# 1

# One-Dimensional Simulation with the FDTD Method

This chapter is a step-by-step introduction to the FDTD method. It begins with the simplest possible problem, the simulation of a pulse propagating in free space in one dimension. This example is used to illustrate the FDTD formulation. Subsequent sections lead to the formulation for more complicated media

## 1.1 ONE-DIMENSIONAL FREE SPACE FORMULATION

The time-dependent Maxwell's curl equations in free space are

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon_0} \nabla \times \boldsymbol{H} \tag{1.1a}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \boldsymbol{E}. \tag{1.1b}$$

$\boldsymbol{E}$ and $\boldsymbol{H}$ are vectors in three dimensions, so in general, Eq. (1.1a) and (1.1b) represent three equations each. We will start with a simple one-dimensional case using only $E_x$ and $H_y$, so Eq. (1.1a) and (1.1b) become

$$\frac{\partial E_x}{\partial t} = -\frac{1}{\varepsilon_0} \frac{\partial H_y}{\partial z} \tag{1.2a}$$

$$\frac{\partial H_y}{\partial t} = -\frac{1}{\mu_0} \frac{\partial E_x}{\partial z}. \tag{1.2b}$$

These are the equations of a plane wave with the electric field oriented in the $x$ direction, the magnetic field oriented in the $y$ direction, and traveling in the $z$ direction.

Taking the central difference approximations for both the temporal and spatial derivatives gives

$$\frac{E_x^{n+1/2}(k) - E_x^{n-1/2}(k)}{\Delta t} = -\frac{1}{\varepsilon_0} \frac{H_y^n(k+1/2) - H_y^n(k-1/2)}{\Delta x} \tag{1.3a}$$

$$\frac{H_y^{n+1}(k+1/2) - H_y^n(k+1/2)}{\Delta t} = -\frac{1}{\mu_0} \frac{E_x^{n+1/2}(k+1) - E_x^{n+1/2}(k)}{\Delta x}. \tag{1.3b}$$

In these two equations, time is specified by the superscripts, i.e., "$n$" actually means a time $t = \Delta t \cdot n$. Remember, we have to discretize everything for formulation into the computer. The term "$n + 1$" means one time step later. The terms in parentheses represent distance, i.e., "$k$" actually means the distance $z = \Delta x \cdot k$. (It might seem more sensible to use $\Delta z$ as the incremental step, since in this case we are going in the $z$ direction. However, $\Delta x$ is so commonly used for a spatial increment that I will use $\Delta x$.) The formulation of Eq. (1.3a) and (1.3b) assumes that the $\boldsymbol{E}$ and $\boldsymbol{H}$ fields are interleaved in both space and time. $\boldsymbol{H}$ uses the arguments $k + 1/2$ and $k - 1/2$ to indicate that the $\boldsymbol{H}$ field values are assumed to be located between the $\boldsymbol{E}$ field values. This is illustrated in Fig. 1.1. Similarly, the $n + 1/2$ or $n - 1/2$ superscript indicates that it occurs slightly after or before $n$, respectively.

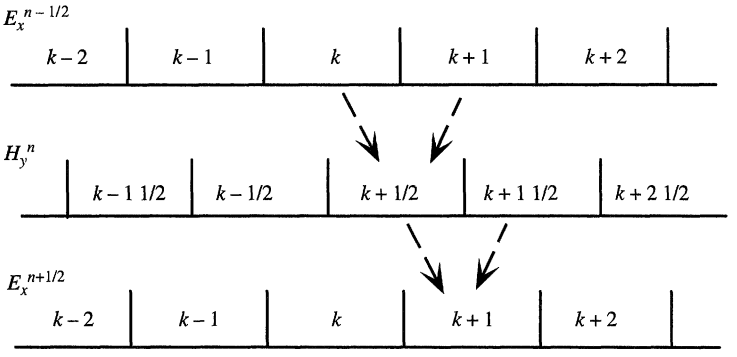


**Figure 1.1** Interleaving of the $E$ and $H$ fields in space and time in the FDTD formulation. To calculate $H_y(k + 1/2)$, for instance, the neighboring values of $E_x$ at $k$ and $k + 1$ are needed. Similarly, to calculate $E_x(k + 1)$, the value of $H_y$ at $k + 1/2$ and $k + 1\ 1/2$ are needed.

Eq. (1.3a) and (1.3b) can be rearranged in an iterative algorithm:

$$E_x^{n+1/2}(k) = E_x^{n-1/2}(k) - \frac{\Delta t}{\varepsilon_0 \cdot \Delta x}[H_y^n(k + 1/2) - H_y^n(k - 1/2)] \tag{1.4a}$$

$$H_y^{n+1}(k + 1/2) = H_y^n(k + 1/2) - \frac{\Delta t}{\mu_0 \cdot \Delta x}[E_x^{n+1/2}(k + 1) - E_x^{n+1/2}(k)]. \tag{1.4b}$$

Notice that the calculations are interleaved in both space and time. In Eq. (1.4a), for example, the new value of $E_x$ is calculated from the previous value of $E_x$ and the most recent values of $H_y$. This is the fundamental paradigm of the finite-difference time-domain (FDTD) method [1].

