# Finite Difference Time Domain Solution of Electromagnetic Scattering on the Hypercube

*Ruel H. Calalo*
*James R. Lyons*
*William A. Imbriale*

Jet Propulsion Laboratory
California Institute of Technology

*ABSTRACT*

Electromagnetic fields interacting with a dielectric or conducting structure produce scattered electromagnetic fields. To model the fields produced by complicated, volumetric structures, the finite difference time domain (FDTD) method employs an iterative solution to Maxwell's time dependent curl equations. Implementations of the FDTD method intensively use memory and perform numerous calculations per time step iteration. We implemented an FDTD code on the California Institute of Technology/ Jet Propulsion Laboratory Mark III Hypercube. This code allows us to solve problems requiring as many as 2,048,000 unit cells on a 32 node Hypercube. For smaller problems, the code produces solutions in a fraction of the time to solve the same problems on sequential computers.

## 1. INTRODUCTION

Several sequential FDTD codes exist. Converting a FDTD code from Lawrence Livermore National Laboratory (LLNL) was our first attempt at producing a parallel FDTD code (1). The modular form and straightforward coding of this program greatly eased the task of parallel decomposition. However, the LLNL code was designed for the analysis of transient currents. A. Taflove provided us with a copy of his FDTD code along with a report describing the FDTD method and its applications to radar cross section (RCS) studies (2). We added to our code the features required to make an RCS code with similar capabilities to Taflove's code. At this point, references to the FDTD code describes our parallel implementation of the finite difference time domain method.

The goal of the FDTD code is to track the propagation of an incident electromagnetic wave into a volume of space containing a dielectric or conducting structure and observe the wave's interaction with the scattering object. Wave tracking is complete when sinusoidal steady state behavior occurs at each lattice cell or after a sufficient number of cycles of the incident field.

FDTD uses a finite difference approximation, in rectangular coordinates, to Maxwell's curl equations to explicitly solve for the fields at the current time step. This iteration scheme replaces the need for a simultaneous solution for all field components (3).

The parallel implementation of FDTD divides the global lattice of discrete field components into blocks of nearly equal dimensions. The code assigns neighboring blocks to nodes directly connected by communication channels. This decomposition scheme assures that each node can perform its field updates with resident information and information communicated by neighboring nodes.

To test the validity of the results from the parallel FDTD code, we compared the reported current on the surface of a perfectly conducting cube with results from a finite difference code obtained from Taflove (2).

## 2. GENERAL THEORY

The equations governing the interaction of electromagnetic radiation with anisotropic dielectric or conducting structures is Maxwell's curl equations (2.1 and 2.2).

$$\overleftrightarrow{\mu}\begin{bmatrix}\partial_t H_x\\ \partial_t H_y\\ \partial_t H_z\end{bmatrix} + \overleftrightarrow{\sigma}_m\begin{bmatrix}H_x\\ H_y\\ H_z\end{bmatrix} = \begin{bmatrix}\partial_z E_y - \partial_y E_z\\ \partial_x E_z - \partial_z E_x\\ \partial_y E_x - \partial_x E_y\end{bmatrix} \qquad (2.1)$$

$$\vec{B} = \overleftrightarrow{\mu}\,\vec{H} \qquad \vec{J}_m = \overleftrightarrow{\sigma}_m\,\vec{H}$$

$$\overleftrightarrow{\varepsilon}\begin{bmatrix}\partial_t E_x\\ \partial_t E_y\\ \partial_t E_z\end{bmatrix} + \overleftrightarrow{\sigma}\begin{bmatrix}E_x\\ E_y\\ E_z\end{bmatrix} = \begin{bmatrix}\partial_y H_z - \partial_z H_y\\ \partial_z H_x - \partial_x H_z\\ \partial_x H_y - \partial_y H_x\end{bmatrix} \qquad (2.2)$$

$$\vec{D} = \overleftrightarrow{\varepsilon}\,\vec{E} \qquad \vec{J} = \overleftrightarrow{\sigma}\,\vec{E}$$

$\mu$, $\sigma_m$, $\varepsilon$, and $\sigma$ are tensors. The FDTD code implements the special case in which these tensors only contain nonzero entries along the diagonal. $\mu(x,y,z)$ is the permeability tensor at coordinates (x,y,z). $\sigma_m$ is the magnetic conductivity. $\varepsilon$ is the permittivity. $\sigma$ is the electric conductivity. B, H, $J_m$, D, E, and J are the usual electromagnetic field quantities.

