4 Three-Dimensional Simulation

At last we have come to three-dimensional simulation. In actuality, three-dimensional FDTD simulation is very much like two-dimensional simulation—it's just harder! It's harder because of logistical problems: you are using all vector fields and each one is in three dimensions. Nonetheless, if you pay attention and build your programs carefully, the process is straightforward.

4.1 FREE SPACE FORMULATION

The original FDTD paradigm was described by the *Yee cell*, (Fig. 4.1), named, of course, after Kane Yee [1]. Note that the E and H fields are assumed interleaved around a cell whose origin is at the location i, j, k. Every E field is located 1/2 cell width from the origin in the direction of its orientation; every H field is offset 1/2 cell in each direction except that of its orientation.

Not surprisingly, we will start with Maxwell's equations:

$$\frac{\partial \tilde{\boldsymbol{D}}}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \cdot \nabla \times \boldsymbol{H} \tag{4.1a}$$

$$\tilde{\boldsymbol{D}}(\omega) = \varepsilon_r^*(\omega) \cdot \tilde{\boldsymbol{E}}(\omega) \tag{4.1b}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \nabla \times \tilde{\boldsymbol{E}}. \tag{4.1c}$$

Once again, we will drop the $\sim$ notation, but it will always be assumed that we are referring to the normalized values.

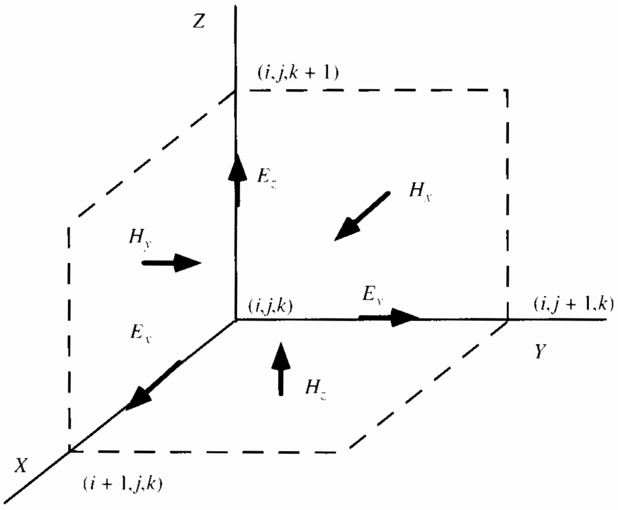

Figure 4.1 The Yee cell.

Eqs. (4.1a) and (4.1c) produce six scalar equations:

$$\frac{\partial D_x}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \tag{4.2a}$$

$$\frac{\partial D_y}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \tag{4.2b}$$

$$\frac{\partial D_z}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \tag{4.2c}$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right) \tag{4.2d}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \tag{4.2e}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right). \tag{4.2f}$$

The first step is to take the finite difference approximations. We will use only Eqs. (4.2c) and (4.2f) as examples:

$$\begin{aligned} D_z^{n+1/2}(i, j, k+1/2) &= D_z^{n-1/2}(i, j, k+1/2) \\ &+ \frac{\Delta t}{\Delta x \cdot \sqrt{\varepsilon_0 \mu_0}} (H_y^n(i+1/2, j, k+1/2) - H_y^n(i-1/2, j, k+1/2) \\ &- H_x^n(i, j+1/2, k+1/2) + H_x^n(i, j-1/2, k+1/2)\,) \end{aligned} \tag{4.3a}$$

$$\begin{aligned} H_z^{n+1}(i+1/2, j+1/2, k) &= H_z^n(i+1/2, j+1/2, k) \\ &- \frac{\Delta t}{\Delta x \cdot \sqrt{\varepsilon_0 \mu_0}} (E_y^{n+1/2}(i+1, j+1/2, k) - E_y^{n+1/2}(i, j+1/2, k) \\ &- E_x^{n+1/2}(i+1/2, j+1, k) + E_x^{n+1/2}(i+1/2, j, k)\,). \end{aligned} \tag{4.3b}$$

