Practical Comparison of Finite Difference Time Domain Schemes

*Joe LoVetri**
Systems Integration Lab
Institute for Information Technology
National Research Council,
Ottawa, Ontario, Canada, K1A 0R8.

Marek S. Wartak
Department of Physics and Computing
Wilfrid Laurier University,
Waterloo, Ontario, Canada,
N2L 3C5.

Abstract

Practical finite difference methods for the time domain Maxwell's equations are considered. As the basis of this analysis, Maxwell's equations are expressed as a system of conservation laws. Practical considerations, such as computational efficiency and memory requirements, for the methods are discussed. The example of the penetration of electromagnetic energy through a shield with a thick gap in two dimensions is used to compare the methods.

1.0 Formulation of Maxwell's Equations as Conservation Law

Maxwell's equations can be cast as a system of hyperbolic conservation laws of the form (see [Shankar 89, 90]), $u_t^j + \partial_x E^j + \partial_y F^j + \partial_z G^j = S^j$, $j = 1, \ldots, n$, where the solution vector {u} is given by $[B, D]^T$, the source vector {S} and the flux vectors {E}, {F} and {G} are given by $\{E(x,t)\} = [\hat{x} \times eD, -\hat{x} \times mB]^T$, $\{F(x,t)\} = [\hat{y} \times eD, -\hat{y} \times mB]^T$, $\{G(x,t)\} = [\hat{z} \times eD, -\hat{z} \times mB]^T$, and $\{S(x,t)\} = [0, -J]^T$.

2.0 Finite Difference Schemes

An approximate solution to the general three dimensional conservation law,

$$u_t + E_x + F_y + G_z = S,$$

with initial conditions $u(x, y, z, 0) = g(x, y, z)$ is sought on the rectangular lattice

$$\Omega_{jkln} = \left\{ (x, y, z, t) \middle| \begin{array}{ll} x_{j-1/2} \le x \le x_{j+1/2} & z_{l-1/2} \le z \le z_{l+1/2} \\ y_{k-1/2} \le y \le y_{k+1/2} & t^n \le t \le t^{n+1} \end{array} \right\},$$

where $x_j = j\Delta x$, $y_k = k\Delta y$, $z_l = l\Delta z$, $t^n = n\Delta t$ and j, k, l, and n are integers. The initial conditions are approximated on Ω_{jkl0} by $u_{jkl}^0 = g_{jkl} = g(x_j, y_k, z_l)$, and a finite difference procedure determines the solution at a new time $t = t^n$ from a previous time $t = t^{n-1}$. A general finite difference scheme can be written as $u_{jkl}^{n+1} = Q(u_{jkl}^n)$, or in a two step form, for $n \ge 0$ where Q is now a polynomial in the backward and forward shift operators E_- and E_+ for each space dimension. The qualities of *consistency*, *stability*, and *convergence* are used to describe a specific difference scheme.

2.1 Leap Frog Scheme

The leap-frog scheme is the simplest to understand and implement [Yee 66]. This is a two step scheme on an interlaced mesh. The interlacing is specific to Maxwell's equa-

91CH3036-1/0000-0418 $1.00 ©1991 IEEE

tions and may not occur for other equations. The solution vector component locations are depicted in Fig. 1, where the **D** and **B** fields are represented at different times t^n and $t^{n-1/2}$.

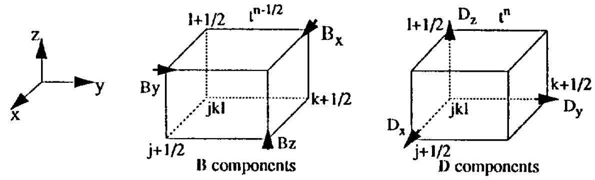

Figure 1. Location of components for leap-frog scheme

Thus the solution vector is discretized as

$$u_{jkl}^n = \begin{bmatrix} B_{jkl}^n \\ D_{jkl}^n \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} (B_x)_{j,k+1/2,l+1/2}^{n-1/2} \\ (B_y)_{j+1/2,k,l+1/2}^{n-1/2} \\ (B_z)_{j+1/2,k+1/2,l}^{n-1/2} \end{bmatrix} & \begin{bmatrix} (D_x)_{j+1/2,k,l}^n \\ (D_y)_{j,k+1/2,l}^n \\ (D_z)_{j,k,l+1/2}^n \end{bmatrix} \end{bmatrix}^T,$$

