SAND2017-12646C

The Parallel Implementation and Accuracy of Matrix Compression in the Method of Moments code EIGER

Joseph D. Kotulski
Electromagnetic Theory
Sandia National Laboratories
Albuquerque, NM USA
jdkotul@sandia.gov

***Abstract*—This paper describes the implementation of the adaptive cross approximation in the method of moments code EIGER. This purely algebraic method provides a mechanism to reduce memory usage and overall computation time. In addition, this work has been targeted for massively parallel platforms to extend the viable frequency range for electromagnetic compatibility and interference problems.**

Keywords—electromagnetic boundary element formulation, method of moments, fast integral equation methods

I. Introduction

The integral equations solved by the Method of Moments (MOM) are an invaluable tool to analyze and then predict the response of systems to electromagnetic environments[1]. Since there is a memory limitation prohibiting the storage of the full matrix for problems large with respect to frequency, alternative techniques have been identified to circumvent this restriction. These alternative techniques are based on methods that use the reduction of the degrees of freedom for the far zone interactions to decrease the storage requirement. The method to be examined is matrix compression using the adaptive cross approximation(ACA)[2], implemented in EIGER[3].

II. Formulation

A. Approximation to the Matrix Equation

The electric and magnetic field integral equations are cast using the standard Stratton-Chu formulation. They are combined to avoid the interior resonance problem associated with closed surfaces using a coupling parameter of 0.5. Once the equations are defined, the Rao-Wilton-Glisson expansions are used for the currents on the surface of the scatterer. Following the application of Galerkin's method a system of linear equations is formed.

$$\sum_{n=1}^{N} Z_{mn} I_n = V_m \quad m = 1,2,\ldots,N \qquad , \qquad (1)$$

where N is the number of unknowns. The ACA method and other "compression" techniques have theoretical basis from the work of Bucci and Franceschetti[4]. Simply stated - separated clusters of unknowns (on elements) have a reduced order to adequately describe their interactions. From the moment method point of view a cluster of elements containing m test elements interacting with a cluster of n source elements can be described by a matrix that has less then m x n contributions. With this thought in mind the matrix equation can be written as:

$$Z_{nxn} I_n = V_m => (Z_{near} + Z_{far}) I = V \qquad . \qquad (2)$$

Where

$$Z_{near} = \sum_{j=1}^{MOM\_blocks} Z_j^{mom} \text{ and } Z_{far} = \sum_{i=1}^{COM\_blocks} \tilde{Z}_i^{com} \qquad . \qquad (3)$$

The *MOM* blocks are full matrix blocks representing the interaction between a group of testing and source functions which are computed in the standard way. The *COM*(compressed) blocks represent the interaction between a group of testing functions and source functions approximated via the ACA. The steps include the identification of the interaction blocks and then the computation of the blocks. The blocks are identified by encapsulating the entire geometry into a group (identified as a box) which is then sub-divided by using an oct-tree (Figure 1). The box at the lowest level of the oct-tree should be large enough allow compression of the block (box to box) interaction.

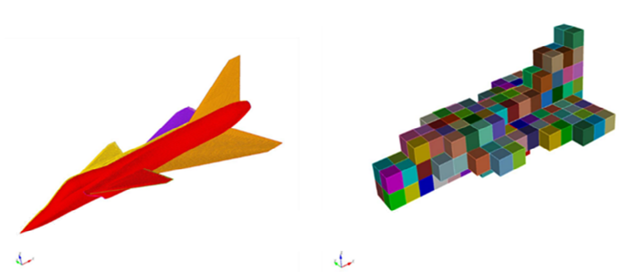

Figure 1. An example of a meshed object and corresponding oct-tree boxes used for the compression algorithm.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Each compressed block is described as

$$\widetilde{Z}_i^{m\times n} = U_i^{m\times r} V_i^{r\times n} = \sum_{j=1}^{r} u_j^{m\times 1} v_j^{1\times n} \quad . \qquad (4)$$

The algorithm is described in detail by Zhao et. al[2]. The key feature consists of building a low-rank matrix in an iterative fashion without building the entire matrix block. At each iteration, the Frobenius norm of the new approximation is compared to the calculated value of the approximate matrix multiplied by an error criterion.

$$\|u_k\| \cdot \|v_k\| \le \varepsilon \left\| \widetilde{Z}^{(k)} \right\| \qquad (5)$$

Where ε is the compression tolerance used in the algorithm. When this criterion is met the for the k'th iteration the process stops, a compressed form of the matrix is obtained, and the memory storage is now *(m+n) x r* elements instead of *m x n.*

The filling of the different matrix blocks (*MOM* and *COM*) has been implemented for use on parallel computing platforms. To efficiently perform this operation a load-balancing scheme has been used in the distribution of the different blocks to multiple processors based on the size of the different blocks. It is straightforward to load-balance the MOM blocks due to the knowledge of the sizes of the blocks. The distribution of the *COM* blocks are based on an estimate of their size from the work of Bucci and Franceschetti[4].