From the continuum equations, we construct six finite difference equations by first expressing equation 2.1 and 2.2 as separate scalar equations in rectangular coordinates. Next, we employ two-point central differencing in both spatial and temporal coordinates to discretize the scalar equations. References (1) and (2) discuss the mathematical details. The following equation, 2.3, is the finite difference equation for the x component of the magnetic field. The finite difference equation for the other five components have a similar form.

$$\tilde{\mu}_{11}(i,j+1/2,k+1/2)\, H_x^{n+1/2}(i,j+1/2,k+1/2) = \tilde{\mu}^{*}_{11}(i,j+1/2,k+1/2)\, H_x^{n-1/2}(i,j+1/2,k+1/2) + \frac{E_y^n(i,j+1/2,k+1) - E_y^n(i,j+1/2,k)}{\Delta z} + \frac{E_z^n(i,j,k+1/2) - E_z^n(i,j+1,k+1/2)}{\Delta y} \qquad (2.3)$$

$$\tilde{\mu}_{11} = \frac{\mu_{11}}{\Delta t} + \frac{\sigma_{m_{11}}}{2} \qquad \tilde{\mu}^{*}_{11} = \frac{\mu_{11}}{\Delta t} - \frac{\sigma_{m_{11}}}{2}$$

I, j, or k indexes the edges of unit cells. Superscripts on the field components index the discrete time steps. $\Delta y$ and $\Delta z$ indicate the discrete increment in the y and z directions. $\Delta t$ indicates the time step increment per iteration. Other subscripted quantities are tensor elements of $\mu$, $\sigma_m$, $\varepsilon$, or $\sigma$. To obtain accurate results, the size of the unit cell (figure 2.1), the spatial finite difference step, must be a small fraction of the electrical size of the scatterer (2). Once we specify the unit cell size, we determine the time step, the temporal finite difference step, by the Courant stability condition (4).

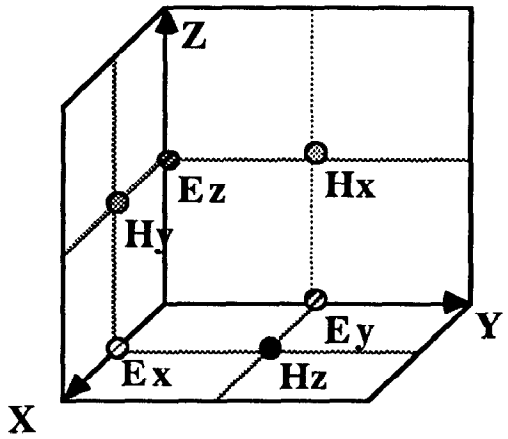



Figure 2.1 Electric and magnetic field components in a typical unit cell.

FDTD constructs a lattice in coordinate space from several unit cells. The discrete electric field components, $E_x$, $E_y$, and $E_z$, lie on the edges of each unit cell. For example, $E_x$ lies on the midpoint of edges in the x direction. The discrete magnetic field components, $H_x$, $H_y$, and $H_z$ occur on the centers of faces of each unit cell. For example, $H_x$ lies on the center of faces perpendicular to the x direction. At discrete electric field locations, FDTD assigns permittivity and electric conductivity. At discrete magnetic field locations, FDTD assigns permeability and magnetic conductivity (3).

The finite difference form of Maxwell's curl equations determine the spatial and temporal update of the discrete field components on the lattice. At time, $n\Delta t$, the magnetic fields, on cell faces and at half a time step back, update neighboring electric field components on cell edges. For example, FDTD uses $H_y$ and $H_z$ components to update $E_x$

components. FDTD, then, increments time by half the time step, $\Delta t$. At time, $(n+1/2)\Delta t$, the new electric field components on cell edges update neighboring magnetic field components on cell centers. FDTD, then, increments time again by half of $\Delta t$. This process continues until the discrete field components have constant maximum amplitudes and constant phase relative to a fixed iteration number (2).

## 3. RADIATION BOUNDARY CONDITIONS

At the lattice boundary we cannot use the FDTD method to update field values, because points outside the lattice would be needed. Therefore, we apply a radiation condition to the scattered electric fields at the lattice truncation planes. Formulas for first and second order radiation conditions, conditions containing first and second derivatives in space and time, and their corresponding central difference expressions are given in reference (5). A discussion of the derivation of radiation conditions from the scalar wave equation is given in reference (6).

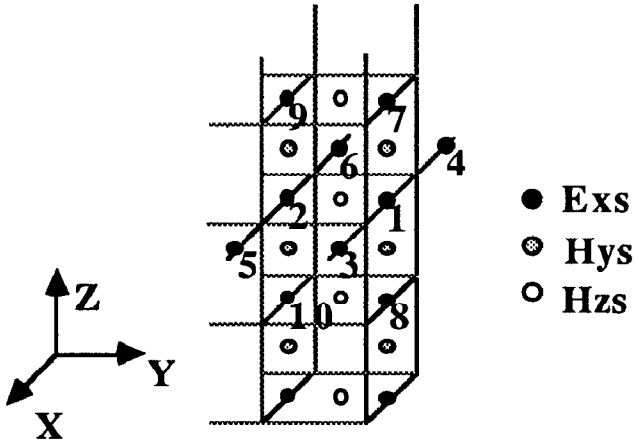



Figure 3.1 $E_X$ points 2 to 10 needed to update $E_X$ point 1 on a truncation plane perpendicular to the y axis.

Figure 3.1 illustrates the update of an x component of the scattered electric field on a truncation plane perpendicular to the y axis and with unit normal vector in the positive y direction. We label this point with the number 1. We cut a section perpendicular to the x axis to illustrate the update of point 1. To update point 1 at time $n\Delta t$, second order correct radiation boundary conditions need the value of the field at points 1 and 2 at time $2\Delta t$ back and time $\Delta t$ back. These conditions also require the value of the field points 3 to 10 at time $\Delta t$ back.

