Efficient Preconditioners for Galerkin Fast Multipole Boundary Element Method for 3D Electrostatic Field

Yuxin Shi, Zezhong Wang

Key Laboratory of High Voltage & EMC, Beijing, 102206, China, xin_yushi@126.com

The fast multipole boundary element method (FMBEM) which is suitable for solving large-scale problems can accelerate computation speed and save memory. The preconditioner which can improves matrix characteristics is a critical part to develop an efficient FMBEM. A new preconditioner based on sparse approximate inverse (SAI) is proposed for the Galerkin FMBEM of electrostatic field. The left-preconditioned generalized minimal residual (GMRES) and right-preconditioned GMRES are compared, and the effect of different sparsity patterns on the preconditioning is studied. Numerical results of different models show that the preconditioner can achieve quickly convergence. The algorithm of FMBEM used this preconditioner can be used for the electrostatic field calculation of the converter valve shielding system in DC transmission system.

***Index Terms*—Boundary element method, electrostatic field, fast multipole method, preconditioner.**

I. Introduction

THE PRECONDITONER is used to improve the matrix characteristics by multiplying the coefficient matrix by right or left preconditioning matrix[1]. The method to get an efficient preconditioning matrix is as follow. Firstly, find a temporary sparsified matrix $\boldsymbol{A}'$ as a good approximation of original matrix $\boldsymbol{A}$. Therefore, preconditioning matrix $\boldsymbol{M}$ is determined to approximate $\boldsymbol{A}'$. There are two kinds of sparsity patterns: based on adjacent nodes [2] and based on leaves for the fast multipole method (FMM) [3]. The sparse approximate inverse (SAI) is the main method to calculate preconditioning matrix [1].The sparsity pattern of $\boldsymbol{A}'$ and $\boldsymbol{M}$ sparse are the same [4]-[5]. For the Galerkin fast multipole boundary element method (GFMBEM) of electrostatic field, A new preconditioner which the sparsity pattern of $\boldsymbol{A}'$ and $\boldsymbol{M}$ are different is proposed. The matrix of direct integration in FMM is selected as $\boldsymbol{A}'$, and is stored using the storage scheme for sparse matrix.

II. Theory

The indirect BEM of electrostatic field is based on coulomb's law. Generating mesh which the element is linear triangular and applying Galerkin weighted residual method, the BEM equation can be obtained as (1).

$$\sum_{e}^{m_{\text{ele}}}\sum_{e'}^{m_{\text{ele}}}\sum_{j}^{n_e}\sum_{i}^{n_{e'}}\iint_{S_e} N_{e,j}\iint_{S_{e'}}\frac{N_{e',i}}{4\pi R}\cdot\frac{\sigma_{e',i}}{\varepsilon_0}dS'dS = \sum_{e}^{m_{\text{ele}}}\sum_{j}^{n_e}\sum_{i}^{n_{e'}}\iint_{S_e} N_{e,j}N_{e,i}\cdot u_{e,i}dS \tag{1}$$

The unknown surface charge density is σ, and the known voltage is u. In (1), R represents the distance between source and field points, ε_0 is the vacuum permittivity, e *and* e' respectively represent field and source element indices, m_{ele} represents the amount of element, $N_{e,i}$ represents the interpolation function of the ith node at element e. Let σ/ε_0 be the unknown variable, denoted by λ. The matrix equation can be expressed as (2).

$$(\boldsymbol{A}_{\text{near}}+\boldsymbol{A}_{\text{far}})\boldsymbol{\lambda}=\boldsymbol{B}\boldsymbol{u} \tag{2}$$

$\boldsymbol{A}_{\text{near}}$ represents coefficient matrix calculated by direct integration, and is sparsified. So $\boldsymbol{A}_{\text{near}}$ can be stored by storage scheme for sparse matrix. Then $\boldsymbol{A}_{\text{near}}$ is selected as $\boldsymbol{A}'$ to calculate $\boldsymbol{M}$. $\boldsymbol{A}_{\text{far}}\cdot\boldsymbol{\lambda}$ is quickly calculated by FMM [6].

The solver is selected restarted GMRES(m). The left preconditioner is shown in (2).

$$\mathbf{M}\cdot\mathbf{A}\boldsymbol{\lambda}=\mathbf{M}\cdot\mathbf{B}\boldsymbol{u} \tag{2}$$

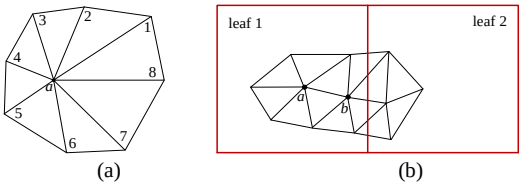
And, right preconditioner is shown in (3).

$$\mathbf{A}\cdot\mathbf{M}\boldsymbol{x}=\mathbf{B}\boldsymbol{u}\ ,\ \boldsymbol{\lambda}=\mathbf{M}\boldsymbol{x} \tag{3}$$

All residual vectors and their norms that are computed by left preconditioner correspond to $\boldsymbol{z}_m=\boldsymbol{M}(\boldsymbol{B}\boldsymbol{u}-\boldsymbol{A}\boldsymbol{\lambda}_m)$, instead of the unpreconditioned residuals $\boldsymbol{z}_m=\boldsymbol{B}\boldsymbol{u}-\boldsymbol{A}\boldsymbol{\lambda}_m$. However the right preconditioner do not change the residuals norm. In order to compare these two method, the relative residual norm is used as the stopping criterion. There are different neighbor definitions based on different pattern.

