111-61

50121

P. 18

NASA Contractor Report 195444

FDTD_ANT User Manual

Martin L. Zimmerman
Analex Corporation
Brook Park, Ohio

May 1995

National Aeronautics and
Space Administration

FDTD_ANT
USER MANUAL

Martin Zimmerman
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

SUMMARY

This manual explains the theory and operation of the finite-difference time domain code FDTD_ANT developed by Analex Corporation at the NASA Lewis Research Center in Cleveland, Ohio. This code can be used for solving electromagnetic problems that are electrically small or medium (on the order of 1 to 50 cubic wavelengths). Calculated parameters include transmission line impedance, relative effective permittivity, antenna input impedance, and far-field patterns in both the time and frequency domains. The maximum problem size may be adjusted according to the computer used. This code has been run on the DEC VAX and 486 PC's and on workstations such as the Sun Sparc and the IBM RS/6000.

INTRODUCTION

The finite-difference time domain (FDTD) method is a general implementation of Maxwell's equations in differential form. Because the method is general, it may be applied to a wide variety of electromagnetics problems, including radiating antennas (refs. 1 and 2), scattering problems (refs. 3 and 4), electromagnetic interference (EMI, ref. 5), and microwave circuit analysis. By using a variety of boundary conditions, the method can be applied to closed- or open-boundary problems.

A low-frequency technique, FDTD is applied to a gridded object; that is, the fields on the object are sampled a number of times per wavelength. Usually, FDTD is applied to a regular, orthogonal grid, but it can be applied to any structured grid. This technique has not been demonstrated with unstructured grids. A typical workstation can deal with a problem of 1 million grid cells, which might extend 5 to 10 wavelengths on a side.

The FDTD method was first formulated for electromagnetics (EM) by Yee in 1966 (ref. 6). Most electromagnetic FDTD codes are still based on the Yee cell, which has the three electric field components (E_x, E_y, E_z) located at the midpoints of the cell edges and the three magnetic field components (H_x, H_y, H_z) located at the centers of the cell faces (fig. 1). The differential forms of Maxwell's equations are then quantized to produce central-differenced

equations. Many papers have been published using this method for electromagnetics, primarily since the mid-1980's when computers became powerful enough to do nontrivial problems.

PROBLEM SIZE

To obtain accurate results, a minimum number of cells per wavelength must be used. There is some debate as to what this number is, but 7 to15 cells per wavelength in the material with the shortest wavelength is a generally accepted limit. Stability is maintained by not permitting energy information to propagate across a cell in less than one time step. The maximum size of the time step is limited by the cell dimensions, particularly the minimum cell dimension. Based on the cell size information, the time step is automatically calculated by FDTD_ANT.

Because the number of cells per wavelength is fixed, as the problem size increases, the memory and cpu time needed for execution also increase. A problem N cells long on a side has N^3 cells. For each cell there are six real field components and six integer material properties. If the computer uses four bytes for a real number and two for an integer (typical FORTRAN sizes), the total storage will be $36N^3$, plus a little more for other arrays of order N or N^2. The execution time is approximately the execution time per cell times the number of cells times the number of time steps. The number of time steps needed increases as N so for a three-dimensional problem, the execution time increases as N^4. In contrast, according to the method of moments (MoM) for each dimension, an N-by-N matrix is inverted, requiring N^2 operations. Therefore, the execution time increases as N^6 for a three-dimensional problem. Note that MoM is a frequency domain method whereas FDTD is time domain. A Fourier transform may be used to obtain frequency information over a wide range of frequencies from given time domain information. This method may provide a more efficient means of obtaining broadband frequency information than a frequency domain method. For a limited number of frequencies, it is often more efficient to use a frequency domain method that solves for a single frequency on each pass. For small problems the time inefficiency due to transformation from one domain to another may not be significant.

BOUNDARY CONDITIONS

At the boundaries of a problem space, the update equations cannot be used; instead, some type of absorbing boundary condition (ABC) is used. The FDTD_ANT code employs first- and second-order Mur boundary conditions, which use extrapolation from interior points to calculate the field values on the boundaries.

Because all ABC's are approximate, some reflection will occur, which limits how close an object in the problem space can be located to the problem space boundary. With the Mur conditions, 7 to 10 cells should be the minimum spacing between the object and the boundary. In some cases, such as the boundary above the radiating aperture of an antenna, the spacing must be larger because of the large amount of energy incident upon the boundary. To model a substrate as being infinite in extent, the substrate should run all the way across the problem space (otherwise there will be reflections from the edge of the substrate). The same modeling method applies to the ground plane and any nonterminated microstrip lines. However, the edges of the antenna or other resonating devices (particularly radiating edges) should obey the minimum spacing of 7 to 10 cells.

INPUTTING INFORMATION

File Commonr.for

This file is the first to be edited before a new problem is run. All the common blocks and constants for the fdtd_ant code are contained in this file. Whenever changes are made to common.for, the code fdtd_ant.for must be recompiled.

Size of the problem domain.—The size of the problem domain is defined by the number of cells in the x-, y-, and z-directions, defined as *NX*, *NY*, and *NZ*. These values are integers.

Cell size.—With one exception, all cells are the same size, but the lengths of the sides in the x-, y-, and z-directions can vary. The side lengths are entered in meters as the parameters *DELX*, *DELY*, and *DELZ*. If a fine grid is used (see Fine-grid regions), there will be two cell lengths in a given direction, one of which is three times larger. In this case, the larger dimension is entered.

In general, no other changes need be made to the file commonr.for. For a given problem, the number of cells and the cell side lengths are only specified once so the code fdtd_ant.for need be compiled only one time.

File fdtd_ant.dat

This file is the main input for the code fdtd_ant.for. When a component is designed, most of the work is adjusting the design by editing fdtd_ant.dat and then running fdtd_ant.exe and s11.exe. Note: all physical dimensions of length are in cells, not in meters.