Eqs. (1.4a) and (1.4b) are very similar, but because $\varepsilon_0$ and $\mu_0$ differ by several orders of magnitude, $E_x$ and $H_y$ will differ by several orders of magnitude. This is circumvented by making the following change of variables [2]:

$$\tilde{E} = \sqrt{\frac{\varepsilon_0}{\mu_0}} E. \tag{1.5}$$

Substituting this into Eqs. (1.4a) and (1.4b) gives

$$\tilde{E}_x^{n+1/2}(k) = \tilde{E}_x^{n-1/2}(k) - \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta x}[H_y^n(k + 1/2) - H_y^n(k - 1/2)] \tag{1.6a}$$

$$H_y^{n+1}(k + 1/2) = H_y^n(k + 1/2) - \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta x}[\tilde{E}_x^{n+1/2}(k + 1) - \tilde{E}_x^{n+1/2}(k)] \tag{1.6b}$$

Once the cell size $\Delta x$ is chosen, then the time step $\Delta t$ is determined by

$$\Delta t = \frac{\Delta x}{2 \cdot c_0} \tag{1.7}$$

where $c_0$ is the speed of light in free space. (The reason for this will be explained later.) Therefore,

$$\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta x} = c_0 \cdot \frac{\Delta x / 2 \cdot c_0}{\Delta x} = \frac{1}{2} \tag{1.8}$$

Rewriting Eqs. (1.6a) and (1.6b) in C computer code gives the following:

```
ex[k] = ex[k] + 0.5*( hy[k-1] - hy[k] )          (1.9a)
hy[k] = hy[k] + 0.5*( ex[k] - ex[k+1] ).         (1.9b)
```

Note that the $n$ or $n + 1/2$ or $n - 1/2$ in the superscripts is gone. Time is implicit in the FDTD method. In Eq. (1.9a), the ex on the right side of the equal sign is the previous value at $n - 1/2$, and the ex on the left side is the new value, $n + 1/2$, which is being calculated. Position, however, is explicit. The only difference is that $k + 1/2$ and $k - 1/2$ are rounded off to $k$ and $k - 1$ in order to specify a position in an array in the program.

The program fd1d_1.1.c at the end of the chapter is a simple one-dimensional FDTD program. It generates a Gaussian pulse in the center of the problem space, and the pulse propagates away in both directions as seen in Fig. 1.2. The $E_x$ field is positive in both directions, but the $H_y$ field is negative in the negative direction. The following things are worth noting about the program:

1. The $E_x$ and $H_y$ values are calculated by separate loops, and they employ the interleaving described above.
2. After the $E_x$ values are calculated, the source is calculated. This is done by simply specifying a value of $E_x$ at the point `k = kc`, and overriding what was previously calculated. This is referred to as a "hard source," because a specific value is imposed on the FDTD grid.

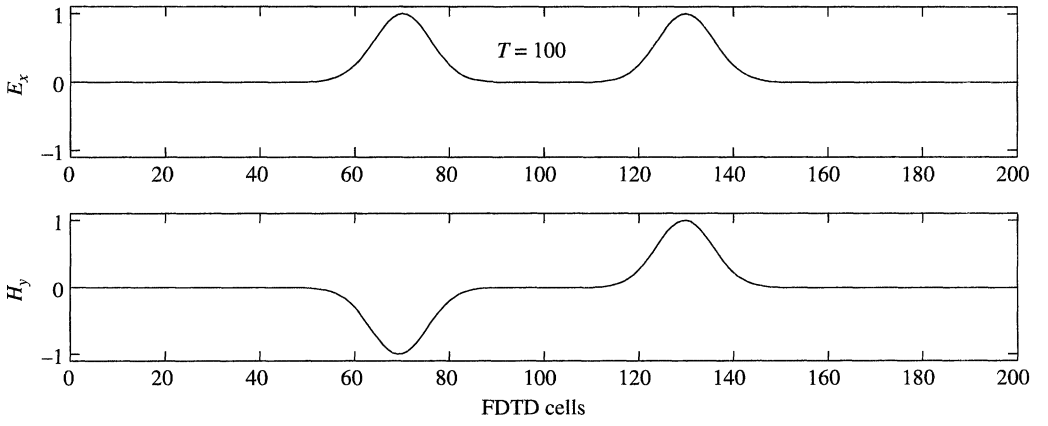


**Figure 1.2** FDTD simulation of a pulse in free space after 100 time steps. The pulse originated in the center and travels outward.

### PROBLEM SET 1.1

1. Get the program fd1d_1.1.c running. What happens when the pulse hits the end of the array? Why?
2. Modify the program so it has two sources, one at `kc - 20` and one at `kc + 20` (Notice that kc is the center of the problem space). What happens when the pulses meet? Explain this from basic EM theory.
3. Instead of $E_x$ as the source, use $H_y$ at `k = kc` as the source. What difference does it make? Try a two-point magnetic source at `kc - 1` and `kc` such that `hy[kc-1] = -hy[kc]`. What does this look like? What does it correspond to physically?

## 1.2 STABILITY AND THE FDTD METHOD

Let us return to the discussion of how we determine the time step. An electromagnetic wave propagating in free space cannot go faster than the speed of light. To propagate a distance of one cell requires a minimum time of $\Delta t = \Delta x/c_0$. When we get to two-dimensional simulation, we have to allow for the propagation in the diagonal direction, which brings the time requirement to $\Delta t = \Delta x/(\sqrt{2}c_0)$. Obviously, three-dimensional simulation requires $\Delta t = \Delta x/(\sqrt{3}c_0)$. This is summarized by the well-known "Courant Condition" [3, 4]:

$$\Delta t \leq \frac{\Delta x}{\sqrt{n} \cdot c_0}, \tag{1.10}$$

where $n$ is the dimension of the simulation. Unless otherwise specified, throughout this book we will determine $\Delta t$ by

$$\Delta t = \frac{\Delta x}{2 \cdot c_0}. \tag{1.11}$$

This is not necessarily the best formula; however, we will use it for simplicity.

### PROBLEM SET 1.2

1. In fd1d_1.1.c, go to the two governing equations, Eq. (1.9a) and (1.9b), and change the factor 0.5 to 1.0. What happens? Change it to 1.1. Now what happens? Change it to .25 and see what happens.

