

**Single Level Dual Rank SVD Algorithm
for Volume Integral Equations of Electromagnetic Scattering**

N. A. Ozdemir*, J.F. Lee

ElectroScience Lab., Electrical Engineering Dept.
The Ohio State University
1320 Kinnear Road, OH 43212
ozdemir.5@osu.edu

1. Introduction

Electromagnetic scattering from arbitrarily shaped inhomogeneous dielectric bodies has been an interesting topic due to its importance in problems that include propagation through inhomogeneities. This type of problems can be formulated in the form of a volume integral equation (VIE) and solved by the method of moments (MoM). However, solving large scale electromagnetic problems by MoM suffers from $O(N^2)$ numerical complexity of memory requirement and CPU time both to assemble the matrix and to perform the matrix-vector multiplication in the iterative solver, where N is the number of unknowns. This paper presents a novel single level dual rank SVD algorithm that efficiently compresses the system matrix to reduce the memory requirement and CPU time for both matrix assembly and matrix-vector multiplication to $O(N^{3/2})$. The dual rank SVD algorithm is based on the rank deficiency feature of the integral equation for well-separated groups of basis functions. The algorithm forms the $Q-R$ factorization of rank deficient local matrices due to non-self group interactions by ranking the most linearly independent basis functions in the transmitting (source) group and the most significant ones in the receiving (observation) group.

There have been a number of successful algorithms that reduce the numerical complexity of integral equation methods such as the fast multipole method (FMM) [1] and the SVD based algorithms presented in [2] and [3]. The method presented in this paper tends to overcome the drawbacks of the previous algorithms [1-3] such that: it is independent of the integral equation kernel; it assembles the local Q and R matrices without a priori knowledge of the local matrix itself and it is free of the previous assumptions about the kernel of the integral equation, namely, the smoothness of the kernel to implement reduced sampling procedure in [3].

2. Volume Integral Equation Formulation

Volume integral equation (VIE) is obtained through mixed potential integral equation. Let V denote the volume of a scatterer characterized by its relative permittivity $\epsilon(\vec{r})$ and relative permeability $\mu(\vec{r})$, which are continuous inside V and let the background medium be free space. Applying the volume equivalence principle and the continuity equation, the scatterer can be replaced with equivalent electric current and charge densities [4]:

$$\vec{J}(\vec{r}) \approx j\omega\kappa(\vec{r})\vec{D}(\vec{r}), \quad (1a)$$

$$\rho(\vec{r}) \approx -\nabla\kappa(\vec{r}) \cdot \vec{D}(\vec{r}) - \kappa(\vec{r})\nabla \cdot \vec{D}(\vec{r}), \quad (1b)$$

where $\kappa(\vec{r}) = \frac{\epsilon(\vec{r}) - \epsilon_0}{\epsilon(\vec{r})}$ is the contrast ratio, $\vec{D}(\vec{r})$ is the total electric flux density and ω is the frequency

of operation. Decomposing the total field into incident and scattered fields we have

$$\vec{E}(\vec{r}) \approx \vec{E}'(\vec{r}) + \vec{E}''(\vec{r}), \quad (2)$$

where

$$\vec{E}' = -j\omega\vec{A}(\vec{r}) - \nabla\Phi(\vec{r}), \quad (3)$$

$$\vec{A}(\vec{r}) = \frac{\mu_0}{4\pi} \int_V \vec{J}(\vec{r}') \frac{e^{-j\mu_0|\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|} dv', \quad (4a)$$

$$\phi(\vec{r}) = \frac{\mu_0}{4\pi} \int_V \frac{\rho(\vec{r}') e^{-j\beta_0 |\vec{r}-\vec{r}'|}}{|\vec{r}-\vec{r}'|} dv' \quad (4b)$$

Discretization of the volume by tetrahedral elements and interpolation of $\vec{D}(\vec{r})$ by the first order divergence conforming vector basis functions followed by testing of Eq. (2) with the basis function chosen leads to a system of linear equations:

$$\langle D/\epsilon, \vec{f}_m \rangle + \langle j\omega\vec{A}, \vec{f}_m \rangle + \langle \nabla\Phi, \vec{f}_m \rangle = \langle E', \vec{f}_m \rangle \quad m=1,2,\dots,N, \quad (5)$$

which can be written in matrix form as

$$[Z] \cdot [D] = [Y] \quad (6)$$

where N is the total number of unknowns, $[Z]$ is the $N \times N$ impedance matrix, D and Y are vectors of length N .

