MoM Solutions for Large Planar Arrays of Three-Dimensional Elements: a Faster AIM Method

B. J. Fasenfest1, F. Capolino2, D.R. Wilton1, D.R. Jackson1, and N. Champagne3

1) Dept. of Electrical and Comp. Eng., University of Houston, Houston, 77204-4005 TX, USA.
2) Dept. of Information Engineering., University of Siena, 53100 Siena, Italy.
3) Lawrence Livermore National Lab., Livermore, CA, USA.

Abstract – A modification of the AIM (Adaptive Integral Method) method based on a different interpolation scheme using Lagrange polynomials and the Toeplitz properties of periodic arrays is presented here. Preliminary results show the effectiveness of the modified AIM method for three-dimensional array elements in free space. Both fill and solve times within a MoM method are improved with respect to more standard MoM solvers.

INTRODUCTION

Numerical modeling analysis of large arrays is still a challenging problem that has recently been tackled by several researchers using various solution methods. Computational resources needed for solving large array problems by the method of moments (MoM) include large times for matrix filling and solving as well as large memory requirements to store the MoM matrices.

In this work we present a modification of the AIM method [1] that is able to efficiently treat large arrays while still maintaining the generality of standard MoM solvers with respect to excitations, array contour geometries, missing or defective array elements, etc. A different approach based on AIM has been recently presented by other researchers in [2]. Another general numerical scheme has been introduced in [3] that uses the fast multipole method to determine the couplings between periodic boxes containing the array elements. There, each box is analyzed using the finite element method. The general scheme presented in this summary is that of the adaptive integral method (AIM) developed in [1], with the significant difference that the Green’s function is approximated using Lagrange interpolating polynomials over array cells. Furthermore, the projection of the basis functions onto the Lagrange polynomials need be performed only once for a single cell because of the assumption of identical array elements. Memory storage, fill time and solve time are significantly reduced by using the present technique.

THE FASTER AIM METHOD

The FAIM (faster AIM) method described in the following is applicable to arrays with arbitrary boundaries. The method is called “faster” mainly for two reasons: (1) it seems that a coarser grid may be used to achieve a given level of accuracy with respect to the original AIM in [1]; and (2) it takes advantage of the reusability properties of periodic arrays of identical elements. The array boundary is defined by the vertices of a closed, piecewise linear curve, which in turn defines an array mask indicating where array elements are present. The mask is generated using a winding number test to determine whether or not an element lies within the array boundary. Typical array boundaries are represented in Fig. 1(a). The mask may also be used to specify missing or defective elements when their effects are under investigation. The displacement between the p'th and p’th array cells is \( (p'_1 - p_1)s_1 + (p'_2 - p_2)s_2 \), where \( s_1 \) and \( s_2 \) are two arbitrary lattice vectors lying in the \( xy \) plane. (Bold letters denote both vectors and double indices; a caret denotes unit vectors.) The vector \( r = p + z\hat{z} \), with \( p = x\hat{x} + y\hat{y} \), is in cell \( p = (p_1, p_2) \) and \( r = p' + z'\hat{z} \) is in cell \( p' = (p'_1, p'_2) \). The array elements within cells \( p \) and \( p' \) are discretized, for instance, using standard RWG basis functions, with primed and unprimed indices denoting source and testing functions, respectively (Fig. 1(c)). Then, a matrix mask (Fig. 1(b)) is synthesized that indicates permissible cell index separations \( p - p' \) between pairs of interacting elements in the array; it is the locus of elements covered by all possible translations of the array mask about a fixed element. I.e., the matrix mask tabulates all possible interactions between elements in the array mask in terms of their separation indices from a central array element. Also shown in Fig. 1(b) is the FFT domain obtained by finding the bounding box of the matrix mask and padding it to the next power of two in each lattice dimension.
A. 

**EFIE Formulation and Lagrange Green’s Function Interpolation (FAIM Filling Acceleration)**

Since we deal with conducting array elements in free space, the MoM is constructed around the EFIE, discretized and written symbolically in the compact form

\[
\begin{bmatrix}
Z_{pp}^m \\
I_p^m
\end{bmatrix} =
\begin{bmatrix}
V_p^m
\end{bmatrix},
\quad
Z_{pp'}^m = -\langle \mathbf{N}_p^m(\mathbf{r}); \mathbf{G}(\mathbf{r} - \mathbf{r}', z, z'); \mathbf{N}_p'(\mathbf{r}') \rangle
\]

