A Quasi Block Cholesky Algorithm for Fast Direct¹ Solution of Integral-Equation Method Based on the PMCHWT Formulation

Shumin Wang

Laboratory of Functional and Molecular Imaging
National Institute of Neurological Disorders and Stroke
National Institutes of Health
10 Center Dr. 10/B1D728, Bethesda, Maryland 20892, USA.
Email: james.wang@ieee.org

I. INTRODUCTION

FAST direct integral-equation methods are important in many engineering applications, for instance, design optimization and simulating high-Q structures [1][2]. Matrix compression is a key because arithmetic operations of compressed matrix blocks involve less floating point operations (FLOPs). Studies show much improved computational efficiency when these methods were applied to metallic structures [1][2]. In this article, we propose a Quasi Block Cholesky (QBC) algorithm for fast direct solution of the Poggio-Miller-Chang-Harrington-Wu-Tsai (PMCHWT) formulation for dielectric bodies [3]. Unlike the EFIE, which results in symmetric impedance matrices, impedance matrices of the PMCHWT formulation are asymmetric but have well-defined symmetric/skew-symmetric patterns. The QBC algorithm explores this feature and halves both memory and CPU time as compared to regular block LU decomposition. With single-level Adaptive Cross Approximation (ACA) method for matrix compression [4], the resultant CPU time scales around $O(N^2)$ and memory scales around $O(N^{3/2})$.

II. THE PMCHWT FORMULATION

For generality, let us consider three dielectric objects. Object 1 contains object 2 and is next to object 3. The material in object l has $(\epsilon_l, \mu_l, \sigma_l)$. Either the RWG or the roof-top basis functions can be used and we denote them by $\vec{\alpha}_i$. Following the procedure of Galerkin's method, and use $\vec{\beta}_j = \hat{n} \times \vec{\alpha}_j$ as testing function, the resulting impedance matrix is written as [3]

$$\begin{bmatrix} Z_{11}^{11} & Z_{12}^{11} & Z_{11}^{12} & Z_{12}^{12} & Z_{11}^{13} & Z_{12}^{13} \\ -Z_{12}^{11T} & Z_{22}^{11} & Z_{21}^{12} & Z_{22}^{12} & Z_{21}^{13} & Z_{22}^{13} \\ Z_{11}^{12T} & -Z_{12}^{12T} & Z_{11}^{22} & Z_{12}^{22} & 0 & 0 \\ -Z_{21}^{12T} & Z_{22}^{12T} & -Z_{12}^{22T} & Z_{22}^{22} & 0 & 0 \\ Z_{11}^{13T} & -Z_{12}^{13T} & 0 & 0 & Z_{11}^{33} & Z_{12}^{33} \\ -Z_{21}^{13T} & Z_{22}^{13T} & 0 & 0 & -Z_{12}^{33T} & Z_{22}^{33} \end{bmatrix} \quad (1)$$

where each entry of each sub-matrix is given by

$$Z_{11,ij}^{mm} = \vec{\alpha}_i \cdot \left[\frac{1}{\epsilon_i^m} \mathcal{L}_i^m(\vec{\alpha}_j) + \frac{1}{\epsilon_e^m} \mathcal{L}_e^m(\vec{\alpha}_j) \right] \quad (2)$$

$$Z_{12,ij}^{mm} = -\vec{\alpha}_i \cdot [\mathcal{K}_i^m(\vec{\alpha}_j) + \mathcal{K}_e^m(\vec{\alpha}_j)] \quad (3)$$

$$Z_{22,ij}^{mm} = \vec{\alpha}_i \cdot \left[\frac{1}{\mu_i^m} \mathcal{L}_i^m(\vec{\alpha}_j) + \frac{1}{\mu_e^m} \mathcal{L}_e^m(\vec{\alpha}_j) \right] \quad (4)$$

$$Z_{11,ij}^{mn} = \pm \vec{\alpha}_i \cdot \frac{1}{\epsilon^{mn}} \mathcal{L}^{mn}(\vec{\alpha}_j) \quad (5)$$

$$Z_{12,ij}^{mn} = \mp \vec{\alpha}_i \cdot \mathcal{K}^{mn}(\vec{\alpha}_j) \quad (6)$$

$$Z_{22,ij}^{mn} = \pm \vec{\alpha}_i \cdot \frac{1}{\mu^{mn}} \mathcal{L}^{mn}(\vec{\alpha}_j) \quad (7)$$

