A Unique Single-Step Algorithm for Time-Stepping Electromagnetic Fields

Russell F. Bowers and Chi H. Chan

Abstract—**A unique algorithm for time-stepping electromagnetic fields is developed that saves a significant amount of computer resources. It is applied to two-dimensional scattering problems and can be applied to any type of grid and finite difference scheme that uses the Yee time-stepping algorithm. The two-step finite difference equations are reduced to a single step which bypasses many redundant calculations. CPU and memory savings are presented.**

I. Introduction

THE majority of finite difference time domain (FDTD) calculations that are performed in the electromagnetics community use the Yee, or leapfrog, time-stepping algorithm [1] using a square or rectangular lattice. It is shown here how the Yee time-stepping method may be reduced from two steps to one step. Not only is this method applicable to rectangular grids, it will be shown how it can be applied to conformal grids as well. First, equations will be shown for conformal triangular element grids in two dimensions, after which rectangular grids are addressed. Computer CPU improvement over the conventional method is shown for both types of grids as well as memory savings for two dimensional problems.

II. Analysis

Since two-dimensional scattering problems are being addressed, Maxwell's equations can be decoupled into the two sets of equations for TM and TE polarizations. To simplify the discussion, the formulation of the equations will be limited to TM polarization. TE polarization can be obtained by duality. To implement Maxwell's equations on a conformal triangular grid, a combination of the differential and integral forms are used [2]. For the triangular discretization as in [3] and [4], the discrete source free equations governing TM polarization are

$$E_j^n = \frac{\delta t}{\epsilon_0 S_j} \sum_{i=1}^{I_j} H_{k(j,i)}^{n-\frac{1}{2}} \delta l_{k(j,i)} + E_j^{n-1} \tag{1}$$

$$H_k^{n+\frac{1}{2}} = \frac{\delta t}{\mu_0} \frac{E_{j2}^n - E_{j1}^n}{\delta r_{j,i}} + H_k^{n-\frac{1}{2}}. \tag{2}$$

The index n is a temporal index; j and k are spatial indices. The variable S_j is the area enclosed by the contour integral around E_j. Since the grid is generated in a random fashion, the H_k fields adjacent to E_j are not necessarily in sequential order, thus the unusual notation of indices involving i, j and k in the above equations. The expression $k(j,i)$ means that an index k corresponds to the ith H field surrounding E_j, and there are I_j such H fields around E_j. The term $\delta l_{k(j,i)}$ is the path along which $H_{k(j,i)}$ is constant on the contour integral. The terms E_{j1}^n and E_{j2}^n are the two E fields on each side of H_k, and $\delta r_{j,i}$ is the distance between them.

Since the conformal grid is irregular, the geometrical parameters of (1) and (2) are not the same for a given set of j, k. The most efficient way to handle this in terms of computer time and memory is to write the equations in matrix form. We will show later that matrix representation does not adversely affect computer memory requirements. The matrix form of the linear equations (1) and (2) is

$$\mathbf{e}^n = \mathbf{A}\mathbf{h}^{n-\frac{1}{2}} + \mathbf{e}^{n-1} \tag{3}$$

$$\mathbf{h}^{n+\frac{1}{2}} = \mathbf{B}\mathbf{e}^n + \mathbf{h}^{n-\frac{1}{2}} \tag{4}$$

where

$$\mathbf{e}^n = (E_1^n, E_2^n, E_3^n, \cdots, E_J^n)^T,$$

and

$$\mathbf{h}^{n+\frac{1}{2}} = \left(H_1^{n+\frac{1}{2}}, H_2^{n+\frac{1}{2}}, H_3^{n+\frac{1}{2}}, \cdots, H_K^{n+\frac{1}{2}}\right)^T.$$

The elements of $\mathbf{A}$ and $\mathbf{B}$ are

$$a_{j,k(j,i)} = \frac{\delta t\, \delta l_{k(j,i)}}{\epsilon_0 S_j} \tag{5}$$

and

$$b_{k,j2} = -b_{k,j1} = \frac{\delta t}{\mu_0 \delta r_{j,i}} \tag{6}$$

where the index convention is the same as before. The matrix $\mathbf{A}$ contains the boundary conditions for the scatterer and, as such, is polarization-dependent. The matrix $\mathbf{B}$ does not depend on polarization.

The matrices $\mathbf{A}$ and $\mathbf{B}$ are very sparse and only the nonzero elements are stored. The number of nonzero elements, I_j, of the jth row of $\mathbf{A}$ correspond to the number of discrete H_k surrounding E_j. The number of nonzero elements of any row of $\mathbf{B}$ is two, because each H_k depends on only two E_j variables. Given the relationship between the two elements of each row of $\mathbf{B}$ in (6), only one of them needs to be stored.

By employing the matrix product $\mathbf{AB}$, the problem size can be signifcantly reduced in terms of computer time and

Manuscript received June 23, 1993.

