LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-JC-154844

General MoM Solutions for Large Arrays

B. Fasenfest, F. Capolino, D. R. Wilton, D. R. Jackson, N. Champagne

International Conference on Electromagnetics in Advanced Applications, Torino, Italy, September 8-12, 2003

July 22, 2003

General MoM Solutions for Large Arrays

B. Fasenfest[1], F. Capolino[2], D.R. Wilton[1], D.R. Jackson[1], and N. Champagne[3]

Abstract – **An effective MoM solution scheme is developed for large arrays using a modification of the AIM (Adaptive Integral Method) method. The method permits the analysis of arrays with arbitrary contours and nonplanar elements. Both fill and solve times within the MoM method are improved with respect to more standard MoM solvers.**

1 INTRODUCTION

This paper focuses on a numerical procedure that addresses the difficulties of dealing with large, finite arrays while preserving the generality and robustness of full-wave methods. We present a fast method based on approximating interactions between sufficiently separated array elements via a relatively coarse interpolation of the Green's function on a uniform grid commensurate with the array's periodicity. The interaction between the basis and testing functions is reduced to a three-stage process. The first stage is a projection of standard (e.g., RWG) subdomain bases onto a set of interpolation functions that interpolate the Green's function on the array face. This projection, which is used in a matrix/vector product for each array cell in an iterative solution process, need only be carried out once for a single cell and results in a low-rank matrix. An intermediate stage matrix/vector product computation involving the uniformly sampled Green's function is of convolutional form in the lateral (transverse) directions so that a 2D FFT may be used. The final stage is a third matrix/vector product computation involving a matrix resulting from projecting testing functions onto the Green's function interpolation functions; the low-rank matrix is either identical to (using Galerkin's method) or similar to that for the bases projection.

The general scheme is that of the adaptive integral method (AIM) developed in [1], with a significant difference being the approximation of the Green's function using Lagrange interpolating polynomials. The MoM scheme presented is suitable for incorporating the solver acceleration method presented recently in [2] that uses a physically-based preconditioner to accelerate the iterative solution process.

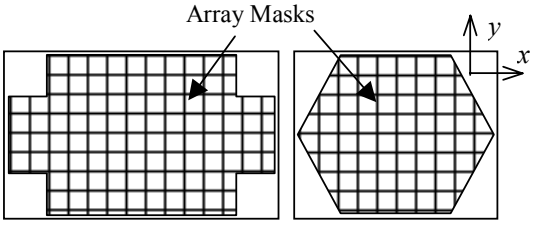

Figure 1: The array mask describes the location of array elements within a rectangular bounding box.

2 THE FAIM METHOD

The FAIM (faster AIM) method discussed below is applicable to arrays with arbitrary boundaries. The array boundary is defined by the vertices of a closed, piecewise linear curve, which in turn defines an *array mask* that indicates where array elements are present. The mask is generated using a winding number test to determine if an element is within the array boundary or not. Typical array boundaries are represented in Fig. 1. Use of the mask also simplifies the specification of missing elements when their effects are to be investigated. The displacement between the $\mathbf{p}'$th and $\mathbf{p}$th array cells is $(p_1' - p_1)\boldsymbol{s}_1 + (p_2' - p_2)\boldsymbol{s}_2$, where $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are two arbitrary lattice vectors lying in the *xy* plane. Bold letters denote vectors or double indexes, and unit vectors are denoted by a caret. The vector $\mathbf{r} = \boldsymbol{\rho} + z\hat{\mathbf{z}}$, with $\boldsymbol{\rho} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}}$, is in cell $\mathbf{p} = (p_1, p_2)$ and $\mathbf{r}' = \boldsymbol{\rho}' + z'\hat{\mathbf{z}}$ is in cell $\mathbf{p}' = (p_1', p_2')$. The array elements within cells $\mathbf{p}$ and $\mathbf{p}'$ are discretized, for instance, using standard RWG basis functions, with primed and unprimed indices denoting source and testing functions, respectively (Fig. 2).

