LETTER

Preconditioners for CG-FMM-FFT Implementation in EM Analysis of Large-Scale Periodic Array Antennas

Huiqing ZHAI[†], *Nonmember*, **Qiaowei YUAN**[††], **Qiang CHEN**[†a)], *Members*, *and* **Kunio SAWAYA**[†], *Fellow*

SUMMARY In this research, a sub-array preconditioner is applied to improve the convergence of conjugate gradient (CG) iterative solver in the fast multipole method and fast Fourier transform (FMM-FFT) implementation on a large-scale finite periodic array antenna with arbitrary geometry elements. The performance of the sub-array preconditioner is compared with the near-group preconditioner in the array antenna analysis. It is found that the near-group preconditioner achieves a little better convergence, while the sub-array preconditioner can be easily constructed and programmed with less CPU-time. The efficiency of the CG-FMM-FFT with high efficient preconditioner has been demonstrated in numerical analysis of a finite periodic array antenna.

key words: *large-scale periodic array antenna, method of moments (MoM), fast multipole method (FMM), fast Fourier transform (FFT), preconditioner*

1. Introduction

An efficient analysis approach is required to evaluate electromagnetic performance of a large-scale finite periodic array antenna in the advanced communication systems.

The method of moments (MoM) is one of the effective methods to analyze the electromagnetic characteristics of antennas. However, when the iterative method such as the conjugate-gradient (CG) method is used to solve the dense matrix equation appearing in the MoM analysis, $O(N^2)$ CPU time and $O(N^2)$ memory is required at least, where N is the number of unknowns. Therefore, acceleration of the iteration process is necessary in order to solve the large-scale array antennas.

The conjugate-gradient method with fast Fourier transform (CG-FFT) is very effective in dealing with the uniform rectangular array because its computational complexity can be reduced to $O(N \log N)$ [1]. However, it is difficult to apply the CG-FFT method to the array antenna whose array elements have an arbitrary structure [2], [4]. The fast multipole method (FMM) [5]–[8] is also a very effective technique in reducing memory requirement and computational complexity, and it has been widely used in the scattering problems of conducting objects with complicated structures of electrically large scale. In this research, CG-FMM-FFT is applied to a large-scale finite periodic array antenna composed of array elements with arbitrary geometry. It has been demonstrated that the memory requirement and computational complexity can be reduced to $O(N)$ and $O(N \log N)$, respectively. Together with the efforts of reducing computational complexity, the sub-array preconditioner is firstly introduced for improving iterative convergence to further reduce the CPU time. Furthermore, the performance of the sub-array preconditioner is also compared with the near-group preconditioner in the CG-FMM-FFT analysis of a large-scale finite periodic array antenna.

Manuscript received July 4, 2006.
Manuscript revised October 30, 2006.
[†]The authors are with the Department of Electrical Communications, Faculty of Engineering, Tohoku University, Sendai-shi, 980-8579 Japan.
[††]The author is with Intelligent Cosmos Research Institute (ICR), Sendai-shi, 989-3204 Japan.
a) E-mail: chenq@ecei.tohoku.ac.jp
DOI: 10.1093/ietcom/e90–b.3.707

2. CG-FMM-FFT Formulation

The FMM is an effective method in saving computational resources because it can accelerate the matrix-vector multiplication which consumes most of the CPU time in the iterative solver, such as the CG iterative method. In the FMM, the unknown current segments in the sub-domain MoM procedure are divided into far groups and near groups. A large boundary of near groups results in a high accuracy, but causes a requirement of a large amount of the computer memory and CPU time. In this research, the neighboring groups are chosen as the near groups and non-neighboring groups are chosen as the far groups. The multiplication of near group impedance matrix $[Z^{near}]$ with current vectors is calculated without approximation, while the multiplication of far group impedance matrix $[Z^{far}]$ with current vector is achieved by using the fast multipole method.

According to the multipole expansion of the Green's function and the numerical evaluation of the spectral integral [5], the mutual impedance between n segment in m group and n' segment in m' group, is expressed by

$$Z_{nn'}^{far} \approx \frac{-jk}{4\pi} \sum_{L_\theta=1}^{L} \sum_{L_\phi=1}^{2L} \frac{\pi}{L} W_{L_\theta} \Big[S_{nm}(\hat{k}) \cdot T_L(kr_{mm'}, \hat{k} \cdot \hat{r}_{mm'}) S^*_{n'm'}(\hat{k}) \Big], \quad (1)$$

where, *translation* function $T_L(kr_{mm'}, \hat{k} \cdot \hat{r}_{mm'})$, *signature* functions $S_{nm}(\hat{k})$ and $S_{n'm'}(\hat{k})$ are given in [5]. W_{L_θ} and π/L denote the associated integral weights of Gaussian quadrature and trapezoidal rule, respectively. The multiplication of the far zone impedance matrix with a current vector can be completed according to the three procedures of *aggregation*, *translation* and *disaggregation* [5]–[8].