Number of time steps.—This number determines how long the program will run. When a pulse excitation is used, the problem must run until all the scattered pulses subside (radiate out of the problem space). For a pulse on a straight transmission line, the problem is run until the incident pulse passes by all the measurement points. Any reflection results from imperfections in the absorbing boundary condition; therefore, it is valid to use only information obtained before these reflections. For a sinusoidal excitation, the problem must run until steady state is reached, at which point the magnitude of the sinusoidal peaks has reached a stable value. The better the component is matched, the faster steady state is acheived. Roughly speaking, the sinusoidal excitation will last as long as the scattered pulse excitation, but it can be much less or much more depending on how well the component is matched at that particular frequency. For a typical resonant antenna, the incident pulse runs for 500 to 700 steps; the scattered pulse, for 2000 to 5000 steps; and the sinusoidal excitation, for 1500 to 5000 steps.

Inputs Related to Source Type Used

Source type.—This input describes the shape of the excitation in time. Sources 1 and 2 are pulses: the first is Gaussian and has the same shape in the frequency and time domains; the second is sinusoidal and is constructed from cosines and sines in the time domain. For this reason, the frequency domain response has sidelobes similar to those of a sinc function. The third source is a sinusoidal excitation that ramps up from zero over two periods.

Source amplitude.—This number is arbitrary and is usually set to 1000.0.

Beta/frequency.—For the pulse excitations (source type 1 or 2), enter the half-width of the pulse in time steps (i.e., the number of time steps taken for the pulse to advance from zero to its maximum value). As the pulse widens in the time domain, it narrows in the frequency domain. As the frequency increases, the wavelength decreases, as does the number of cells per wavelength for a given grid. At frequencies where the grid is coarse, the response will be inaccurate and potentially unstable. Therefore, there should be very little energy in the higher frequencies. If the frequency domain pulse is too narrow, the response at frequencies of interest may be lost in the noise. A rule of thumb is that the frequency response be down by about 80 dB by the time the frequency reaches the point where there are only four cells per wavelength. A good choice for the pulse half-width is 125 to 250 steps.

For the sinusoidal excitation (source type 3), enter the frequency in gigahertz.

Feed type.—This integer flag indicates the physical shape over which the forcing function is applied.

- Feed type 0 is used for scattering problems. The forcing function is a plane wave with a certain polarization and propagation direction. This feed type is handled somewhat differently from the way other feed types are because it has different defining parameters.
- Feed type 1, a wire, is a delta gap voltage generator that has arbitrary length but zero thickness. The wire is uniformly excited.
- Feed type 2, a microstrip, has zero size in one dimension and is planar in extent. It is uniformly excited. When used with a microstrip line, the feed stretches the width of the microstrip line and extends from the microstrip line to the ground plane.
- Feed type 3, a waveguide, is planar in extent, but the excitation is weighted as a half-sinusoid to represent the lowest order of the resonant transverse mode in a recangular waveguide.
- Feed type 4, a large rectangular aperture, is the same as a microstrip feed but can be used in cases when the feed size is larger than 10-by-10 cells.

For a coaxial feed, the impedance of the coaxial line in ohms is entered.

Feed location.—This input describes the physical dimensions of the feed region. There may be more than one region so on the first line, the number of feeds is entered. On the subsequent lines (one per feed), the following information is entered for feed types other than 0: the starting values of the region in the x-, y-, and z-directions; the width of the region in cells; and the relative phase of each feed in degrees. For example, a phased array could be modeled as a number of radiating elements, each with its own feed line and feed source, phased to give the desired results. A coaxial microstrip line is modeled as having two feed sources, one on each side of the main line, with opposite phases. Note that the wire, microstrip, waveguide, and coaxial feeds are limited to a 10-by-10 cell size.

For the plane wave (feed type 0), the entered information is the direction of the incident wave (defined as the direction the wave comes from, not the direction of the propagation vector) and the theta and phi components of the E-field polarization vector.

Source polarization.—This input is the E-field polarization for the source region. Enter 1, 2, or 3 for the x-, y-, or z-directions, respectively. For the coaxial feed, enter the axis along which the coaxial center conductor lies (since the coaxial feed has two E-field components in cartesian coordinates). This information is not used for the plane wave case.

4

Information Related to Program Output

<u>Types of data output</u> .—Enter the integers *IDATA* and *NSTEP* on one line.

- For *IDATA* = 1, data are only saved after every *NSTEP* time steps.
- For *IDATA* = 2, data are saved after every *NSTEP* time steps; near-field information is also saved.
- For *IDATA* = 3, only near-field information is saved.

<u>Number of data points sampled</u>.—Enter the number of data points that are saved after every *NSTEP* time steps.

<u>Output data descriptions</u>.—Note that these do not have to be a single point. If the entry describes more than one point, the output value will be the summation over all the points in the entry. Each sampled value is on a separate line. The parameters are the starting x-, y-, z-cell location, the width of the data sampling area in cells, and the data type. A width of zero corresponds to one point, with the value sampled at the starting cell location. Therefore, if the width is two cells, the output will be a summation over three points. The symbol * represents a digit placeholder.

A three-digit integer indicates the data type as follows:

First digit

0**		energy density or Poynting vector
1**	E_x	
2**	E_y	
3**	E_z	
4**	H_x	
5**	H_y	
6**	H_z	
7**	I_x	current through a rectangular loop in the y, z-plane
8**	I_y	
9**	I_z	

Second digit

01*	energy density $0.5 \cdot (\varepsilon \cdot |E|^2 + \mu \cdot |H|^2)$
02*	Poynting vector $E_T \times H_T$, where the subscript T represents components transverse to indicated direction.

If the first digit is not zero, the second digit is usually zero.

Third digit

02*	for Poynting vector 1, 2, or 3 indicates that the vector lies along x-, y-, or z-axis, respectively. For other cases the third digit is usually zero.

This is not a complete listing of all possible values; these are the most commonly used.

<u>Number of near-field data sets</u>.—Enter the number of near-field data sets saved at the end of the calculation for converting to far-field patterns (or for estimating the power contained in a problem).

<u>Near-field data descriptions</u>.—For each data set, two output files are created. The files for the first data set are for011.dat and for012.dat. For a second data set, the files are for013.dat and for014.dat, etc. These input parameters are very similar to the parameters used for the output data (Output data descriptions). The final value, however, is a one-digit integer. If the data type is 1, the first file for each data set contains the complex E-field components. The second file contains the complex H-field components. If the data type is 2, the first file contains the magnitude of one of the tangential E-field components, and the other contains the magnitude of the second tangential E-field component. Data type 1 is used to obtain far-field information; data type 2 is used to obtain graphical output describing the near fields (e.g., contour or vector plots).