2.1 Array and Matrix Masks and the FFT Domain

The array mask indicates the presence or absence of an array element at each cell location within an arbitrary contour. From it, a *matrix mask* is synthesized that indicates permissible interactions between the elements present in the array. The matrix mask for a hexagonal array, for example, is shown in Fig. 3; the matrix tabulates all possible interactions between elements in the array mask in terms of their separation indices from a central array element. Not only does the matrix mask

[1] Dept. of Electrical and Computer Engr., University of Houston, Houston, TX 77204-4005, USA
[2] Dip. di Ingegneria dell'Informazione, University of Siena, Via Roma 56, 53100 Siena, Italy
[3] Lawrence Livermore National Lab., Livermore, CA 94550, USA

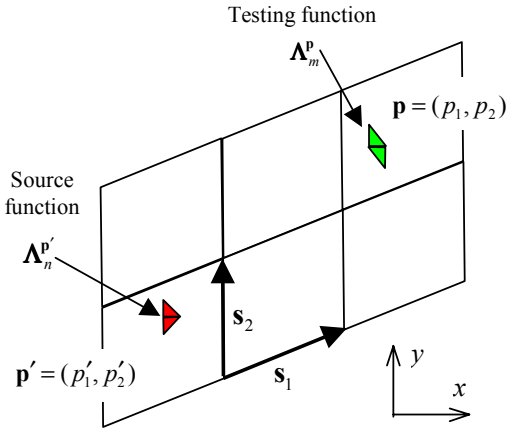

Figure 2: Array cell index definitions and arbitrary skew lattice vectors.

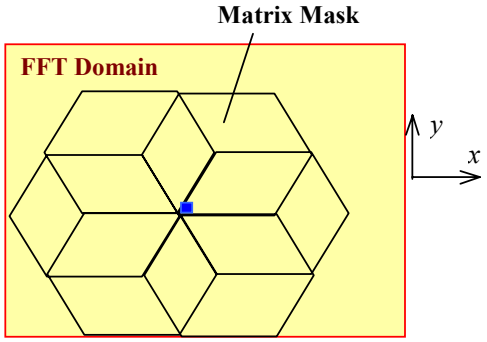

Figure 3: Matrix mask for a hexagonal array obtained by translating the array mask, and the FFT domain for a rectangular array lattice obtained by padding to the next power of 2 the number of array cells (blocks) of the matrix mask in both x and y directions.

determine what values of the Green's function are needed for interpolation, but also facilitates the storage of block-Toeplitz matrices of rank $N \times N$, with N denoting the total number of degrees of freedom per array element. Also shown in the figure is the FFT domain obtained by finding the bounding box of the matrix mask and padding the number of cells to the next power of two in each lattice dimension.

2.2 EFIE Formulation

Though a mixed potential representation is actually used, the EFIE integral equation for the array is written symbolically in the compact form

$$\int_S \mathcal{G}(\boldsymbol{\rho} - \boldsymbol{\rho}', z, z') \cdot \mathbf{J}(\mathbf{r}')\, dS' = -\mathbf{E}_{\tan}^{\text{inc}}, \tag{1}$$

where the integration domain S represents all conducting elements over the entire finite array. This integral equation has the discretized form

$$-\sum_{\mathbf{p}'} \sum_{n=1}^{N} < \mathbf{\Lambda}_m^{\mathbf{p}}(\mathbf{r}); \mathcal{G}(\boldsymbol{\rho} - \boldsymbol{\rho}', z, z'); \mathbf{\Lambda}_n^{\mathbf{p}'}(\mathbf{r}') > \mathrm{I}_n^{\mathbf{p}'} = V_m^{\mathbf{p}} \quad \forall\, m, \mathbf{p},$$

where $\mathbf{\Lambda}_n^{\mathbf{p}'}$ and $\mathbf{\Lambda}_m^{\mathbf{p}}$ are RWG basis and testing functions, respectively. In supermatrix form, the discretized EFIE may be written symbolically as

$$\left[Z_{mn}^{\mathbf{p}\mathbf{p}'} \right] \left[\mathrm{I}_n^{\mathbf{p}'} \right] = \left[V_m^{\mathbf{p}} \right], \tag{2}$$

with

$$Z_{mn}^{\mathbf{p}\mathbf{p}'} = - < \mathbf{\Lambda}_m^{\mathbf{p}}(\mathbf{r}); \mathcal{G}(\boldsymbol{\rho} - \boldsymbol{\rho}', z, z'); \mathbf{\Lambda}_n^{\mathbf{p}'}(\mathbf{r}') > .$$