The plus and minus signs in Eqs. (5)–(7) depend on the relationship between two objects. If object m is adjacent to object n , the first set of signs are taken. If object m contains object n , the second set of signs are taken. $\mu_{i,e}^m$ and $\epsilon_{i,e}^m$ denote the interior and the exterior material of object m . $\mathcal{L}_l$ and $\mathcal{K}_l$ are defined by

$$\mathcal{L}_l(\vec{\alpha}_j) = \frac{(\nabla \nabla \cdot + k_l^2) \int_{S'} G_l(\vec{r}, \vec{r}') \vec{\alpha}_j dS'}{j\omega} \quad (8)$$

$$\mathcal{K}_l(\vec{\alpha}_j) = \nabla \times \int_{S'} G_l(\vec{r}, \vec{r}') \vec{\alpha}_j dS' \quad (9)$$

where $k_l = \omega \sqrt{\mu_l \epsilon_l}$ and $G_l(\vec{r}, \vec{r}')$ is Green's function in homogeneous medium

$$G_l(\vec{r}, \vec{r}') = \frac{e^{-jk_l|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \quad (10)$$

III. THE QUASI BLOCK CHOLESKY ALGORITHM

If the impedance matrix in Eq. (1) is either symmetric or skew-symmetric, efficient decomposition algorithms are available [5]. However, Eq. (1) is neither symmetric nor skew-symmetric. A closer examination reveals that the lower-half of $[Z]$ can be obtained from the *transverse* of the upper-half of $[Z]$ with the following sign pattern

$$\begin{bmatrix} + & & & & & & \\ - & + & & & & & \\ + & - & + & & & & \\ - & + & - & + & & & \\ + & - & + & - & + & & \\ - & + & - & + & - & + & \end{bmatrix} \quad (11)$$

In the following, we denote this special pattern as checkerboard symmetry pattern. One may wonder whether this checkerboard symmetry can be utilized in a way similar to the Cholesky decomposition to halve the memory and CPU time. This is the motivation of deriving the QBC algorithm.

With checkerboard symmetry pattern, each lower-half matrix block $[Z_{ji}]$ satisfies

$$[Z_{ji}] = (-1)^{(i+j)} [Z_{ij}]^T \quad (12)$$

with block indices i and j start from zero. The QBC algorithm is stated as follows.

The Quasi Block Cholesky algorithm

1. for $i=0:M-1$
2. for $j=i:M-1$

3. $[Z_{ij}] = [Z_{ij}] - \sum_{k=0}^{j-1} (-1)^{(i+k)} [Z_{ki}]^T [Z_{kj}];$
4. *end*
5. *Decompose* $[Z_{ii}]$ *by* $[G_i][G_i]^T = [Z_{ii}];$
6. *Replace* $[Z_{ii}]$ *by* $[G_i]^T;$
7. *for* $j=i+1:M-1$
8. *Solve* $[G_i][Z'_{ij}] = [Z_{ij}];$
9. *Overwrite* $[Z_{ij}]$ *with* $[Z'_{ij}];$
10. *end*
11. *end*

The QBC algorithm has a computational sequence similar to the Crout algorithm. It starts by updating each row of matrix blocks in the upper half of $[Z]$, i.e., $[Z_{ij}]$ with $i \geq j$. The updating (in line 3) only accesses the upper half of $[Z]$, as the lower half matrix blocks are the transpose of the upper half matrix with the checkerboard sign pattern. After updating, the diagonal block is decomposed into the product of the Cholesky triangle $[G_i]$ and its transpose by the Cholesky decomposition in line 5. $[G_i]$ is subsequently used in line 8 to obtain a new matrix $[Z'_{ij}]$ by forward substitution. Finally, $[Z_{ij}]$ with $i > j$ is replaced by $[Z'_{ij}]$ and the algorithm proceeds.