3. Dual Rank SVD Algorithm

The dual rank SVD algorithm factorizes the local impedance matrix Z_{mn}^q due to the groups of basis functions i and j into Q_{mn}^q and R_{mn}^q matrices, where m and n are the number of receiving and transmitting basis functions, respectively, and r is the rank of interaction:

$$Z_{mn}^q = \begin{bmatrix} z_1 & z_n \end{bmatrix} = \begin{bmatrix} v_1 \\ v_n \end{bmatrix} \approx Q_{mn}^q \cdot R_{mn}^q \quad (7)$$

Here, z_p is the column vector due to the p^{th} basis function in the transmitting group (transmitter) and v_q is the row vector due to the q^{th} basis function in the receiving group (receiver). The basic advantage of this algorithm is that it does not require a priori knowledge of Z^q .

In order to be able to implement the dual rank SVD algorithm, the problem domain is recursively partitioned into groups of tetrahedrons of size at most the prescribed threshold α . Then the algorithm consists of two main parts: construction of Q^q by dual ranking in an iterative scheme and computation of R^q by LU decomposition.

The objective of the dual ranking algorithm to construct Q^q is to determine the most linearly independent orthonormal column vectors that span the column space of Z^q within a prescribed tolerance δ in an iterative scheme. To achieve this objective, the algorithm ranks the basis functions in both the transmitter and receiver groups on the basis of the linear independency of the column vectors and the significance of the row vectors they give rise to, respectively. There are three steps in constructing Q^q : selection of the transmitter, updating the Q^q matrix and selection of the receiver. Iterations continue until the columns of Q^q can span the column space of Z^q within a prescribed tolerance δ . For the k^{th} iteration the algorithm can be briefly prescribed as:

1. Selection of the transmitter:

- Form m_{i+1} column vectors from $v_{i_1}, v_{i_2}, \dots, v_{i_k}$:

$$m_i = \begin{bmatrix} v_{i_k}(i) \\ v_{i_k}(i) \end{bmatrix} \quad i=1,\dots,n$$

- Perform modified Gram-Schmidt (MGS) orthogonalization based upon transmitter ranking

$$\{\bar{m}_{i_1}, \bar{m}_{i_2}, \dots, \bar{m}_{i_{k+1}}\} = MGS\{m_{i_1}, m_{i_2}, \dots, m_{i_{k+1}}\}$$

- Find the index of the next transmitter:

$$m'_i = m_i - \sum_{j=1}^{k-1} \langle m_i, \bar{m}_{j,i} \rangle \bar{m}_{j,i} \quad i = 1, \dots, n; i \notin J$$

$$\|m'_{i_k}\| = \max_{i \notin J} \|m'_i\|$$

2. Updating Q :

- Compute J_k^* column of Z , z_{j_k} , and orthonormalize it with respect to the previously computed $(k-1)$ columns of Q :

$$q'_k = z_{j_k} - \sum_{i=1}^{k-1} \langle z_{j_k}, q_i \rangle q_i; \quad q_k = \frac{q'_k}{\|q'_k\|}$$

- If $\frac{\|q'_k\|}{\|z_{j_k}\|} \leq \delta$, end the iterations; else continue.

3. Selection of the receiver

- Form n_{rd} row vectors from $q_1, q_2, \dots, q_k$:

$$\|n_{i_k}\| = \max_i \|n_i\|; \quad i = 1, 2, \dots, m \text{ and } i \notin I$$

The r columns of R that are the expansion coefficients of the corresponding r columns of Z with respect to the column vectors in Q are obtained while updating the Q matrix. The rest $(n-r)$ columns of R are obtained by solving the r equations extracted from (7).

4. Numerical Complexity of the Algorithm

Assuming the average number of basis functions in each group is m , the number of groups is $N_g = N/m$. If average rank r is also assumed to be constant for all non-self group interactions, the numerical complexity for the memory requirement and matrix-vector multiplication can be estimated by

$$C = N_g m^2 + (N_g^2 - N_g) 2rm = Nm + 2rN^2/m - 2rN. \quad (8)$$

C can be optimized by choosing m proportional to $\sqrt{N}$. With this choice of m the numerical complexity is $O(N^{3/2})$. The numerical complexity of the matrix assembly may also shown to be $O(N^{3/2})$, albeit with a large-scale factor.