and the scheme can be written as

$$D_{jkl}^{n+1} = D_{jkl}^n + \frac{\Delta t}{2}(s_{jkl}^{n+1} + s_{jkl}^n) - \rho_x(E_j^n - E_{j-1}^n) - \rho_y(F_k^n - F_{k-1}^n) - \rho_z(G_l^n - G_{l-1}^n),$$

$$B_{jkl}^{n+1} = B_{jkl}^n + \frac{\Delta t}{2}(s_{jkl}^{n+1} + s_{jkl}^n) - \rho_x(E_{j+1}^n - E_j^n) - \rho_y(F_{k+1}^n - F_k^n) - \rho_z(G_{l+1}^n - G_l^n),$$

where B_{jkl}^n implies u_{jkl}^n for any of the **B** components and D_{jkl}^n implies u_{jkl}^n for the any of the **D** components and $\rho_\xi = \Delta t/\Delta\xi$. The notation for the flux vectors **E**, **F**, and **G** is obvious, for example

$$E_j^n = E(u_{jkl}^n) = \begin{bmatrix} \begin{bmatrix} 0 \\ (-eD_z)_{j,k,l+1/2}^n \\ (eD_y)_{j,k+1/2,l}^n \end{bmatrix} & \begin{bmatrix} 0 \\ (mB_z)_{j+1/2,k+1/2,l}^{n-1/2} \\ (-mB_y)_{j+1/2,k,l+1/2}^{n-1/2} \end{bmatrix} \end{bmatrix}^T.$$

2.2 Lax-Wendroff: Two Step Rotated Richtmyer Scheme

The formulation of the Lax-Wendroff method in two step form was first presented by Richtmyer [Richtmyer 67]. The analysis of these schemes has been investigated by Wilson [Wilson 72] with the rotated Richtmyer scheme given as

$$u_{jkl}^{n*} = (\mu_x\mu_y\mu_z)u_{jkl}^n - \frac{1}{2}(\rho_x\mu_y\mu_z\delta_x E_{jkl}^n + \rho_y\mu_x\mu_z\delta_y F_{jkl}^n + \rho_z\mu_x\mu_y\delta_z G_{jkl}^n)$$

$$u_{jkl}^{n+1} = u_{jkl}^{n*} - \left[\rho_x\mu_y\mu_z\delta_x E_{jkl}^{n*} + \rho_y\mu_x\mu_z\delta_y F_{jkl}^{n*} + \rho_z\mu_x\mu_y\delta_z G_{jkl}^{n*}\right]$$

where $E_{jkl}^n = E(u_{jkl}^n)$, $E_{jkl}^{n*} = E(u_{jkl}^{n*})$, with similar relations for the flux vectors **F**, and **G**. The symbols μ and δ represent the *averaging* and the *central difference* operators respectively.The two dimensional versions of these schemes can be easily obtained from the above equations by dropping the flux vector G and all difference operators in the z-coordinate direction. Where now the solution vector is discretized as $u^n_{jkl} = \left[B^n_{jkl}\ D^n_{jkl}\right]^T$.

2.3 Upwind Schemes

The upwind difference methods based on the schemes of Warming and Beam (see [Warming 76] and the flux vector splitting techniques of Steger and Warming (see [Steger 81]) are also available. The flux vector splitting can be performed on each of the flux vectors **E**, **F**, and **G** individually based on the Jacobian matrices of each. For example

$$E = T_E(\Lambda_E^+ + \Lambda_E^-)T_E^{-1}u = (A_E^+ + A_E^-)u = E^+ + E^-,$$

and the Warming and Beam scheme predictor can be written in two dimensions as

$$\overline{u^{n+1}_{jkl}} = u^n_{jkl} - \rho_x(\nabla_x(E_j^+)^n + \Delta_x(E_j^-)^n) - \rho_y(\nabla_y(F_k^+)^n + \Delta_y(F_k^-)^n),$$

and the corrector as

$$u^{n+1}_{jkl} = \frac{1}{2}(u^n_{jkl} + \overline{u^{n+1}_{jkl}}) - \frac{\rho_x}{2}(\nabla_x(E_j^+)^{n+1} + \nabla_x^2(E_j^+)^{\overline{n+1}} - \Delta_x^2(E_j^-)^{\overline{n+1}} + \Delta_x(E_j^-)^{n+1})$$
$$- \frac{\rho_y}{2}(\nabla_y(F_k^+)^{n+1} + \nabla_y^2(F_k^+)^{\overline{n+1}}) + \frac{\rho_y}{2}(\Delta_y^2(F_k^-)^{\overline{n+1}} - \Delta_y(F_k^-)^{n+1}),$$