From the difference equations, the computer equations can be written:

```
dx[i][j][k] = dx[i][j][k]
                    + .5*( hz[i][j][k]  - hz[i][j-1][k]
                         - hy[i][j][k]  + hy[i][j][k-1]) ;
dy[i][j][k] = dy[i][j][k]
                    + .5*( hx[i][j][k]  - hx[i][j][k-1]
                         - hz[i][j][k]  + hz[i-1][j][k]) ;
dz[i][j][k] = dz[i][j][k         ]
                    + .5*( hy[i][j][k]  - hy[i-1][j][k]
                         - hx[i][j][k]  + hx[i][j-1][k]) ;

hx[i][j][k] = hx[i][j][k]
                    + .5*( ey[i][j][k+1]  - ey[i][j][k]
                         - ez[i][j+1][k]  + ez[i][j][k]) ;
hy[i][j][k] = hy[i][j][k]
                    + .5*( ez[i+1][j][k]  - ez[i][j][k]
                         - ex[i][j][k+1]  + ex[i][j][k]) ;
hz[i][j][k] = hz[i][j][k]
                    + .5*( ex[i][j+1][k]  - ex[i][j][k]
                         - ey[i+1][j][k]  + ey[i][j][k]) ;
```

The relationship between E and D, corresponding to Eq. (4.1b), is exactly the same as the one-dimensional or two-dimensional cases, except now there will be three equations!

The program fd3d_4.1.c at the end of the chapter is a very basic three-dimensional FDTD program with a source in the middle of the problem space. This is similar to the two-dimensional program fd2d_3.1.c, except that the source is not a simple point source. In three dimensions, the E field attenuates as the square of the distance as it propagates out from the point source, so we would have trouble just seeing it. Instead, we use a dipole antenna as the source. A simple dipole antenna is illustrated in Fig. 4.2. It consists of two metal arms. A dipole antenna functions by having current run through the arms, which results in radiation. FDTD simulates a dipole in the following way: The metal of the arms is specified by setting the `gaz` parameters to zero in the cells corresponding to metal (See Eqs. (2.9) and (2.10)). This insures that the corresponding E_z field at this point remains zero, as it would if that point were inside metal. The source is specified by setting the E_z field in the gap to a certain value. In fd3d_4.1.c, we specify a Gaussian pulse. (In a real dipole antenna, the E_z field in the gap would be the result of the current running through the metal arms.) Notice that we could have specified a current in the following manner: Ampere's circuital law says [2]

$$\oint_C \boldsymbol{H} \cdot dl = I,$$

i.e., the current through a surface is equal to the line integral of the H field around the surface. We could specify the current by setting the appropriate H field values around the gap, as illustrated in Fig. 4.2. I think you can see that specifying the E field in the gap is easier.

Figure 4.3 shows the propagation of the E_z from the dipole in the XY plane level with the gap of the dipole. Of course, there is radiation in the Z direction as well. This illustrates a major problem in three-dimensional simulation: unless one has unusually good graphics, visualizing three dimensions can be difficult.

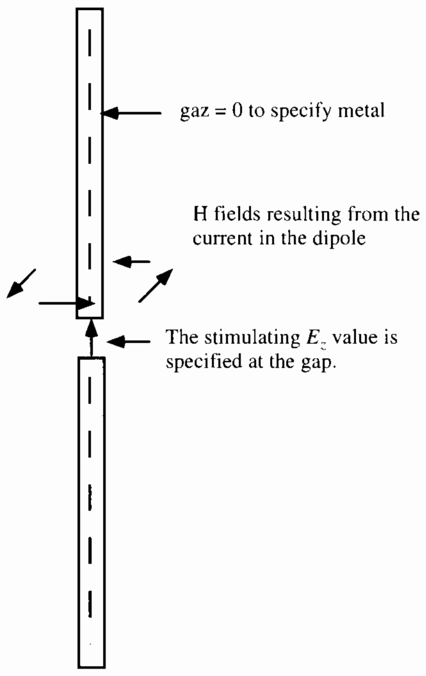
gaz = 0 to specify metal
H fields resulting from the current in the dipole
The stimulating E_z value is specified at the gap.

Figure 4.2 A dipole antenna. The FDTD program specifies the metal arms of the dipole by setting gaz = 0. The source is specified by setting the E_z field to a value at the gap.