Information Describing Problem Geometry

<u>Fine-grid regions</u>.—The fine-grid option allows the cell dimensions to be specified in a certain region as one-third the size of the values in file commonr.for. This option can be used to model some aspect of the object modeled in finer detail. It can also be used to enlarge the cells between the object and the outer boundary so that the absorbing boundary conditions are applied farther away from the object. Enter the starting and ending points of the fine grid for the x-direction, then the starting and ending points for the y-direction followed by the z-direction. If the starting cell point is 0, it is assumed that there is no fine grid for that direction. Note that this entry format is different from that used in other places in this file.

<u>Object descriptions</u>.—The model to be analyzed is described in this section. Each line is used to describe a simple object that has a basic shape and material type. Complex objects are built as a collection of simple objects. The order in which objects are entered is very important. If two objects share a boundary (meaning the material properties of that object are applied to the same points), the common points will have the material type of the last object entered. For this reason, when a metal-clad substrate is used, the substrate should be entered before the ground plane and any surface properties. When two nonmetal objects touch, the boundary is often described as having a dielectric constant that is the average of the values for the two materials (e.g., a Duroid 5880/air interface has a permittivity ε of 1.6). After the last object is entered, another line that begins with a −1 is used to tell the program that there are no more objects.

Because of the quantized nature of the material properties (only defined once for each cell), there is some uncertainty about where one material begins and another ends; therefore, a finite-length object will typically produce results as though it were one cell longer than it is modeled. For a short dimension, this error is closer to a half-cell. For example, if the cell size is 0.3 mm, a patch 3.0 mm long would be better modeled with 9 cells than with 10 cells. A microstrip line that is 0.75 mm wide can best be modeled by a metal object that is 2 cells wide.

There are eight integer parameters for each object. The first number is the object type and the second is the material type. The remaining parameters describe the location and size of the object:

- Object type 1, a rectangular cube or plane (no more than one width can be zero). Enter the x-, y-, z-cell starting coordinates and the x-, y-, and z-widths of the object in cells. Note that the far wall of the object is located at the cell with coordinate value equal to the

starting coordinate plus the object width. If a space has $NX = 50$ and a substrate crosses the entire space in the x-direction, the starting x-value is 1 and the width is 49, not 50.

- Object type 2, a wire (two of the dimensional widths = 0). The dimensional parameters are the same as those for the rectangular cube.
- Object type 3, a cylinder with its axis along a principal axis. The first three location parameters are the x-, y-, and z-cell coordinates of the center of the circle that forms the lower end of the cylinder. The next value is the direction of the cylinder axis and is 1, 2, or 3 for the x-, y-, or z-axes, respectively. The next value is the height of the cylinder in cells, and the last value is the radius of the cylinder in cells.
- Object type 4, a sphere. The first three location parameters describe the center of the sphere, and the next value is the radius of the sphere in cells. The final two values on the input line must be present but are not used.

The material type is 0 for freespace and 1 for perfect electrical conductor (p.e.c.). The remaining values describe dielectric materials. The permittivity and conductivity are set in subroutine SETUP in file fdtd_ant.for. At this time, material type 2 corresponds to RT/Duroid 5880 ($\varepsilon_r = 2.2$) and material type 4 corresponds to ($\varepsilon_r = 1.6$).

Boundary conditions.—The fdtd_ant.for code uses Mur absorbing boundary conditions, both first and second order. Second-order Mur is more accurate, but it cannot be used unless the boundary surface is a homogeneous material. If cells along the boundary face do not all have the same material property, first-order Mur must be used. Three-integer parameters are entered, indicating that first-order (=1) or second order (=2) Mur is used along the YZ-(x = constant) faces, XZ-faces and XY-faces. For example, if a substrate is built and extends across the problem space in the XY-plane, the entered values are 1, 1, 2.

File s11.dat

The s11.dat file is used in processing the incident and scattered pulses to produce frequency domain information. Some information is used in strip.com and some is used in s11.com. Physical length dimensions are entered in millimeters.

Distance between measurements on microstrip line.—To calculate the relative effective permittivity, voltage measurements are needed at two points on the microstrip line. Enter the distance in millimeters between the two measurement points.

Height of cell in substrate.—The code can sample currents but not voltages, both of which are needed for computing the transmission line impedances. The electric field has units of volts per cell which can be converted to volts by multiplying by the length of a cell in the direction between the transmission line and the ground plane.

Distance from measurement point to antenna.—The scattered response must be measured some distance away from the scattering object to avoid including transient effects. However, to obtain the correct phase for the load impedance, the impedance at the measurement point must be transformed. The distance between the measurement point and the antenna (in millimeters) must be known to complete this transformation.

Maximum frequency.—When the time domain information is Fourier transformed to the frequency domain, the maximum frequency is the inverse of the time step. This frequency is much larger than the highest frequency for which the code is accurate. Enter in gigahertz the maximum frequency for which frequency domain information should be calculated.

7

Maximum number of points.—This number serves much the same purpose as the maximum frequency (i.e., limiting the amount of frequency domain information calculated). Usually it is much easier for the user to decide a maximum frequency than to decide a maximum number of points. The code uses the limit that is reached first.

File nff1.dat

File s11.dat is the input file for the near-field to far-field transformation code nff1.for. Another input file used by nff1.for is nfe.inp, which contains the E-field, near-field information (that must be copied from the for011.dat, etc. files that fdtd_ant.exe produces).

Far-field variables.—Choose 1 for theta and phi and 2 for elevation, azimuth.

Theta (elevation) range.—Enter the initial value of theta, the final value of theta, and the number of theta divisions (not points). Zero divisions means that only the initial value of theta is used. One division means that calculations are performed at the starting and ending theta values.

Phi (azimuth) range.—Enter the initial value of phi, the final value of phi, and the number of phi divisions.

Aperture power.—Enter the maximum value for the Poynting vector evaluated over the aperture of the antenna. This aperture should be the area over which near fields are sampled for near-field to far-field calculations. This number is used to normalize the directive gain.