Provided an $N_x \times N_y$ uniform rectangular array antenna is considered. The array is divided into $N_x \times N_y$ groups and

Copyright © 2007 The Institute of Electronics, Information and Communication Engineers

each group just includes one array element that is expanded by K basis functions. The analysis model has the number of groups of $M = N_x \times N_y$, and number of whole unknowns of $N = M \times K$.

Because the *signature* function is independent of the group number, $S^*_{n'm'}$ can be denoted by $\xi^*_{n'}$ and S_{nm} can be denoted by ξ_n. Based on the far zone impedance matrix element of Eq. (1), the voltage at segment n in group m can be expressed in the form of a matrix equation as

$$\begin{bmatrix} V_{1n} \\ V_{2n} \\ V_{3n} \\ \vdots \\ V_{Mn} \end{bmatrix} = \frac{-jk}{4\pi} \sum_{L_\theta=1}^{L} \sum_{L_\varphi=1}^{2L} \frac{\pi}{L} W_{L_\theta} \cdot \begin{bmatrix} T_L^{(1,1)} & T_L^{(1,2)} & T_L^{(1,3)} & \cdots & T_L^{(1,M)} \\ T_L^{(2,1)} & T_L^{(2,2)} & T_L^{(2,3)} & \cdots & T_L^{(2,M)} \\ T_L^{(3,1)} & T_L^{(3,2)} & T_L^{(3,3)} & \cdots & T_L^{(3,M)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_L^{(M,1)} & T_L^{(M,2)} & T_L^{(M,3)} & \cdots & T_L^{(M,M)} \end{bmatrix} \cdot \left\{ \xi_1^* \begin{bmatrix} I_{11} \\ I_{21} \\ I_{31} \\ \vdots \\ I_{M1} \end{bmatrix} + \xi_2^* \begin{bmatrix} I_{12} \\ I_{22} \\ I_{32} \\ \vdots \\ I_{M2} \end{bmatrix} + \cdots + \xi_K^* \begin{bmatrix} I_{1K} \\ I_{2K} \\ I_{3K} \\ \vdots \\ I_{MK} \end{bmatrix} \right\} \xi_n, \quad (2)$$

where $n = 1, \cdots, K$. Because the group centers are periodically located structure, the *translation* function $T_L^{(m,m')}$ between group m and group m' is a two-dimensional (2-D) Toeplitz matrix. Therefore, 2-D FFT can be employed to speed up the multiplication of $T_L^{(m,m')}$ with current vector $[I_{(1,i)}, I_{(2,i)}, I_{(3,i)}, \cdots, I_{(M,i)}], i \in [1, K]$ to further reduce the memory requirement and CPU time of each iterative step. Because of the FFT operation, the memory requirement and computational complexity for each evaluation of the matrix-vector multiplication can be reduced to $O(N)$ and $O(N \log N)$, respectively.

3. Preconditioner

In order to effectively reduce the whole iterative steps of the iterative method, it is required to find a preconditioner matrix $[P]$ which makes the number of the iterative steps for solving $[P][Z][I] = [P][V]$ less than that required for the original matrix equation $[Z][I] = [V]$. In this research, the sub-array preconditioner is applied to the CG-FMM-FFT implementation for analysis of a periodic array antenna composed of array elements with arbitrary geometry.

The sparse sub-array preconditioner, which has been originally used in the iterative method based on the Gauss-Seidel scheme [3] and has also been used in the CG-FFT method [4] for analysis of array atennas, is an inversed matrix of the impedance matrix, i.e. $P = [Z_1]^{-1}$. $[Z_1]$ is expressed by

$$[Z_1] = \begin{bmatrix} [Z_s] & 0 & 0 & \cdots & 0 \\ 0 & [Z_s] & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & [Z_s] & 0 \\ 0 & 0 & \cdots & 0 & [Z_s] \end{bmatrix}, \quad (3)$$

where $[Z_s]$ denotes $K \times K$ impedance matrix in one sub-array with K unknowns. Since all sub-arrays are the same and the number of unknowns in one sub-array is much smaller than the total number of unknowns, the implementation of the preconditioner costs little extra memory and CPU time.