The first pattern is based on adjacent nodes. Two nodes are called neighbors if they are in one element. Each node is also a neighbor of itself. As shown in Fig. 1(a), node a have 8 neighbors. The preconditioning matrix is denoted by $\boldsymbol{M}_{\text{node}}$. In GFMBEM, the sparsity pattern of $\boldsymbol{B}$ is also based on adjacent nodes.

Secondly, the pattern of $\boldsymbol{M}_{\text{leaf}}$ is based on leaves for FMM. Because the oc-tree is created based on the center of element in GFMBEM, and a node is considered to be in the leaf of its element. So a node may be in different leaves, such as node b in Fig. 1(b) belongs to leaf 1 and 2. Therefore, the nodes in leaf 1 are neighbors of node a, and all nodes in leaf 1 and leaf 2 are the neighbors of node b. Besides, the node indices do not to be arranged to be sequential. Compared with the method in [4], the calculation is reduced.



(a) (b)

Fig. 1 Definition of neighbors, (a) based on adjacent nodes, (b) based on leaves.

Thridly, The pattern of $\boldsymbol{M}_{\text{near}}$ is same as $\boldsymbol{A}_{\text{near}}$. The amount of

nonzero entries and calculation complexity in the three sparsity pattern increases gradually.

Since $\boldsymbol{A}_{\text{near}}$ is stored, and the sparsity pattern of $\boldsymbol{M}$ is selected, $\boldsymbol{M}$ can be obtained from $\boldsymbol{A}_{\text{near}}$ by using SAI based on the minimization of Frobenius norm. The right-approximate inverse of $\boldsymbol{A}_{\text{near}}$ can be expressed as (4).

$$\left\|\boldsymbol{I}-\boldsymbol{A}_{\text{near}}\cdot\boldsymbol{M}\right\|_{\text{F}}^{2}=\sum_{j=1}^{N}\left\|\boldsymbol{e}_{j}-\boldsymbol{A}_{\text{near}}\cdot\boldsymbol{m}_{j}\right\|_{2}^{2} \tag{4}$$

where $\boldsymbol{e}_j$ and $\boldsymbol{m}_j$ respectively represent the jth column of $\boldsymbol{I}$ and $\boldsymbol{M}$. N represents the amount of unknowns. The SAI algorithm to calculate $\boldsymbol{M}$ is as follows [4].

Let $S=\{(i,j)|1\le i,j\le N,\ \boldsymbol{M}(i,j)\neq 0\}$ represent the set of indices of nonzero entries in $\boldsymbol{M}$. For every m_j.

(1) Let $J=\{i|(i,j)\in S\}$ represent the set of row indices of nonzero entries in m_j.

(2) Let $I=\{i|\boldsymbol{A}_{\text{near}}(i,J)\neq 0\}$represent the set of row indices of nonzero entries in Jth colunm of $\boldsymbol{A}_{\text{near}}$.

(3) Let $\boldsymbol{A}'_{\text{near}}=\boldsymbol{A}_{\text{near}}(I,J)$ represents the sub-matrix of $\boldsymbol{A}_{\text{near}}$ formed from I and J.

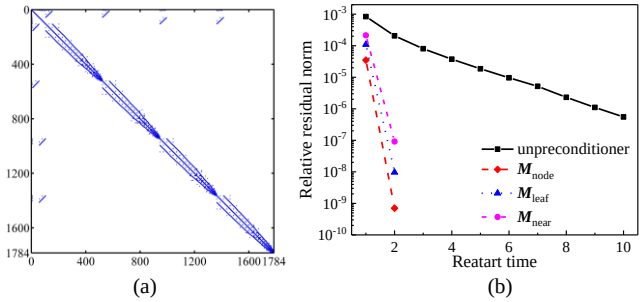
(4) min $\|\boldsymbol{A}'_{\text{near}}(I,J)\bullet\boldsymbol{m}_j\ (J)=\boldsymbol{e}_j(I)\|_2$ to obtain $\boldsymbol{m}_j\ (J)$. Since $\boldsymbol{A}'_{\text{near}}$ is small, the least square problem can be solved by QR decomposition.

