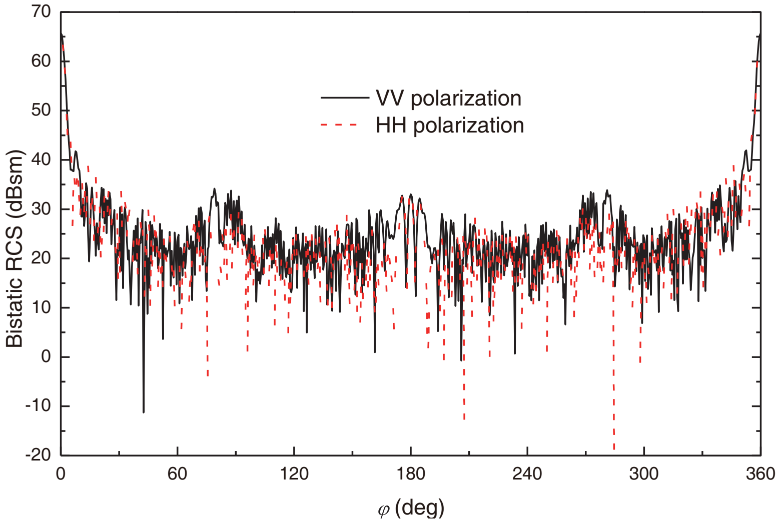


Figure 6. Bistatic RCS of a complex vessel.

Table 3. Details with different value of L_{SAI} for the complex vessel.

SAI type	L_{SAI} (λ)	SAI-Mem (GB)	T_{ser} (sec)	T_{par} (sec)	η	Peak memory (GB)	Total time (min)	iterations
none	——					13.8	92.8	1000*, 0.17
prop	0.15	1.29	1830	133	86%	15.1	95.7	1000*, 0.005
	0.2	2.29	4329	21.3	89%	16.2	763	757
	0.25	383	20981	99.5	91%	17.9	947	739

4. CONCLUSIONS

A new sparse pattern selecting strategy with only one tuning parameter for SAI preconditioner based on the geometric information is proposed by using a new octree. With effective utilization of near-field matrix in MLFMA, the proposed select strategy exhibits a significant reduction in the iteration number and total solution time demonstrated by numerical results. Besides, the one tuning parameter can be automatically set according to the average mesh size.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61701447 and Grant 62071436, and in part by the Key Laboratory of All Optical Network and Advanced Telecommunication Network, Ministry of Education (Beijing Jiaotong University ZG19002).

REFERENCES

1. Chew, W. C., J. M. Jin, E. Michielssen, and J. M. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House, Boston, MA, 2001.
2. Harrington, R. F., *Field Computation by Moment Methods*, MacMillan, New York, 1968.
3. Lee, J., J. Zhang, and C. C. Lu, "Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems," *Journal of Computational Physics*, Vol. 185, No. 1, 158–175, Feb. 2003.

4. Malas, T. and L. Gürel, "Incomplete LU preconditioning with the multilevel fast multipole algorithm for electromagnetic scattering," *SIAM Journal on Scientific Computing*, Vol. 29, No. 4, 1476–1494, 2007.
5. Saad, Y. and J. Zhang, "Enhanced multi-level block ILU preconditioning strategies for general sparse linear systems," *Journal of Computational and Applied Mathematics*, Vol. 130, No. 1, 99–118, 2001.
6. Lee, J., J. Zhang and C. C. Lu, "Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics," *IEEE Trans. Antennas Propag.*, Vol. 52, No. 9, 2277–2287, Sep. 2004.
7. Rui, P. L. and R. S. Chen, "An efficient sparse approximate inverse preconditioning for FMM implementation," *Microwave Opt. Technol. Lett.*, Vol. 49, No. 7, 1746–1750, Jul. 2007.
8. Carpentieri, B., I. S. Duff, and L. Giraud, "Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism," *Numer. Linear Algebra Appl.*, Vol. 7, 667–685, 2000.
9. Malas, T. and L. Gürel, "Accelerating the multilevel fast multipole algorithm with the sparse-approximate-inverse (SAI) preconditioning," *SIAM Journal on Scientific Computing*, Vol. 31, No. 3, 1968–1984, 2009.
10. Carpentieri, B., "Algebraic preconditioners for the Fast Multipole Method in electromagnetic scattering analysis from large structures: Trends and problems," *Electronic Journey of Boundary Elements*, Vol. 7, No. 1, 13–49, Mar. 2009.
11. Delgado, C., E. García, A. Somolinos, and M. F. Cátedra, "Hybrid parallelisation scheme for the application of distributed near-field sparse approximate inverse preconditioners on high-performance computing clusters," *IET Microw. Antennas Propag.*, Vol. 14, No. 4, 320–328, Mar. 2020.
12. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, No. 5, 409–418, May 1982.
13. Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia, 2003.