CHAPTER 10

Thin-wire modeling

So far we have assumed that all of the objects in the finite-difference time-domain (FDTD) problem space conform to the FDTD grid. Some subcell modeling techniques are developed to model geometries that do not conform to the grid or have dimensions smaller than cell dimensions. One of the most common such geometries is a thin wire that has a radius less than a cell size. In this chapter we discuss one of the subcell modeling techniques that is proposed to model thin wires in FDTD simulations.

There are various techniques proposed for modeling thin wires. However, we discuss the one proposed in [39], which is based on Faraday's law contour-path formulation. This model is easy to understand and to implement in the FDTD program.

10.1 Thin-wire formulation

Figure 10.1 illustrates a thin wire of radius a with its axis coinciding with a field component $E_z(i, j, k)$ on the FDTD grid. In the given example the radius a is smaller than the x and y cell dimensions. There are four magnetic field components circulating around $E_z(i, j, k)$: namely, $H_y(i, j, k)$, $H_x(i, j, k)$, $H_y(i-1, j, k)$, and $H_x(i, j-1, k)$ as shown in Figure 10.2. The thin wire model proposes special updating equations for these magnetic field components. We now demonstrate the derivation of the updating equation for the component $H_y(i, j, k)$; the derivation of updating equations for other field components follows the same procedure.

Figure 10.1 shows the magnetic field component $H_y(i, j, k)$ and four electric field components circulating around $H_y(i, j, k)$. Faraday's law given in integral form as

$$-\mu\int_S \frac{\partial\vec{H}}{\partial t}\ .\ d\vec{s} = \oint_L \vec{E}\cdot d\vec{l} \tag{10.1}$$

can be applied on the enclosed surface shown in Figure 10.1 to establish the relation between $H_y(i, j, k)$ and the electric field components located on the boundaries of the enclosed surface. Before utilizing (10.1) it should be noted that the variation of the fields around the thin wire is assumed to be a function of $1/r$, where r represents the distance to the field position from the thin-wire axis [39]. In more explicit form H_y can be expressed as

$$H_y(r) = \frac{H_{y0}}{r}, \tag{10.2}$$

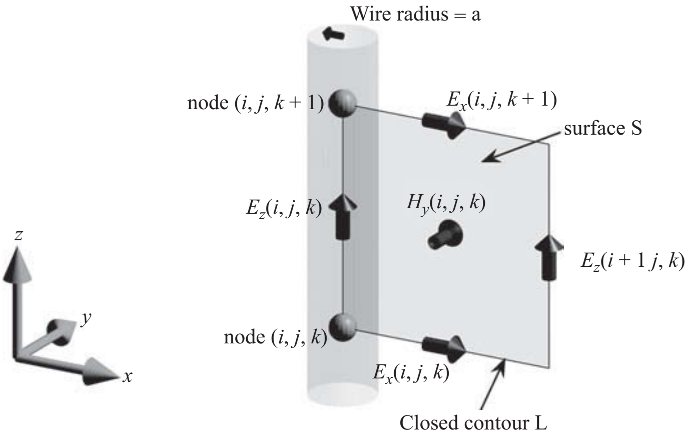

Figure 10.1 A thin wire with its axis coinciding with $E_z(i, j, k)$ and field components surrounding $H_y(i, j, k)$.

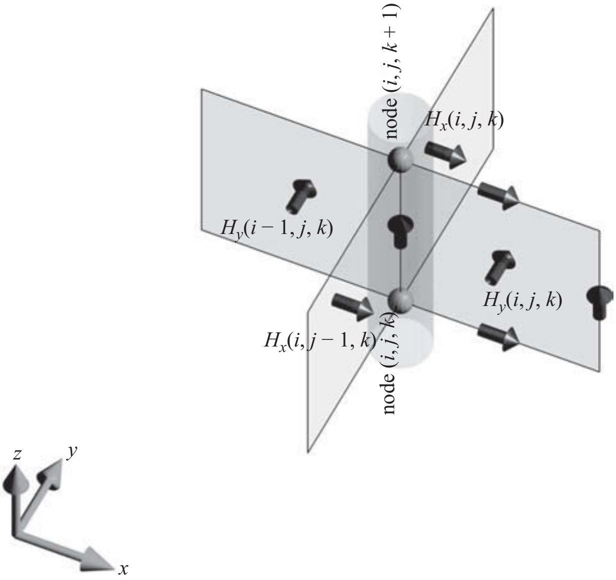

Figure 10.2 Magnetic field components surrounding the thin wire.

and, similarly, E_x can be expressed as

$$E_x(r) = \frac{E_{x0}}{r}, \tag{10.3}$$

where H_{y0} and E_{x0} are constants. In Figure 10.1 one can observe that the field components $H_y(i, j, k)$, $E_x(i, j, k)$, and $E_x(i, j, k+1)$ are located at $r = \Delta x/2$. For instance,

$$H_y\left(\frac{\Delta x}{2}\right) = \frac{2H_{y0}}{\Delta x} = H_y(i, j, k). \tag{10.4}$$

Then

$$H_{y0} = \frac{H_y(i, j, k)\Delta x}{2}, \tag{10.5}$$

hence,

$$H_y(r) = \frac{H_y(i, j, k)\Delta x}{2r}. \tag{10.6}$$

Similarly,

$$E_x(r)\big|_{j,k} = \frac{E_x(i, j, k)\Delta x}{2r}, \tag{10.7}$$

and

$$E_x(r)\big|_{j,k+1} = \frac{E_x(i, j, k+1)\Delta x}{2r}. \tag{10.8}$$

Using (10.6), (10.7), and (10.8) in (10.1) one can obtain

$$\begin{aligned} -\mu \int_{z=0}^{z=\Delta z} \int_{r=a}^{r=\Delta x} \frac{\partial}{\partial t} \frac{H_y(i, j, k)\Delta x}{2r} drdz \\ = \int_{z=0}^{z=\Delta z} E_z(i, j, k)dz + \int_{z=\Delta z}^{z=0} E_z(i+1, j, k)dz \\ + \int_{r=a}^{r=\Delta x} \frac{E_x(i, j, k+1)\Delta x}{2r} dr + \int_{r=\Delta x}^{r=a} \frac{E_x(i, j, k)\Delta x}{2r} dr. \end{aligned} \tag{10.9}$$

Notice that the electric field should vanish inside the wire; therefore, the integration limits are from *a* to Δ*x*. Due to the same reasoning, $E_z(i, j, k)$ is zero as well. Then evaluation of (10.9) yields

$$\begin{aligned} \frac{-\mu\Delta z\Delta x}{2} \ln\left(\frac{\Delta x}{a}\right) \frac{\partial H_y(i, j, k)}{\partial t} = -\Delta z E_z(i+1, j, k) \\ + \ln\left(\frac{\Delta x}{a}\right) \frac{\Delta x}{2} E_x(i, j, k+1) - ln\left(\frac{\Delta x}{a}\right) \frac{\Delta x}{2} E_x(i, j, k). \end{aligned} \tag{10.10}$$

After applying the central difference approximation to the time derivative of magnetic field component and rearranging the terms, one can obtain the new value of $H_y(i, j, k)$ in terms of other components as

$$\begin{aligned} H_y^{n+1/2}(i,j,k) = H_y^{n-1/2}(i,j,k) + \frac{2\Delta t}{\mu\Delta x \ln\left(\frac{\Delta x}{a}\right)} E_z^n(i+1,j,k) \\ - \frac{\Delta t}{\mu\Delta z}\left(E_x^n(i,j,k+1) - E_x^n(i,j,k)\right). \end{aligned} \tag{10.11}$$

