

A Modified Adaptive Integral Method for Analysis of Large-scale Finite Periodic Array

Mingxuan Zheng, Huiling Zhao, and Zhonghui Zhao

School of Electronics and Information
Northwestern Polytechnic University, Xi'an, Shaanxi, 710129, China
x408859786@mail.nwpu.edu.cn

Abstract — A fast algorithm based on AIM is proposed to analyze the scattering problem of the large-scale finite array. In this method, by filling zeros into the local transformation matrix, the near and far fields are isolated thoroughly to eliminate the near correction process. In the far part, a 5-level block-toeplitz matrix is employed to avoid saving the idle grids without adding artificial interfaces. In the near part, only one local cube is required to compute the local translation matrix and near impedance matrix, which can be shared by all elements. Furthermore, the block Jacobi preconditioning technique is applied to improve the convergence, and the principle of pattern multiplication is used to accelerate the calculation of the scattering pattern. Numerical results show that the proposed method can reduce not only the CPU time in filling and solving matrix but also the whole memory requirement dramatically for the large-scale finite array with large spacings.

Index Terms — Adaptive integral method, diagonal block preconditioning, large-scale finite periodic array, multilevel block-toeplitz, scattering problem.

I. INTRODUCTION

Periodic arrays [1] have been widely used in microwave engineering, antennas and metamaterials design. It is well known that the method of moment (MoM) [2] possess high accuracy, but inefficient in solving large-scale problems. In the past three decades, a lot of fast algorithms based on MoM have been proposed, such as multilevel fast multipole algorithm (MLFMA) [3], integral-equation fast Fourier transform (IE-FFT) algorithm [4,5], the adaptive integral method (AIM) [6], etc. For large-scale quasi-plane problems, these methods could reduce computational complexity from $O(N^2)$ to $O(N\log N)$, and memory requirements from $O(N^2)$ to $O(N)$. Among these methods, IE-FFT and AIM are FFT-based methods, and less dependent on the integral kernels. This feature enables them to be implemented with the standard and efficient FFT libraries available online.

The IE-FFT algorithm employs Cartesian grids for interpolating Green's functions, which has been successfully used in analyzing the large planar microstrip antenna arrays [7] and integrated circuits [8]. Different from IE-FFT, AIM is based on an "equivalent" source approximation, where the unstructured basis functions are mapped to the uniform grids by multipole expansion [6] without interpolating. This suggests that AIM is more accurate in analysis of arbitrary three-dimensional (3D) electromagnetic problems. Many hybrid methods have been proposed to solve the problems of periodic structures. For example, AIM is combined with the model-based parameter estimation [9] for the infinite periodic structures, with multiresolution sparse matrix (MR SM/AIM) [10] for large finite 2.5D antenna arrays, with the MultiLayer method [11] for the multilayer printed arrays, with Synthetic Function eXpansion (SFX) domain-decomposition approach [12] and with characteristic basis function method for aperiodic tiling-based antenna arrays [13].

Nevertheless, there are two drawbacks in AIM based hybrid methods when the finite periodic array is rather large with large spacings. For the first drawback, referring Fig. 1, there are a lot of idle grids which do not contribute to far field interactions, but still waste a lot of computational source. Consequently, the speed of matrix-vector product (MVP) is slow down. To address this problem, the circulant AIM (CAIM) [14] and subdomain AIM (SAIM) [15] are developed. However, CAIM is only efficient for quasi-cylindrical structures and SAIM requires to build complicated artificial interfaces among different subdomains, which is less efficient for large-scale arrays. The second drawback is that it will cost a lot of time to correct the nonzeros in near field correction process. Moreover, it is hard to find a suitable preconditioner [16] for the corrected matrix.

In this paper, a novel array AIM is proposed for the large-scale finite periodic array with large spacings. Different from SAIM, the proposed method avoids saving the idle grids through adopting a 5-level block-toeplitz matrix without adding artificial interfaces. In

addition, the near correction is eliminated, thus the near and far matrix can be isolated thoroughly. To the best of author's knowledge, it is seldom reported on the AIM-based technique without near correction. To further improve the convergence of iteration, the block diagonal preconditioner is applied to reduce the cost of computation and memory. The rest of paper is organized as follows. In Section II, the details of the proposed method are illustrated, including the process of filling matrix, preconditioning, solving and post computation. Section III gives some numerical simulation results to validate the accuracy and efficiency of the proposed method. Finally, the summary and conclusions are presented in Section IV.