## 4. RADAR CROSS SECTION

Standard radar cross section (RCS) calculations require steady state far fields. However, the FDTD code computes the time evolution of the near fields. Therefore, the FDTD code incorporates plane wave incident field, magnitude and phase calculations for

sinusoidal steady state near fields, and near to far field transformations.

The code evaluates the steady state magnitude and phase in the following manner. The algorithm marches in time through several cycles of the incident field. After several cycles, the fields reach steady state. The code, then, monitors the peaks and valleys of the sinusoidally varying waveform by computing the first and second time derivatives at various points in the lattice. The code computes the magnitudes as one half the difference between peak and valley amplitudes. It computes the phase by recording the time step number at which a peak occurs and subtracting from a reference time step number. The FDTD code converts the resulting time difference to radians by muliplication with the incident field angular frequency.

Equations 4.1 illustrate the computation of the RCS, $\sigma$.

$$\sigma = 4\pi r^2 \frac{E_\theta^* E_\theta + E_\phi^* E_\phi}{E^{i*} E^i} \qquad r \to \infty$$

$$E_\theta = -ik_o \eta_o \left( A_x \cos\theta \cos\phi + A_y \cos\theta \sin\phi - A_z \sin\theta + \eta_o^{-1} \left( -F_x \sin\phi + F_y \cos\phi \right) \right)$$

$$E_\phi = -ik_o \eta_o \left( -\eta_o^{-1} \left( F_x \cos\theta \cos\phi + F_y \cos\theta \sin\phi - F_z \sin\theta \right) - A_x \sin\phi + A_y \cos\phi \right) \tag{4.1}$$

$$\vec{A} = \frac{e^{-ik_o r}}{4\pi r} \iint_s \left( \vec{n} \times \vec{H}^s \right) e^{ik_o r' \cos\xi} ds$$

$$\vec{F} = \frac{e^{-ik_o r}}{4\pi r} \iint_s \left( -\vec{n} \times \vec{E}^s \right) e^{ik_o r' \cos\xi} ds$$

$$r' \cos\xi = \left( x' \cos\phi + y' \sin\phi \right) \sin\theta + z' \cos\theta$$

The variables in the equations are the following: r is the distance of the observation point, r' is the distance of the source point, $k_o$ is the wave number of the incident field, $\eta_o$ equals 376.73 ohms, $\theta$ and $\phi$ are the angles of the observation point, and the $E^s$ and $H^s$ vectors are the complex fields obtained from magnitude and phase evaluations on the integration planes. The integration is over a cubic surface surrounding a volume containing the scatterer. The vector n is the unit outward normal on this volume.

The mathematical details of the near to far field transformations, as well as a more in depth discussion of the second order radiation condition and the magnitude and phase computation is found in reference (2).

## 5. PARALLEL DECOMPOSITION

The FDTD code uses a spatial decomposition of the lattice of unit cells. The program fills the computation space with several unit cells. It, then, divides this global lattice into blocks of nearly equal dimensions. The FDTD code assigns neighboring blocks of the global lattice to Hypercube nodes directly connected by communication channels. This decomposition scheme assures that each node can perform its discrete field updates, described in section 2 above, with resident information and information communicated by neighboring nodes.