Equation (10.12) can be written in the same form as the general updating equation for H_y (1.30), such that

$$\begin{aligned} H_y^{n+\frac{1}{2}}(i,j,k) &= C_{hyh}(i,j,k) \times H_y^{n-\frac{1}{2}}(i,j,k) \\ &+ C_{hyez}(i,j,k) \times \left(E_z^n(i+1,j,k) - E_z^n(i,j,k)\right) \\ &+ C_{hyex}(i,j,k) \times \left(E_x^n(i,j,k+1) - E_x^n(i,j,k)\right), \end{aligned} \tag{10.12}$$

where

$$C_{hyh}(i,j,k) = 1, C_{hyez}(i,j,k) = \frac{2\Delta t}{\mu_y(i,j,k)\Delta x ln\left(\frac{\Delta x}{a}\right)},$$

$$C_{hyex}(i,j,k) = -\frac{\Delta t}{\mu_y(i,j,k)\Delta z}.$$

Then one only need to modify the updating coefficients before the FDTD time-marching loop according to (10.12) to have $H_y^{n+\frac{1}{2}}(i, j, k)$ updated due to a thin wire between nodes (i, j, k) and $(i, j, k+1)$. Furthermore, we have enforced $E_z(i, j, k)$ to be zero. This can be accomplished similarly by setting the updating coefficients for $E_z^n(i, j, k)$ in (1.28) as zero. That means $C_{eze}(i, j, k)$, $C_{ezby}(i, j, k)$, and $C_{ezby}(i, j, k)$ should be assigned zeros before the FDTD time-marching procedure begins.

As mentioned before, there are four magnetic field components circulating around $E_z^n(i, j, k)$ as illustrated in Figure 10.2; therefore, all these magnetic field components need to be updated based on thin-wire modeling. The updating equation for $H_y(i, j, k)$ is given in (10.11). Similarly, updating equations can be obtained for the other three components as

$$\begin{aligned} H_y^{n+\frac{1}{2}}(i-1,j,k) &= C_{hyh}(i-1,j,k) \times H_y^{n-\frac{1}{2}}(i-1,j,k) \\ &+ C_{hyez}(i-1,j,k) \times \left(E_z^n(i,j,k) - E_z^n(i-1,j,k)\right) \\ &+ C_{hyex}(i-1,j,k) \times \left(E_x^n(i-1,j,k+1) - E_x^n(i-1,j,k)\right), \end{aligned} \tag{10.13}$$

where

$$C_{hyh}(i-1,j,k) = 1, C_{hyez}(i-1,j,k) = \frac{2\Delta t}{\mu_y(i-1,j,k)\Delta x ln\left(\frac{\Delta x}{a}\right)},$$

$$C_{hyex}(i-1,j,k) = -\frac{\Delta t}{\mu_y(i-1,j,k)\Delta z}.$$

$$\begin{aligned} H_x^{n+\frac{1}{2}}(i,j,k) &= C_{hxh}(i,j,k) \times H_x^{n-\frac{1}{2}}(i,j,k) \\ &+ C_{hxey}(i,j,k) \times \left(E_y^n(i,j,k+1) - E_y^n(i,j,k)\right) \\ &+ C_{hxez}(i,j,k) \times \left(E_z^n(i,j+1,k) - E_z^n(i,j,k)\right), \end{aligned} \tag{10.14}$$

where

$$C_{hxh}(i,j,k) = 1, \quad C_{hyey}(i,j,k) = \frac{\Delta t}{\mu_x(i,j,k)\Delta z},$$

$$C_{hxez}(i,j,k) = -\frac{2\Delta t}{\mu_x(i,j,k)\Delta y \ln\left(\frac{\Delta y}{a}\right)}.$$

and

$$\begin{aligned} H_x^{n+\frac{1}{2}}(i,j-1,k) &= C_{hxh}(i,j-1,k) \times H_x^{n-\frac{1}{2}}(i,j-1,k) \\ &+ C_{hxey}(i,j-1,k) \times \left(E_y^n(i,j-1,k+1) - E_y^n(i,j-1,k)\right) \\ &+ C_{hxez}(i,j-1,k) \times \left(E_z^n(i,j,k) - E_z^n(i,j-1,k)\right), \end{aligned} \tag{10.15}$$

where

$$C_{hxh}(i,j-1,k) = 1, \quad C_{hxey}(i,j-1,k) = \frac{\Delta t}{\mu_x(i,j-1,k)\Delta z},$$
$$C_{hxez}(i,j-1,k) = -\frac{2\Delta t}{\mu_x(i,j-1,k)\Delta y \ln\left(\frac{\Delta y}{a}\right)}.$$

10.2 MATLAB® implementation of the thin-wire formulation

Having derived the updating equations for modeling thin wires, these equations can be implemented in the FDTD program. The first step is the definition of the thin-wire parameters. The thin wires are defined as a part of the geometry, and their definition implementation is provided in the subroutine ***define_geometry***, as shown in Listing 10.1.

Listing 10.1 define_geometry.m

```
disp('defining the problem geometry');

bricks  = [];
spheres = [];
thin_wires = [];

% define a thin wire
thin_wires(1).min_x = 0;
thin_wires(1).min_y = 0;
thin_wires(1).min_z = 1e-3;
thin_wires(1).max_x = 0;
thin_wires(1).max_y = 0;
thin_wires(1).max_z = 10e-3;
thin_wires(1).radius = 0.25e-3;
thin_wires(1).direction = 'z';

% define a thin wire
thin_wires(2).min_x = 0;
thin_wires(2).min_y = 0;
thin_wires(2).min_z = -10e-3;
thin_wires(2).max_x = 0;
thin_wires(2).max_y = 0;
thin_wires(2).max_z = -1e-3;
thin_wires(2).radius = 0.25e-3;
thin_wires(2).direction = 'z';
```

A new parameter called **thin_wires** is defined and initialized as an empty structure array. This parameter is used to hold the respective thin-wire parameter values. It is assumed that thin wires are aligned parallel to either the *x*, *y*, or *z* axis; therefore, the **direction** field of **thin_wires** is set accordingly. Then other parameters **min_x**, **min_y**, **min_z**, **max_x**, **max_y**, and **max_z** are assigned values to indicate the position of the thin wire in three-dimensional space. Since a thin wire is like a one-dimensional object, its length is determined by the given position parameters, while its thickness is determined by its radius. Therefore, **radius** is another field of **thin_wires** that holds the value of the radius. Although there are six position parameters, two of these are actually redundant in the current implementation. For instance, if the thin wire is aligned in the *x* direction, the parameters **min_x**, **min_y**, **min_z**, and **max_x** are sufficient to describe the location of the thin wire. The other two parameters **max_y** and **max_z** are assigned the same values as **min_y** and **min_z** as shown in Listing 10.1.

After the definition process, the initialization process is performed. Since the thin wire is a new type of geometry that would determine the size of the problem space, necessary codes must be implemented in the subroutine ***calculate_domain_size***. The code for this task is shown in Listing 10.2.

The next step in the thin-wire implementation is the initialization of the thin-wire parameters through the updating coefficients. A new subroutine named ***initialize_thin_wire_updating_coefficients*** is implemented for initialization of the thin-wire updating coefficients. This subroutine is named ***initialize_updating_coefficients*** and is called after the initialization subroutines for the updating coefficients of other types of objects are called. The implementation of ***initialize_thin_wire_updating_coefficients*** is shown in Listing 10.3. In this subroutine, the electric and magnetic field coefficients associated with thin wires are updated based on the equations derived in Section 10.1.

As the updating coefficients associated with the thin wires are updated appropriately, the implementation of the thin-wire formulation is completed. The thin-wire implementation only requires an extra preprocessing step. Then during the FDTD time-marching loop the updating coefficients will manipulate the fields for modeling the thin-wire behavior.