II. FORMULATION

A. The conventional AIM

For an arbitrary 3D perfect electric conductor (PEC), the conventional MoM under Galerkin's testing procedure is applied to form a matrix equation as:

$$\mathbf{Z}\mathbf{I} = \mathbf{V}, \quad (1)$$

where \mathbf{I} is the current coefficients of basis functions and \mathbf{V} denotes the excitation. To reduce the memory requirements, the impedance matrix \mathbf{Z} is divided into near and far parts as $\mathbf{Z} = \mathbf{Z}^{near} + \mathbf{Z}^{far}$ in AIM [6], where \mathbf{Z}^{near} is a sparse matrix and \mathbf{Z}^{far} is compressed as the multiplications of several sparse matrices:

$$\mathbf{Z}^{near} = \mathbf{Z}^{MoM} - \mathbf{Z}^{AIM}, \quad (2)$$

$$\mathbf{Z}^{far} = \sum_{q=x,y,z,D} \Lambda_q \mathbf{G} \Lambda_q^T, \quad (3)$$

where \mathbf{Z}^{MoM} is the near interaction calculated by MoM, \mathbf{Z}^{AIM} is the inaccurate contribution from grids, \mathbf{G} is the block-toeplitz transformation matrix of Green's function on the auxiliary grids, and Λ_q are the sparse translation matrices. To solve equation (1) efficiently, preconditioning techniques, such as threshold-based incomplete LU (ILUT) [17], shifted symmetric successive over-relaxation (SSOR) [18] or parallel sparse approximate inverse (PSAI) [19], are generally applied to improve the convergence of iteration.

However, for large-scale finite periodic array, the conventional AIM will become less efficient. Firstly, filling \mathbf{Z}^{near} costs a lot of time especially for the calculation of \mathbf{Z}^{AIM} . It is worth noting that computing \mathbf{Z}^{AIM} with FFT is less efficient when the number of grids is quite large. Instead, \mathbf{Z}^{AIM} should be calculated according to (3) in order to improve the filling speed. Secondly, the construction of the preconditioning matrix will cost more time if \mathbf{Z}^{near} or \mathbf{Z}^{MoM} has more nonzeros. The reasons of the long time filling and preconditioning are the unseparated near and far fields. In other word, the MVP for \mathbf{Z}^{far} and \mathbf{I} brings the inaccurate values to the near part, which forces \mathbf{Z}^{MoM} need to be corrected. Finally, although each element is identical, the multipole expansion remains to be repeated many times because

the distance between the basis functions and the surrounding grids may be different for each element. Moreover, the huge number of idle grids will also influence the peak memory requirements and speed of FFT.

To overcome these problems, a modified AIM called array AIM is proposed for the finite periodic array with large spacings in the next part.

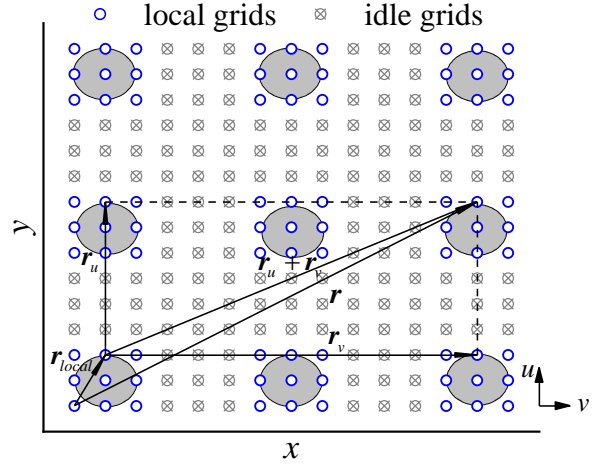


Fig. 1. Geometry of local and idle grids for an array in xy plane. The array elements are depicted in gray.

B. Array AIM for the finite periodic array

In this part, the Green's function matrix is reconstructed at first in order to avoid saving the idle grids. Then the near correction is eliminated by isolating the near and far fields. The modified MVP is given at the end of the part.

For a common 3D electromagnetic problem, \mathbf{G} is constructed as the form of three-level block-toeplitz matrix in the traditional AIM. However, for the array with large spacings as depicted in Fig. 1, \mathbf{G} has to keep many idle or unoccupied auxiliary point sources to maintain the Toeplitz property [20].

To avoid saving these idle grids, firstly each element is enclosed in a small local cube instead of a huge global cube for the whole array. Then the translation matrices Λ_i are computed just in the small local cube by the multipole expansion. Since each element is identical, Λ_i are calculated only once and shared by all elements.