where $E_j = E_{jkl}$, $F_k = F_{jkl}$, and $G_l = G_{jkl}$. If *numerical* fluxes for each of the flux vectors are denoted by h_E, h_F, and h_G, the forward and backward flux differences become

$$\Delta E_j^- = \delta E^-_{j+1/2} = ((h_E)_{j+1/2} - E(x_j)),\quad \nabla E_j^+ = \delta E^+_{j-1/2} = -((h_E)_{j-1/2} - E(x_j)),$$

which can be used in the above predictor corrector scheme. The numerical fluxes can now be written by making use of the method of characteristics (see [Shankar 90]).

3.0 Results and Conclusions

One and two dimensional versions of the three finite difference schemes discussed above were implemented in the C programming language on a Sun SPARCstation™ using the techniques described in [LoVetri 90]. The three algorithms are compared for efficiency in terms of CPU time and memory requirements for the same electromagnetic problem. That is, a Gaussian pulse incident on a thick conducting shield with a gap is used as the test problem.

The Lax-Wendroff and the upwind schemes require twice and three times as much memory as the leap frog scheme respectively. This is due to the fact that the Lax-Wendroff method is a two-step method and temporary memory is required for the storage of the intermediate calculation. Only the intermediate vectors are required to determine the solution at the next time step so that the old solution can be immediately overwritten. In the Upwind scheme, both the intermediate vector and the solution vector at the previous time step is required and thus the solution cannot be overwritten.

The required CPU time is more difficult to compare due to different Courant number for each scheme. A plot of the CPU time required to reach the same *absolute* time for the problem is shown in figure 2. Also shown is the CPU time per node in the lattice which contains a disturbance.

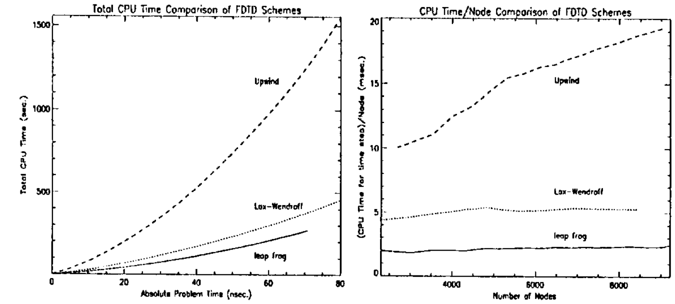

Figure 2. Comparison Total CPU time and Time per node in the 100 X 100 lattice

Thus, it can be seen that the time per node is relatively constant for each method, but is considerably greater for the upwind scheme. The advantage of the last two schemes is that the components of the solution vector are not staggered over an interlaced mesh, thus making it more accurate at the boundaries where boundary conditions are imposed. It is also apparent when comparing the solutions generated from these algorithms, that the upwind algorithm resolves discontinuities in the medium better than the other techniques (e.g. discontinuous dielectric boundaries inside the computational domain). More test problems will be tried in order to more fully compare these algorithms.

References

[LoVetri 90] LoVetri, J., Costache, G. I., *Efficiency Issues in the Implementation of Finite Difference Time Domain Codes*, submitted this symposium proceedings, 1990.

[Richtmyer 67] Richtmyer, R. D., and Morton, K. W., **Difference Methods for Initial-Value Problems**, Interscience Publishers, New York, 1967.

[Shankar 90] Shankar, V., Mohammadian, A. H., Hall, W. F., *A Time-Domain Finite-Volume Treatment for the Maxwell Equations*, Electromagnetics, Vol. 10, No. 1-2, pp 127 - 145, 1990.

[Steger 81] Steger, J. L., and Warming, R. F., *Flux Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite-Difference Methods*, J. of Comp. Phys., Vol. 40, pp 263 - 293, 1981.

[Warming 76] Warming, R. F., and Beam, R. M., *Upwind Second-Order Difference Schemes and Applications in Aerodynamic Flows*, AIAA Journal, Vol. 14, No. 9, pp 1241 - 1249, September, 1976.

[Wilson 72] Wilson, J. C., *Stability of Richtmyer Type Difference Schemes in any Finite Number of Space Variables and Their Comparison with Multistep Strang Schemes*, J. Inst. Maths. and Applics., Vol. 10, pp 238 - 257, 1972.

[Yee 66] Yee, S. K., *Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media*, IEEE Trans. on Ant. and Prop., Vol. AP-14, No. 3, pp 302 -307, May, 1966.