Listing 10.2 calculate_domain_size.m

```
disp('calculating the number of cells in the problem space');

number_of_spheres = size(spheres,2);
number_of_bricks  = size(bricks,2);
number_of_thin_wires  = size(thin_wires,2);
for i=1:number_of_thin_wires
    min_x(number_of_objects) = thin_wires(i).min_x;
    min_y(number_of_objects) = thin_wires(i).min_y;
    min_z(number_of_objects) = thin_wires(i).min_z;
    max_x(number_of_objects) = thin_wires(i).max_x;
    max_y(number_of_objects) = thin_wires(i).max_y;
    max_z(number_of_objects) = thin_wires(i).max_z;
    number_of_objects = number_of_objects + 1;
end
```

Listing 10.3 initialize_thin_wire_updating_coefficients.m

```
disp('initializing␣thin␣wire␣updating␣coefficients');

dtm = dt/mu_0;

for ind = 1:number_of_thin_wires
    is = round((thin_wires(ind).min_x - fdtd_domain.min_x)/dx)+1;
    js = round((thin_wires(ind).min_y - fdtd_domain.min_y)/dy)+1;
    ks = round((thin_wires(ind).min_z - fdtd_domain.min_z)/dz)+1;
    ie = round((thin_wires(ind).max_x - fdtd_domain.min_x)/dx)+1;
    je = round((thin_wires(ind).max_y - fdtd_domain.min_y)/dy)+1;
    ke = round((thin_wires(ind).max_z - fdtd_domain.min_z)/dz)+1;
    r_o = thin_wires(ind).radius;

    switch (thin_wires(ind).direction(1))
        case 'x'
            Cexe (is:ie-1,js,ks) = 0;
            Cexhy(is:ie-1,js,ks) = 0;
            Cexhz(is:ie-1,js,ks) = 0;
            Chyh (is:ie-1,js,ks-1:ks) = 1;
            Chyez(is:ie-1,js,ks-1:ks) = dtm ...
                ./ (mu_r_y(is:ie-1,js,ks-1:ks) * dx);
            Chyex(is:ie-1,js,ks-1:ks) = -2 * dtm ...
                ./ (mu_r_y(is:ie-1,js,ks-1:ks) * dz * log(dz/r_o));
            Chzh( is:ie-1,js-1:js,ks) = 1;
            Chzex(is:ie-1,js-1:js,ks) = 2 * dtm ...
                ./ (mu_r_z(is:ie-1,js-1:js,ks) * dy * log(dy/r_o));
            Chzey(is:ie-1,js-1:js,ks) = -dtm ...
                ./ (mu_r_z(is:ie-1,js-1:js,ks) * dx);
        case 'y'
            Ceye (is,js:je-1,ks) = 0;
            Ceyhx(is,js:je-1,ks) = 0;
            Ceyhz(is,js:je-1,ks) = 0;
            Chzh (is-1:is,js:je-1,ks) = 1;
            Chzex(is-1:is,js:je-1,ks) = dtm ...
                ./ (mu_r_z(is-1:is,js:je-1,ks) * dy);
            Chzey(is-1:is,js:je-1,ks) = -2 * dtm ...
                ./ (mu_r_z(is-1:is,js:je-1,ks) * dx * log(dx/r_o));
            Chxh (is,js:je-1,ks-1:ks) = 1;
            Chxey(is,js:je-1,ks-1:ks) = 2 * dtm ...
                ./ (mu_r_x(is,js:je-1,ks-1:ks) * dz * log(dz/r_o));
            Chxez(is,js:je-1,ks-1:ks) = -dtm ...
                ./ (mu_r_x(is,js:je-1,ks-1:ks) * dy);

        case 'z'
            Ceze (is,js,ks:ke-1) = 0;
            Cezhx(is,js,ks:ke-1) = 0;
            Cezhy(is,js,ks:ke-1) = 0;
            Chxh (is,js-1:js,ks:ke-1) = 1;
            Chxey(is,js-1:js,ks:ke-1) = dtm ...
                ./ (mu_r_x(is,js-1:js,ks:ke-1) * dz);
            Chxez(is,js-1:js,ks:ke-1) = -2 * dtm ...
                ./ (mu_r_x(is,js-1:js,ks:ke-1) * dy * log(dy/r_o));
            Chyh (is-1:is,js,ks:ke-1) = 1;
            Chyez(is-1:is,js,ks:ke-1) = 2 * dtm ...
                ./ (mu_r_y(is-1:is,js,ks:ke-1) * dx * log(dx/r_o));
            Chyex(is-1:is,js,ks:ke-1) = -dtm ...
                ./ (mu_r_y(is-1:is,js,ks:ke-1) * dz);
    end
end
```

10.3 Simulation examples

10.3.1 Thin-wire dipole antenna

Simulation of a thin-wire dipole antenna is presented in this section. The problem space is composed of cells with $\Delta x = 0.25$ mm, $\Delta y = 0.25$ mm, and $\Delta z = 0.25$ mm. The boundaries are CPML with 8 cells thickness and a 10 cells air gap on all sides. The dipole is composed of two thin wires having 0.05 mm radius and 9.75 mm length. There is a 0.5 mm gap between the wires where a voltage source is placed. Figure 10.3 illustrates the problem geometry. The definition of the geometry is illustrated in Listing 10.4. The definition of the voltage source is given in Listing 10.5, and the definitions of the output parameters are given in Listing 10.6. One can notice in Listing 10.6 that a far-field frequency is defined as 7 GHz to obtain the far-field radiation patterns at this frequency.

The simulation of this problem is performed for 4,000 time steps. The same problem is simulated using WIPL-D [40], which is a three-dimensional EM simulation software. Figure 10.4 shows the S_{11} of the thin-wire antenna calculated by FDTD and WIPL-D. The magnitude and phase curves show a good agreement. The input impedance of a dipole antenna is very much affected by the thickness of the dipole wires. The input impedances obtained from the two simulations are compared in Figure 10.5, and they show very good agreement over the simulated frequency band from zero to 20 GHz.

Since this is a radiation problem the far-field patterns are also calculated, as indicated before, at 7 GHz. The directivity patterns are plotted in Figures 10.6, 10.7, and 10.8 for *xy*, *xz*, and *yz* planes, respectively.

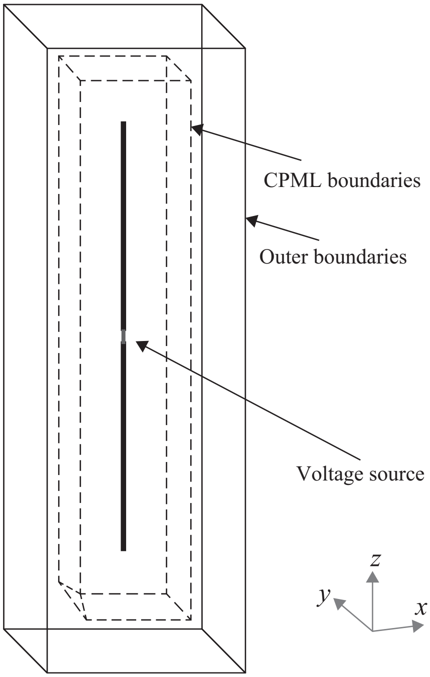

Figure 10.3 A thin-wire dipole antenna.