It can be proved that the checkerboard symmetry pattern is preserved during matrix decomposition, thus the algorithm can proceed as proposed and only the upper half of the impedance matrix needs to be stored and processed. In the following, we demonstrate its validity and performance by numerical examples.

IV. RESULTS

The VV-polarized bi-static RCS of a 1-m diameter lossless dielectric sphere with $\epsilon_r = 4.0$ was computed [6]. The sphere was modeled by 2,480 triangular patches and a total of 7,440 unknowns. In Figure 1, the results at 300 MHz obtained by conventional LU decomposition of uncompressed matrix (the conventional method) and the QBC method with ACA tolerance $\tau = 0.01$ are compared with Mie results, which were taken from Ref. [6]. The QBC results are very close to the Mie results.

Fig. 1. Comparison of the bi-static RCS at 300 MHz of a 1-m diameter lossless dielectric sphere with $\epsilon = 4.0$ (left) and the computational complexity (right).

A further comparison of the QBC results with different ACA tolerance τ and the conventional method are listed in Table 1, where the results of the conventional

method were used as reference to evaluate the 2-norm relative errors (ϵ) of the QBC method. All results were obtained on a 2.6 GHz Intel CPU with single-precision numbers. In this table, t_1 is matrix construction time and t_2 is matrix decomposition time. Overall, the QBC method is an order of magnitude faster than the conventional method in this example.

TABLE I
COMPUTATIONAL EFFICIENCY AND ACCURACY OF THE QBC METHOD

	RAM(MB)	t_1 (s)	t_2 (s)	ϵ
QBC with $\tau = 10^{-2}$	101	52.6	92	2.49%
QBC with $\tau = 10^{-3}$	124	71.3	154	0.13%
QBC with $\tau = 10^{-4}$	150	91	216	0.007%
Full matrix	632	60	1502	(N/A)

The computational complexity was illustrated in Fig. 1 as we increased the number of unknowns. Two scenarios were considered. First of all, mesh density (as a function of wavelength) was kept constant and the diameter of the sphere increased. Then, the diameter of the sphere was kept constant and mesh density increased. The CPU time for matrix decomposition scales by $O(N^{2.3})$ in the first scenario and by $O(N^{1.85})$ in the second scenario. As to RAM consumption, it scales by $O(N^{1.6})$ in the first scenario and by $O(N^{1.4})$ in the second scenario. Increasing mesh density is less computationally expensive than increasing problem domain size.

V. CONCLUSIONS

We proposed a QBC algorithm for fast direct solution of the PMCHWT formulation for dielectric bodies. This algorithm explores the checkerboard symmetry of the impedance matrix and halves both memory and CPU time as compared to regular block LU decomposition. In the examples we have tested, the resultant CPU time scales around $O(N^2)$ and memory scales around $O(N^{3/2})$.

REFERENCES

- [1] A. Heldring, J.M. Rius, J.M. Tamayo, J. Parrón and E. Úbeda *Fast Direct solution of Method of moments Linear System*, IEEE Trans. Antennas and Propagat., vol. 55, no. 11, pp. 3220–3228., Nov. 2007.
- [2] R.J. Adams, Y. Xu, X. Xu, J.S. Choi, S.D. Gedney and F.X. Canning *Modular Fast Direct Electromagnetic Analysis Using Local-Global Solution Modes*, IEEE Trans. Antennas and Propagat., vol. 56, no. 8, pp. 2427–2441., Aug. 2008.
- [3] W.C. Chew, J.M. Jin, E. Michielssen, and J.M. Song eds *Fast and efficient algorithms in computational electromagnetics*, Boston, MA: Artech House Publishers, 2000.
- [4] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., vol. 86, no. 4, pp. 565–589, Jun. 2000.
- [5] I.S. Duff, A.M. Erisman, and J.K. Reid, *Direct methods for sparse matrices*, Oxford: Oxford Science Publications, 1986.
- [6] X.Q. Sheng, J.-M. Jin, J. Song, W.C. Chew and C.-C. Lu *Solution of Combined-Field Integral Equation Using Multilevel Fast Multipole Algorithm for Scattering by Homogeneous Bodies*, IEEE Trans. Antennas and Propagat., vol. 46, no. 11, pp. 1718–1726., Nov. 1998.