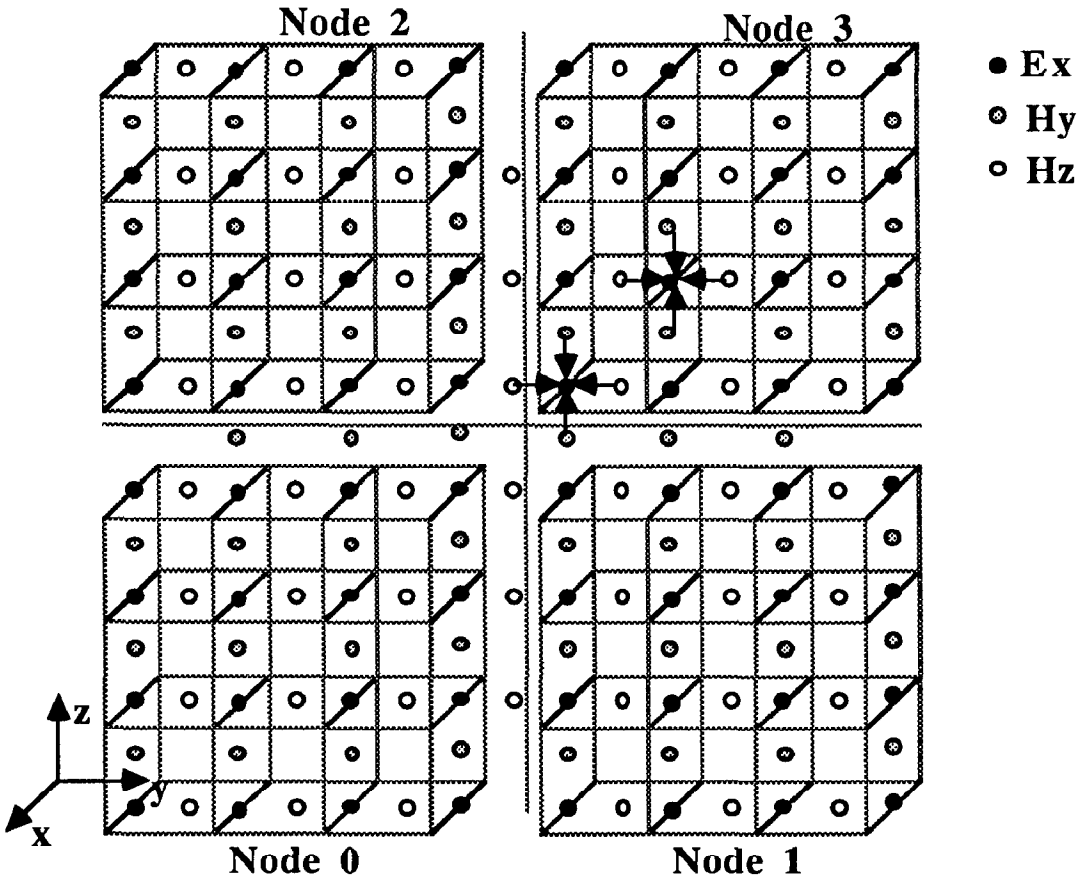



Figure 5.1 The decomposition of the global lattice among four nodes of the Hypercube. Black arrows symbolize the magnetic field components used to update the $E_x$ component within nodes and between the boundaries of nodes.

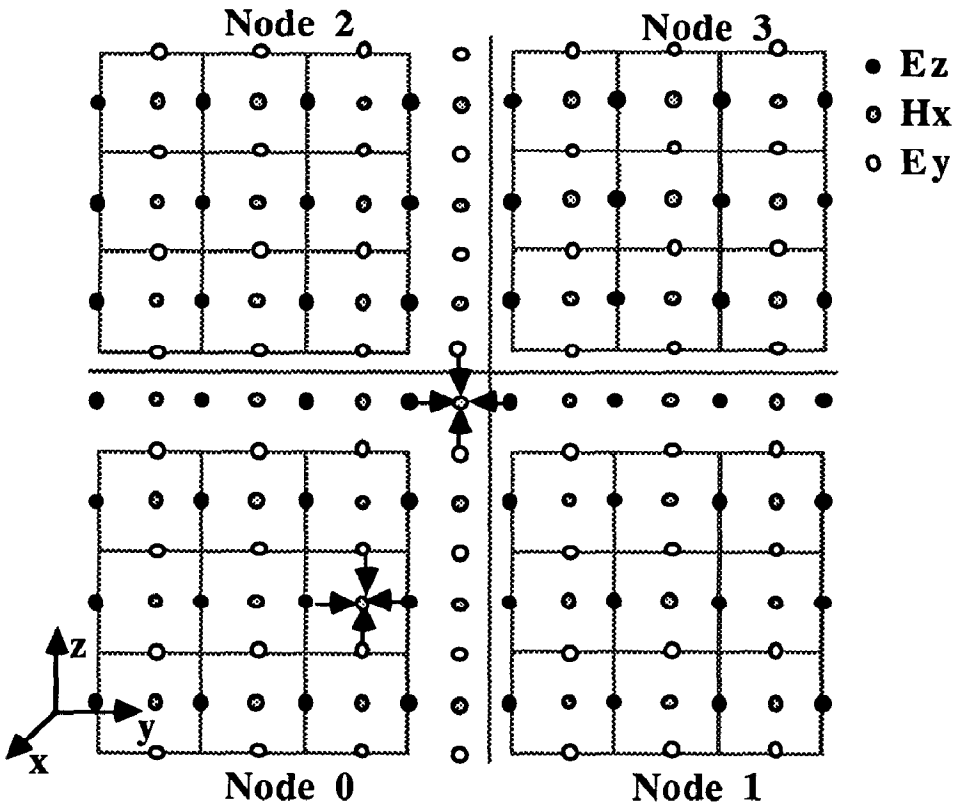



Figure 5.2 The decomposition of the global lattice among four nodes of the Hypercube. Black arrows symbolize the electric field components used to update the $H_X$ component within nodes and between the boundaries of nodes.

Figures 5.1 and 5.2 illustrate a decomposition of a 7 x 7 x 7 unit cell lattice. In this example, the FDTD code uses four nodes of the Hypercube. It assigns a 7 x 4 x 4 cell block to node 0, a 7 x 3 x 4 cell block to node 1, a 7 x 4 x 3 cell block to node 2, and a 7 x 3 x 3 cell block to node 3. Both figures show a cut in a plane perpendicular to the x axis. In this example, there are seven such planes along the x direction. With eight active nodes, the program also divides the global lattice in the x direction. The code will assign four cells in the x direction to the nodes responsible for negative x values and three cells in the x direction to the nodes responsible for positive x values.

Figure 5.1 illustrates the update of the x component of the electric field, $E_X$. To update the discrete values of $E_X$ on the lattice, FDTD requires the value of $E_X$ at the previous time step, the values of neighboring $H_Y$ and $H_Z$ discrete fields at half a time step back, and the values $\varepsilon_{11}$ and $\sigma_{11}$. Refer to the finite differenc forms of equation 2.1. For an $E_X$ component in the middle of a block assigned to a particular node, the update is the same as the sequential case. For an $E_X$ component on a boundary between nodes, neighboring nodes must communicate before the update. The black arrows pointing to a central $E_X$ component illustrate the two types of updates that can occur within node 3.