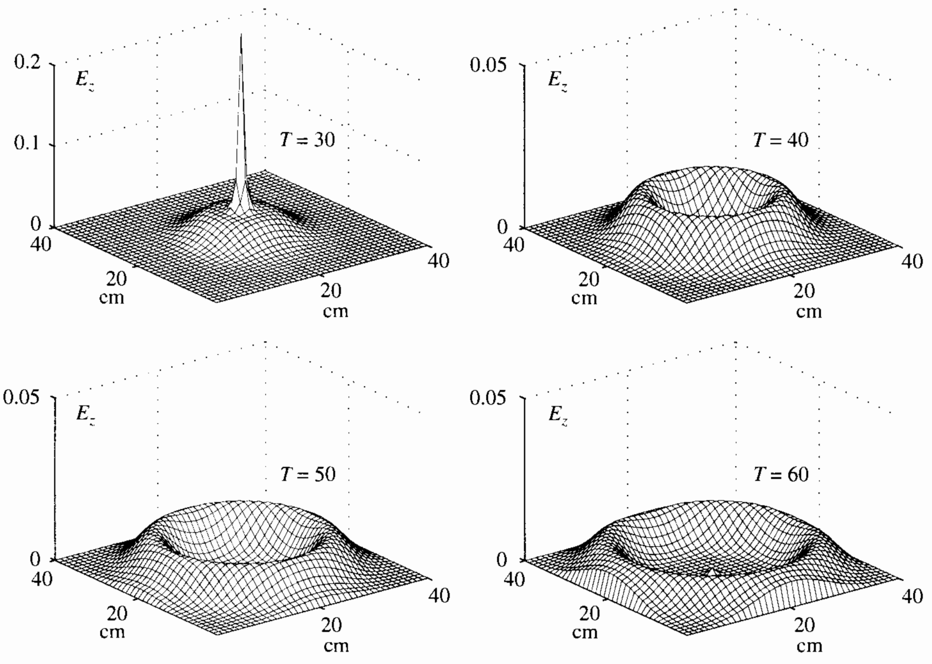

Figure 4.3 *E* field radiation from a dipole antenna in a three-dimensional FDTD program.

PROBLEM SET 4.1

1. Get the program fd3d_4.1.c running. Duplicate the results of Fig. 4.3. (Remember, there is no PML yet!)

4.2 THE PML IN THREE DIMENSIONS

The development of the PML for three dimensions closely follows the two-dimensional version. The only difference is that you deal with three directions instead of two [3]. For instance, Eq. (3.23a) becomes

$$j\omega \cdot \left(1+\frac{\sigma_x(x)}{j\omega\varepsilon_0}\right)\left(1+\frac{\sigma_y(y)}{j\omega\varepsilon_0}\right)\left(1+\frac{\sigma_z(z)}{j\omega\varepsilon_0}\right)^{-1} D_z = c_0 \cdot \left(\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y}\right) \tag{4.4}$$

and implementing it will closely follow the two-dimensional development. Start by rewriting Eq. (4.4) as

$$\begin{aligned} j\omega \cdot \left(1+\frac{\sigma_x(x)}{j\omega\varepsilon_0}\right)\left(1+\frac{\sigma_y(y)}{j\omega\varepsilon_0}\right) D_z &= c_0\left(1+\frac{\sigma_z(z)}{j\omega\varepsilon_0}\right)\cdot\left(\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y}\right) \\ &= c_0\cdot curl\_h + c_0 \cdot \frac{\sigma_z(z)}{\varepsilon_0}\frac{1}{j\omega} curl\_h, \end{aligned} \tag{4.5}$$

We will define

$$I_{Dz} = \frac{1}{j\omega} curl\_h,$$

which is an integration when it goes to the time domain, so Eq. (4.5) becomes

$$j\omega \cdot \left(1+\frac{\sigma_x(x)}{j\omega\varepsilon_0}\right)\left(1+\frac{\sigma_y(y)}{j\omega\varepsilon_0}\right) D_z = c_0 \cdot \left(curl\_h + \frac{\sigma_z(z)}{\varepsilon_0} I_{Dz}\right).$$