## 1.3 THE ABSORBING BOUNDARY CONDITION IN ONE DIMENSION

Absorbing boundary conditions are necessary to keep outgoing $\boldsymbol{E}$ and $\boldsymbol{H}$ fields from being reflected back into the problem space. Normally, in calculating the $\boldsymbol{E}$ field, we need to know the surrounding $\boldsymbol{H}$ values; this is a fundamental assumption of the FDTD method. At the edge of the problem space we will not have the value to one side. However, we have an advantage because we know there are no sources outside the problem space. Therefore, the fields at the edge must be propagating outward. We will use these two facts to estimate the value at the end by using the value next to it [5].

Suppose we are looking for a boundary condition at the end where $k = 0$. If a wave is going toward a boundary in free space, it is traveling at $c_0$, the speed of light. So in one time step of the FDTD algorithm, it travels

$$\text{distance} = c_0 \cdot \Delta t = c_0 \cdot \frac{\Delta x}{c_0} = \frac{\Delta x}{2}.$$

This equation basically explains that it takes two time steps for a wave front to cross one cell. So a common sense approach tells us that an acceptable boundary condition might be

$$E_x^n(0) = E_x^{n-2}(1). \tag{1.12}$$

It is relatively easy to implement this. Simply store a value of $E_x(1)$ for two time steps, and then put it in $E_x(0)$. Boundary conditions such as these have been implemented at both ends of the $E_x$ array in the program fd1d_1.2.c. (In the printout labeled fd1d_1.2.c at the end of the chapter, the entire program hasn't been reproduced, only those parts which are different from fd1d_1.1.c. Furthermore, key points are highlighted in boldface.) Figure 1.3 shows the results of a simulation using fd1d_1.2.c. A pulse that originates in the center propagates outward and is absorbed without reflecting anything back into the problem space.

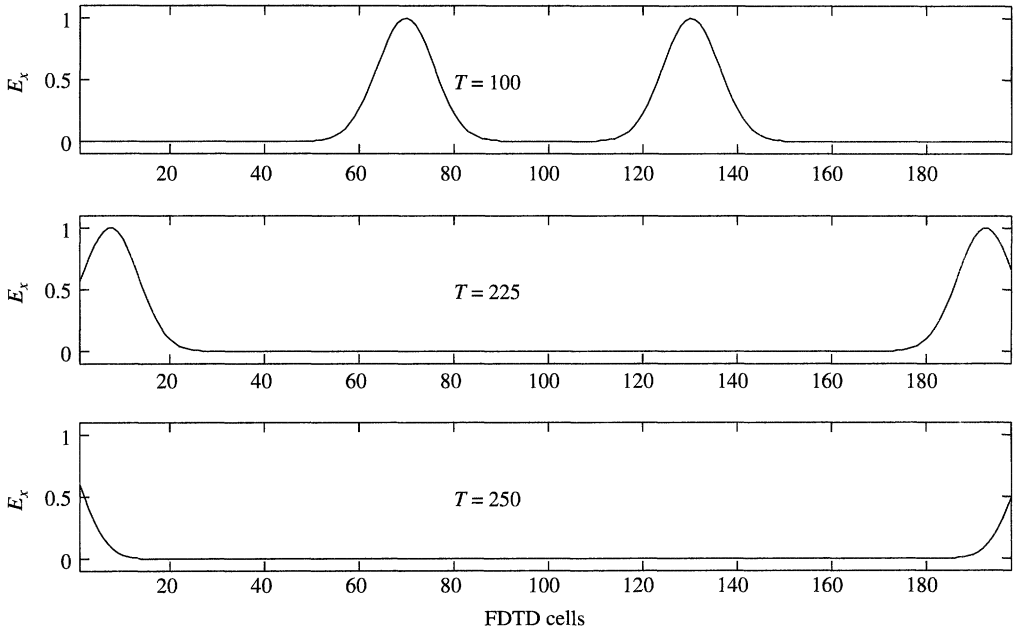


**Figure 1.3** Simulation of an FDTD program with absorbing boundary conditions. Notice that the pulse is absorbed at the edges without reflecting anything back.

## PROBLEM SET 1.3

1. The program fd1d_1.2.c has absorbing boundary conditions at both ends. Get this program running and test it to ensure the boundary conditions are completely absorbing the pulse.

## 1.4 PROPAGATION IN A DIELECTRIC MEDIUM

In order to simulate a medium with a dielectric constant other than one, which corresponds to free space, we have to add the relative dielectric constant $\varepsilon_r$ to Maxwell's equations:

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon_r \varepsilon_0} \nabla \times \boldsymbol{H} \tag{1.13a}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \boldsymbol{E}. \tag{1.13b}$$

We will stay with our one-dimensional example and make the change of variables in Eq. (1.5),

$$\frac{\partial \tilde{E}_x(t)}{\partial t} = \frac{1}{\varepsilon_r \sqrt{\varepsilon_0 \mu_0}} \cdot \frac{\partial H_y(t)}{\partial z}$$

$$\frac{\partial H_y(t)}{\partial t} = -\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \cdot \frac{\partial \tilde{E}_x(t)}{\partial z},$$

and then go to the finite difference approximations

$$\frac{\tilde{E}_x^{n+1/2}(k) - \tilde{E}_x^{n-1/2}(k)}{\Delta t} = \frac{1}{\varepsilon_r \sqrt{\varepsilon_0 \mu_0}} \cdot \frac{H_y^n(k+1/2) - H_y^n(k+1/2)}{\Delta x} \tag{1.14a}$$

$$\frac{H_y^{n+1}(k+1/2) - H_y^n(k+1/2)}{\Delta t} = -\frac{1}{\mu_0} \frac{\tilde{E}_x^{n+1/2}(k+1) - \tilde{E}_x^{n+1/2}(k)}{\Delta x}. \tag{1.14b}$$