Figure 5.2 illustrates the update of the x component of the magnetic field, $H_X$. To update the discrete values of $H_X$ on the lattice, the FDTD code requires the value of $H_X$ at the previous time step, the values of neighboring $E_Y$ and $E_Z$ discrete fields at half a time step back and the values $\mu_{11}$ and $\sigma_{m11}$. Refer to the finite differenc forms of equation

2.1. For an $H_x$ component in the middle of a block assigned to a particular node, the update is the same as the sequential case. For an $H_x$ component on a boundary between nodes, neighboring nodes must communicate before the update. The black arrows pointing to a central $H_x$ component illustrate the two types of updates that can occur within node 0.

Second order correct radiation boundary conditions work well with the spatial decomposition of the global lattice. A node, responsible for field points on the truncation planes, the planes marking the boundary of the computation lattice, has enough resident and communicated field information to update truncation plane points using second order correct radiation conditions.

Refer back to the example illustrated in figure 3.1. If any of the points 3 to 10 reside in neighboring nodes, the node containing point 1 must communicate with its neighbors before the update occurs.

The spatial decomposition of the global lattice also works well for calculating the radar cross section, RCS. Several nodes in the hypercube contribute to the integration that yields the electric and magnetic vector potentials. The FDTD code combines the local contributions to these potentials to calculate $E_\theta$ and $E_\phi$. These globally accumulated field values contribute to give the RCS, $\sigma$. Refer to equations 4.1.

RCS calculations depend on the magnitude and phase of discrete field components on an integration surface. The FDTD code uses a cube surrounding the scattering object as the integration surface (2). Portions of the cubic integration surface reside in different nodes. The program obtains the magnitude of field components on this integration surface by recording the peak and valley of the steady state sinusoidal wave forms. When a peak occurs, the program also records the time step. From this time step information, the time step increment $\Delta t$, and the angular frequency of the incident field, the program calculates the phase relative to a reference time step.

## 6. PERFORMANCE OF FDTD CODE ON THE HYPERCUBE

To analyze the performance of the FDTD code, we identify the computing intensive parts. FDTD contains input/output, initialization, and setup subroutines. However, the time step iteration loop, in which the program performs field updates, radiation boundary condition updates, internode communication for several iterations, and magnitude and phase, comprises almost all of the execution time.

One method of efficiency measurement fixes the number of unit cells in each node while increasing the number of active nodes. If the code were 100% efficient and if the number of active nodes increases by a factor N, the execution time of the code should remain constant because the total computational load also increases by a factor N. However, the time may not remain constant because of the added internode communication. Note that, with each increase in the number of active processors, FDTD solves a different problem because the global lattice size increases.

We concentrate on the various components of the iteration loop and give results for this method of efficiency measurement.