The implementation of this into FDTD parallels that of the two-dimensional PML, except the right side contains the integration term I_{Dz}. Therefore, following the same math we used in Chapter 3, we get

$$curl\_h = \begin{bmatrix} H_y^n(i+1/2,j,k+1/2) - H_y^n(i-1/2,j,k+1/2) \\ -H_x^n(i,j+1/2,k+1/2) \\ +H_x^n(i,j-1/2,k+1/2) \end{bmatrix} \tag{4.6a}$$

$$I_{Dz}^n(i,j,k+1/2) = I_{Dz}^{n-1}(i,j,k+1/2) + curl\_h \tag{4.6b}$$

$$\begin{aligned} D_z^{n+1/2}(i,j,k+1/2) &= gi3(i)\cdot gj3(j)\cdot D_z^{n-1/2}(i,j,k+1/2) \\ &\quad + gi2(i)\cdot gj2(j)\cdot .5\cdot (curl\_h + gk1(k)\cdot I_{Dz}^n(i,j,k+1/2)\ ). \end{aligned} \tag{4.6c}$$

The one-dimensional g parameters are defined the same as Eq. (3.25).

We also mentioned in the last chapter that this is a "seamless" transition from the main problem space to the PML, but it suffers from one main drawback: the integral parameter I_{Dz} is an additional three-dimensional array that is dimensioned throughout the problem space, but used only at two edges. The three-dimensional implementation will have a total of six such arrays, which clearly is a waste of computer resources. For this reason, I_{Dz} will be broken up into two smaller three-dimensional arrays, one defined at the low values of k (`idzl`) and one defined at the high values of k (`idzh`).

```
for( i=1; i< IE,i++) {
   for( j=1; j< JE,j++) {
      for(k=0; k <ka; k++ ) {
         curl_h = ( hy[i][j][k] - hy[i-1][j][k]
                  - hx[i][j][k] + hx[i][j-1][k]);
      idzl[i][j][k] = idzl[i][j][k]+ curl_h;
      dz[i][j][k] = gi3[i]*gj3[j]*dz[i][j][k]
       + gi2[i]*gj2[j].5*(curl_h + gk1[k]*idzl[i][j][k] );
}   }  }
```

```
for( i=1; i< IE,i++) {
    for( j=1; j< JE,j++) {
       for(k=ka; k <=kb; k++ ) {
       dz[i][j][k] = gi3[i]*gj3[j]*dz[i][j][k]
                           + gi2[i]*gj2[j].5*curl_h ;
}   }  }

for( i=1; i< IE,i++) {
    for( j=1; j< JE,j++) {
       for(k=kb+1; k < KE; k++ )
       kzh = KE - k - 1;{
           curl_h = ( hy[i][j][k] - hy[i-1][j][k]
                    - hx[i][j][k] + hx[i][j-1][k])
       idzh[i][j][kzh] = idzh[i][j][kzh]+ curl_h
       dz[i][j][k] = gi3[i]*gj3[j]*dz[i][j][k]
       + gi2[i]*gj2[j].5*(curl_h + gk1[k]*idzh[i][j][kzh]);
}   }  }
```

Figure 4.4 shows the *E* field emanating from a dipole source in a program using a seven-point PML. Notice that the part of the field not in the PML radiates concentrically from the source, as it should.