Listing 10.4 define_geometry.m

```
disp('defining␣the␣problem␣geometry');

bricks  = [];
spheres = [];
thin_wires = [];

% define a thin wire
thin_wires(1).min_x = 0;
thin_wires(1).min_y = 0;
thin_wires(1).min_z = 0.25e-3;

thin_wires(1).max_x = 0;
thin_wires(1).max_y = 0;
thin_wires(1).max_z = 10e-3;
thin_wires(1).radius = 0.05e-3;
thin_wires(1).direction = 'z';

% define a thin wire
thin_wires(2).min_x = 0;
thin_wires(2).min_y = 0;
thin_wires(2).min_z = -10e-3;
thin_wires(2).max_x = 0;
thin_wires(2).max_y = 0;
thin_wires(2).max_z = -0.25e-3;
thin_wires(2).radius = 0.05e-3;
thin_wires(2).direction = 'z';
```

Listing 10.5 define_secources_and_lumped_elements.m

```
disp('defining␣sources␣and␣lumped␣element␣components');

voltage_sources = [];
current_sources = [];
diodes = [];
resistors = [];
inductors = [];
capacitors = [];

% define source waveform types and parameters
waveforms.gaussian(1).number_of_cells_per_wavelength = 0;
waveforms.gaussian(2).number_of_cells_per_wavelength = 15;

% voltage sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
% resistance : ohms, magitude   : volts
voltage_sources(1).min_x = 0;
voltage_sources(1).min_y = 0;
voltage_sources(1).min_z = -0.25e-3;
voltage_sources(1).max_x = 0;
voltage_sources(1).max_y = 0;
voltage_sources(1).max_z = 0.25e-3;
voltage_sources(1).direction = 'zp';
voltage_sources(1).resistance = 50;
voltage_sources(1).magnitude = 1;
voltage_sources(1).waveform_type = 'gaussian';
voltage_sources(1).waveform_index = 1;
```

Listing 10.6 define_output_parameters.m

```
disp('defining output parameters');

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_voltages = [];
sampled_currents = [];
ports = [];
farfield.frequencies = [];

% figure refresh rate
plotting_step = 10;

% mode of operation
run_simulation = true;
show_material_mesh = true;
show_problem_space = true;

% far field calculation parameters
farfield.frequencies(1) = 7.0e9;
farfield.number_of_cells_from_outer_boundary = 13;

% frequency domain parameters
frequency_domain.start = 20e6;
frequency_domain.end   = 20e9;
frequency_domain.step  = 20e6;

% define sampled voltages
sampled_voltages(1).min_x = 0;
sampled_voltages(1).min_y = 0;
sampled_voltages(1).min_z = -0.25e-3;
sampled_voltages(1).max_x = 0;
sampled_voltages(1).max_y = 0;
sampled_voltages(1).max_z = 0.25e-3;
sampled_voltages(1).direction = 'zp';
sampled_voltages(1).display_plot = false;

% define sampled currents
sampled_currents(1).min_x = 0;
sampled_currents(1).min_y = 0;
sampled_currents(1).min_z = 0;
sampled_currents(1).max_x = 0;
sampled_currents(1).max_y = 0;
sampled_currents(1).max_z = 0;
sampled_currents(1).direction = 'zp';
sampled_currents(1).display_plot = false;

% define ports
ports(1).sampled_voltage_index = 1;
ports(1).sampled_current_index = 1;
ports(1).impedance = 50;
ports(1).is_source_port = true;
```

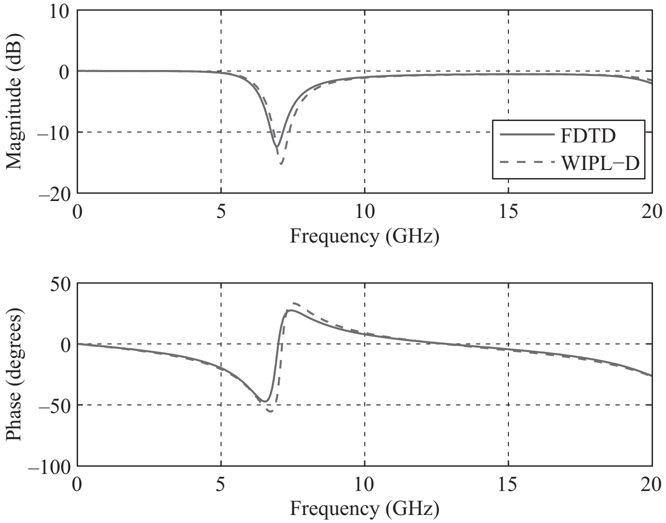

Figure 10.4 S_{11} of the thin-wire dipole antenna.

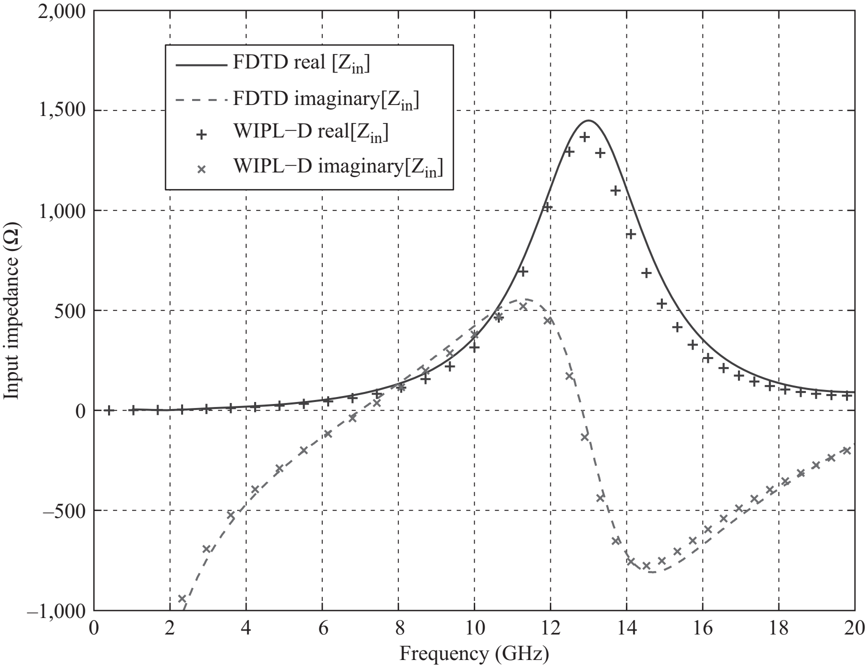

Figure 10.5 Input impedance of the thin-wire dipole antenna.

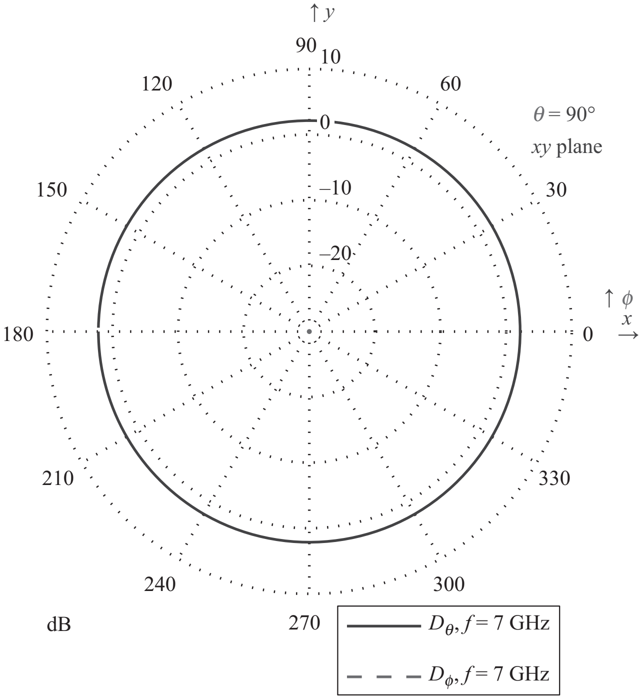

Figure 10.6 Radiation pattern in the *xy* plane cut.