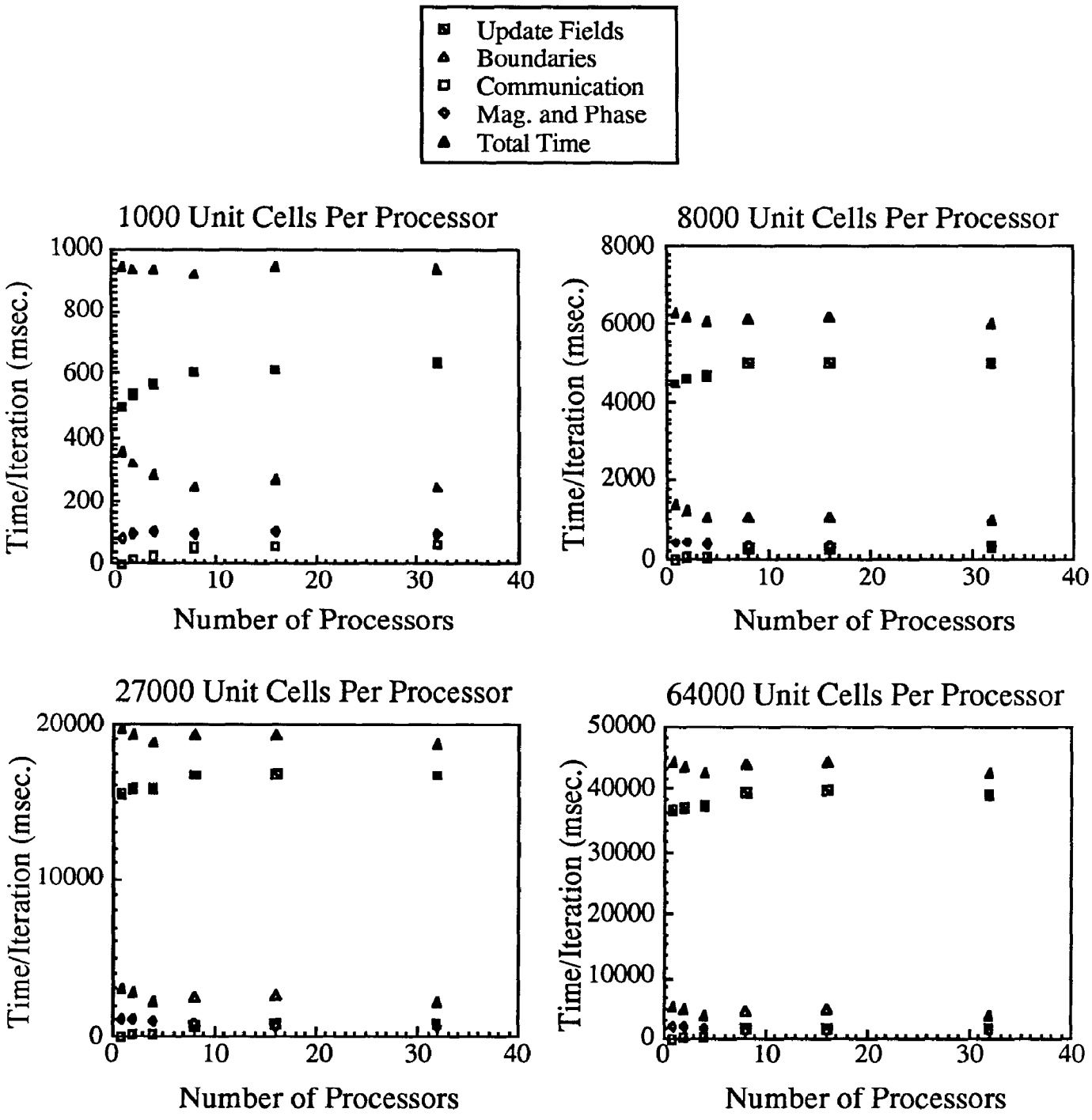



Figure 6.1 Four sets of timing runs for fixed number of unit cells per node of the Hypercube.

Figure 6.1 shows four sets of timing runs. The horizontal axes indicate the number of active processors, 1, 2, 4, 8, 16, or 32. The vertical axes show the execution time in milliseconds. Hollow squares indicate the maximum reported internode communication time per iteration. Because nodes responsible for the boundaries of the global lattice may communicate less, nodes report distinct internode communication time. Hollow triangles indicate the maximum reported time per iteration to perform second order correct radiation

boundary updates. Filled squares indicate the maximum reported time per iteration to perform electric and magnetic field updates within the volume of the computation lattice. Diamonds indicate the maximum reported time per iteration to perform magnitude and phase tracking on the integration surface. Lastly, filled triangles indicate the total time per iteration. We respectively used 1000, 8000, 27000, and 64000 unit cells per node.

The various components of the iteration loop predictably follow certain trends. The communication time per iteration increases with the number of processors. The communication time is also a small fraction of the total time per iteration and is a small fraction of the time per iteration to perform field updates.

The communication time should hit an upper limit for a given density of cells per node because a node can communicate in at most six directions for the three dimensional FDTD method and because the amount of communicated information in a given direction remains constant for a fixed density. The existence of this upper limit is possible if nodes do not wait while other nodes in the configuration communicate. All nodes involved in a communication step should first write a packet of information, read the packet sent by its neighbor, and, then, continue to send and read additional packets. The current version of FDTD does not have this communication scheme.

As the number of nodes increases, the time per iteration to perform radiation boundary updates decreases for node configurations of 2, 4, 8, and 16. For 32 active nodes, the time per iteration increases from the time reported for 16 nodes. Although the number of unit cells remains constant in each node as we increase the number of nodes, FDTD still breaks up the truncation planes among the nodes on the periphery of the global lattice. This division is the reason for the decreased times in 2, 4, 8, and 16 nodes. Second order boundary condition updates also require internode communication. This communication is the reason for the increased time in 32 nodes.

There is a slight increase in the time per iteration for field updates as the number of nodes increases. As the number of nodes increases, a node performs less boundary condition updates. Because the number of cells remains constant, the node must now perform additional field updates within the volume of the computation lattice.

For a fixed density of cells per node, the four plots indicate that these competing times results in faster total execution time per iteration in 4, 8, 16 and 32 nodes compared to 1 and 2 nodes. Internode communication has minimal effects on total execution time for problems saturating the capacity of each active node. With the above mentioned improvements in internode communication, this efficiency should remain high for arbitrarily large node configurations.

The parallel finite difference code performs well on the Mark III Hypercube. The efficiency for fixed global lattice size from 1 to 32 nodes is approximately 80%. The efficiency for fixed number of unit cells per node from 1 to 32 nodes is above 90%.

There does not exist a sequential version of the FDTD code. However, for those people interested in a comparison between sequential and parallel execution times for codes using the FDTD method, we compare two codes, Taflove's sequential code and the parallel FDTD code, of comparable capabilities. For the conducting cube case, presented in the next section, the parallel code runs approximately 22.7 times faster on a VAX/750 and 8.8 times faster on a VAX/785 than Taflove's code. Both VAX times are total cpu usage. Lastly, for a 32 node Hypercube with 4 Mbytes of RAM in each node, the parallel code allows 2,048,,000 unit cells in the global lattice.

## 7. RESULTS FOR DIFFERENT SCATTERERS

To test the validity of the results from the FDTD code, we compare the program's reported currents on the surface of a perfectly conducting cube with the results on page 163 of reference (2).