From the previous section

$$\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta x} = \frac{1}{2},$$

so Eq. (1.14) becomes

$$\tilde{E}_x^{n+1/2}(k) = \tilde{E}_x^{n+1/2}(k) + \frac{1/2}{\varepsilon_r}[H_y^n(k+1/2) - H_y^n(k+1/2)] \tag{1.15a}$$

$$H_y^{n+1}(k+1/2) = H_y^n(k+1/2) - \frac{1}{2}[\tilde{E}_x^{n+1/2}(k+1) - \tilde{E}_x^{n+1/2}(k)]. \tag{1.15b}$$

From these we can get the computer equations

```
ex[k] = ex[k] + cb[k]*(hy[k-1] - hy[k] )        (1.16a)
hy[k] = hy[k] + 0.5*(ex[k] - ex[k+1] ),         (1.16b)
```

where

```
cb[k] = .5/epsilon                              (1.17)
```

over those values of $k$ which specify the dielectric material.

The program fd1d_1.3.c simulates the interaction of a pulse traveling in free space until it strikes a dielectric medium. The medium is specified by the parameter `cb` in Eq. (1.17). Figure 1.4 shows the result of a simulation with a dielectric medium having a relative dielectric constant of 4. Note that a portion of the pulse propagates into the medium and a portion is reflected, in keeping with basic EM theory [6].

## PROBLEM SET 1.4

1. The program fd1d_1.3.c simulates a problem partly containing free space and partly dielectric material. Run this program and duplicate the results of Fig. 1.4.
2. Look at the relative amplitudes of the reflected and transmitted pulses. Are they correct? Check them by calculating the reflection and transmission coefficients. (See the appendix at the end of this chapter.)
3. Still using a dielectric constant of 4, let the transmitted pulse propagate until it hits the far right wall. What happens? What could you do to correct this?

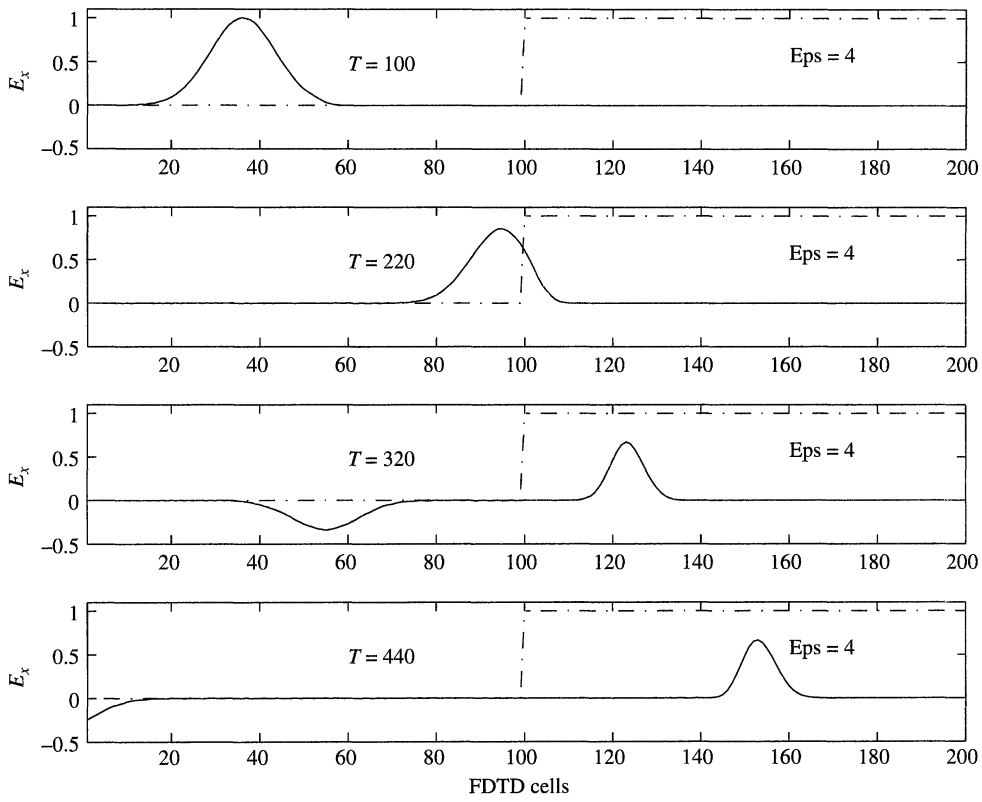


**Figure 1.4** Simulation of a pulse striking a dielectric material with a dielectric constant of 4. The source originates at cell number 5.

## 1.5 SIMULATING DIFFERENT SOURCES

In the first two programs, a source is assigned a value to $E_x$; this is referred to as a *hard source*. In fd1d_1.3.c, however, a value is added to $E_x$ at a certain point; this is called a *soft source*. The reason is that with a hard source, a propagating pulse will see that value and be reflected, because a hard value of $E_x$ looks like a metal wall to FDTD. With the soft source, a propagating pulse will just pass through.

Up until now, we have been using a Gaussian pulse as the source. It is very easy to switch to a sinusoidal source. Just replace the parameter *pulse* with the following:

```
pulse = sin[2*pi*freq_in*dt*T]
ex[5] = ex[5] + pulse.
```

The parameter *freq_in* determines the frequency of the wave. This source is used in the program fd1d_1.4.c. Figure 1.5 shows the same dielectric medium problem with a sinusoidal source. A frequency of 700 MHz is used. Notice that the simulation was stopped before the wave reached the far right side. Remember that we have an absorbing boundary condition, but it was only for free space!