(1)

where \( \mathbf{N}_p^m \) and \( \mathbf{N}_p' \) are RWG basis and testing functions, respectively, the right hand side \( V_p^m = \langle \mathbf{N}_p^m ; \mathbf{E}^{inc} \rangle \) is the projection of the incident field onto the RWG basis functions, and \( \mathbf{G}(\mathbf{r} - \mathbf{r}', z, z') \) is the dyadic Green’s function. For elements separated by at least one array cell, i.e., for different \( p \) and \( p' \), we approximate the Green’s function \( \mathbf{G}(\mathbf{r} - \mathbf{r}', z, z') \) via Lagrange polynomial interpolation, with interpolation points \( \mathbf{g}_{i'j',i,j}^m \). The coupling between the \( m \) and \( n \) elements in the matrix block \( Z_{pp'}^m \) corresponding to array cell \( p \) and source cell \( p' \) may be represented as (tilde ~ denotes an approximated block)

\[
Z_{pp'}^m \approx \tilde{Z}_{pp'}^m \equiv - \sum_{i',j',i,j} \langle \mathbf{N}_p^m(\mathbf{r}); L_i(\mathbf{p})L_j(z) \rangle \cdot \mathbf{g}_{i'j',i,j}^m \cdot \langle \mathbf{N}_p'(\mathbf{r}'); L_i'(\mathbf{p}')L_j'(z') \rangle
\]

(2)

where the indices on \( \mathbf{g}_{i'j',i,j}^m \) denote sampled values of the coordinates and \( L_i(\mathbf{p}) \) and \( L_j(z) \) are 2D and 1D Lagrange interpolation polynomials defined on the interpolation points within a cell. It is significant that in the evaluation of the interaction \( Z_{pp'}^m \) between two array cells \( p \) and \( p' \), 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.

---

**Figure 1:** (a) The array mask describes the location of array elements within a rectangular bounding box. (b) Matrix mask for a hexagonal array obtained by translating the array mask about one array element of the array, 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. (c) Array cell index definitions and arbitrary skew lattice vectors. 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. There are \( N \) basis functions within each array cell.
B. FAIM Solution Acceleration. Fast Computation of Matrix-Vector Products

The form (2) of the mutual coupling 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. With good accuracy, the original discretized EFIE in (1) is thus rewritten as
\[
\Delta Z_{pp}^{mn} \mathbf{I}^p + \Delta Z_{pp}^{mn} \mathbf{I}^p = \Delta \mathbf{V}^p
\]
where the block Toeplitz difference matrix \( \Delta Z_{pp}^{mn} = Z_{pp}^{mn} - \bar{Z}_{pp}^{mn} \) is taken as zero for elements satisfying \( |p-p| \geq c \) (c was unbolded here) and is hence sparse. To evaluate the matrix/vector product, we note that the product \( \Delta \mathbf{V}^p \) can be performed quickly since \( \Delta Z_{pp}^{mn} \) is sparse, whereas \( \Delta Z_{pp}^{mn} \mathbf{I}^p \) is of convolutional form and can be evaluated quickly using a 2D FFT as follows:

\[
\left[ \Delta Z_{pp}^{mn} \right] \mathbf{I}^p = \sum_{i,j} < \mathcal{A}_i, L_j > \, \text{MASK}, \text{FFT}^{-1} \left( \text{FFT}, \mathcal{Q}_{i,j}^{\mathbf{I}^p} \right) \text{FFT} \left( \sum_{p \in \mathcal{P}} < \mathcal{A}_i, L_j, \mathcal{A}_p > \mathbf{I}^p \right)
\]

(3)

where the double bars over 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). \( \text{FFT}^{-1} \) denotes the inverse fast Fourier transform, and \( \text{MASK} \) is the array mask restricting the result to array elements within the array boundary. The degree to which (3) approximates (1) depends on how many elements are chosen to be different from zero in \( \Delta Z_{pp}^{mn} \) (also called the strong interaction matrix) and on how many interpolation points are used for \( \Delta Z_{pp}^{mn} \mathbf{I}^p \). It has been observed that, with good accuracy, \( \Delta Z_{pp}^{mn} \) may be set to zero for elements separated by at least a wavelength.

**NUMERICAL RESULTS**

Numerous results will be presented to demonstrate the efficiency of the FAIM method. Results for structures such as phased arrays of printed dipoles show that very significant speed-ups in the matrix fill time and the solution time (related to the matrix-vector inner product calculation time) can be obtained by using the FAIM method, compared with direct MoM solutions or with block conventional methods, or with the conventional AIM method.

**CONCLUSIONS**

A faster AIM method is developed for arrays with arbitrary geometries. The method is similar to the traditional 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. Numerical simulations suggest that fewer integration points are required to maintain the same accuracy as compared to the original AIM method.

**REFERENCES**