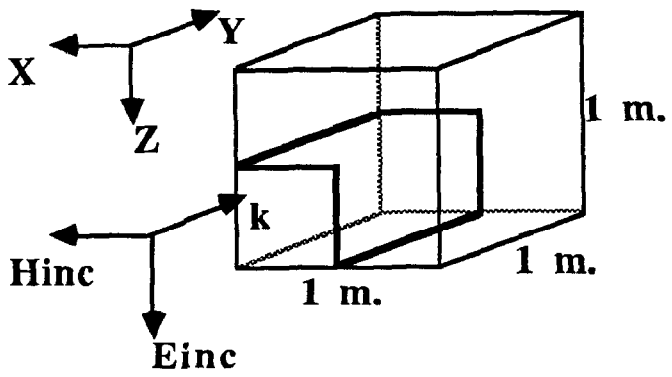
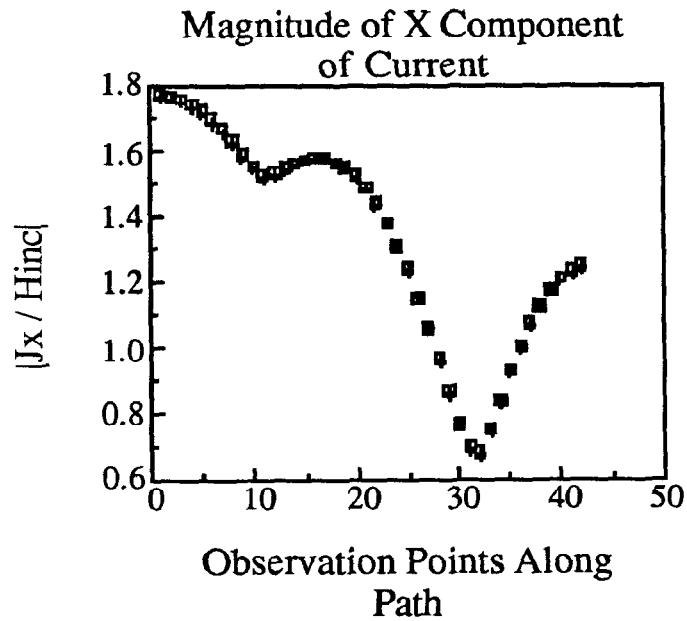
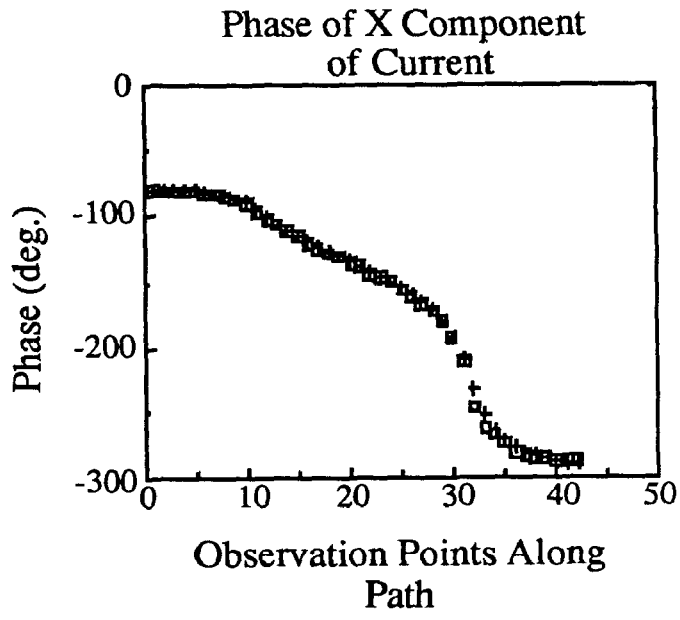



Figure 7.1 The perfectly conducting cube scatterer. The paths for evaluating the currents on the surface are bold. Arrows indicate the polarization and direction of the incident plane wave.

We illustrate the scattering object in figure 7.1. The scattering object is a perfectly conducting cube in vacuum. The cube is one meter in each direction. The unit cell size is 0.05 x 0.05 x 0.05 m. The computation lattice size is 2 x 2 x 2 m. Total number of cells is 40 x 40 x 40. The wave number times the length of a cube side, ks, equals 2. A plane wave is incident on the cube's front face. The direction of propagation of this plane wave is the y direction. The electric field is polarized in the z direction with magnitude 1 V/m. The magnetic field is polarized in the x direction with magnitude $376.73^{-1}$ A/m. The frequency is 95.43 Mhz. The wavelength is 3.141593 m. The number of iterations is 630 time steps with time increment 83.39 x $10^{-12}$ sec.