R. F. Bowers is with the Boeing Company, Box 3707 M/S 4M-09, Seattle, WA 98124.

C. H. Chan is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195.

IEEE Log Number 9212837.

1051-8207/93$03.00 © 1993 IEEE

memory. If it is assumed that the initial condition is specified at $t = 0$ by $\mathbf{e}^0$ and that all fields at $t < 0$ are zero, then by using (4),

$$\mathbf{h}^{\frac{1}{2}} = \mathbf{B}\mathbf{e}^0 \tag{7}$$

$$\mathbf{h}^{\frac{3}{2}} = \mathbf{B}\mathbf{e}^1 + \mathbf{h}^{\frac{1}{2}}. \tag{8}$$

Substituting (7) into (8) gives

$$\mathbf{h}^{\frac{3}{2}} = \mathbf{B}\left(\mathbf{e}^1 + \mathbf{e}^0\right). \tag{9}$$

Continuing on to observe the general form that the equations take as the nth time step is approached by using the same type of substitutions,

$$\begin{aligned}\mathbf{h}^{\frac{5}{2}} &= \mathbf{B}\mathbf{e}^2 + \mathbf{h}^{\frac{3}{2}}\\ &= \mathbf{B}\mathbf{e}^2 + \mathbf{B}\left(\mathbf{e}^1 + \mathbf{e}^0\right)\\ &= \mathbf{B}\left(\mathbf{e}^2 + \mathbf{e}^1 + \mathbf{e}^0\right).\end{aligned}$$

Finally, employing (3) gives

$$\begin{aligned}\mathbf{e}^3 &= \mathbf{A}\mathbf{h}^{\frac{5}{2}} + \mathbf{e}^2\\ &= \mathbf{A}\mathbf{B}\left(\mathbf{e}^2 + \mathbf{e}^1 + \mathbf{e}^0\right) + \mathbf{e}^2\\ &\vdots\\ \mathbf{e}^n &= \mathbf{A}\mathbf{B}\left(\mathbf{e}^{n-1} + \mathbf{e}^{n-2} + \mathbf{e}^{n-3} + \cdots + \mathbf{e}^0\right) + \mathbf{e}^{n-1}\end{aligned} \tag{10}$$

which has no dependence on $\mathbf{h}^{n+\frac{1}{2}}$ for any n. Using a shorthand notation of $\psi_{\mathbf{e}^{n-1}}\psi_{\mathbf{e}}{}^{n-1} = \sum_{i=0}^{n-1} \mathbf{e}^i$ and **C** = **AB**, and substituting into (10),

$$\mathbf{e}^n = \mathbf{C}\psi_{\mathbf{e}^{n-1}} + \mathbf{e}^{n-1}. \tag{11}$$

The two new terms of (11), $\psi_{\mathbf{e}^{n-1}}$ and **C**, have significant implications. First, $\psi_{\mathbf{e}^{n-1}}$ is simply a running sum over time of all E_j fields in the problem and is easily implemented. It is true that it is a new term that takes up memory and time to calculate; however, more memory and time are saved by eliminating the intermediate calculations for H_k. The nonzero elements of **C** are

$$\left.\begin{aligned}c_{j,m} = &\quad (c\delta t)^2 \frac{\delta l_{j,m}}{\delta r_{j,m} S_j}\\ c_{j,j} = &\quad -\sum_{m=1}^{J} c_{j,m}\end{aligned}\right\} m \neq j. \tag{12}$$

where c is the speed of light in free space. Equation (12) states that the diagonal element is the negative of the sum of all the off-diagonal elements on the same row.

The lack of $\mathbf{h}^{n+\frac{1}{2}}$ in (11) is because the dependence has shifted from the surrounding H_k variables to the surrounding E_j variables plus an additional self term. Thus, the number of nonzero terms of the jth row of **C** is only one more (the self term) than the jth row of **A**.

In practice, the triangular conformal grid is surrounded by a rectangular grid [4] for the reason of implementing the radiation boundary condition, e.g. Liao's boundary condition [5]. In a typical problem the ratio of the number of E_j in the triangular grid to the number of E_j in the rectangular grid is approximately 5%. If we let J' represent the number of all nodes, then $J \approx .05J'$ and $K \approx .05K'$. Since the matrix **C** has J rows, then **C** has $2K + J$ nonzero elements compared to $4K$ nonzero elements of **A** and **B** combined. It is typical that the number of H_k fields in a scattering problem is approximately twice as many as the E_j fields ($K' \approx 2J'$). Remembering that only K elements need to be stored for **B**, Table I compares the required program array sizes for the two matrix method and the single matrix method. This table also includes pointer arrays for making sense of storing a sparse matrix in a one-dimensional array. It is obvious, from Table I, that the majority of the savings comes from reducing the number of pointer arrays needed for the matrices.