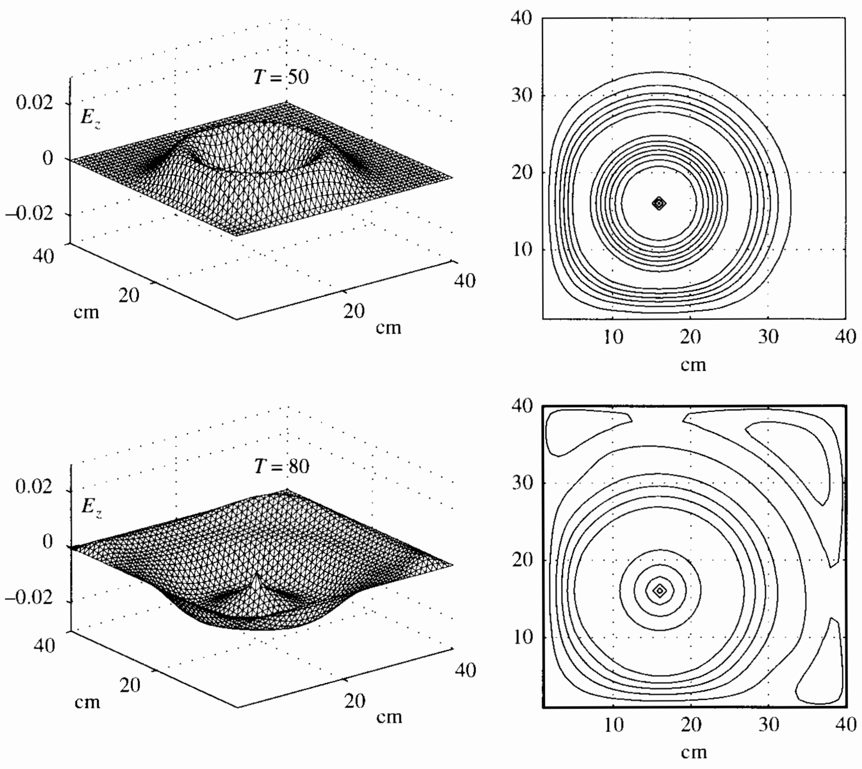

Figure 4.4 Radiation from a dipole antenna in an FDTD program with a seven-point PML. The contour diagrams on the right illustrate the fact that the fields remain concentric until they reach the PML.

PROBLEM SET 4.2

1. Add the PML to your three-dimensional program. The program fd3d_4.2.c at the end of the chapter has the PML, but it also has the things described in the next section. You should be able to pick out those things having to do with the PML. Duplicate the results of Fig. 4.4.

4.3 TOTAL/SCATTERED FIELD FORMULATION IN THREE DIMENSIONS

Generating plane waves in three dimensions is similar to two dimensions. The three-dimensional problem is illustrated in Fig. 4.5. A plane wave is generated in one plane of the three-dimensional problem space, in this case an XZ plane, at $j = ja$ and subtracted out at $j = jb$. Therefore, in free space with no obstacle in the total field, we should see only E_z and H_x. The plane wave is generated at one side and subtracted out the other side by adding to D or H fields that are on the boundary and subtracting from D or H fields that are next to the boundary. Therefore, Eqs. (3.26) through (3.28) are still used, but they are imposed on an entire plane instead of just a line as in two dimensions. Another difference in three dimensions is the additional surfaces at $k = ka$ and $k = kb$. Figure 4.6 illustrates the $k = ka$ boundary. The calculation of $D_y(i, j, ka)$, which is in the scattered field, requires the values of $H_x(i, j, ka)$, which is in the total field. The difference between the two is the incident component of the H_x field. A similar difference exists at the $k = kb$ boundary, where $D_y(i, j, kb + 1)$ is just above the scattered field. Therefore, besides equations similar to Eqs. (3.26), (3.27), and (3.28) the following are necessary:

$$D_y(i, j + 1/2, ka) = D_y(i, j + 1/2, ka) - .5 \cdot H_{x\_inc}(j) \tag{4.7a}$$

$$D_y(i, j + 1/2, kb + 1) = D_y(i, j + 1/2, kb + 1) + .5 \cdot H_{x\_inc}(j). \tag{4.7b}$$

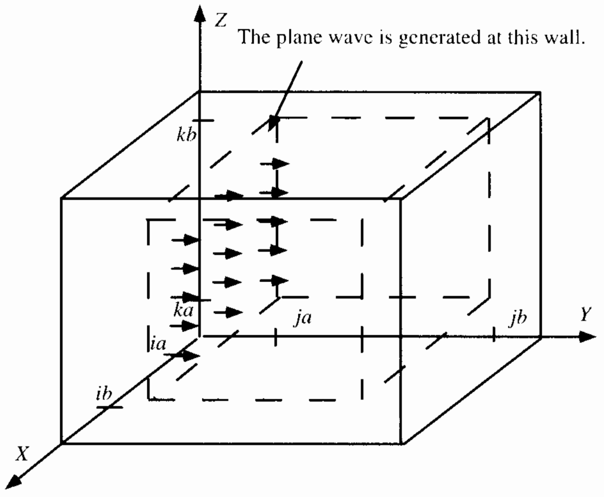

Figure 4.5 Total/scattered field in 3D.