TEST EXAMPLE FOR FDTD_ANT

This example presents the steps to model a microstrip endfire dipole antenna (fig. 2). Note that italics indicates text typed at the prompt on the VAX. Some commands will change slightly if the code is run on a UNIX machine. For example, *run fdtd_ant* becomes *fdtd_ant.x*, and *@s11* becomes *s11*.

Obtaining Information About The Transmission Line

Set up the commonr.for file for the correct number of cells and correct cell size (cell dimensions in meters).

nx = 50   ny = 60   nz = 40
delx = 0.0003 dely = 0.0003 delz = 0.000254 (10 mil)

Set up the problem space for a through transmission line in fdtd_ant.dat.

nstop = 700
source type = 1 (Gaussian)
beta = 200
1 feed = 20, 3, 20, 2, 0, 2, 0

The four data points for saving are

21, 06, 20, 0, 0, 1, 300	(reference E_z-field)
21, 36, 20, 0, 0, 1, 300	(second E_z-field for calculating eff. rel. perm.)
20, 06, 22, 2, 0, 0, 800	(I_y for calculating line impedance)
16, 06, 20, 9, 0, 9, 022	(P_y power in the pulse–not needed)

Because this is a *thru* line, the substrate must go across the entire problem space. The objects to build are

1, 2, 1, 1, 20, 49, 59, 2	!substrate (mat 2 is $\varepsilon = 2.2$)
1, 4, 1, 1, 22, 49, 59, 0	!top face of substrate (mat 4 is $\varepsilon = 1.6$)
1, 1, 1, 1, 20, 49, 59, 0	!ground plane on bottom of substrate
1, 1, 20, 1, 22, 2, 59, 0	!microstrip line (runs in the y-direction)

Examining the built object reveals that all the measurement points are centered around the microstrip line.

Any file entry not listed should already be correct.

Recompile fdtd_ant (if changes have been made to commonr.for). Then execute fdtd.exe.

fort fdtd_ant
link fdtd_ant
run fdtd_ant

Figure 3 plots the first two columns of data in fdtd_ant.out, the voltage pulses measured at y = 6 and 36. As seen from the figure, only 500 time steps, not 700, were necessary.

Copy output file fdtd_ant.out to pulse.dat

copy fdtd_ant.out pulse.dat

Set up file s11.dat if necessary (should already be correct).

Run the command file strip.com.

@strip

The line impedance as a function of frequency is in file znot.out (fig. 4) and the relative effective permittivity as a function of frequency is in file epeff.out (fig. 5).

Obtaining Pulse Response Of Antenna

Edit the input file fdtd_ant.dat.

nstop = 3000

Model the entire antenna now. Note that all the lines needed but are not presently in use may be stored at the bottom of the file fdtd_ant.dat.

1, 2, 1, 1, 20, 49, 49, 2	!substrate (now only goes to y = 50)
1, 4, 1, 1, 22, 49, 49, 0	!top face of substrate
1, 4, 1, 1, 20, 49, 49, 0	!bottom face of substrate
1, 1, 1, 1, 20, 49, 30, 0	!ground plane
1, 1, 19, 31, 20, 11, 11, 0	!base of dipole
1, 1, 19, 42, 20, 05, 08, 0	!left of slot
1, 1, 12, 46, 20, 07, 04, 0	!left arm
1, 1, 25, 42, 20, 05, 08, 0	!right of slot
1, 1, 30, 46, 20, 07, 04, 0	!right arm
1, 1, 20, 01, 22, 02, 38, 0	!feed line
1, 1, 20, 39, 22, 02, 09, 0	!1/4 wave transformer (note: same as feedline)
1, 1, 21, 47, 22, 07, 02, 0	!slot crossing piece
1, 1, 27, 39, 22, 02, 09, 0	!tuning stub

The rectangular objects described above make up the design shown in figure 2. One aspect not shown in the drawing is that the two 90° bends in the microstrip line have the cell at the tip of the corner missing. This area corresponds to the corner chamfer that would be used to reduce the reflections caused by this bend.

Run the file fdtd_ant.exe

run fdtd_ant

The first column of fdtd_ant.out is plotted in figure 6, which is the total field response, not the reflected field response. As can be seen, only about 2000 time steps were really needed (3000 were used).

Run the file s11.com.

@s11

The file s11.out contains the return loss, magnitude of s11, real and imaginary parts of Z/Z_0, and real and imaginary parts of Z as a function of frequency. The return loss is plotted in figure 7.

Calculating The Far-Field Patterns

Edit the input file fdtd_ant.dat.

    nstop = 2000
    source type = 3 (ramped cosine)
    freq = 15.003 (replaces beta = 200)

Data points saved are

16, 06, 20, 9, 022	!power on microstrip line
02, 52, 03, 47, 0, 37, 022	!endfire power in x, z-plane beyond substrate edge
03, 35, 18, 44, 24, 0, 023	!power in x, y-planes below substrate
03, 35, 23, 44, 24, 0, 023	!power in x, y-plane above substrate

Three data sets are saved at the end:

02, 52, 03, 47, 0, 37, 1	!endfire
03, 35, 18, 44, 24, 0,1	!bottom
03, 35, 23, 44, 24, 0, 1	!top

Run the file fdtd_ant.exe.

run fdtd_ant

Six output files are produced for011.dat, for012.dat, ..., for016.dat. The H-fields for the first data set are in for011.dat, and the E-fields are in for012.dat, etc. For a UNIX machine, for011.dat becomes fort.11.

Obtain the power normalization numbers.

To obtain the correct directivity, it is necessary to know the maximum power level in the radiating aperture (data set 1). Read this from fdtd_ant.out (15377). The maximum power on the feedline can also be read (23444). These numbers are in units (watts/area of cell face) and must be multiplied by the cell cross-sectional area before being compared. This area is (0.0003 * 0.0003) for the aperture and (0.0003 * 0.000254) for the feedline. The feed efficiency, defined as the ratio of these two powers, is 77 percent.

Set up the input files for nff1.exe.

copy for012.dat nfe.inp

In file nff1.dat, enter the aperture power density (15377). Set the other values to determine the number of points plotted out. Using the H-fields gives a different answer from that obtained using the E-fields, and because the E-field results match more closely with experiment, only the E-fields are used.