Secondly, the three-level block-toeplitz \mathbf{G} used in SAIM is replaced by five-level block-toeplitz \mathbf{G}_{Array} without adding artificial surfaces. The interactions between any two elements are computed directly by (3) because they are well separated to meet the far-field condition. Based on the denotation for multilevel block-toeplitz matrix in [20], \mathbf{G}_{Array} can be expressed as $\mathbf{T}^5(N_u, N_v, N_z, N_y, N_x)$, where N_u, N_v are the number of elements at two orthogonal directions u, v in array plane

respectively, and N_x, N_y, N_z are the number of grids in x, y, z directions in the local cube. In the final level, the Green's function between two different grids is calculated as:

$$g(\mathbf{r}) = \exp(-jk|\mathbf{r}|)/4\pi|\mathbf{r}|, \mathbf{r} = \mathbf{r}_{local} + \mathbf{r}_u + \mathbf{r}_v, \quad (4)$$

where k is the wave number, \mathbf{r} denotes the distance vector measured from source point to field point, \mathbf{r}_{local} is the position vector if the two points are translated into one cube, \mathbf{r}_u and \mathbf{r}_v are the position vectors between two elements at u and v directions respectively, as illustrated in Fig. 1. It should be noted that $\mathbf{r}_{u,v}$ and \mathbf{r}_{local} may not be orthogonal.

Finally, if the far part of impedance matrix calculated by using \mathbf{G}_{Array} is expressed as \mathbf{Z}_{Array} , then its expression is:

$$\begin{aligned} \mathbf{Z}_{Array} &= \sum_{q=x,y,z,D} \Lambda'_q \mathbf{G}_{Array} \Lambda_q'^T \\ &= \sum_{q=x,y,z,D} [\text{diag}(\Lambda_{l,q})] \mathbf{G}_{Array} [\text{diag}(\Lambda_{l,q})]^T, \end{aligned} \quad (5)$$

where Λ'_q is a block diagonal matrix, in which there are $N_u N_v$ local sparse matrices Λ_l . \mathbf{G}_{Array} is saved as the form of $\tilde{\mathbf{G}}_{Array} = \text{FFT}(\mathbf{G}_{Array})$, whose length is $N_A = \prod_{p=u,v,x,y,z} (2N_p - 1)$ to meet the circular Toeplitz property. And the submatrices of \mathbf{G}_{Array} for the different elements are asymmetric three-level block-toeplitz matrices although scalar Green's function is applied.

Next, we focus on calculation of self-impedance matrix \mathbf{Z}_l for each element. To eliminate the near correction, the global and local fields should be isolated.

It is well known that the diagonal values of \mathbf{G} are set to zeros to avoid the singularity when the source and field grids are the same. From another point of view, the self-interactions from the same grids are blocked by zeros. Inspired by this point, the blocked region could be enlarged to cover the whole local cube so that the global and local fields are separated thoroughly. This could be implemented by adding a constraint condition to (4) as:

$$\text{s.t. } g(\mathbf{r}) = 0 \text{ when } \mathbf{r}_u = \mathbf{r}_v = \mathbf{0}. \quad (6)$$

For example, consider a linear array with only three elements. The array impedance matrix in (5) is rewritten as:

$$\mathbf{Z}_{Array} = \sum_{q=x,y,z,D} [\text{diag}(\Lambda_{l,q})] \begin{bmatrix} \mathbf{0} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{0} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{0} \end{bmatrix} [\text{diag}(\Lambda_{l,q})]^T, \quad (7)$$

where \mathbf{G}_{mn} denotes the transformation matrix between m and n elements. The diagonal submatrices of \mathbf{G}_{Array} are all zeros to isolate global and local regions. In this way, \mathbf{Z}^{AIM} in (2) is not needed and \mathbf{Z}^{near} is filled by MoM for the local region since usually the array element is relatively small. At last, the whole impedance matrix \mathbf{Z} is written as:

$$\mathbf{Z} = \mathbf{Z}_{self} + \mathbf{Z}_{Array} = [\text{diag}(\mathbf{Z}_l)] + \mathbf{Z}_{Array}, \quad (8)$$

where \mathbf{Z}_{self} is a diagonal matrix and \mathbf{Z}_l is a dense matrix shared by all the elements. As a result, the MVP can be computed by:

$$\begin{aligned} \mathbf{V}' &= \mathbf{Z}_{self} \mathbf{I} + \mathbf{Z}_{Array} \mathbf{I} \\ &= \begin{bmatrix} \mathbf{Z}_l \mathbf{I}_1 \\ \mathbf{Z}_l \mathbf{I}_2 \\ \vdots \\ \mathbf{Z}_l \mathbf{I}_M \end{bmatrix} + \sum_{q=x,y,z,D} [\text{diag}(\Lambda_{l,q})] \text{IFFT} \left(\tilde{\mathbf{G}}_{Array} \circ \text{FFT} \begin{bmatrix} \Lambda_{l,q}^T \mathbf{I}_1 \\ \Lambda_{l,q}^T \mathbf{I}_2 \\ \vdots \\ \Lambda_{l,q}^T \mathbf{I}_M \end{bmatrix} \right), \end{aligned} \quad (9)$$

where \circ denotes Hadamard product operator, \mathbf{I}_m is the current coefficients of the m th element, M is the number of total elements and \mathbf{V}' is the output vector of the MVP. The MVP is speeded up by 5D FFT and IFFT in this paper. From (9) it is obvious that both \mathbf{Z}_l and Λ_l are reused by each element. Hence, only one copy of \mathbf{Z}_l and Λ_l are required, which could reduce a lot of memory if the number of the array elements is quite huge.