Note that in fd1d_1.4.c the cell size `ddx` and the time step `dt` are specified explicitly. We do this because we need `dt` in the calculation of pulse. The cell size `ddx` is only specified because it is needed to calculate `dt` from Eq. (1.7).

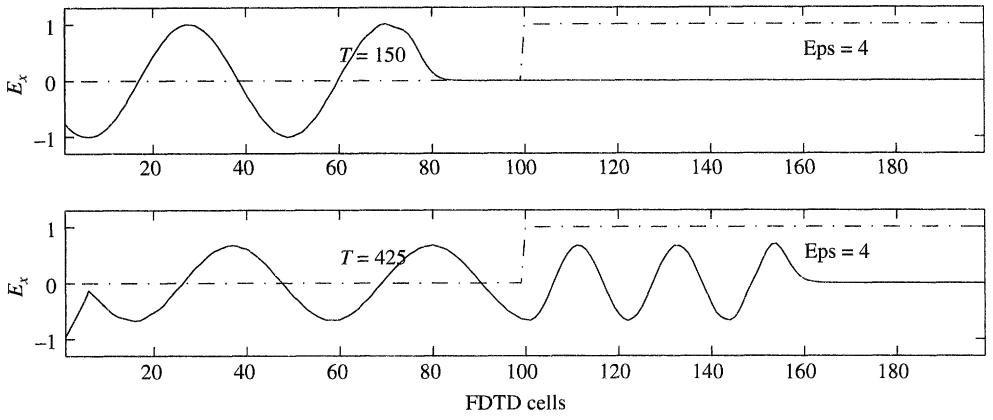


**Figure 1.5** Simulation of a propagating sinusoidal wave of 700 MHz striking a medium with a relative dielectric constant of 4.

## PROBLEM SET 1.5

1. Modify your program fd1d_1.3.c to simulate the sinusoidal source (see fd1d_1.4.c).
2. Keep increasing your incident frequency from 700 MHz upward at intervals of 300 MHz. What happens?
3. A type of propagating wave function that is of great interest in areas such as optics is the "wave packet," which is a sinusoidal function in a Gaussian envelope. Modify your program to simulate a wave packet.

## 1.6 DETERMINING CELL SIZE

Choosing the cell size to be used in an FDTD formulation is similar to any approximation procedure: enough sampling points must be taken to ensure that an adequate representation is made. The number of points per wavelength is dependent on many factors [3, 4]. However, a good rule of thumb is 10 points per wavelength. Experience has shown this to be adequate, with inaccuracies appearing as soon as the sampling drops below this rate.

Naturally, we must use a worst-case scenario. In general, this will involve looking at the highest frequencies we are simulating and determining the corresponding wavelength. For instance, suppose we are running simulations at 400 MHz. In free space, EM energy will propagate at the wavelength

$$\lambda_0 = \frac{c_0}{400\ \text{MHz}} = \frac{3 \times 10^8\ \text{m/sec}}{4 \times 10^8\ \text{sec}^{-1}} = .75\ \text{m}. \tag{1.18}$$

If we were only simulating free space we could choose

$$\Delta x = \lambda_0/10 = 7.5\ \text{cm}.$$

However, if we are simulating EM propagation in biological tissues, for instance, we must look at the wavelengths in the tissue with the highest dielectric constant, because this will have the corresponding shortest wavelength. For instance, muscle has a relative dielectric constant of about 50 at 400 MHz, so

$$\lambda_m = \frac{c_0/\sqrt{50}}{400\ \text{MHz}} = \frac{.424 \times 10^8\ \text{m/sec}}{4 \times 10^8\ \text{sec}^{-1}} = 10.6\ \text{cm}, \tag{1.19}$$

and we would probably select a cell size of one centimeter.

## PROBLEM SET 1.6

1. Simulate a 3-GHz sine wave impinging on a material with a dielectric constant of $\varepsilon_r = 20$.

## 1.7 PROPAGATION IN A LOSSY DIELECTRIC MEDIUM

So far, we have simulated EM propagation in free space or in simple media that are specified by the relative dielectric constant $\varepsilon_r$. However, there are many media that also have a loss term specified by the conductivity. This loss term results in the attenuation of the propagating energy.

Once more we will start with the time-dependent Maxwell's curl equations, but we will write them in a more general form, which will allow us to simulate propagation in media that have conductivity:

$$\varepsilon \frac{\partial \boldsymbol{E}}{\partial t} = \nabla \times \boldsymbol{H} - \boldsymbol{J} \tag{1.20a}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \boldsymbol{E}. \tag{1.20b}$$

$\boldsymbol{J}$ is the current density, which can also be written

$$\boldsymbol{J} = \sigma \cdot \boldsymbol{E},$$

where $\sigma$ is the conductivity. Putting this into Eq. (1.20a) and dividing through by the dielectric constant we get

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon_0 \varepsilon_r} \nabla \times \boldsymbol{H} - \frac{\sigma}{\varepsilon_0 \varepsilon_r} \boldsymbol{E}.$$

We now revert to our simple one-dimensional equation:

$$\frac{\partial E_x(t)}{\partial t} = -\frac{1}{\varepsilon_r \varepsilon_0} \cdot \frac{\partial H_y(t)}{\partial z} - \frac{\sigma}{\varepsilon_r \varepsilon_0} E_x(t),$$

and make the change of variable in Eq. (1.5), which gives

$$\frac{\partial \tilde{E}_x(t)}{\partial t} = -\frac{1}{\varepsilon_r \sqrt{\varepsilon_0 \mu_0}} \cdot \frac{\partial H_y(t)}{\partial z} - \frac{\sigma}{\varepsilon_r \varepsilon_0} \tilde{E}_x(t) \tag{1.21a}$$

$$\frac{\partial H_y(t)}{\partial t} = -\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \cdot \frac{\partial \tilde{E}_x(t)}{\partial z}. \tag{1.21b}$$