4.3.1 A Plane Wave Impinging on a Dielectric Sphere

Now that we have a program that generates a plane wave in three dimensions, we will want to start putting objects in the problem space to see how the plane wave interacts with them. In two dimensions, we chose a cylinder because we had an analytic solution with which

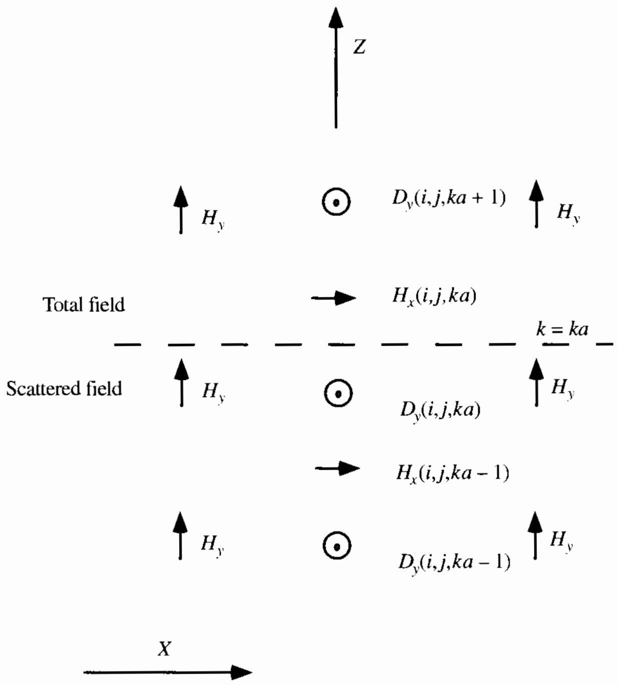

Figure 4.6 Total/scattered field boundary at $k = ka$.

we could check the accuracy of our calculation via a Bessel function expansion. It turns out that the interaction of a plane wave with a dielectric sphere can be determined through an expansion of the modified Bessel functions [4].

Specifying a sphere in three dimensions is very similar to specifying a cylinder in two dimensions. The major difference is that it must be done for all three electric fields. The following code specifies the parameters needed for the E_z field calculation:

```
for ( i=ia; i < ib; i++ ) {
   for ( j = ja; j < jb; j++ ) {
     for ( k = ka; k < kb; k++ ) {
          xdist = (ic-i);
          ydist = (jc-j);
          zdist = (kc-k-.5);
          dist
            = sqrt(pow(xdist,2.)+pow(ydist,2.)+pow(zdist,2.))
          if( dist <= radius) {
             gaz[i][j][k] = 1./(epsilon + (sigma*dt/epsz));
             gbz[i][j][k] = sigma*dt/epsz;
          }
} } }
```

Besides the obvious differences, e.g., three loops instead of two, there is another that should be pointed out; the parameter `zdist` calculates the distance to the point as if it were 1/2 a cell further in the z direction. That's because it is! Look at the diagram of the Yee cell in Fig. 4.1. Each E field is assumed offset from i, j, k by a half a cell in its own direction.

In Fig. 4.7, we show a comparison between FDTD results and Bessel function expansion results for a plane wave incident on a dielectric sphere. The sphere had a diameter of 20 cm, a dielectric constant of 30, and a conductivity of 0.3. The FDTD program is $40 \times 40 \times 40$, and