C. Block Jacobi preconditioning

In order to improve the convergence of iteration, block Jacobi (BJ) preconditioner is applied to alleviate the ill-condition of the impedance matrix. This preconditioner is generally more robust because of the diagonal dominance of \mathbf{Z}_{self} . Thus, with the help of BJ, the MVP in (9) could be multiplied by \mathbf{Z}_{self}^{-1} as:

$$\begin{aligned} \mathbf{Z}_{self}^{-1} \mathbf{V}' &= \mathbf{Z}_{self}^{-1} (\mathbf{Z}_{self} \mathbf{I} + \mathbf{Z}_{Array} \mathbf{I}) \\ &= \mathbf{I} + \mathbf{Z}_{self}^{-1} \mathbf{V}_{Array} \\ &= \mathbf{I} + \begin{bmatrix} \mathbf{Z}_l^{-1} \mathbf{V}_{Array,1} \\ \mathbf{Z}_l^{-1} \mathbf{V}_{Array,2} \\ \vdots \\ \mathbf{Z}_l^{-1} \mathbf{V}_{Array,M} \end{bmatrix}, \end{aligned} \quad (10)$$

where $\mathbf{V}_{Array} = \mathbf{Z}_{Array} \mathbf{I}$ could be divided into M parts for each element. Since \mathbf{Z}_l in (9) is replaced by \mathbf{Z}_l^{-1} in (10) and \mathbf{Z}_l^{-1} is shared by all elements, the preconditioner doesn't consume extra memory. Moreover, all the multiplications of matrix-vector are moved to the global part without adding extra computations, which is hard to be implemented by other preconditioning technique. Furthermore, \mathbf{Z}_l^{-1} could be precomputed by inverting \mathbf{Z}_l directly if \mathbf{Z}_l is very small. Otherwise, one can also use LU decomposition on \mathbf{Z}_l and solve $\mathbf{Z}_l \mathbf{I}_m = \mathbf{V}_{Array,m}$ by direct method to get \mathbf{I}_m , where $m = 1, 2, 3 \dots M$.

D. Scattering field computation

To save the memory and time of pre-process, we just mesh one local array element as reference since all elements are the same. Thus, the total unknowns N for array is equal to $N_l \times M$, where N_l is the unknowns for the reference element. Actually, it is not necessary to

construct all basis functions for the whole array. \mathbf{Z}_l and Λ_l can be calculated from N_l local basis functions and \mathbf{G}_{Array} is independent of the basis functions. Nevertheless, ignoring the current, the contribution of each element to the scattering field is different due to its different position in the array.

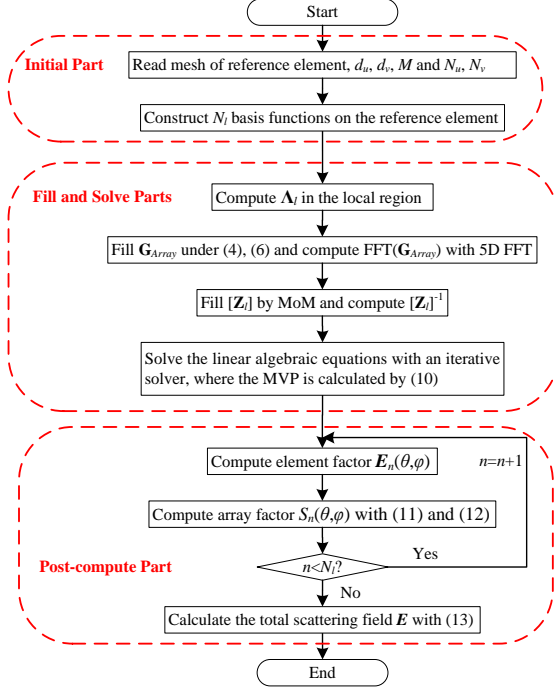


Fig. 2. Flowchart of the array AIM.