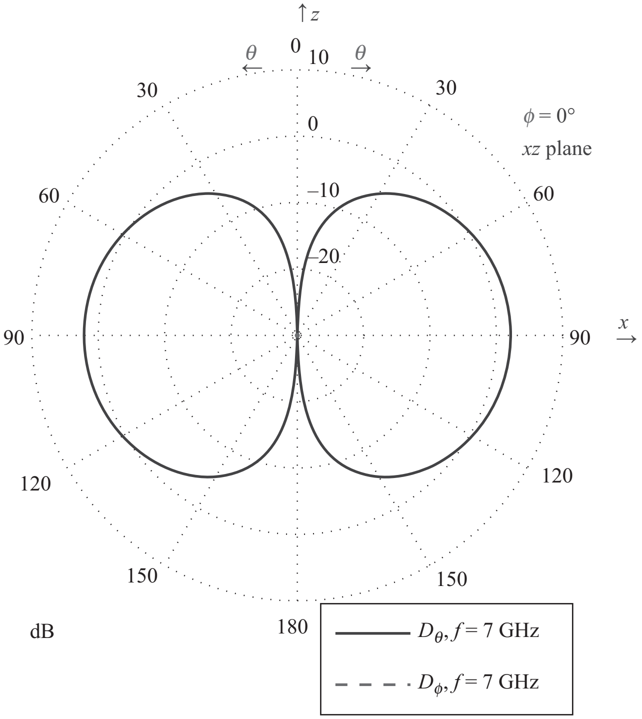

Figure 10.7 Radiation pattern in the *xz* plane cut.

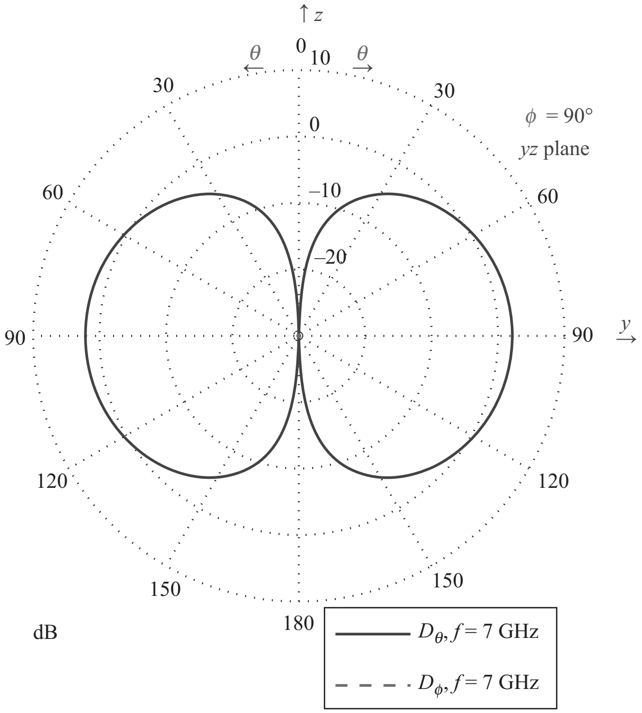

Figure 10.8 Radiation pattern in the *yz* plane cut.

10.4 An improved thin-wire model

In this section, we will discuss an improvement on the thin-wire model of [39], discussed in the previous sections, as presented in [41]. As in [39], the formulation that updates the magnetic field components circulating around a thin-wire is modified to account for the thin-wire in [41], albeit with a correction. In this section the formulation by [39] is referred to as the original thin-wire formulation while the one by [41] is referred to as the improved thin-wire formulation.

In Section 10.1, the field variation around a thin-wire is assumed to be a function of $1/r$ as in (10.2) and (10.3), which follows a static field distribution around a perfect electric conductor wire. The magnetic field component $H_y(i,j,k)$ in Figure 10.1, as calculated using (10.12), is the point-wise value of the field at the center of the surface S. It is also a constant value of magnetic field on a circular arc illustrated in Figure 10.9 due to the $1/r$ dependence assumption. When $H_y(i,j,k)$ is used in the update of an electric field as in (1.26) or (1.28), it is assumed to be an average of H_y along the edge of a rectangular path surrounding the electric field components, as illustrated in Figure 10.10 for E_z. This is due to the fact that Ampere's law is being imposed on a Cartesian grid in FDTD. Since magnetic field is constant on the circular loop, thus variable on the Cartesian cell edge due to the $1/r$ dependence, $H_y(i,j,k)$ is not an accurate average value of the field along the edge. As a solution to this contradiction, the looping magnetic field is projected on the respective cell edge and an average value of H_y that can be used to update the neighboring electric field components is obtained.

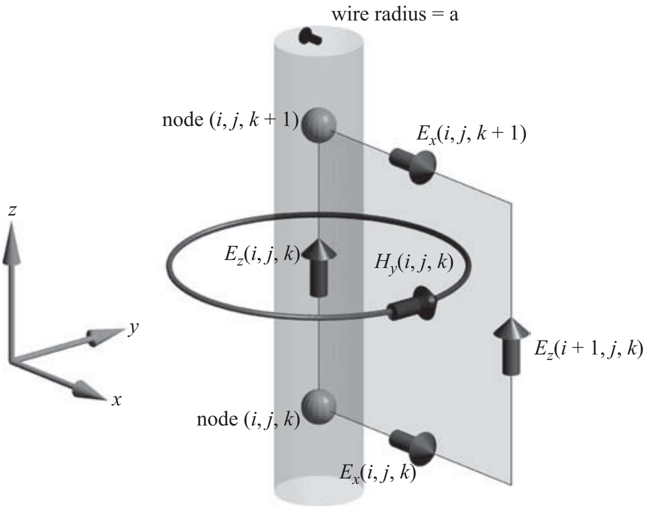

Figure 10.9 Constant magnetic field on a loop circulating around a thin wire.

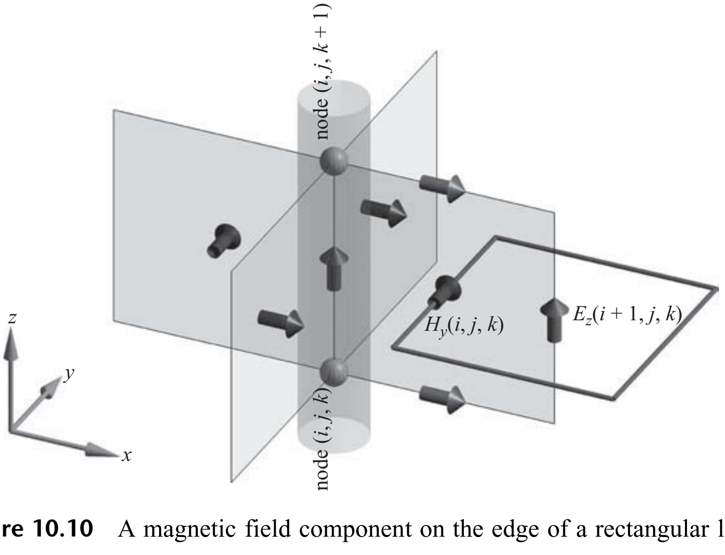

Figure 10.10 A magnetic field component on the edge of a rectangular loop.