2.2 Green's Function Interpolation (FAIM Filling Acceleration)

For elements separated by at least one cell, we approximate the Green's function $\mathcal{G}(\boldsymbol{\rho} - \boldsymbol{\rho}', z, z')$ via Lagrange polynomial interpolation in a separable form as

$$\mathcal{G}(\boldsymbol{\rho} - \boldsymbol{\rho}', z, z') \cong \sum_{\mathbf{i},\mathbf{i}',j,j'} L_{\mathbf{i}}(\boldsymbol{\rho}) L_j(z)\, \mathcal{G}_{\mathbf{i}\text{-}\mathbf{i}',j,j'}\, L_{\mathbf{i}'}(\boldsymbol{\rho}') L_{j'}(z'), \tag{3}$$

where the indices on $\mathcal{G}_{\mathbf{i}\text{-}\mathbf{i}',j,j'}$ denote sampled values of the coordinates and

$$\mathcal{G}_{\mathbf{i}\text{-}\mathbf{i}',j,j'} \equiv \mathcal{G}(\boldsymbol{\rho}^{(\mathbf{i})} - \boldsymbol{\rho}'^{(\mathbf{i}')}, z^{(j)}, z'^{(j')}) \tag{4}$$

and $L_{\mathbf{i}}(\boldsymbol{\rho}) L_j(z)$ are Lagrange interpolation polynomials defined on the interpolation points within a cell (see Fig. 4). It is significant that in the evaluation of the interaction $Z_{mn}^{\mathbf{p}\mathbf{p}'}$ between two array cells $\mathbf{p}$ and $\mathbf{p}'$, the interpolation scheme generally requires many fewer Green's function evaluations per cell than in the usual case where subdomain interactions are evaluated directly, or even than the usual AIM approach requires. This becomes especially true as the complexity of the array elements increases. Furthermore, in the case of layered media, the Green's function itself is evaluated by interpolating constituent terms that are tabulated along a single radial line.

2.4 FAIM Formulation

In the FAIM formulation, the m and n elements in the matrix block $Z_{mn}^{\mathbf{p}\mathbf{p}'}$ corresponding to array cell $\mathbf{p}$ and source cell $\mathbf{p}'$ may be represented as

$$Z_{mn}^{\mathbf{p}\mathbf{p}'} \approx \tilde{Z}_{mn}^{\mathbf{p}\mathbf{p}'} \equiv - \sum_{\mathbf{i},\mathbf{i}',j,j'} < \mathbf{\Lambda}_m^{\mathbf{p}}(\mathbf{r}), L_{\mathbf{i}}(\boldsymbol{\rho}) L_j(z) > \cdot \cdot \mathcal{G}_{\mathbf{i}\text{-}\mathbf{i}',j,j'} \cdot < L_{\mathbf{i}'}(\boldsymbol{\rho}') L_{j'}(z'), \mathbf{\Lambda}_n^{\mathbf{p}'}(\mathbf{r}') > \tag{5}$$

and the corresponding matrix equation would be

$$\sum_{\mathbf{p}'}\sum_{n=1}^{N}\tilde{Z}_{mn}^{\mathbf{pp}'}\cdot \mathrm{I}_n^{\mathbf{p}'} = V_m^{\mathbf{p}} \quad \forall\, m,\mathbf{p}. \tag{6}$$

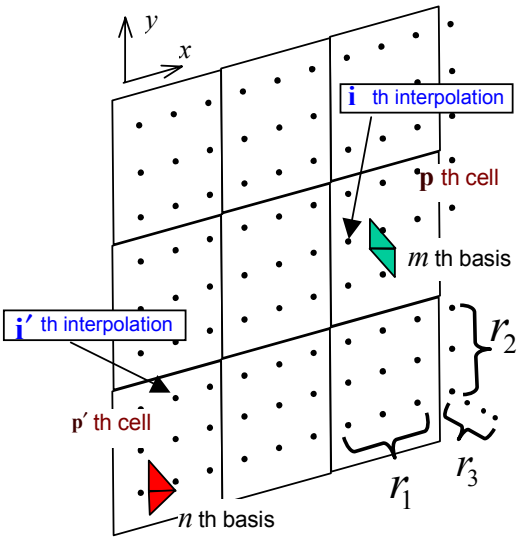

Figure 4: The periodic grid on which the Green's function is evaluated and sampled is shown superimposed on the array cells. Within an array cell, the Green's function is evaluated at $r_1 \times r_2 \times r_3$ points.