As a comparison, the second preconditioner, called near-group preconditioner [9], is constructed by using near group impedance matrix $[Z^{near}]$ instead of $[Z_1]$. $[Z^{near}]$ can be generally expressed by

$$[Z^{near}] = \begin{bmatrix} [G_1] & [G_2] & 0 & \cdots & 0 \\ [G_2] & [G_1] & [G_2] & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & [G_2] & [G_1] & [G_2] \\ 0 & \cdots & 0 & [G_2] & [G_1] \end{bmatrix}, \quad (4)$$

where

$$[G_1] = \begin{bmatrix} [Z_s] & [Z_{1,2}] & 0 & \cdots & 0 \\ [Z_{2,1}] & [Z_s] & [Z_{2,3}] & 0 & \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & & & \\ 0 & \cdots & 0 & [Z_{N_x,N_x-1}] & [Z_s] \end{bmatrix}, \quad (5)$$

and

$$[G_2] = \begin{bmatrix} [Z_{1,N_x+1}] & 0 & \cdots & 0 \\ 0 & [Z_{2,N_x+2}] & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & [Z_{N_x,2N_x}] \end{bmatrix}. \quad (6)$$

The size of $[G_1]$ or $[G_2]$ is $(N_xK) \times (N_xK)$, the size of $[Z_{i,j}]$ is $K \times K$, and the size of $[Z^{near}]$ is $N \times N = (N_xN_yK) \times (N_xN_yK)$. Because $[Z^{near}]$ includes the mutual impedance in the "near group" which is composed of one sub-array and the neighboring sub-arrays, the number of non-zero elements in $[Z^{near}]$ is greater than that in $[Z_1]$, which means more electromagnetic mutual coupling is considered in the near-group preconditioner at the cost of enlarging the non-zero entries.

However, it is not practical to evaluate the inverse matrix of $[Z^{near}]$ directly because $[Z^{near}]$ has the same size of $[Z]$ and the non-zero elements in $[Z^{near}]$ are greater than those in $[Z_1]$. In [9], the incomplete lower-upper (ILU) decomposition method is applied to $[Z^{near}]$ and the preconditioner is supposed to be $P = ([L_1][U_1])^{-1}$, where $[L_1]$ is the lower triangular matrix and $[U_1]$ is the upper triangular matrix of (ILU) decomposition of $[Z^{near}]$. Because $[Z^{near}]$ is a sparse matrix, the ILU decomposition does not cost much

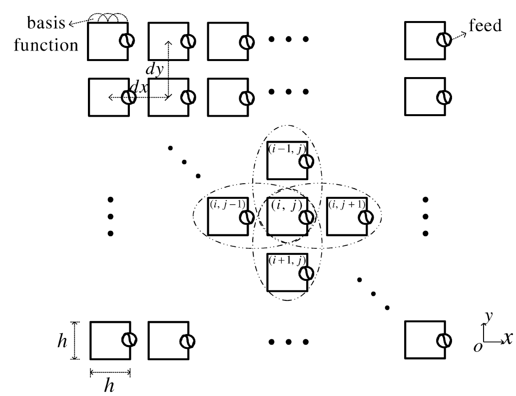

Fig. 1 Geometry of rectangular loop array.

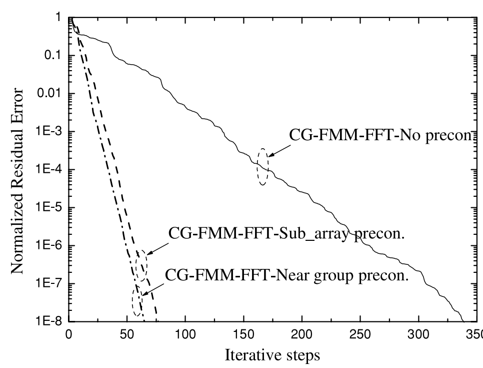

Fig. 2 Convergence comparison of different methods.

CPU time. Therefore, the near group preconditioner is efficiently evaluated by $P = ([L_1][U_1])^{-1}$ in the present research. In the next section, comparison of performance between the near group preconditioner and the sub-array preconditioner to CG-FMM-FFT is carried out.

4. Numerical Results

An $N_x \times N_y = 10 \times 10$ array composed of thin-wire rectangular loop elements is analyzed by using the CG-FMM-FFT with the two preconditioners. The geometry of the array is shown in Fig. 1, where $d_x = d_y = 0.75\lambda$ and $h = 0.5\lambda$. The radius of wire is $2.5 \times 10^{-4}\lambda$. All the array elements are fed by voltage with the same amplitude and phase.