Run file nff1.exe.

run nff1

This file was run twice, once for the phi = 0° cut (fig. 8) and once for the phi = 90° cut (fig. 9). The value of phi is set in file nff1.dat.

Obtaining Near-Field Information (Optional)

If desired, instead of calculating the complex near-field values, it is possible to save the magnitudes of the tangential components of the E-fields in files for011.dat, for012.dat, etc.

Edit file fdtd_ant.dat.

In the section for data sets to be saved at the end, the final number on each line is a one, which is changed to a two to save the magnitudes of the tangential E-fields.

Run file fdtd_ant.

run fdtd_ant

The data in files for011.dat, etc. may be plotted using a commercial package, as were the plots in figure 10. Note that the drawing of the antenna structure and all of the graphs were produced with commercial packages and were not part of this code.

CONCLUDING REMARKS

This document describes the use of a FORTRAN FDTD code that may be used for antenna design and other electromagnetic source problems. This code is based upon the Luebbers and Kunz FDTDA code and has been heavily modified by Analex Corporation under a NASA Lewis Research Center contract. It is available for limited distribution from the NASA Lewis Research Center. This code has been successfully run on VAX, UNIX (IBM RS/6000 and SUN SparcStation), and 486-based PC computers.

REFERENCES

1. Reneix, A.; and Jecko, B.: Analysis of Microstrip Patch Antennas Using Finite-Difference Time Domain Method. IEEE Trans. Antennas Propag., vol. AP-37, Nov. 1989, pp.1361–1369.
2. Maloney, J.G.; Smith, G.S.; and Scott Jr., W.R.: Accurate Computation of the Radiation From Simple Antennas Using the Finite-Difference Time-Domain Method. IEEE Trans. Antennas Propag., vol. AP-40, Mar. 1992, pp. 334–340.
3. Taflove, A.; Umashankar, K.; and Jurgens, T.: Validation of FD-TD Modeling of the Radar Cross Section of Three-Dimensional Structures Spanning up to Nine Wavelengths. IEEE Trans. Antennas Propag., vol. AP-33, June 1985, pp. 662–666.
4. Luebbers, R., et al.: A Frequency-Dependent Finite Difference Time Domain Formulation for Dispersive Materials. IEEE Trans. Electromag. Compat., vol. EMC-32, Aug. 1990, pp. 222–227.
5. Umashankar, K.; and Taflove, A.: A Novel Method to Analyze Electromagnetic Scattering of Complex Objects. IEEE Trans. Electromag. Compat., vol. EMC-24, Nov. 1982, pp. 397–405.
6. Yee, K.S.: Numerical Solution of Initial Boundary Value Problems in Isotropic Media. IEEE Trans. Antennas Propag., vol. AP-14, no. 3, May, 1966, pp. 302–307.
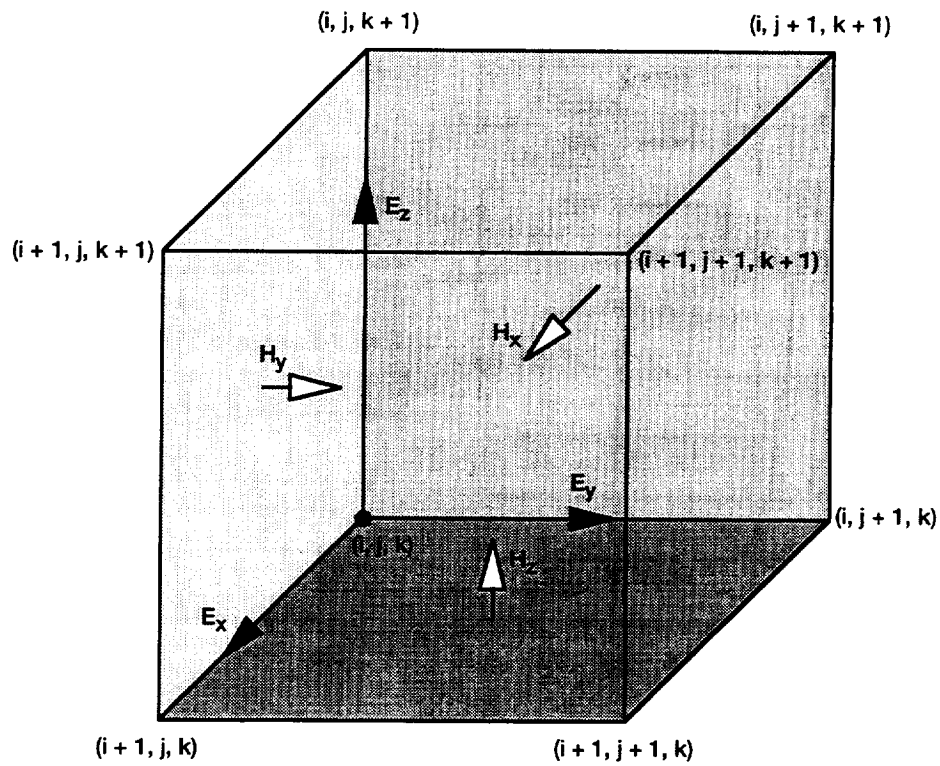
Figure 1.—Yee cell used in electromagnetic implementation of finite-difference time domain (FDTD) code.
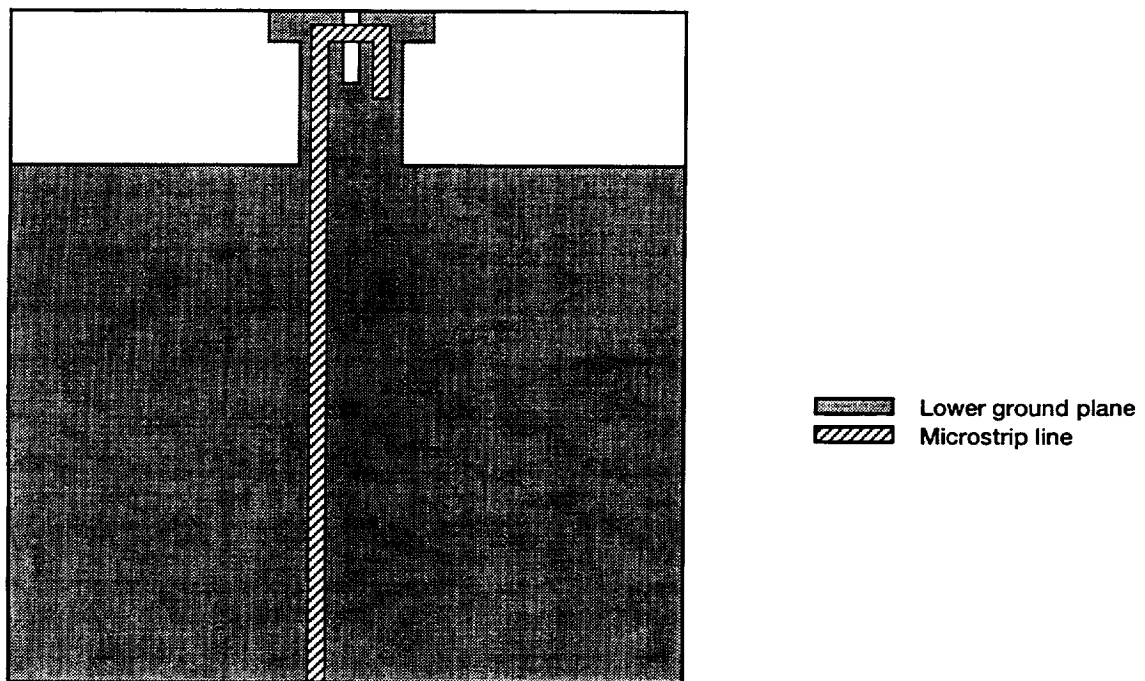