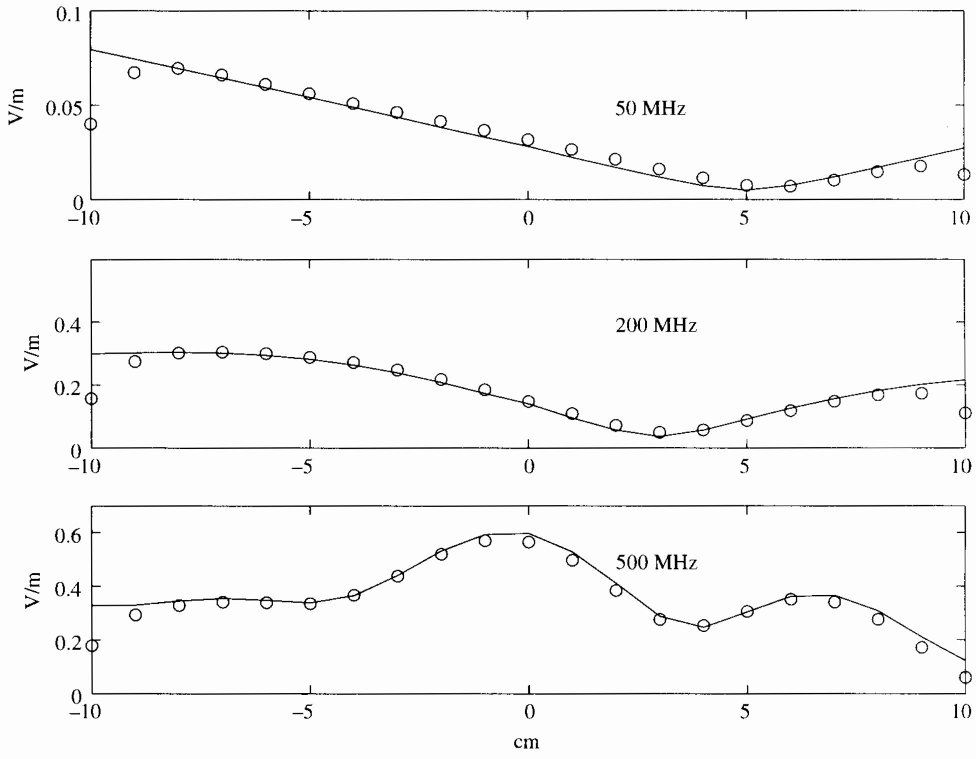

Figure 4.7 Comparison of FDTD calculation (circles) with Bessel function expansion calculations (lines) along the main axis of a dielectric sphere, 20 cm in diameter, with $\varepsilon_r = 30$ and $\sigma = 0.3$. The program for the FDTD calculation used the simple "in or out" strategy to determine the parameters.

uses a seven-point PML. Apparently, the "in and out" method of determining the parameters like gaz and gbz is not as forgiving as it was in two dimensions!

This method of determining parameters in three dimensions will give the same "staircasing" that we faced in two dimensions. We can do an averaging over subcells to improve efficiency. We might jump to the conclusion that because this meant averaging subcells in a plane in two dimensions, we will have to average subcubes in three dimensions. It turns out that this is not the case. Figure 4.8 illustrates the calculation of E_z, which uses the surrounding H_x and H_y values. Since this calculation is confined to a plane, we may think of the calculation

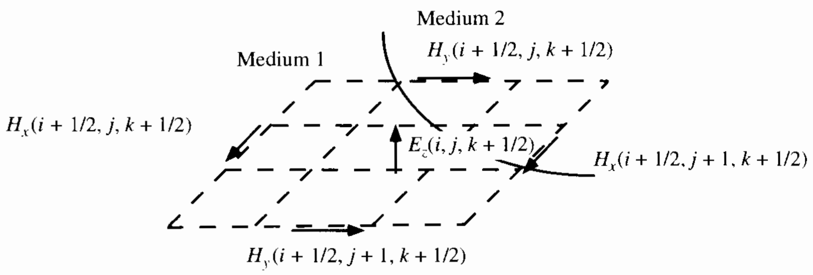
Medium 2
Medium 1
$H_y(i + 1/2, j, k + 1/2)$
$H_x(i + 1/2, j, k + 1/2)$
$E_z(i, j, k + 1/2)$
$H_x(i + 1/2, j + 1, k + 1/2)$
$H_y(i + 1/2, j + 1, k + 1/2)$

Figure 4.8 E_z is calculated by the surrounding H_x and H_y parameters. The parameters used to calculate E_z are determined by the percentage of subcells that lie in each medium. In the above example, six subcells are in medium 1 and three subcells are in medium 2.