The left-approximate inverse can be calculated by the same algorithm by replacing $\boldsymbol{A}_{\text{near}}$ and $\boldsymbol{M}$ with $\boldsymbol{A}^{\text{T}}_{\text{near}}$ and $\boldsymbol{M}^{\text{T}}$, respectively. In order to cooperate with the algorithm, $\boldsymbol{A}_{\text{near}}$, $\boldsymbol{B}$, and $\boldsymbol{M}$ are stored by the compressed sparse row (CSR) format in left-preconditioned GMRES(m), and the compressed sparse column (CSC) format in right-preconditioned GMRES(m).

III. Numerical Results

The code is written in FORTRAN language. In all examples, the expansion order of FMM is 4, m is 8. The stopping criterion are: (1) the relative residual norm is less than 1×10^{-6}, (2) the max restart number is over 21.

The first example is two spheres model. The node number is 1,784, and layers number of oc-tree is 4. The nonzero entries of $\boldsymbol{M}_{\text{node}}$, $\boldsymbol{M}_{\text{leaf}}$, and $\boldsymbol{M}_{\text{near}}$ is respectively 12,457, 97,971, and 899,133. The structure of $\boldsymbol{M}_{\text{node}}$ is shown in Fig. 2(a), and the convergence using different left preconditioning matrix is shown in Fig. 2(b).

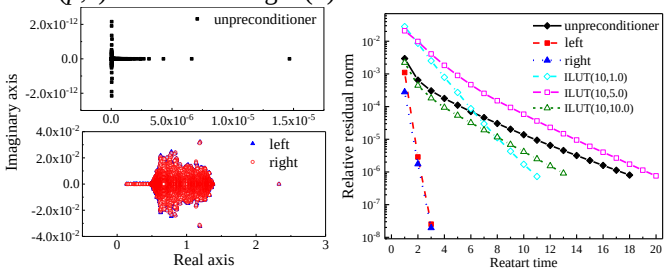


(a) (b)

Fig. 2 Results of example 1, (a)sparsity structure of $\boldsymbol{M}_{\text{node}}$, (b)convergence of different preconditioners.

The three preconditionering matrix can achieve quick convergence. The same convergence rate can be achieved by $\boldsymbol{M}_{\text{node}}$, though the calculation complexity is least.

The second example is three paralleled torus. The node number is 6,720, and layers number is 5. Using this example, the different between left and right preconditioner is research. As we known, The eigenvalues are more concentrated near 1, the convergence rate is more quick. The distribution of eigenvalues is shown in Fig. 3(a) where the coefficient matrix is calculated by traditional BEM. Besides, $\boldsymbol{M}_{\text{node}}$ is selected as preconditioning matrix, the convergence compared with ILUT(p,τ) is shown in Fig. 3(b).



(a) (b)

Fig. 3 Results of left and right preconditioner, (a) eigenvalues distribution, (b)convergence.

The eigenvalues distribution of left and right preconditioner is almost the same, and the matrix characteristics is both improved. Therefore, the convergence rate is same. This two preconditioners are both efficient.

The third example is simplified model of corona rings in converter valve. The unknowns number is 19,200. The preconditioner is left preconditioning based on adjacent nodes. Comparing with the proposed method, the temporary sparsified matrix $\boldsymbol{A}'$ is also based on adjacent nodes in [4]. The results of the two methods are shown in table I.

TABLE I
Comparison of two method

Method	Method in [4]			Proposed method		
layers	3	5	7	3	5	7
Number of restart for convergence	9	9	9	2	2	2
Calculating time of M (s)	28.5	28.5	29.3	78.9	35.8	33.6
Total time (s)	7463.4	2152.6	3027.9	6375.6	1097.6	1109.6

When the number of layers is small, the entries amount of $\boldsymbol{A}_{\text{near}}$ is more, so the calculating time of $\boldsymbol{M}$ is more. The proposed method is more efficient. Though calculating time of $\boldsymbol{M}$ is slightly larger, the total time is obvious smaller.

References

[1] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., Beijing: science press, 2009, pp. 261-333.

[2] Z. J. Chen, H. Xiao, and X. Yang, "Error analysis and novel near-field preconditioning techniques for Taylor series multipole-BEM," *Eng. Analysis with Boundary Elements*, vol. 34, no. 2, pp. 173-181, Feb, 2010.

[3] J. M. 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, pp. 1488–1493, Oct. 1997.

[4] H. T. Wang, Z. H. Yao, and P. B. Wang, "On the preconditioners for fast multipole boundary element methods for 2D multi-domain elastostatics," *Eng. Analysis with Boundary Elements*, vol, 29, no. 7, pp. 673-688, Jul. 2005.

[5] J. Lee, J. Zhang, and C. Lu, "Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetic," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 9, pp. 2277-2287, Sep. 2004.

[6] K. Yoshida, "Application of fast multipole method to boundary integral equation method," Ph.D. dissertation, Dept. of Global Environment Eng., Kyoto Univ., Kyoto, Japan, 2001.