Two important facts that significantly speed the computation are the following:

1) $\left\langle L_{\mathbf{i}'} L_{j'}, \mathbf{\Lambda}_n^{\mathbf{p}'} \right\rangle_{r_1 \times r_2 \times r_3 \times N}$ vanishes unless $\left\lfloor \frac{\boldsymbol{i}'}{\boldsymbol{r}} \right\rfloor \equiv \left(\left\lfloor \frac{i_1'}{r_1} \right\rfloor, \left\lfloor \frac{i_2'}{r_2} \right\rfloor \right) = \mathbf{p}'$ where $\lfloor x \rfloor \equiv \mathrm{Floor}(x)$ is the greatest integer less than or equal to its argument. The same holds for $\left\langle L_{\mathbf{i}} L_j, \mathbf{\Lambda}_m^{\mathbf{p}} \right\rangle_{r_1 \times r_2 \times r_3 \times N}$.

2) $\left\langle L_{\mathbf{i}'} L_{j'}, \mathbf{\Lambda}_n^{\mathbf{p}'} \right\rangle_{r_1 \times r_2 \times r_3 \times N}$ and $\left\langle L_{\mathbf{i}} L_j, \mathbf{\Lambda}_m^{\mathbf{p}} \right\rangle_{r_1 \times r_2 \times r_3 \times N}$ are identical for every array element $\mathbf{p}'$ or $\mathbf{p}$, so they need only be computed once. Furthermore, if Galerkin's method is used ($\mathbf{\Lambda}_m^{\mathbf{p}} = \mathbf{\Lambda}_n^{\mathbf{p}'}$), then the quantities are equal.

2.4 FAIM Solution Acceleration. Fast Computation of Matrix-Vector Products

The form (6) of the EFIE is not sufficiently accurate if the cell (index) separation is not sufficiently large since a low order interpolation of the Green's function is not accurate near the source point. To avoid this inaccuracy while minimizing the number of interpolating polynomials within each cell, the self block coupling and that between neighboring blocks is found by standard MoM, i.e., using standard integration rules and Green's function evaluation for the interaction between each RWG basis and test function. The original discretized EFIE is thus rewritten as

$$\left[\Delta Z_{mn}^{\mathbf{pp}'}\right]\left[\mathrm{I}_n^{\mathbf{p}'}\right] + \left[\tilde{Z}_{mn}^{\mathbf{pp}'}\right]\left[\mathrm{I}_n^{\mathbf{p}'}\right] = \left[V_m^{\mathbf{p}}\right], \tag{7}$$

where the block Toeplitz *difference matrix* $\Delta Z_{mn}^{\mathbf{pp}'} = Z_{mn}^{\mathbf{pp}'} - \tilde{Z}_{mn}^{\mathbf{pp}'}$ is taken as zero for elements satisfying $|\mathbf{p}-\mathbf{p}'| \geq \mathbf{c}$ and is hence sparse. We also note that generally $\mathcal{G}_{\mathbf{i}-\mathbf{i}',j,j'} = \infty$ when $\mathbf{i} = \mathbf{i}'$, $j = j'$, but this infinite value can be replaced by a finite value and (6) remains valid. To evaluate the matrix/vector product, we note that the product $\left[\Delta Z_{mn}^{\mathbf{pp}'}\right]\left[\mathrm{I}_n^{\mathbf{p}'}\right]$ can be performed quickly since $\Delta Z_{mn}^{\mathbf{pp}'}$ is sparse, whereas $\left[\tilde{Z}_{mn}^{\mathbf{pp}'}\right]\left[\mathrm{I}_n^{\mathbf{p}'}\right]$ is of convolutional form and can be evaluated quickly using a 2D FFT as follows:

$$\left[\tilde{Z}_{mn}^{\mathbf{pp}'}\right]\left[\mathrm{I}_n^{\mathbf{p}'}\right] = -\sum_{\mathbf{i},\mathbf{i}',j,j'} < \mathbf{\Lambda}_m^{\mathbf{p}}, L_{\mathbf{i}} L_j > \cdot \mathsf{MASK}_{\mathbf{i}}\, \mathsf{FFT}_{\mathbf{i}}^{-1} \cdot \left[\left(\mathsf{FFT}_{\mathbf{i}}\, \overline{\overline{\mathcal{G}_{\mathbf{i},j,j'}}} \right) \bullet \mathsf{FFT}_{\mathbf{i}} \left(\overline{\overline{\sum_{\mathbf{p}'}\sum_{n=1}^{N} < L_{\mathbf{i}} L_{j'}, \mathbf{\Lambda}_n^{\mathbf{p}'} > \mathrm{I}_n^{\mathbf{p}'}}} \right) \right], \tag{8}$$

where the double bars on a quantity indicate that its length is extended so as to obtain a circular convolutional form and then zero-padded to obtain vectors of length 2^k for applying the fast Fourier transform (FFT), FFT^{-1} denotes the inverse fast Fourier transform, and $\mathsf{MASK}_{\mathbf{i}}$ is the array mask restricting the result to array elements within the array boundary.