Lower ground plane
Microstrip line

Figure 2.—Layout of 15-GHz endfire dipole on 20-mil polytetrafluoroethylene substrate with relative permittivity of 2.2.
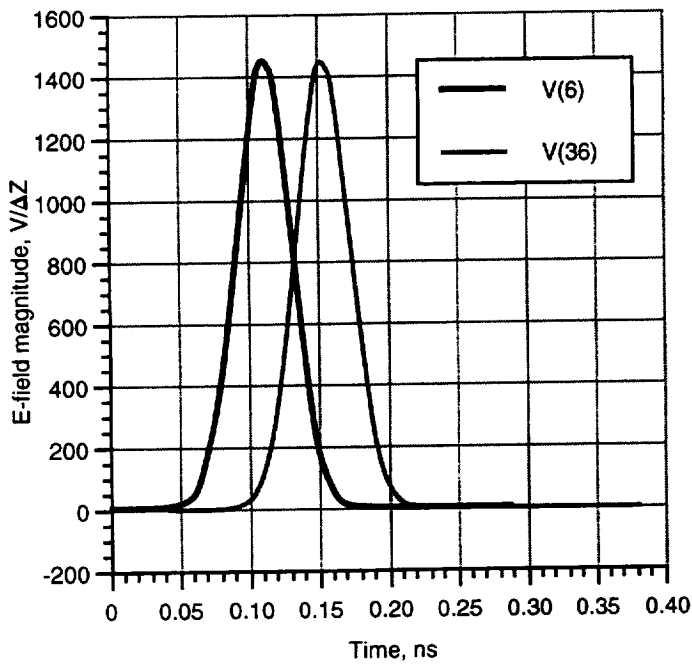
13

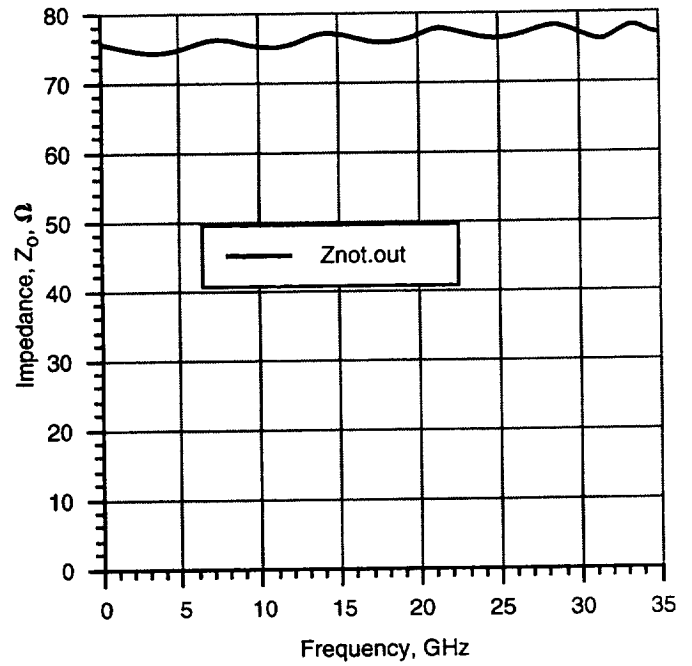Figure 3.—Incident electric field component pulse E_z on microstrip line at y = 6 and 36.

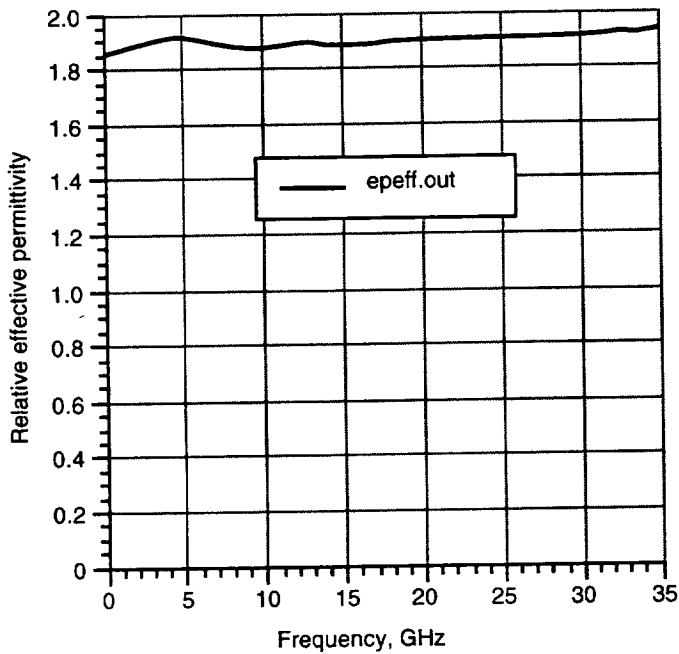Figure 4.—Impedance Z_0 on microstrip line (file Znot.out).