This issue could be solved easily with the help of the principle of pattern multiplication. Firstly let $\mathbf{E}_n(\theta, \varphi)$ be the scattering electric field of the n th basis functions at the angles of elevation θ and azimuth φ without multiplying the current I_n , where $n \in [1, N_l]$ is an integer. Secondly, the array factor $S_n(\theta, \varphi)$ for the n th basis functions can be evaluated by:

$$S_n(\theta, \varphi) = \sum_{m=1}^M I_{n+(m-1)N_l} \exp(-jkd_m), \quad (11)$$

$$d_m = -(\Delta N_{u,m} d_u \cos \varphi + \Delta N_{v,m} d_v \sin \varphi) \sin \theta, \quad (12)$$

where d_m is the path-difference between the m th element and the reference element. The array factor only includes the current excitations and phase factors of different elements, which is independent of the basis functions. Finally, the total scattering field at (θ, φ) can be expressed as:

$$\mathbf{E}(\theta, \varphi) = \sum_{n=1}^{N_l} \mathbf{E}_n(\theta, \varphi) S_n(\theta, \varphi), \quad (13)$$

Once the currents are calculated by the proposed method, the scattering field can be evaluated quickly via (11), (12) and (13). To better understand the whole procedure of the proposed method, the flowchart is shown in Fig. 2.

E. Memory requirements and computational complexity

In the proposed method, the memory consumptions are primarily from \mathbf{Z}_l^{-1} , Λ_l and $\tilde{\mathbf{G}}_{Array}$. Suppose that each basis function is expanded by K grids and the ratio of idle to local grids along u, v direction are c , thus the memory cost of the proposed method is:

$$16(N_l^2 + 2KN_l + N_A) \approx 16(N_l^2 + 2KN_l + 32N_u N_v N_x N_y N_z), \quad (14)$$

where the first and second terms are the memory cost of \mathbf{Z}_l^{-1} and Λ_l respectively. The last term for $\tilde{\mathbf{G}}_{Array}$ is rather larger than the others and 16 means all the matrices are saved in double precision. While the memory cost of AIM is approximated as:

$$16(2TMN_l + 2MKN_l + 8(1+c)^2 N_u N_v N_x N_y N_z), \quad (15)$$

where T is the average number of the basis functions in the near field. The first term of (15) represents the memory cost by \mathbf{Z}^{near} and the preconditioning matrix, which is much larger than \mathbf{Z}_l^{-1} since $2TM \gg N_l$ for the large-scale array. The second and third terms of (15) denote the memory requirements of Λ and $\tilde{\mathbf{G}}$. Compared with Λ_l , Λ is M times larger than Λ_l , and the length of $\tilde{\mathbf{G}}$ is determined by the unoccupied ratio c . If $c < 1$, $\tilde{\mathbf{G}}$ is less than $\tilde{\mathbf{G}}_{Array}$, but when $c > 1$ for the array with large spacings, $\tilde{\mathbf{G}}_{Array}$ is less than $\tilde{\mathbf{G}}$ since $\tilde{\mathbf{G}}$ is increased by $(1+c)^2$ times.

On the other hand, the computational complexity of \mathbf{Z}_l and \mathbf{Z}^{near} are $O(N_l^2)$ and $O(TMN_l)$ respectively. This suggests that the speed of filling \mathbf{Z}_l is much faster than filling \mathbf{Z}^{near} . The constructing time of preconditioner could also be further reduced. Moreover, it is easy to infer the time cost for calculating Λ_l is M times less than calculating Λ . The filling time of $\tilde{\mathbf{G}}_{Array}$ and $\tilde{\mathbf{G}}$ are quite short. Similarly, the improvement on the speed of MVP is determined by c because the computational complexity of $\tilde{\mathbf{G}}_{Array}$ and $\tilde{\mathbf{G}}$ are $O(N_A \log(N_A))$ and $O(N_C \log(N_C))$ respectively, where $N_C = (1+c)^2 N_A / 4$.

Therefore, for a given element, the filling time and

memory requirements of \mathbf{Z}_l^{-1} and $\mathbf{\Lambda}_l$ in the proposed method are independent of the size and interval of the array, which are much less than AIM. Significant improvements on memory cost of $\tilde{\mathbf{G}}_{Array}$ and the speed of the MVP would be obtained if c is larger than 1.

III. NUMERICAL RESULTS

To demonstrate the performance of the proposed method, the scattering of 2D arrays are considered. The arrays are supposed to be located at xoy plane. A $-x$ -polarized plane wave propagation along $-z$ direction impinges the array with frequency 0.3GHz. AIM parameters are: the grid interval is 0.05λ , the grid order is 2 and the near threshold is 0.3λ . The iterative solver is the stabilized biconjugate gradient (Bi-CGSTAB) with the relative residual error less than 10^{-4} . All the numerical experiments are performed on the desktop PC of four cores 4790 Intel processors with math kernel library to calculate FFT and IFFT.