3 NUMERICAL RESULTS

To assess the interpolation accuracy, the method is tested on an array of 5x5 dipoles in free space, each discretized using 24 triangles and RWG basis functions. The dipoles are illuminated by a plane wave with E field along the dipole axis and at a frequency $f = 380$MHz. The dipoles have length $l = 0.494\lambda$ and width $w = 0.025\lambda$. The average percent error in the current at the center of each dipole of the array is plotted versus the Lagrange polynomial interpolation order in Fig. 5. The error at

each element is calculated relative to a reference solution using an element-by element MoM scheme, and then averaged over all the elements. The computation time per iteration taken by the BiCGstab iterative method increases with interpolation order since more terms are involved in each FFT matrix-vector multiplication.

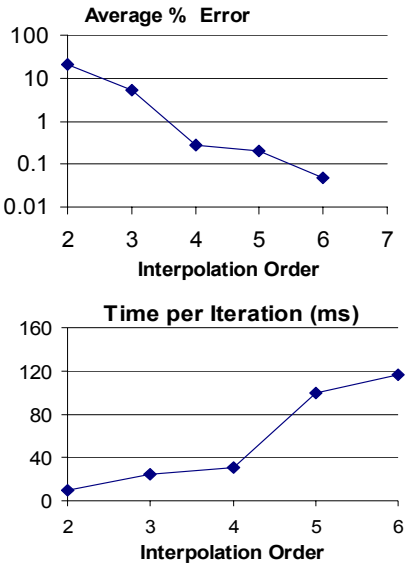

Figure 5: Average error in dipole currents and time per iteration vs. interpolation order.

The FAIM method has been tested on a 20×20 array of printed dipoles on a grounded dielectric substrate. The geometrical configuration is that of [3], where it was shown that the infinite array has a scan blindness near 45 degrees. However, we note in Fig. 6 that while the reflection coefficient of the center element in the finite array is nearly the same as that of the infinite array, the average reflection coefficient of the finite array is somewhat lower near the scan blindness angle due to edge effects.

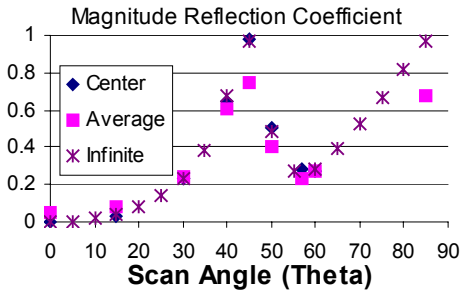

Figure 6: Magnitude of dipole reflection coefficient for infinite array compared to the average and center element values for a 20 × 20 array.

The finite array was also treated using an "exact" MoM calculation utilizing an accelerated but exact fill. The average error in currents between the FAIM and MoM calculation was less than a 0.1%, while the FAIM calculation ran about 7 times faster.

4 CONCLUSIONS

The FAIM method is developed for arrays with arbitrary geometries and is summarized by the following steps. An array mask function is used to determine the arbitrary array contour and to specify the domain for which the Green's function is interpolated using Lagrange polynomials to accelerate the matrix fill; an FFT is then used to accelerate the matrix-vector products in an iterative solver. Preliminary results show the effectiveness of the method for large array problems.

The name FAIM (faster AIM) arises from the fact that the method is similar to the AIM method, but with the difference that Lagrange interpolating polynomials are used to approximate the Green's function using relatively fewer interpolation points per cell. Further, the bases are projected onto the interpolation polynomials in contrast to determining their multipole moments relative to the interpolation grid. Preliminary numerical simulations suggest that fewer integration points are required to keep the same accuracy compared to the original AIM method.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

References

[1] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large scale electromagnetic Scattering and Radiation Problems," *Radio Science*, v. 31, pp. 1225-1251, 1996.

[2] F. Capolino, D. R. Wilton, and D. R. Jackson, "Physical Preconditioning for Numerical Modeling of Large Periodic Arrays", XXVIIth *URSI General Assembly*, Maastricht, The Netherlands, 17-24 Aug. 2002.

[3] D. Pozar, D. Schaubert, "Scan Blindness in Infinite Phased Arrays of Printed Dipoles", *IEEE Transactions on Antennas and Propagation*, v. AP-32, pp.602-610, June 1984.