In the analysis, the rectangular loop element is divided into 16 sinusoidal segments [10]. The corresponding source position for each element is also denoted in Fig. 1. The normalized residual error of each iterative step for the sub-array preconditioner and the near-group preconditioner is given in Fig. 2. For the sub-array preconditioner, each array element looks like a sub-array. For the near-group preconditioner, the self-impedance of one sub-array and the mutual-impedance between the neighboring sub-arrays are included in $[Z^{near}]$. As shown in Fig. 1, a sub-array in position (i, j) has neighboring sub-arrays in positions of $(i, j-1), (i, j+1), (i-1, j)$ and $(i+1, j)$.

It is found that both the CG-FMM-FFT with the sub-array preconditioner and that with the near-group preconditioner can reduce the whole iterative steps greatly, as shown in Fig. 2.

Table 1 Required time for different preconditioners. (Pentium-IV 2.6 GHz PC/1.5G Byte memory) [sec]

Methods	Construct time	Time/step	Total steps	Total time
No precon.	0	0.45	339	152.55
Sub-array	0.02	0.48	77	36.98
Near-group	0.89	0.49	65	33.39

Table 1 gives the required CPU time for both preconditioners in detail.

From Table 1, it is also concluded that the use of preconditioners can reduce the total CPU time greatly but the CPU time saving effect of these two preconditions is not large. The two preconditioners have their own features. The sub-array preconditioner needs more iterative steps but costs less CPU time for the preconditioner construction, while the near-group preconditioner needs less iterative steps at the cost of more CPU time for constructing itself.

5. Conclusions

In this research, a preconditioner called the sub-array preconditioner has been used to improve the convergence of iterative solvers in the CG-FMM-FFT implementation to analyze a large-scale finite periodic array antenna composed of array elements with arbitrary geometry. The performance of the sub-array preconditioners has been compared with the near-group preconditioner in the array antenna analysis. The two preconditioners can reduce the total CPU time greatly but the CPU time saving effect of these two preconditions is not large. The near-group preconditioner achieves a little better convergence, while the sub-array preconditioner needs less CPU-time for constructing itself.

Acknowledgement

This research was partly supported by the Japan Aerospace Exploration Agency (JAXA) and the program of the JSPS Postdoctoral Fellowship for Foreign Researchers.

References

[1] T.K. Sarkar, E. Arvas, and S.M. Rao, "Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies," IEEE Trans. Antennas Propag., vol.34, no.5, pp.635–640, May 1986.

[2] W.B. Lu, T.J. Cui, X.X. Yin, Z.G. Qian, and W. Hong, "Fast algorithms for large-scale periodic structures using subentire domain basis functions," IEEE Trans. Antennas Propag., vol.53, no.3, pp.1154–1162, March 2005.

[3] Q. Chen, Q. Yuan, and K. Sawaya, "Fast algorithm for solving matrix equation in MoM analysis of large-scale array antennas," IEICE Trans. Commun., vol.E85-B, no.11, pp.2482–2488, Nov. 2002.

[4] H. Zhai, Q. Chen, Q. Yuan, K. Sawaya, and C.H. Liang, "Analysis of large-scale periodic array antennas by CG-FFT combined with equivalent sub-array preconditioner," IEICE Trans. Commun., vol.E89-B, no.3, pp.922–928, March 2006.

[5] R. Coifman, V. Rokhlin, and S. Wandzura, “The fast multipole method for the wave equation: A pedestrian prescription,” IEEE Antennas Propag. Mag., vol.35, no.3, pp.7–12, June 1993.
[6] J.M. Song and W.C. Chew, “Multilevel fast-mutipole algorithm for solving combined field integral equations of electromagnetic scattering,” Microwave Opt. Technol. Lett., vol.10, no.1, pp.14–19, Sept. 1995.
[7] R.L. Wagner, J.M. Song, and W.C. Chew, “Monte Carlo simulation of electromagnetic scattering from two-dimensional random rough surface,” IEEE Trans. Antennas Propag., vol.45, no.2, pp.235–245, Feb. 1997.
[8] H. Zhai, Q. Yuan, Q. Chen, and K. Sawaya, “A numerical study on large-scale periodic array antenna by FMM and FFT,” IEEE Antennas and Propagat. Soc. Symp., pp.4035–4038, July 2006.
[9] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing, Boston, MA, 1996.
[10] J.H. Richmond and N.H. Greay, “Mutual impedance of nonplanar-skew sinusoidal dipoles,” IEEE Trans. Antennas Propag., vol.23, no.5, pp.412–414, May 1975.