A. The accuracy of the proposed method

The first example is a 9×9 conducting cylinder array with the array length $l_{u,v}=6.5\text{m}$ and the interval $d_{u,v}=0.7\text{m}$. The geometry parameters of PEC cylinder are the height $h=0.8\text{m}$ and diameter $D=0.2\text{m}$ as shown in left side of Fig. 3. Each cylinder is discretized by 334 triangle patches, leading to 501 basis functions, and correspondingly 40581 unknowns for the whole array. The bistatic radar cross section (RCS) are plotted in Fig. 4. Clearly, the results of the proposed method agree well with that of AIM and MLFMA (FEKO). Figure 5 gives the relative error of this example, which is defined as:

$$error = 10 \log_{10} \left(\left| \frac{\sigma_{proposed} - \sigma_{MLFMA}}{\sigma_{MLFMA}} \right| \right), \quad (16)$$

where $\sigma_{proposed}$ and σ_{MLFMA} represent the RCS pattern computed by the proposed method and MLFMA respectively. The average relative errors are about -15dB. On some angles, the relative errors exceed -10dB due to the round-off error.

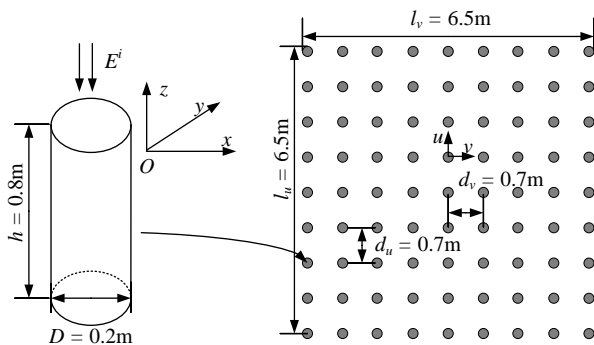


Fig. 3. Geometry and dimension of the 9×9 PEC cylinder periodic array.

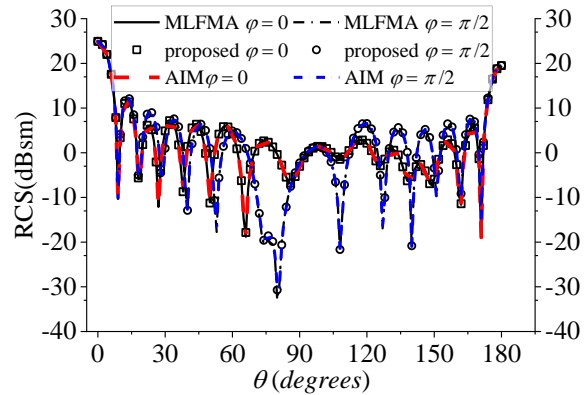


Fig. 4. Bistatic RCS pattern of the 9×9 cylinder array in the vertical plane.

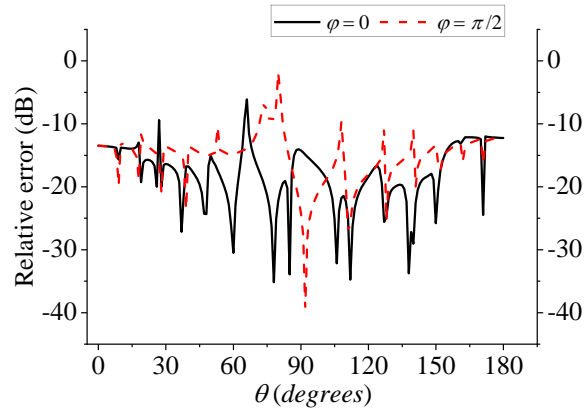


Fig. 5. The relative error of the 9×9 cylinder array in the vertical plane. (MLFMA as reference).

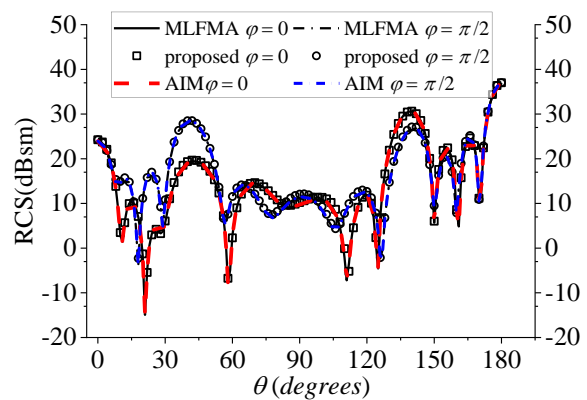


Fig. 6. Bistatic RCS pattern of the 4×4 sphere array in the vertical plane.

The second example is a 4×4 conducting sphere array. The array has length $l=5.7\text{m}$, the interval $d_{u,v}=1.5\text{m}$ and the PEC sphere radius $r=0.6\text{m}$. Incident wave is the

same as last example. The geometry of this example is omitted for the similarity with the first one. The unknowns of the sphere and the whole array are 1287 and 20592 respectively. Figure 6 demonstrates that the bistatic RCS of proposed method agrees well with that of AIM and MLFMA.

