CHAPTER 6

S-Parameters

Scattering parameters (S-parameters) are used to characterize the response of radiofrequency and microwave circuits, and they are more commonly used than other types of network parameters (e.g., Y-parameters, Z-parameters) because they are easier to measure and work with at high frequencies [12]. In this chapter, we discuss the methods to obtain the S-parameters from a finite-difference time-domain (FDTD) simulation of a single or multiport circuit.

6.1 Scattering parameters

S-parameters are based on the power waves concept. The incident and reflected power waves a_i and b_i associated with port i are defined as

$$a_i = \frac{V_i + Z_i \times I_i}{2\sqrt{|Re\{Z_i\}|}}, \quad b_i = \frac{V_i - Z_i^* \times I_i}{2\sqrt{|Re\{Z_i\}|}}, \tag{6.1}$$

where V_i and I_i are the voltage and the current flowing into the ith port of a junction and Z_i is the impedance looking out from the ith port as illustrated in Figure 6.1 [13]. In general, Z_i is complex; however, in most of the microwaves applications it is real and equal to 50 Ω. Then the S-parameters matrix can be expressed as

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1N} \\ S_{21} & S_{22} & \cdots & S_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{NN} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}. \tag{6.2}$$

By definition, the subscripts mn indicate output port number, m, and input port number, n, of the scattering parameter S_{mn}. If only the port n is excited while all other ports are terminated by matched loads, the output power wave at port m, b_m, and the input power wave at port n, a_n, can be used to calculate S_{mn} using

$$S_{mn} = \frac{b_m}{a_n}. \tag{6.3}$$

This technique can be applied to FDTD simulation results to obtain S-parameters for an input port n. A multiport circuit can be constructed in an FDTD problem space where all ports are terminated by matching loads and only the reference port n is excited by a source.

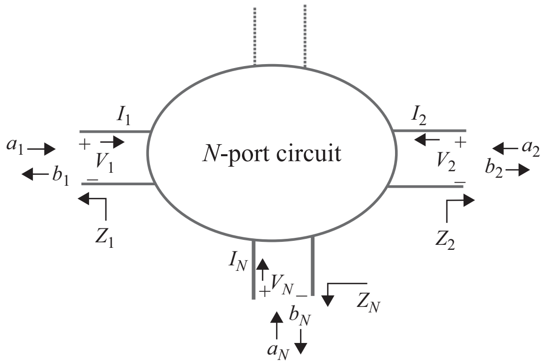

Figure 6.1 An N-port network.

Then sampled voltages and currents can be captured at all ports during the FDTD time-marching loop. S-parameters are *frequency-domain* outputs of a network. After the FDTD iterations are completed, the sampled voltages and currents can be transformed to the frequency domain using the algorithms described in Chapter 5. Then the frequency-domain sampled voltages and currents can be used in (6.1) to obtain incident and reflected power waves, a_i and b_i, from which the S-parameters can be obtained for the reference port using (6.3). One should keep in mind that in an FDTD simulation only one of the ports can be excited. Therefore, to obtain a complete set of S-parameters for all of the port excitations in the circuit, more than one FDTD run may be required depending on the type and symmetry conditions of the problem.

The S-parameters are complex quantities, as they are obtained using (6.3). Generally, S-parameters are plotted by their magnitudes in decibels such that $|S_{mn}|_{\text{dB}} = 20\log_{10}(|S_{mn}|)$ and by their phases.

6.2 S-Parameter calculations

In this section, we illustrate the implementation of S-parameter calculations through an example. S-parameters are associated with ports; therefore, ports are the necessary components for the S-parameter calculations. To define a port we need a sampled voltage and a sampled current defined at the same location and associated with the port. Consider the two-port circuit shown in Figure 6.2, which has been published in [14] as an example for the application of the FDTD method to the analysis of planar microstrip circuits. The problem space is composed of cells having $\Delta x = 0.4064$ mm, $\Delta y = 0.4233$ mm, and $\Delta z = 0.265$ mm. An air gap of 5 cells is left between the circuit and the outer boundary in the *xn*, *xp*, *yn*, and *yp* directions and of 10 cells in the *zp* direction. The outer boundary is perfect electric conductor (PEC) and touches the ground of the circuit in the *zn* direction. The dimensions of the microstrips are shown in Figure 6.2(b) and are implemented in Listing 6.1. The substrate is $3 \times \Delta z$ thick and has a dielectric constant of 2.2. The microstrip filter is terminated by a voltage source with 50 Ω internal resistance on one end and by a 50 Ω resistor on the other end. The voltage source is excited by a Gaussian waveform with 20 cells per wavelength accuracy parameter. Two sampled voltages and two sampled currents are defined 10 cells away from the ends of the microstrip terminations as implemented in Listing 6.2.