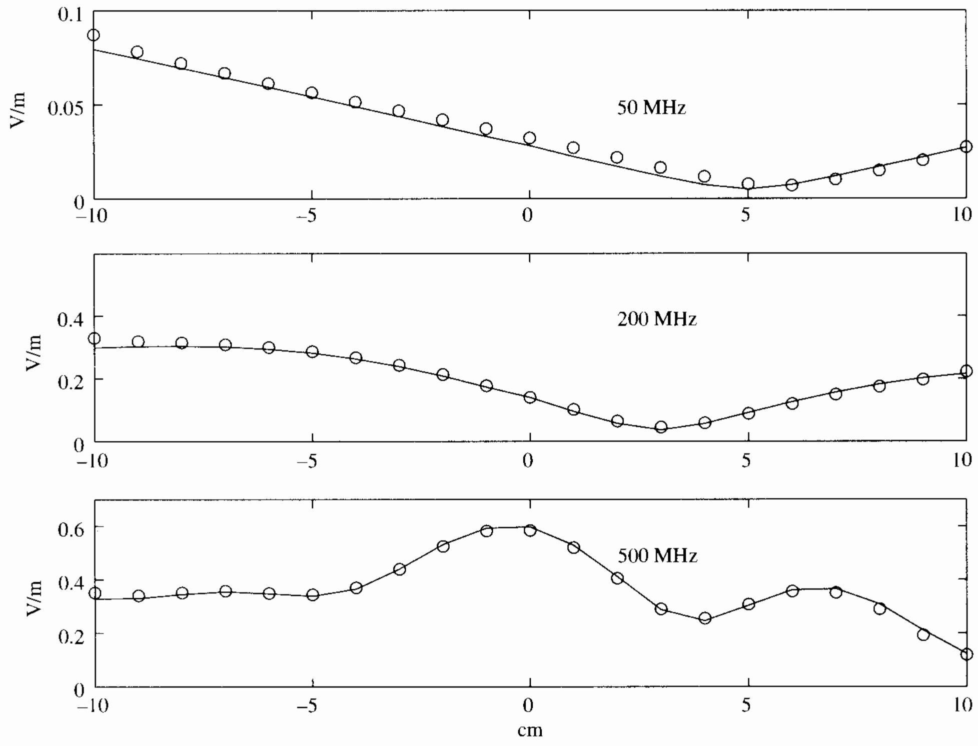

Figure 4.9 Comparison of FDTD calculation (circles) with Bessel function expansion calculations (lines) along the main axis of a dielectric sphere, 20 cm in diameter, with $\varepsilon_r = 30$ and $\sigma = 0.3$. The program used for the FDTD calculation averaged over nine subcells within each cell to determine the parameters.

of E_z as being a two-dimensional problem. Therefore, in determining the parameters gaz and gbz, we will go by how much of the area of the plane containing H_x and H_y is in each of the different media. The following code averages the properties over nine subcells:

```
for ( j = ja; j < jb; j++ ) {
    for ( i=ia; i < ib; i++ ) {
    eps  = epsilon(0);
    cond = sigma(0);
        for ( jj = -1; jj <= 1; jj++ ) {
        for ( ii = -1; ii <= 1; ii++ ) {
          xdist = (ic-i) + .333*ii;
          ydist = (jc-j) + .333*jj;
          zdist = (jc-j-.5);
          dist =
               sqrt(pow(xdist,2.)+pow(ydist,2.)+pow(zdist,2.);
          for (n=1; n <+ numcyl: n++1) {
             if( dist <= radius) {
             eps  = eps  + (1./9)*(epsilon[n] - epsilon[n-1]);
             cond = cond + (1./9)*(sigma[n] - sigma[n-1]);
          } }
        }}
        gaz[i][j] = 1./(eps + (sigma*dt/epsz));
        gbz[i][j] = cond*dt/epsz;
} }
```

Figure 4.9 repeats the FDTD/Bessel comparison, but with the averaged parameter values. Clearly, it results in an improvement.

PROBLEM SET 4.3

1. Add the plane wave propagation to your 3D FDTD program. Make sure that it can propagate a wave through and subtract it out the other end. (See the program fd3d_4.2.c).
2. The program fd3d_4.2.c creates the spheres by "in or out." Get this program running and duplicate the results of Fig. 4.7.
3. Use the averaging technique to get a more accurate calculation of the parameters, and duplicate the results of Fig. 4.9.

REFERENCES

[1] K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. on Antennas and Propagation*, vol. AP-17, 1996, pp. 585–589.

[2] D. K. Cheng, *Field and Wave Electromagnetics*, Menlo Park, CA: Addison-Wesley, 1992.

[3] D. M. Sullivan, A simplified PML for use with the FDTD method, *IEEE Microwave and Guided Wave Letters*, vol. 6, Feb. 1996, pp. 97–99.

[4] R. Harrington, *Time-Harmonic Electromagnetic Fields*, New York: McGraw-Hill, 1961.