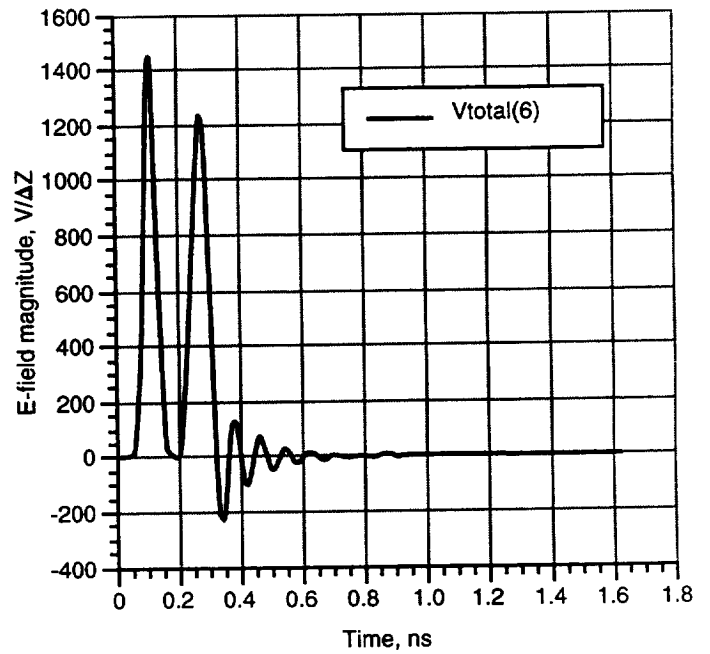Figure 5.—Relative effective permittivity of microstrip line (file epeff.out).

Figure 6.—Magnitude of total E_z-field response due to pulse hitting the antenna.