Next take the finite difference approximations for both the temporal and spatial derivatives similar to Eq. (1.3a):

$$\begin{aligned} \frac{\tilde{E}_x^{n+1/2}(k) - \tilde{E}_x^{n-1/2}(k)}{\Delta t} = &-\frac{1}{\varepsilon_r \sqrt{\varepsilon_0 \mu_0}} \cdot \frac{H_y^n(k+1/2) - H_y^n(k-1/2)}{\Delta x} \\ &- \frac{\sigma}{\varepsilon_r \varepsilon_0} \frac{\tilde{E}_x^{n+1/2}(k) + \tilde{E}_x^{n-1/2}(k)}{2}. \end{aligned} \tag{1.22}$$

Notice that the last term in Eq. (1.21a) is approximated as the average across two time steps in Eq. (1.22). From the previous section

$$\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \frac{\Delta t}{\Delta x} = \frac{1}{2},$$

so Eq. (1.22) becomes

$$\tilde{E}_x^{n+1/2}(k)\left[1+\frac{\Delta t\cdot\sigma}{2\varepsilon_r\varepsilon_0}\right]=\tilde{E}_x^{n-1/2}(k)\left[1-\frac{\Delta t\cdot\sigma}{2\varepsilon_r\varepsilon_0}\right]-\frac{1/2}{\varepsilon_r}[H_y^n(k+1/2)-H_y^n(k-1/2)]$$

or

$$\tilde{E}_x^{n+1/2}(k)=\frac{\left(1-\dfrac{\Delta t\cdot\sigma}{2\varepsilon_r\varepsilon_0}\right)}{\left(1+\dfrac{\Delta t\cdot\sigma}{2\varepsilon_r\varepsilon_0}\right)}\tilde{E}_x^{n-1/2}(k)-\frac{1/2}{\varepsilon_r\cdot\left(1+\dfrac{\Delta t\cdot\sigma}{2\varepsilon_x\varepsilon_0}\right)}[H_y^n(k+1/2)-H_y^n(k-1/2)].$$

From these we can get the computer equations

```
ex[k] = ca[k]*ex[k] + cb[k] *( hy[k-1] - hy[k] )      (1.23a)
hy[k] = hy[k] + 0.5*( ex[k] - ex[k+1] ),              (1.23b)
```

where

```
eaf = dt*sigma/(2*epsz*epsilon)                       (1.24a)
ca[k] = (1.  - eaf)/(1.  + eaf)                       (1.24b)
cb[k] =0.5/(epsilon*(1.  + eaf)).                     (1.24c)
```

The program fd1d_1.5.c simulates a sinusoidal wave hitting a lossy medium that has a dielectric constant of 4 and a conductivity of 0.04. The pulse is generated at the far left side and propagates to the right. This is illustrated in Fig. 1.6. Notice that the waveform in the medium is absorbed before it hits the boundary, so we don't have to worry about absorbing boundary conditions.

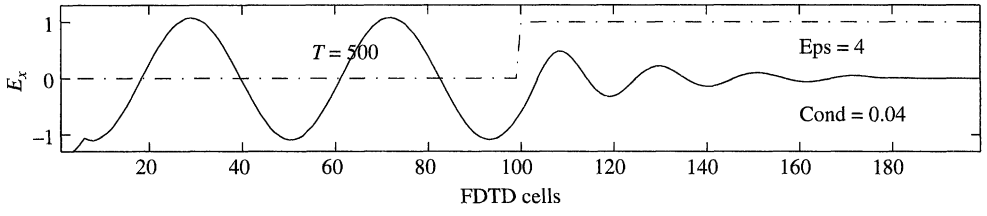


**Figure 1.6** Simulation of a propagating sinusoidal wave striking a lossy dielectric material with a dielectric constant of 4 and a conductivity of 0.04 (S/m). The source is 700 MHz, and originates at cell number 5.

## PROBLEM SET 1.7

1. Run program fd1d_1.5.c to simulate a complex dielectric material. Duplicate the results of Fig. 1.6.
2. Verify that your calculation of the sine wave in the lossy dielectric is correct, i.e., it is the correct amplitude going into the slab, and then it attenuates at the proper rate. You can best do this by writing a little program that calculates the parameters given in the appendix of this chapter.
3. How would you write an absorbing boundary condition for a lossy material?
4. Simulate a pulse hitting a metal wall. This is very easy to do, if you remember that metal has a very high conductivity. For the complex dielectric, just use $\sigma$ = 1.e6, or any large number. (It does not have to be the correct conductivity of the metal, just very large.) What does this do to the FDTD parameters *ca* and *cb*? What result does this have for the field parameters $E_x$ and $H_y$? If you didn't want to specify dielectric parameters, how else could you simulate metal in an FDTD program?