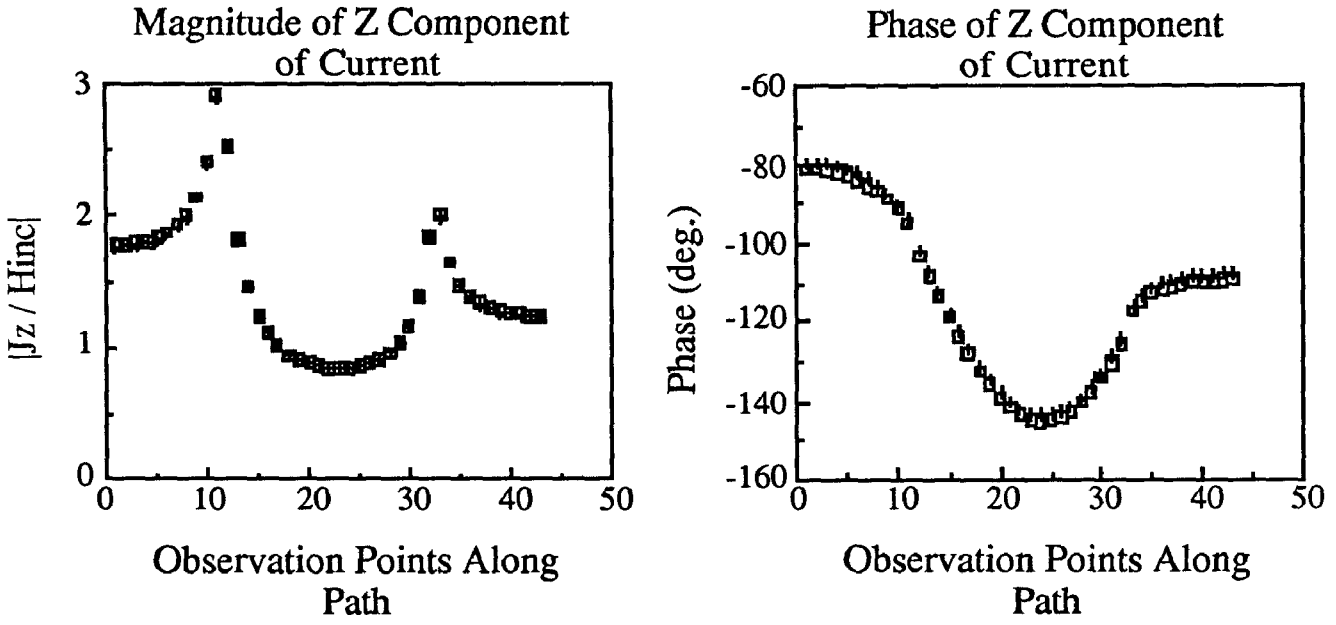



Figure 7.2 The magnitude and phase for the x component of the current on the path below cube scatterer. The magnitude and phase for the z component of the current on the path on the side of cube scatterer.

In Figure 7.2, we evaluate the currents on the surface of the perfectly conducting cube scatterer. We used Taflove's sequential FDTD code and plotted the results as hollow squares. We ran the parallel FDTD code and plotted the results as plus marks (+). The top two plots show the magnitude and phase of the x component of the current on the bottom path for the perfectly conducting cube scatterer. The last two plots show the magnitude and phase of the z component of the current on the side path. The horizontal axes indicate observation points taken at intervals of 0.05 m on the current path. The vertical axes indicate either the normalized current or the phase. The Taflove and the FDTD numbers agree well.

Along with the currents, we found a monostatic radar cross section of 2.14 $m^2$ from the Taflove code and 2.11 $m^2$ from the parallel FDTD code.

We also scattered a plane wave from a anisotropic flat plate scatterer according to the parameters specified in reference (2). The plate is 0.1 x 0.00625 x 0.3 m. The unit cell size is 0.00625 x 0.00625 x 0.00625 m. The computation lattice size is 0.2 x 0.1125 x 0.4 m. Total number of cells is 32 x 18 x 64. The plate has relative permittivity equal to 1 in the z direction and 40 in the x and y directions. It has electric conductivity of 3.72 x $10^7$ mhos/m in the z direction and 28 mhos/m in the x and y directions. A plane wave is incident in the plus y direction. The frequency is 1.0 Ghz. The number of iterations is 576 time steps with time increment 10.42 x $10^{-12}$ sec.

We also obtained a monostatic radar cross section of 1.94 $m^2$ from the Taflove code and 1.90 $m^2$ from the parallel FDTD code.

## 8. CONCLUSION

One can code the FDTD method on the Hypercube very efficiently. In the particular

parallel implementation, described above, communication is only between neighboring processors resulting in low communication time compared to computations occuring in the iteration loop. The large memory available on the Hypercube and the ability to scale the number of Hypercube nodes allows the solution of electrically large three dimensional anisotropic scatterers.

## 9. REFERENCES


(1) K. S. Kunz, "Generalized Three-Dimensional Experimental Lightning Code (G3DXL) User's Manual," Kunz Associates, Inc., February 1986.

(2) K. R. Umashankar and A. Taflove, "Analytical Models for Electromagnetic Scattering," Part II: Finite Difference Time Domain Developments, Final Report on IITRI Project E06538, Electronics Department, IIT Research Institute, June 1984.

(3) K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," IEEE Trans. Antennas Prop., Vol. AP-14, May 1966, pp. 302-307.

(4) A. Taflove and M. E. Brodwin, "Numerical Solution of Steady-State Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations," IEEE Trans. Microwave Theory Tech., Vol. MTT-23, August 1975, pp. 623-630.

(5) G. Mur, "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations," IEEE Trans. on Electromagnetic Compatibility, Vol. EMC-23, November 1981, pp. 377-382.

(6) B. Engquist and A. Majda, "Absorbing Boundary Conditions for the Numerical Simulation of Waves," Math Comp., Vol. 31, July 1977, pp. 629-651.