TABLE I
MEMORY REQUIREMENTS FOR TWO MATRIX AND SINGLE MATRIX METHODS FOR A TWO-DIMENSIONAL CONFORMAL GRID

Variable array	Two matrix method	Single matrix method
$\vec{E}$and$\vec{H}$ fields	$3J'$	J'
ψ	0	J'
A	$0.2J'$	$0.25J'$
B	$0.1J'$	0
Sparse matrix pointers	$8.05J'$	$4.05J'$
Total	$11.35J'$	$6.3J'$

$J = .05J'$

TABLE II
MEMORY REQUIREMENTS FOR TWO MATRIX AND SINGLE MATRIX METHODS FOR A TWO-DIMENSIONAL RECTANGULAR GRID

Variable array	Two matrix method	Single matrix method
$\vec{E}$and$\vec{H}$ fields	$3J'$	J'
ψ	0	J'
Total	$3J'$	$2J'$

This explicit single matrix equation method may be applied to any type of grid, including rectangular. The application of this method to a rectangular grid is much easier because the grid spacing is regular; therefore, all rows of the matrix have the same nonzero values. Only three coefficients need to be calculated for all rectangular elements. In particular, the coefficients are $(c\,\delta t/\delta x)^2$ for adjacent fields in the $\pm\hat{x}$ direction, $(c\,\delta t/\delta y)^2$ for adjacent fields in the $\pm\hat{y}$ direction and $-2(c\,\delta t)^2(1/\delta x^2 + 1/\delta y^2)$ for the self term. The memory requirement comparison for a two-dimensional rectangular grid is shown in Table II.

It should be noted that for a three-dimensional rectangular element problem, there will be no memory savings. The running sum $\psi_{\mathbf{e}}^{n-1}$ will take the memory freed up by the three field vectors being eliminated. For a three-dimensional conformal grid, the memory savings will only come from eliminating the need for pointer arrays used to delineate the H field dependencies.

Since the single matrix approach avoids having to explicitly calculate all H fields, a significant savings in computer CPU time should be realized. The timing results of the implementation of this explicit single matrix equation method compared to the two matrix method is shown in Table III. The CPU times are the average cost of updating a single E. This is obtained by the expression CPU$/(J' \times N)$. Conformal grid as well as rectangular grid performance is shown. Note that the rectangular element program runs faster than the conformal element program, because it is a simple program that does not use complicated array indexing and is "hard-

TABLE III
CPU TIME VS. MATRIX METHODS AND GRID TYPES ON A SUN SPARC-2 WORKSTATION

Grid	$\frac{CPU}{J'\times N}$	
	Two matrix	Single matrix
Conformal	8.0 μs	4.7 μs
Rectangular	3.6 μs	1.7 μs

coded" for only two shapes of scatterers. The conformal grid program handles arbitrarily shaped scatterers and, as such, has more overhead. However, rectangular element solutions cannot converge quickly enough for scattering phenomena that are difficult to predict, such as sharp edges and creeping wave phenomena [4].

It is common that intermediate information needs to be gathered such as surface currents or levels in the far field. This information requires knowing the H fields at certain locations, the number of which is typically far less than the original number of H fields. They may be quickly and easily calculated at the selected areas without having to calculate all of the other H fields.

The use of this single matrix approach does not deteriorate the accuracy of the calculations. There are no approximations in going from (3) and (4) to (11). Implementation of material parameters would simply change the matrix elements of (12); however, the algorithm would become more complicated if frequency-dependent materials are used.

III. CONCLUSION

Though there are other single-step methods, this single-step method for updating time-dependent fields is a worthwhile contribution to the techniques of FDTD electromagnetic field calculations. Through application of this method, an efficient algorithm is produced that makes conformal grid calculations much more worthwhile while at the same time preserving the properties of leapfrog time-stepping [1]. Representing the FDTD equations in matrix form presents a simple method to estimate the stability condition of a problem [4]. The single matrix representation also allows for easier portability and better performance of FDTD problems on parallel computers, especially ones that have fast vector operations.

REFERENCES

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Ant. and Prop.*, vol. AP-14, pp. 302–307, May 1966.

[2] A. Taflove, K. R. Umashankar, B. Beker, F. Harfoush, and K. S. Yee, "Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens," *IEEE Trans. Ant. and Prop.*, vol. 36, pp. 247–257, Feb. 1988.

[3] C. F. Lee, R. T. Shin, J. A. Kong, and B. J. McCartin, "Finite difference time domain techniques for two dimensional triangular grids," in *Proc. Progress in Electromag. Res. Symp.*, pp. 189–190, July 1989.

[4] R. F. Bowers, "An explicit single matrix equation method for time marching electromagnetic fields and its application to conformal grids," master's thesis, University of Washington, Seattle, June 1993.

[5] W. C. Chew, *Waves and Fields in Inhomogeneous Media.* New York: Von Nostrand Reinhold, 1990, pp. 251–256.