Consider the circular contour on which $H_y(i,j,k)$ is constant and the cross-section of a cell where $\Delta x \neq \Delta y$ in Figure 10.11(a). Figure 10.11(b) shows the arc between the angles 0 and ϕ' radians zoomed in. Notice that the constant value of $H_y(i,j,k)$ is H_ϕ on the arc while the magnetic field on the right edge is a function of y as $H_y(y)$. The distance between $H_y(y)$ and the cell center is $r = \sqrt{(\Delta x/2)^2 + y^2}$. To accurately compute $H_y(y)$, the H_ϕ is to be

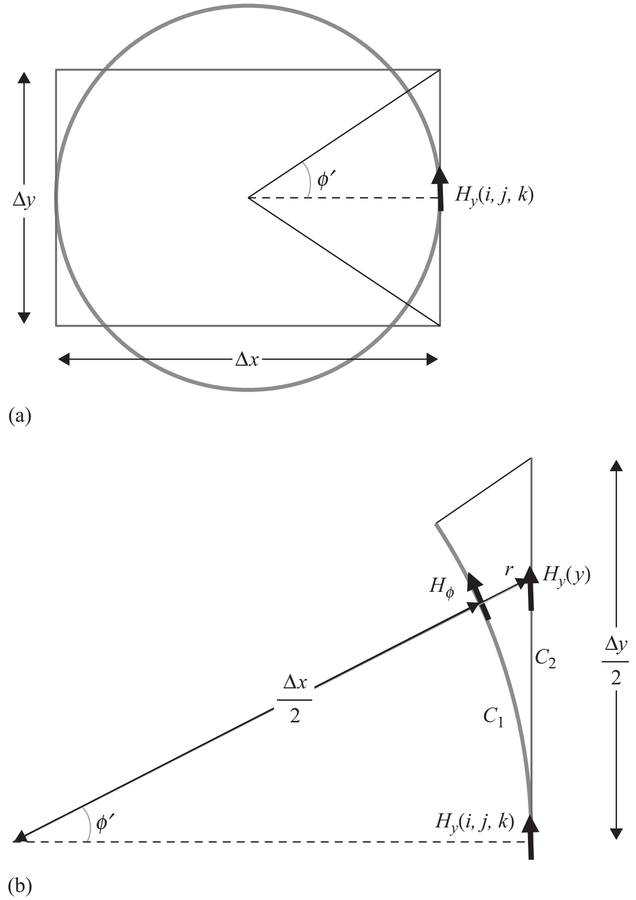

Figure 10.11 Projection of magnetic field from a circular loop to a rectangular edge: (a) large view and (b) details of the right edge.

scaled by the factor $\frac{\Delta x}{2r} = \frac{\Delta x/2}{\sqrt{(\Delta x/2)^2+y^2}}$ and then projected on the y axis using $\widehat{\phi} \cdot \hat{y} = \cos\phi$ which then leads to

$$H_y(y) = \cos\phi \frac{\Delta x/2}{\sqrt{(\Delta x/2)^2 + y^2}} H_\phi = \frac{(\Delta x/2)^2}{(\Delta x/2)^2 + y^2} H_\phi. \tag{10.16}$$

To find an average value for $H_y(y)$ on the right edge, we can integrate $H_y(y)$ along the right edge and divide by Δy as

$$H_{y,avg} = \frac{1}{\Delta y} \int_{y=-\Delta y/2}^{\Delta y/2} H_y(y) dy = \frac{H_\phi}{\Delta y} \int_{y=-\Delta y/2}^{\Delta y/2} \frac{(\Delta x/2)^2}{(\Delta x/2)^2 + y^2} dy. \tag{10.17}$$

The solution of the integral (10.17) yields

$$H_{y,avg} = k_{Hy} H_\phi = k_{Hy} H_y(i,j,k), \tag{10.18}$$

where

$$k_{Hy} = \frac{\Delta x}{\Delta y}\tan^{-1}\left(\frac{\Delta y}{\Delta x}\right). \tag{10.19}$$

Notice that k_{Hy} is just a scaling factor. Thus, to scale $H_y(i,j,k)$ by k_{Hy} one can simply scale the coefficients $C_{hyez}(i,j,k)$ and $C_{hyex}(i,j,k)$ in (10.12) by k_{Hy} as

$$C_{hyez}(i,j,k) = k_{Hy}\frac{2\Delta t}{\mu_y(i,j,k)\Delta x \ln\left(\frac{\Delta x}{a}\right)}, \tag{10.20}$$

$$C_{hyex}(i,j,k) = -k_{Hy}\frac{\Delta t}{\mu_y(i,j,k)\Delta z}. \tag{10.21}$$

Note that $H_y(i-1,j,k)$ in Figure 10.2 as well need to be scaled by k_{Hy}, whereas $H_x(i,j,k)$ and $H_x(i,j-1,k)$ need to be scaled by a factor of $k_{Hx} = \frac{\Delta y}{\Delta x}\tan^{-1}\left(\frac{\Delta x}{\Delta y}\right)$.

The presented modification to the original thin-wire formulation in terms of the calculation of magnetic field components led to more accurate update values of electric field components adjacent to the thin wire. The expressions of the electric field components updates remain unchanged as the standard FDTD updating equations in the original thin-wire formulation.

In the standard FDTD updating equation for an electric field component it is assumed that the field component is an average value of the field distribution on the face of a cell face through which it flows. For instance, the radial electric field component $E_x(i,j,k)$, illustrated in Figure 10.12, is an average value of E_x on the surface S1. However, when developing the thin-wire updating equation for $H_y(i,j,k)$, the radial component of electric field as well is assumed to vary with $1/r$. Thus, when using $E_x(i,j,k)$ and $E_x(i,j,k+1)$ in (10.12), their $1/r$ dependent values at half-cell distance from the thin-wire axis need to be used instead of their

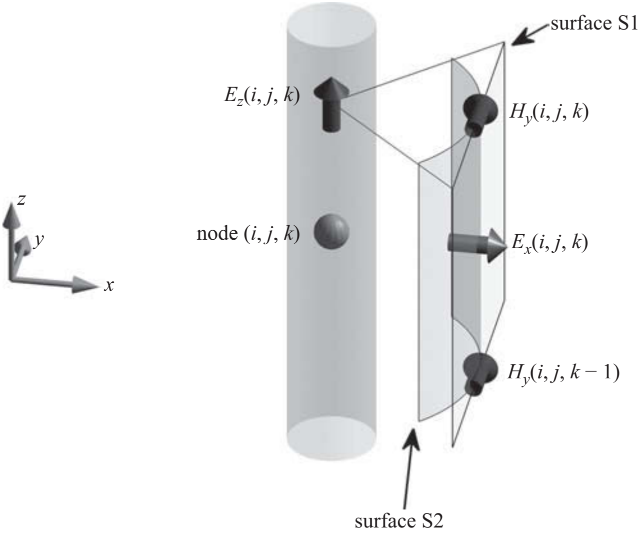

Figure 10.12 Electric field component E_x at the center of a cell face.

average values on respective cell faces. The radial electric field component E_r is constant on the cylindrical surface S2 in Figure 10.12, thus its projection on the surface S1 is a function of y. Note that $E_x(i,j,k)$ is the average of the projection of E_r on surface S1, while the value of E_r at the position of $E_x(i,j,k)$ is needed in (10.12). To obtain E_r one needs to project $E_x(i,j,k)$ from surface S1 to surface S2. Since the fields are constant in the z-direction, the required projection for $E_x(i,j,k)$ is the reverse of the projection done with $H_y(i,j,k)$ illustrated in Figure 10.11. Thus, utilizing (10.19), a scaling factor can be defined as

$$k_{Ex} = \left(\frac{\Delta x}{\Delta y}\tan^{-1}\left(\frac{\Delta y}{\Delta x}\right)\right)^{-1}, \tag{10.22}$$

which can be used to scale $E_x(i,j,k)$ and $E_x(i,j,k+1)$ before they are used to update $H_y(i,j,k)$. This scaling can be performed simply by multiplying the coefficient $C_{hyex}(i,j,k)$ given in (10.21) by k_{Ex}. Since k_{Hy} and k_{Ex} are inverse, scaling $C_{hyex}(i,j,k)$ by k_{Ex} returns it to its original value before it was scaled by k_{Hy} as

$$C_{hyex}(i,j,k) = -\frac{\Delta t}{\mu_y(i,j,k)\Delta z}. \tag{10.23}$$

Thus, the only modification needed for the improved thin-wire formulation over the original thin-wire formulation in updating $H_y(i,j,k)$ is an adjustment to the updating coefficient as in (10.20). Similar adjustments are needed for the respective updating coefficients that update $H_y(i-1,j,k)$, $H_x(i,j,k)$, and $H_x(i,j-1,k)$.