14
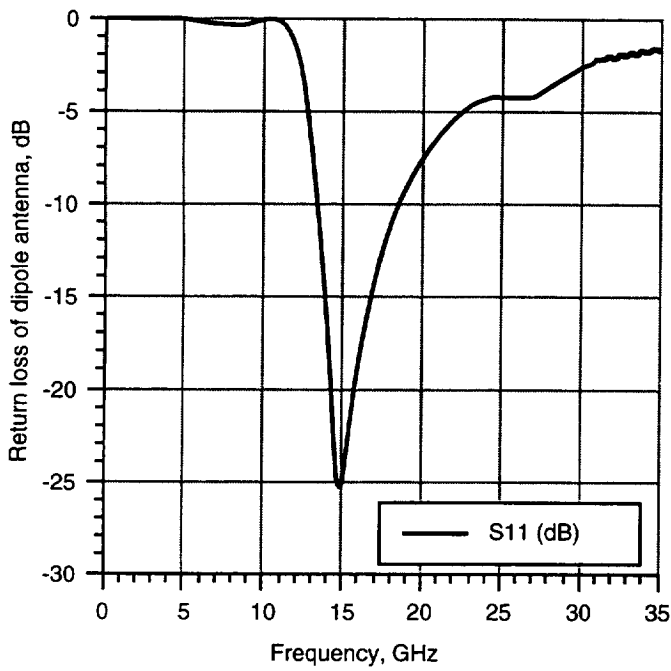
Figure 7.—Return loss s11 for endfire dipole antenna.
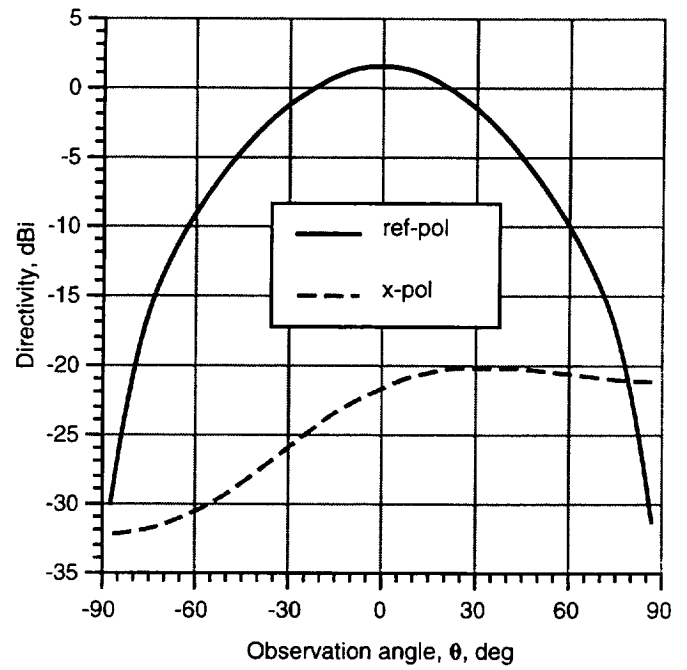
Figure 8.—Phi = 0° (plane of substrate) far-field pattern cut at 15.003 GHz.
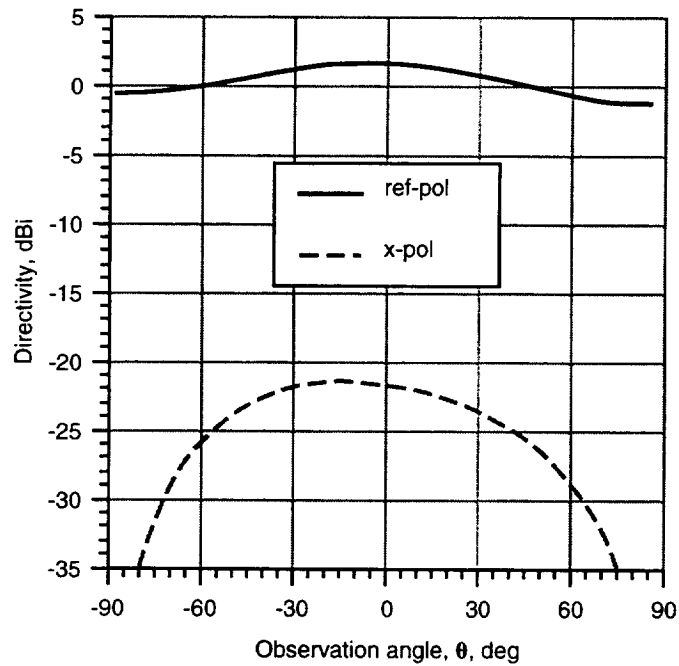
Figure 9.—Phi = 90° (normal to substrate) far-field pattern cut at 15.003 GHz.
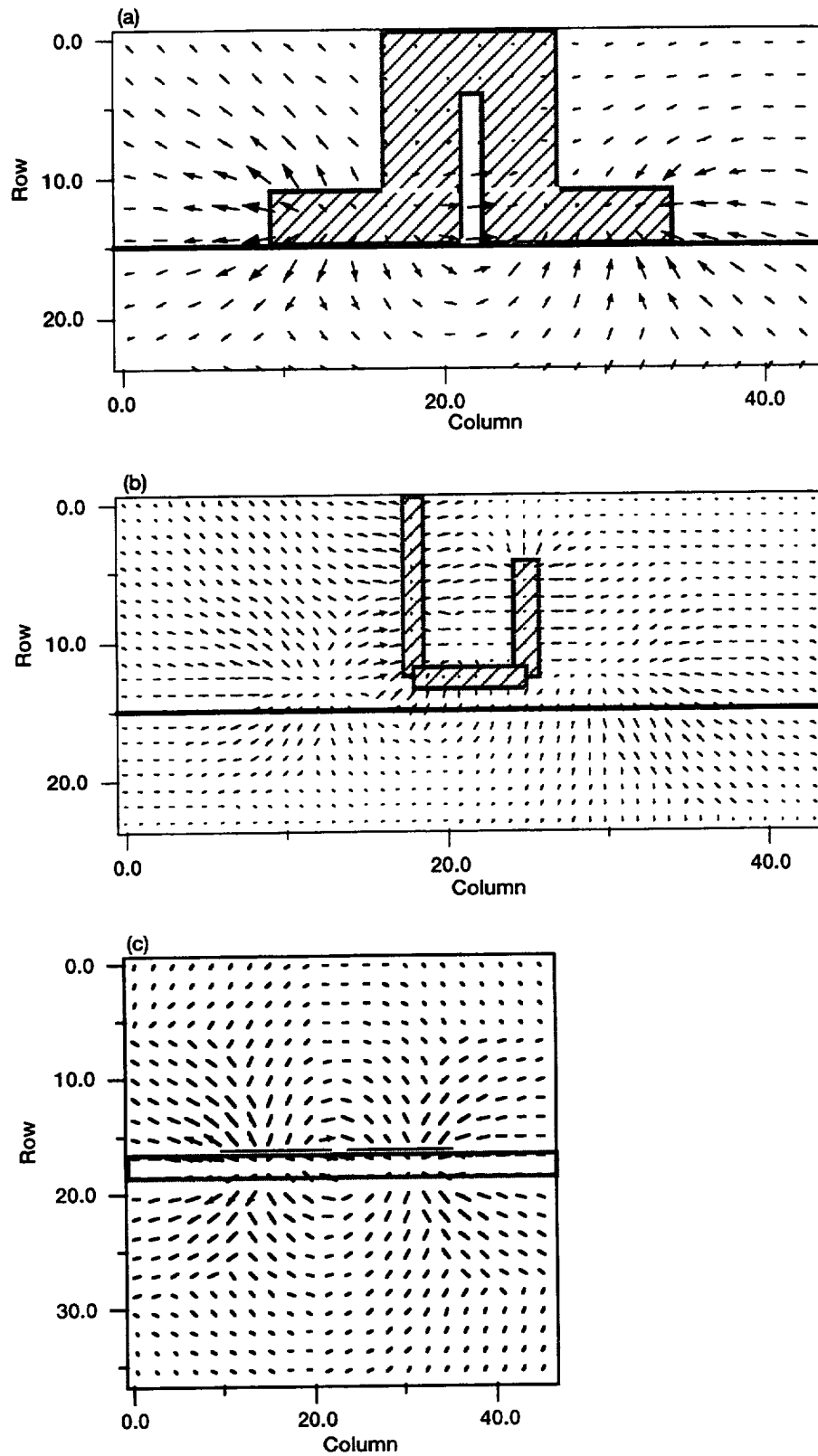
Figure 10.—Tangential near-E-fields in vicinity of dipole. (a) E-fields just below bottom of substrate where dipole is located. Each unit (row, column) represents one cell. (b) E-fields just above top of substrate where feed line is located. Note: dipole fields can also be seen. (c) E-fields in radiating aperture of dipole. Note: this graph is not to the same scale as (a) and (b).

16

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1995	3. REPORT TYPE AND DATES COVERED Final Contractor Report

4. TITLE AND SUBTITLE

FDTD_ANT User Manual

5. FUNDING NUMBERS

WU–235–01–0C
C–NAS3–25776

6. AUTHOR(S)

Martin L. Zimmerman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER

E–9492

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–195444

11. SUPPLEMENTARY NOTES

Project Manager, Charles A. Raquet, Space Electronics Division, NASA Lewis Research Center, organization code 5640, (216) 433–3471.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 61

This publication is available from the NASA Center for Aerospace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This manual explains the theory and operation of the finite-difference time domain code FDTD_ANT developed by Analex Corporation at the NASA Lewis Research Center in Cleveland, Ohio. This code can be used for solving electromagnetic problems that are electrically small or medium (on the order of 1 to 50 cubic wavelengths). Calculated parameters include transmission line impedance, relative effective permittivity, antenna input impedance, and far-field patterns in both the time and frequency domains. The maximum problem size may be adjusted according to the computer used. This code has been run on the DEC VAX and 486 PC's and on workstations such as the Sun Sparc and the IBM RS/6000.

14. SUBJECT TERMS

FDTD; Antenna; Electromagnetics; CEM

15. NUMBER OF PAGES
18

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	