B. The performance of the proposed method for different unoccupied ratio c

To investigate the influence of the unoccupied ratio c , the interval $d_{u,v}$ of the 9×9 cylinder array above are chosen as 0.5m, 0.6m, 0.8m, 1.2m and 2m, which are corresponding to $c=0.42, 0.71, 1.28, 2.42$ and 4.7 respectively. The number of the local grids for the element is $N_x=N_y=7, N_z=19$ and the rest parameters of the

array are the same as Fig. 3.

Table 1 summarizes the performance of the proposed method and AIM in different intervals. We can see from Table 1 that the filling time and memory requirements of the proposed method are independent of c . While in AIM the memory requirement of \mathbf{G} and the per iteration time increase dramatically with the increasing of c . It can be observed that the proposed method could achieve more than 99.6% reduction in filling time.

In addition, in terms of the memory and per iteration time, AIM performs a better performance for $c<1$. But when $c>1$, the proposed method are more efficient than AIM, especially for $d=2m$, about 10 times improvements are achieved.

Table 1: CPU time and memory requirements of the proposed and conventional AIM for the 9×9 cylinder array in different intervals

Interval $d(m)$	Unoccupied Ratio c	Memory (MB)		Filling Time(s)		Each Iteration Time(s)	
		\mathbf{G}_{Array}	\mathbf{G}	Proposed	AIM	Proposed	AIM
0.5	0.42	27.57	16.25	0.465	74.53	0.544	0.47986
0.6	0.71	27.57	22.86	0.417	73.356	0.544	0.53645
0.8	1.28	27.57	39.45	0.461	75.217	0.566	1.05973
1.2	2.42	27.57	86.14	0.435	78.098	0.586	1.33586
2	4.7	27.57	233.51	0.464	78.696	0.586	7.848

Table 2: CPU time and memory requirements of the proposed and conventional AIM for the cylinder array in different sizes

Array size	Unknowns	Total Memory (MB)		Filling Time(s)		Total Time(s)	
		Proposed	AIM	Proposed	AIM	Proposed	AIM
9×9	40581	31.81	761.95	0.465	81.20	10.614	98.16
16×16	128256	95.93	2468.38	0.465	523.53	57.582	596.61
27×27	365229	272.25	8397.46	0.465	1652.4	271.09	2000.85
40×40	801600	599.71	20663.87	0.465	12404.23	3448.8	17186.4

C. The performance of the proposed method for the different array sizes

In this part, the array AIM and conventional AIM are applied to solve the scattering problem of the cylinder array with different sizes. The sizes of the array are selected as 9×9, 16×16, 27×27 and 40×40, the rest parameters of the array are the same as Fig. 3. The unknowns of the four arrays are 40581, 128256, 365229 and 801600, respectively. RCS patterns of these arrays are omitted for redundancies.

Table 2 lists the CPU time and memory requirements of the proposed method and AIM for the four cases. It can be observed from Table 2 that the proposed method could reduce more than 96% total memory requirements in comparison with the conventional AIM. The reason of the memory reduction is that the local impedance matrix and the BJ preconditioner only consume a little memory.

Furthermore, the filling time of AIM rises dramatically as the array size increases, while the proposed method holds. Thus for the largest array size 40×40, the filling speed of the proposed method is about 26000 times faster than AIM. As a result, the total solution time of the proposed method is greatly less than that of AIM. It should be pointed that the solving time of the two methods are nearly the same since interval $d=0.7m$ means $c=1$ for all the cases.

Therefore, for the large-scale array with large spacings, the array AIM is much more efficient than the conventional AIM both on computational complexity and memory requirements.

IV. CONCLUSION

In this paper, a modified AIM has been proposed for the fast analysis of the electromagnetic scattering problem of the large-scale finite array. The whole array

is divided into array and local parts. In array part, a 5-level block-toeplitz \mathbf{G}_{Array} is constructed to calculate the interactions between different elements, because \mathbf{G}_{Array} could reduce memory requirement by eliminating the idle grids. In local part, the local translation matrix Λ_l and self-impedance matrix \mathbf{Z}_l are shared by all elements to reduce the computational source. Through filling zeros into the diagonal submatrices of \mathbf{G}_{Array} , the local and array parts are isolated to eliminate the near correction. The numerical simulations have demonstrated that, without losing accuracy, the proposed method combined with block Jacobi preconditioning could not only accelerate the speed of filling matrix and iteration tremendously but also reduce the whole memory requirements dramatically, especially for the large-scale array with large spacings.

Furthermore, the proposed method is suitable for both periodic or non-periodic array. To form a non-periodic array, some periodic elements would be masked by adjusting translation matrix Λ'_q of (5). This work will be studied in the future.