It should be noted that the presented improvement of thin-wire formulation is not the only contribution presented in [41]. Also presented in [41] is an end-cap model that includes the effect of charge accumulation at wire end caps which further improves the accuracy of modeling unconnected thin wires. Moreover, Mâkinen et al. employed the precepts of the improved thin-wire formulation to develop a stabilized resistive voltage source (RVS) model in [42] to feed thin wires that further improves the stability and accuracy of thin-wire simulations.

10.5 MATLAB® implementation of the improved thin-wire formulation

The only modification needed for the improved thin-wire formulation is the scaling of the coefficients used to update the magnetic field components circulating around the thin wire due to the electric field components parallel to the thin-wire axis. Listing 10.3 shows the initialization of thin wire updating coefficients based on the original thin-wire formulation as in (10.20). This code is modified to accommodate the scaling factors introduced by the improved thin-wire formulation as shown in Listing 10.7.

10.6 Simulation example

Simulation of a thin-wire dipole antenna is presented in Section 10.3.1. The same example is repeated here with the simulation performed using the improved thin-wire formulation. The

Listing 10.7 initialize_thin_wire_updating_coefficients.m

```
disp('initializing thin wire updating coefficients');

dtm = dt/mu_0;

for ind = 1:number_of_thin_wires
    is   = round((thin_wires(ind).min_x - fdtd_domain.min_x)/dx)+1;
    js   = round((thin_wires(ind).min_y - fdtd_domain.min_y)/dy)+1;
    ks   = round((thin_wires(ind).min_z - fdtd_domain.min_z)/dz)+1;
    ie   = round((thin_wires(ind).max_x - fdtd_domain.min_x)/dx)+1;
    je   = round((thin_wires(ind).max_y - fdtd_domain.min_y)/dy)+1;
    ke   = round((thin_wires(ind).max_z - fdtd_domain.min_z)/dz)+1;
    r_o = thin_wires(ind).radius;

   switch (thin_wires(ind).direction(1))
      case 'x'
         khy = (dz/dy)*atan(dy/dz);
         khz = (dy/dz)*atan(dz/dy);
         Cexe (is:ie-1,js,ks)  = 0;
         Cexhy(is:ie-1,js,ks)  = 0;
         Cexhz(is:ie-1,js,ks)  = 0;
         Chyh (is:ie-1,js,ks-1:ks) = 1;
         Chyex(is:ie-1,js,ks-1:ks) = −2 * dtm * khy...
            ./ (mu_r_y(is:ie-1,js,ks-1:ks) * dz * log(dz/r_o));
         Chzh( is:ie-1,js-1:js,ks) = 1;
         Chzex(is:ie-1,js-1:js,ks) = 2 * dtm * khz...
            ./ (mu_r_z(is:ie-1,js-1:js,ks) * dy * log(dy/r_o));
      case 'y'
         khz = (dx/dz)*atan(dz/dx);
         khx = (dz/dx)*atan(dx/dz);
         Ceye (is,js:je-1,ks)  = 0;
         Ceyhx(is,js:je-1,ks)  = 0;
         Ceyhz(is,js:je-1,ks)  = 0;
         Chzh (is-1:is,js:je-1,ks) = 1;
         Chzey(is-1:is,js:je-1,ks) = -2 * dtm * khz...
            ./ (mu_r_z(is-1:is,js:je-1,ks) * dx * log(dx/r_o));
         Chxh (is,js:je-1,ks-1:ks) = 1;
         Chxey(is,js:je-1,ks-1:ks) = 2 * dtm *khx...
            ./ (mu_r_x(is,js:je-1,ks-1:ks) * dz * log(dz/r_o));
      case 'z'
         khx = (dy/dx)*atan(dx/dy);
         khy = (dx/dy)*atan(dy/dx);
         Ceze (is,js,ks:ke-1)  = 0;
         Cezhx(is,js,ks:ke-1)  = 0;
         Cezhy(is,js,ks:ke-1)  = 0;
         Chxh (is,js-1:js,ks:ke-1) = 1;
         Chxez(is,js-1:js,ks:ke-1) = −2 * dtm * khx...
            ./ (mu_r_x(is,js-1:js,ks:ke-1) * dy * log(dy/r_o));
         Chyh (is-1:is,js,ks:ke-1) = 1;
         Chyez(is-1:is,js,ks:ke-1) = 2 * dtm * khy...
            ./ (mu_r_y(is-1:is,js,ks:ke-1) * dx * log(dx/r_o));
   end
end
```

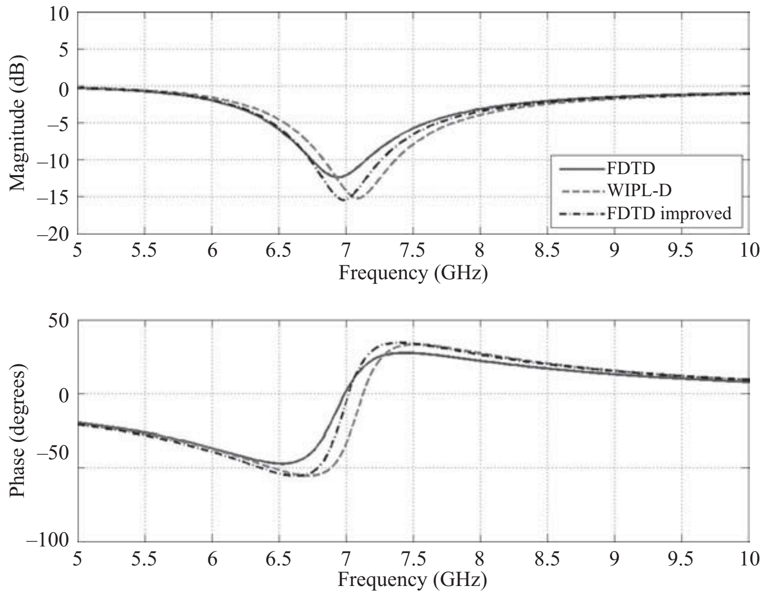

Figure 10.13 Power reflection coefficient of the thin-wire dipole antenna.

calculated power reflection coefficient of the dipole antenna is plotted in Figure 10.13 along with the results previously presented in Figure 10.4. One can notice the improvement of agreement between the FDTD and WIPL-D results when the improved thin-wire formulation is used. Moreover, the input impedance as well calculated and compared with the results in Figure 10.5. The comparison is shown in Figure 10.14.

As mentioned in Section 10.5, accuracy of thin-wire simulations can further be improved by including the end-cap model [41] and the stabilized RVS model [42]. Although these models are not presented here, they are implemented in an FDTD code, although not presented here, and the presented thin-wire dipole antenna is resimulated. Figure 10.15 shows the power reflection coefficient of the thin-wire dipole antenna including the effects of the end-caps and the stabilized RVS model, and illustrates the further improvement in accuracy.

10.7 Exercises

10.1 Construct a problem space composed of cells with size 1 mm on a side. Define a dipole antenna using thin wires similar to the dipole antenna discussed in Section 10.3.1. Set the length of each wire to 10 mm, and leave a 1 mm gap between the wires. Place a voltage source, a sampled voltage, and a sampled current in the gap between the wires, and associate them to a port. Set the radius of the wire as 0.1 mm, run the simulation, and obtain the input reflection coefficient of the antenna. Then increase the radius of the wire by 0.2 mm, repeat the simulation, and observe the change in the input reflection coefficient with incremented increase of wire radius by 0.2 mm until the increased wire radius causes instability in the simulation. Can you determine the maximum wire radius for accurate results in this configuration?