Now with the matrix blocks computed equation 1 can be solved using an iterative solver. Two such solvers have been implemented and tested – generalized minimum residual method(GMRES) and the transpose free quasi-minimal residual method (TFQMR). The results shown here are for the GMRES algorithm with restart[5].

B. Parallel Considerations

Two different algorithms were used to distribute the matrix fill for both *MOM* and *COM* blocks to the different processors. First, a simplified technique was employed that only used the number of blocks without regard to the size of work in each block. The rule for this algorithm was that no processor would have no more, or less, than one block with respect to any of the other processors. This implementation resulted in a highly imbalanced time for the matrix fill on the different processors. To improve this the computational workload in each block was calculated for the *MOM* blocks and estimated for the *COM* blocks using [4]. This resulted in a very balanced fill for the matrix blocks.

III. RESULTS

The algorithm was applied to the VFY-218 with a grid density that would be used for up to 1.2 GHz. This problem was also solved using LU decomposition for comparison. The job statistics are shown in Table 1 and are for a compression tolerance (ε) of 1.5e-03 and a solver tolerance of 1.e-06 using the GMRES solver. The incident plane wave electric field is polarized in the phi direction and is incident on the nose of the aircraft.

Table 1.Statistics for the Vfy-218 (934,128 unknowns).

Solution Method	Memory Used (GBytes)	Solution Time(hours)	Number of Processors
Direct Solve(LU)	13962	4.6	5040
ACA (GMRES)	436	13.5	80

The comparison of the bistatic RCS between the direct solve and using ACA is shown in Figure 2.

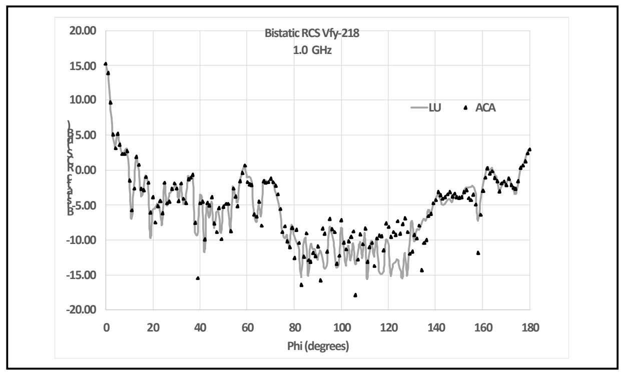

Figure 2. Comparison of bistatic RCS for the VFY-218 at 1GHz for the LU and ACA solution.

IV. CONCLUSIONS

This paper demonstrates the utility of the method of moments when coupled with the ACA and an iterative solver. Further work is continuing to improve the solution accuracy, developing methods of evaluating the accuracy, and solver efficiency.

REFERENCES

[1] J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in R. Mittra (ed.), Computer Techniques for Electromagnetics, Oxford, UK, Pergamon Press, 1973.

[2] K. Zhao, M. N. Vouvakis, J.-F. Lee, "The adaptive cross approximation for accelarated method of moments compiuttation of EMC problem," IEEE Trans. Electromagnetic Compat., vol 47, pp. 763-773,Nov. 2005.

[3] D. R. Wilton, W. A. Johnson, R. Jorgenson, R. Sharpe, J. Rockway, "EIGER: a new generation of computational electromagnetics tools," ElectroSft: Software for EE Analysis and Design, Miniato, Italy, pp. 28-30, May 1996.

[4] O. Bucci and Giorgio Franceschetti, "On the degrees of freedom of scattered fields," IEEE Trans,. Antennas and Prop., vol 30, pp. 918-926, July 1989.

[5] Y. Saad, Iterative Methods for Sparse Linear Systems, Sec. Ed., SIAM, 2003, pp. 164-184.