## APPENDIX 1.A

When a plane wave traveling in medium 1 strikes a medium 2, the fraction that is reflected is given by the reflection coefficient $\Gamma$, and the fraction that is transmitted into medium 2 is given by the transmission coefficient $\tau$. These are determined by the intrinsic impedances $\eta_1$ and $\eta_2$ of the respective media [6, p. 398]:

$$\Gamma = \frac{E_{ref}}{E_{inc}} = \frac{\eta_2 - \eta_1}{\eta_2 + \eta_1} \tag{1.A.1}$$

$$\tau = \frac{E_{trans}}{E_{inc}} = \frac{2\eta_2}{\eta_2 + \eta_1}. \tag{1.A.2}$$

The impedances are given by

$$\eta = \sqrt{\frac{\mu}{\varepsilon_0 \varepsilon_r^*}} \tag{1.A.3}$$

where $\varepsilon_r^*$ is the complex relative dielectric constant

$$\varepsilon_r^* = \varepsilon_r + \frac{\sigma}{j\omega\varepsilon_0}.$$

For the case where $\mu = \mu_0$, Eqs. (1.A.1) and (1.A.2) become

$$\Gamma = \frac{\dfrac{1}{\sqrt{\varepsilon_2^*}} - \dfrac{1}{\sqrt{\varepsilon_1^*}}}{\dfrac{1}{\sqrt{\varepsilon_2^*}} + \dfrac{1}{\sqrt{\varepsilon_1^*}}} = \frac{\sqrt{\varepsilon_1^*} - \sqrt{\varepsilon_2^*}}{\sqrt{\varepsilon_1^*} + \sqrt{\varepsilon_2^*}} \tag{1.A.4}$$

$$\tau = \frac{2/\sqrt{\varepsilon_2^*}}{\dfrac{1}{\sqrt{\varepsilon_2^*}} + \dfrac{1}{\sqrt{\varepsilon_1^*}}} = \frac{2 \cdot \sqrt{\varepsilon_1^*}}{\sqrt{\varepsilon_1^*} + \sqrt{\varepsilon_2^*}}. \tag{1.A.5}$$

The amplitude of an electric field propagating in the positive $z$ direction in a lossy dielectric medium is given by

$$E_x = E_0 e^{-\alpha z} e^{-j\beta z},$$

where $E_0$ is the amplitude at $z = 0$. The parameters $\alpha$ and $\beta$ are determined by the dielectric constant $\varepsilon_r$, the conductivity $\sigma$, and the radian frequency $\omega = 2\pi f$ of the propagating wave, via the following two formulas [6, p. 420]:

$$\alpha = \frac{\omega}{c_0}\sqrt{\frac{\varepsilon_r}{2}}\left[\sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon_0\varepsilon_r}\right)^2} - 1\right]^{1/2} \quad \text{(Np/m)} \tag{1.A.6}$$

$$\beta = \frac{\omega}{c_0}\sqrt{\frac{\varepsilon_r}{2}}\left[\sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon_0\varepsilon_r}\right)^2} + 1\right]^{1/2} \quad \text{(rad/m)}. \tag{1.A.7}$$

## REFERENCES

[1] K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas and Propagat.*, vol. 17, 1966, pp. 585–589.

[2] A. Taflove and M. Brodwin, Numerical solution of steady state electromagnetic scattering problems using the time-dependent Maxwell's equations, *IEEE Trans. Microwave Theory Tech.*, vol. 23, 1975, pp. 623–730.

[3] A. Taflove, *Computation Electrodynamics: The Finite-Difference Time-Domain Method.* Boston, MA: Artech House, 1995.

[4] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics.* Boca Raton, FL; CRC Press, 1993.

[5] G. Mur, Absorbing boundary conditions for the finite-difference approximation of the time domain electromagnetic field equations, *IEEE Trans. Electromagn. Compat.*, vol. 23, 1981, pp. 377–384.