5. Numerical Results

A numerical example is presented in this section to demonstrate both the accuracy and efficiency of the dual rank SVD algorithm for solving the volume integral equation of electromagnetic scattering. In Fig. 1(a) and (b), CPU time and memory requirements to generate the $Q-R$ matrices for homogeneous dielectric cubes of edge lengths 0.5, 0.75, 1, 1.25 λ_0 , which correspond to 1449, 3753, 10408, 20982 unknowns, are plotted. The time it takes for a single-vector matrix multiplication in the iterative solver for the same number of unknowns is shown in Fig. 1(c). From the reference lines in Fig. 1, we can see that the numerical complexity of the dual rank SVD algorithm is indeed $O(N^{1.5})$ as we expect. In Fig. 2, far field patterns calculated using the dual rank SVD algorithm is compared with the direct MoM implementation for electromagnetic scattering from a dielectric cube. The cube has edge length of 0.75 λ_0 and the dielectric constant is $\epsilon_r = 4.0$. The incident wave frequency is 1.125 GHz and the cube is illuminated at $\theta = 180^\circ$. The cube is discretized into 1777 tetrahedron elements with 3753 unknowns. We also observe in Fig. 2 that MoM and SVD results are in good agreement for far field patterns. It is the authors' intention to present numerical results for electromagnetic scattering from electrically larger dielectrics of arbitrary shape at the conference.

References

- [1] R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," IEEE Antennas Propagat. Mag., Vol. 35, No.3, June 1993, pp 7-12.
- [2] S. Kapur, J. Zhao, "A Fast Method of Moments Solver for Efficient Parameter Extraction of MCMs", DAC 1997, pp. 141-146.
- [3] S. Kapur, D. E. Long, "IES³: A Fast Integral Equation Solver for Efficient 3-Dimensional Extraction," ICCAD 1997, pp. 448-455.
- [4] D. H. Schaubert, D.R. Wilton, and A. W. Glisson, "A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies," IEEE Trans. Antennas and Propagat., vol. 32, pp. 77-85, Jan. 1984.

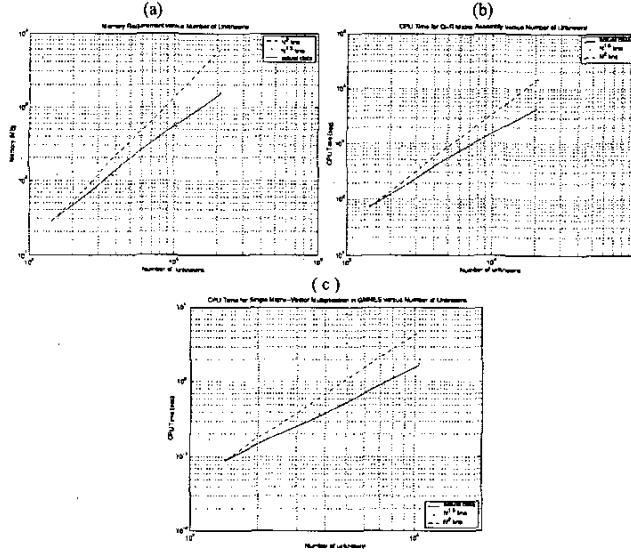


Figure 1: (a) Memory requirement, CPU time for (b) matrix assembly (c) matrix-vector multiplication

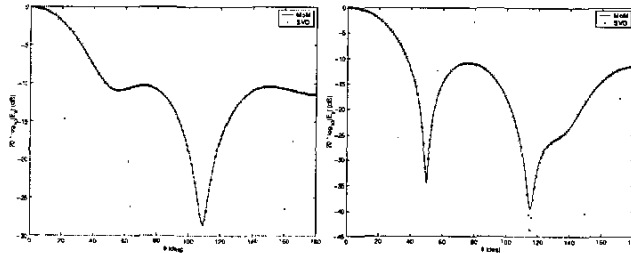


Figure 2: Comparison of far fields $20\log_{10}|E_\theta|$ and $20\log_{10}|E_\phi|$ at $\phi=0^\circ$ and $\phi=90^\circ$, respectively, for MoM and SVD