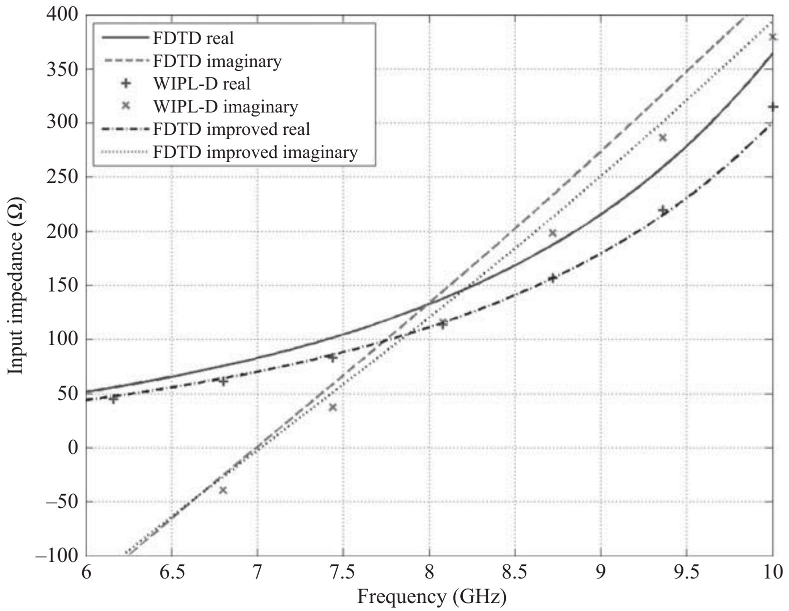

Figure 10.14 Input impedance of the thin-wire dipole antenna.

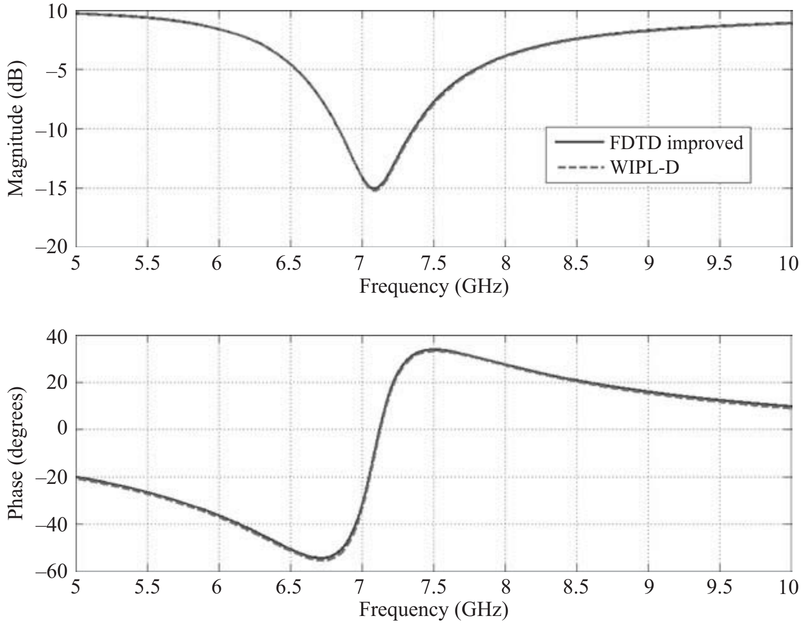

Figure 10.15 Power reflection coefficient of the thin-wire dipole antenna: FDTD implementation includes the end-caps model [41] and the stabilized RVS model [42].

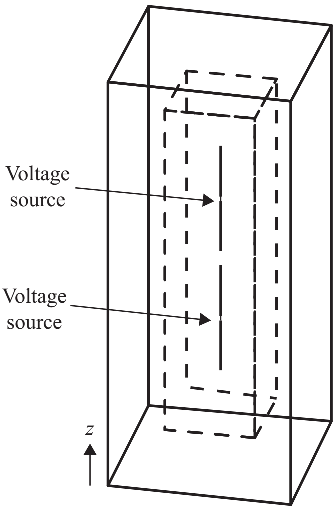

Figure 10.16 An antenna array composed of two thin-wire dipole antennas.

10.2 In this example we construct an antenna array composed of two thin-wire dipole antennas. Consider the dipole antenna that you constructed in Exercise 10.1. Run the dipole simulation using the 0.1 mm wire radius, and record the operation frequency. Rerun the simulation, and obtain the radiation pattern at the frequency of operation. Then define another thin-wire dipole antenna with the same dimensions as the current one, and place it 2 mm above the current one, such that the center to center distance between antennas will be 23 mm. Place a voltage source at the feeding gap of the second antenna that has the same properties of the voltage source of the first antenna. Do not define any ports, and if there are already defined ports, remove them. The geometry of the problem is illustrated in Figure 10.16. Run the simulation, and calculate the radiation patterns at the frequency of operation. Examine the directivity patterns of the single- and two-dipole configurations.

10.3 Consider the dipole antenna array that you constructed in Exercise 10.2. Invert the polarity of one of the voltage sources. For instance, if the direction of the voltage source is "***zp***," set the direction as "***zn***." This will let the antennas be excited by 180° phase difference. Run the simulation, and obtain the radiation pattern at the frequency of operation.

10.4 In this exercise we construct a square loop inductor. Construct a problem space composed of cells with size 1 mm on a side. Place three thin wires, each of which is 10 mm long, such that they will form the three sides of a square loop. Then place two thin wires, each of which is 4 mm long, on the fourth side, leaving a 2 mm gap in between at the center of the fourth side. Place a voltage source, a sampled voltage, and a sampled current in this gap. Set the radius of the wires as 0.1 mm. The geometry of the problem is illustrated in Figure 10.17. Run the simulation, and calculate the input impedance. At low frequencies the imaginary part of the input impedance is positive and linear, which implies that the inductance is constant at low frequencies. Calculate

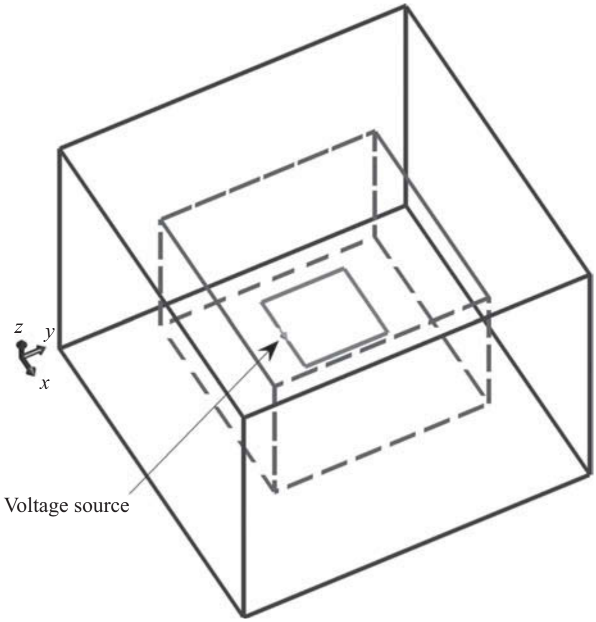

Figure 10.17 A thin-wire loop.

the inductance of the loop. The inductance of such a square loop can be calculated as 30.7 nH using the formula in [43].

10.5 Consider the square loop that you constructed in Exercise 10.4. Now reduce the cell size to 0.5 mm on a side, and reduce the length of the source gap to 1 mm. Run the simulation, and then calculate the inductance of the loop. Examine if the calculated inductance got closer to the expected value.