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

```
/* FD1D_1.1.c.  1D FDTD simulation in free space */

# include <math.h>
# include <stdlib.h>
# include <stdio.h>

#define KE  200                        /* KE is the number of cells to be used */

main ()
{
    float ex[KE],hy[KE];
    int n,k,kc,ke,NSTEPS;
    float T;
    float t0,spread,pulse;
    FILE *fp, *fopen();

     /* Initialize */
     for ( k=1; k < KE; k++ )
     {    ex[k] = 0;
          hy[k] = 0. }

    kc = KE/2;                 /* Center of the problem space */
    t0 = 40.0;                 /* Center of the incident pulse */
    spread = 12;               /* Width of the incident pulse */
    T = 0;
    NSTEPS = 1;

while ( NSTEPS > 0 ) {
       printf( "NSTEPS --> ");     /* NSTEPS is the number of times the */
       scanf("%d", &NSTEPS);       /* main loop has executed */
       printf("%d \n", NSTEPS);
       n= 0;

    for ( n=1; n <=NSTEPS ; n++)
    {
       T = T + 1;                  /* T keeps track of the total number */
                                   /* of times the main loop is executed */
    /*  Main FDTD Loop */

       /* Calculate the Ex field */
       for ( k=1; k < KE; k++ )
       { ex[k] = ex[k] + .5*( hy[k-1] - hy[k] ) ; }

       /* Put a Gaussian pulse in the middle */

       pulse =  exp(-.5*(pow( (t0-T)/spread,2.0) ));
       ex[kc] =  pulse;
       printf( "%5.1f  %6.2f\n",t0-T,ex[kc]);
```

```
        /* Calculate the Hy field */
        for ( k=0; k < KE-1; k++ )
        { hy[k] = hy[k] + .5*( ex[k] - ex[k+1] ) ; }

        }

    /* End of the Main FDTD Loop */

        /* At the end of the calculation, print out
              the Ex and Hy fields */
        for ( k=1; k <= KE; k++ )
        { printf( "%3d   %6.2f  %6.2f\n",k,ex[k],hy[k]); }

        /* Write the E field out to a file "Ex" */
        fp = fopen( "Ex","w");
        for ( k=1; k <= KE; k++ )
        { fprintf( fp,"  %6.2f \n",ex[k]); }
        fclose(fp);

        /* Write the H field out to a file "Hy" */
        fp = fopen( "Hy","w");
        for ( k=1; k <= KE; k++ )
        { fprintf( fp,"  %6.2f \n",hy[k]); }
        fclose(fp);

        printf( "T = %5.0f\n",T);
}

}
```

```
/* FD1D_1.2.c.  1D FDTD simulation in free space */
/* Absorbing Boundary Condition added */


main ()
{
  float ex[KE],hy[KE];
      float ex_low_m1,ex_low_m2,ex_high_m1,ex_high_m2;
.
.
    for ( n=1; n <=NSTEPS ; n++)
    {
       T = T + 1;

    /*  Main FDTD Loop */

       /* Calculate the Ex field */
       for ( k=1; k < KE; k++ )
       { ex[k] = ex[k] + .5*( hy[k-1] - hy[k] ) ; }

       /* Put a Gaussian pulse in the middle */

       pulse =  exp(-.5*(pow( (t0-T)/spread,2.0) ));
       ex[kc] = ex[kc] + pulse;
       printf( "%5.1f %6.2f %6.2f'cn",t0-T,arg,ex[kc]);

           /* Absorbing Boundary Conditions */

            ex[0]         =       ex_low_m2;
            ex_low_m2     =       ex_low_m1;
            ex_low_m1     =       ex[1];

            ex[KE-1]      =       ex_high_m2;
            ex_high_m2    =       ex_high_m1;
            ex_high_m1    =       ex[KE-2];

       /* Calculate the Hy field */
       for ( k=0; k < KE-1; k++ )
       { hy[k] = hy[k] + .5*( ex[k] - ex[k+1] ) ; }
       }
  /* End of the Main FDTD Loop */
```

```
/* FD1D_1.3.c. */
/* Simulation of a pulse hitting a dielectric medium */

main ()
{
  float ex[KE],hy[KE];
        int n,k,kc,ke,kstart,nsteps;
        float ddx,dt,T,epsz,epsilon,sigma,eaf;
        float cb[KE];
.
.
    for ( k=1; k <= KE; k++ ) {  /* Initialize to free space */
      cb[k] = .5;
    }
        printf( "Dielectric starts at --> ");
        scanf("%d", &kstart);
        printf( "Epsilon --> ");
        scanf("%f", &epsilon);
        printf("%d %6.2f   \n", kstart,epsilon);

    for ( k=kstart; k <= KE; k++ ) {
      cb[k] = .5/epsilon;
    }
       for ( k=1; k <= KE; k++ )
       { printf( "%2d   %4.2f\n",k,cb[k]); }

/* Main part of the program */

while ( nsteps > 0 ) {
       printf( "nsteps --> ");
       scanf("%d", &nsteps);
       printf("%d \n", nsteps);

    for ( n=1; n <=nsteps ; n++)
    {
       T = T + 1;

       /* Calculate the Ex field */
       for ( k=0; k < KE; k++ )
        { ex[k] = ex[k] + cb[k]*( hy[k-1] - hy[k] ) ; }

       /* Put a Gaussian pulse at the low end */

       pulse =  exp(-.5*(pow((t0-T)/spread,2.0)));
        ex[5] =  ex[5] + pulse;
       printf( "%5.1f %6.2f %6.2f\n",T,pulse,ex[5]);
```

```
/* FD1D_1.4.c. */
/* Simulation of a sinusoidal wave hitting a dielectric medium */

float ddx, dt;
float freq_in;

      ddx = .01;                    /* Set the cell size to 1 cm */
      dt = ddx/(2*3e8)              /* Calculate the time step */
.
     /* These parameters specify the input pulse */
        printf( "Input freq (MHz)--> ");
        scanf("%f", &freq_in);
        freq_in = freq_in*1e6;
        printf(" %8.0f   \n", freq_in);

    T = 0;
    nsteps = 1;

    /* Main part of the program */

while ( nsteps > 0 ) {
       printf( "nsteps --> ");
       scanf("%d", &nsteps);
       printf("%d \n", nsteps);

    for ( n=1; n <=nsteps ; n++)
    {
       T = T + 1;

       /* Calculate the Ex field */
       for ( k=0; k < KE; k++ )
       { ex[k] = ex[k] + cb[k]*( hy[k-1] - hy[k] ) ; }

       /* Put a sinusoidal source at cell 5  p0
*/

         pulse = sin(2*pi*freq_in*dt*T);
         ex[5] = ex[5] + pulse;
       printf( "%5.1f %6.2f %6.2f\n",T,pulse,ex[5]);
```

```
/*FD1D_1.5.c.  1D FDTD simulation of a lossy dielectric medium */

    float ca[KE],cb[KE];

      epsz = 8.85419e-12;

    for ( k=1; k <= KE; k++ ) {  /* Initialize to free space */
      ca[k] = 1.;
      cb[k] = .5;
    }
       printf( "Dielectric starts at --> ");
       scanf("%d", &kstart);
       printf( "Epsilon --> ");
       scanf("%f", &epsilon);
          printf( "Conductivity --> ");
          scanf("%f", &sigma);
       printf("%d %6.2f  %6.2f  \n", kstart,epsilon, sigma);

       eaf = dt*sigma/(2*epsz*epsilon);
        printf(" %6.4f  \n", eaf);
      for ( k=kstart; k <= KE; k++ ) {
        ca[k] = (1. - eaf)/(1 + eaf) ;
        cb[k] = .5/(epsilon*(1 + eaf) );
    }
    /* Main part of the program */

       /* Calculate the Ex field */
       for ( k=0; k < KE; k++ )
          {  ex[k] = ca[k]*ex[k] + cb[k]*( hy[k-1] - hy[k] ) ; }
```