REFERENCES

- [1] L. Liu and Z. Nie, "An efficient numerical model for the radiation analysis of microstrip patch antennas," *ACES Journal*, vol. 34, no. 10, p. 6, 2019.
- [2] R. F. Harrington, *Field Computation by Moment Methods*. New York, 1993.
- [3] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488-1493, 1997.
- [4] B. J. Fasenfest, F. Capolino, D. R. Wilton, D. R. Jackson, and N. J. Champagne, "A fast MoM solution for large arrays: Green's function interpolation with FFT," *IEEE Antennas Wireless Propag. Lett.*, vol. 3, no. 1, pp. 161-164, Dec. 2004.
- [5] S. M. Seo and J.-F. Lee, "A fast IE-FFT algorithm for solving PEC scattering problems," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1476-1479, 2005.
- [6] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225-1251, Sep. 1996.
- [7] S. M. Seo, "A fast IE-FFT algorithm to analyze electrically large planar microstrip antenna arrays," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 6, pp. 983-987, June 2018.
- [8] B. J. Rautio, V. I. Okhmatovski, A. C. Cangellaris, J. C. Rautio, and J. K. Lee, "The unified-FFT algorithm for fast electromagnetic analysis of planar integrated circuits printed on layered media inside a rectangular enclosure," *IEEE Trans. Microw. Theory Tech.*, vol. 62, no. 5, pp. 1112-1121, May 2014.
- [9] X. Wang, D. H. Werner, and J. P. Turpin, "Application of AIM and MBPE techniques to accelerate modeling of 3-D doubly periodic structures with nonorthogonal lattices composed of bianisotropic media," *IEEE Trans. Antennas Propagat.*, vol. 62, no. 8, pp. 4067-4080, Aug. 2014.
- [10] P. De Vita, A. Freni, F. Vipiana, P. Pirinoli, and G. Vecchi, "Fast analysis of large finite arrays with a combined multiresolution-SM/AIM approach," *IEEE Trans. Antennas Propagat.*, vol. 54, no. 12, pp. 3827-3832, Dec. 2006.
- [11] P. D. Vita, F. D. Vita, A. D. Maria, and A. Freni, "An efficient technique for the analysis of large multilayered printed arrays," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 104-107, 2009.
- [12] A. Freni, P. D. Vita, P. Pirinoli, L. Matekovits, and G. Vecchi, "Fast-factorization acceleration of MoM domain-decomposition: SFX-AIM and general," in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, pp. 276-277, 2012.
- [13] X. Wang, D. H. Werner, and J. P. Turpin, "Investigation of scattering properties of large-scale aperiodic tilings using a combination of the characteristic basis function and adaptive integral methods," *IEEE Trans. Antennas Propagat.*, vol. 61, no. 6, pp. 3149-3160, June 2013.
- [14] C. C. Ioannidi and H. T. Anastassiou, "Circulant adaptive integral method (CAIM) for electromagnetic scattering from large targets of arbitrary shape," *IEEE Trans. Magn.*, vol. 45, no. 3, pp. 1308-1311, Mar. 2009.
- [15] X. Wang, S. Zhang, Z.-L. Liu, and C.-F. Wang, "A SAIM-FAFFA method for efficient computation of electromagnetic scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 64, no. 12, pp. 5507-5512, Dec. 2016.
- [16] W.-B. Ewe, L.-W. Li, Q. Wu, and M.-S. Leong, "Preconditioners for adaptive integral method implementation," *IEEE Trans. Antennas Propagat.*, vol. 53, no. 7, pp. 2346-2350, July 2005.
- [17] M. Zhang, T. S. Yeo, and L. W. Li, "Thresholdbased incomplete LU factorization preconditioner for adaptive integral method," *Proc of 2007 Asia Pacific Microwave Conference*, Bangkok, Thailand, pp. 913-916, Dec. 11-14, 2007.
- [18] J. Q. Chen, Z. W. Liu, K. Xu, D. Z. Ding, Z. H. Fan, and R. S. Chen, "Shifted SSOR preconditioning technique for electromagnetic wave scattering problems," *Microw. Opt. Tech. Lett.*, vol. 51, no. 4, pp. 1035-1039, 2009.
- [19] M. Li, M. Chen, W. Zhuang, Z. Fan, and R. Chen, "Parallel SAI preconditioned adaptive integral

- method for analysis of large planar microstrip antennas," *ACES Journal*, vol. 25, no. 11, p. 10, 2010.
- [20] B. E. Barrowes, F. L. Teixeira, and J. A. Kong, "Fast algorithm for matrix-vector multiply of asymmetric multilevel block-Toeplitz matrices in 3-D scattering," *Microw. Opt. Technol. Lett.*, vol. 31, no. 1, pp. 28-32, 2001.