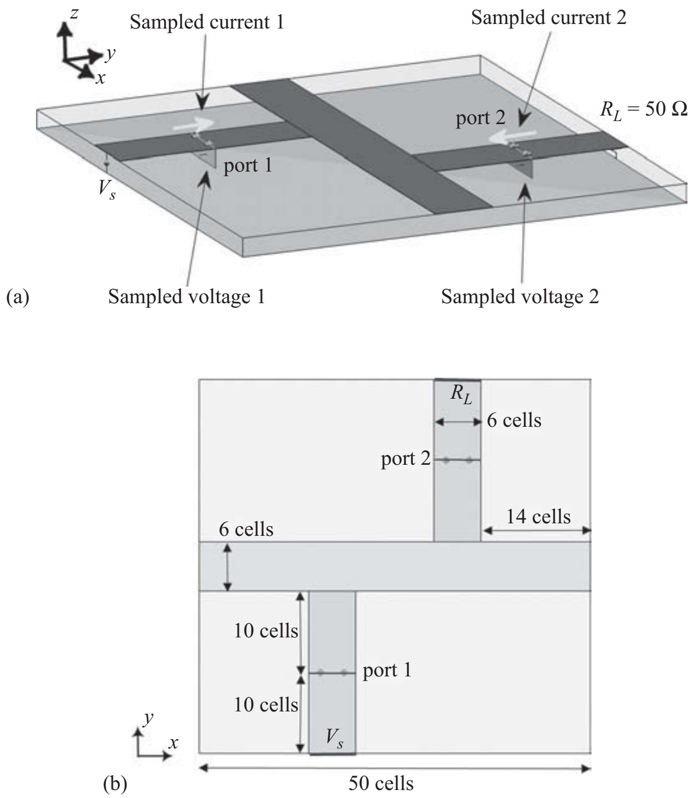

Figure 6.2 A microstrip low-pass filter terminated by a voltage source and a resistor on two ends: (a) three-dimensional view and (b) dimensions.

Listing 6.1 define_geometry.m

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

bricks  = [];
spheres = [];

% define a substrate
bricks(1).min_x = 0;
bricks(1).min_y = 0;
bricks(1).min_z = 0;
bricks(1).max_x = 50*dx;
bricks(1).max_y = 46*dy;
bricks(1).max_z = 3*dz;
bricks(1).material_type = 4;

% define a PEC plate
bricks(2).min_x = 14*dx;
bricks(2).min_y = 0;
bricks(2).min_z = 3*dz;
```

```
bricks(2).max_x = 20*dx;
bricks(2).max_y = 20*dy;
bricks(2).max_z = 3*dz;
bricks(2).material_type = 2;

% define a PEC plate
bricks(3).min_x = 30*dx;
bricks(3).min_y = 26*dy;
bricks(3).min_z = 3*dz;
bricks(3).max_x = 36*dx;
bricks(3).max_y = 46*dy;
bricks(3).max_z = 3*dz;
bricks(3).material_type = 2;

% define a PEC plate
bricks(4).min_x = 0;
bricks(4).min_y = 20*dy;
bricks(4).min_z = 3*dz;
bricks(4).max_x = 50*dx;
bricks(4).max_y = 26*dy;
bricks(4).max_z = 3*dz;
bricks(4).material_type = 2;

% define a PEC plate as ground
bricks(5).min_x = 0;
bricks(5).min_y = 0;
bricks(5).min_z = 0;
bricks(5).max_x = 50*dx;
bricks(5).max_y = 46*dy;
bricks(5).max_z = 0;
bricks(5).material_type = 2;
```

We associate a sampled voltage and sampled current pair to a port, and thus we have two ports. In Listing 6.2 a new parameter **ports** is defined and initialized as an empty array. At the end of the listing, the sampled voltages and currents are associated to ports through their indices. For instance, the subfield **sampled_voltage_index** of **ports(2)** is assigned the value 2, implying that **sampled_voltages(2)** is the sampled voltage associated to the second port. The subfield **impedance** is assigned the impedance of the respective port, which is supposed to be equal to the microstrip-line characteristic impedance and the resistance of the voltage source and resistor terminating the microstrip line. One additional subfield of **ports** is **is_source_port**, which indicates whether the port is the source port or not. In these two ports example, the first port is the excitation port due to the voltage source; therefore, **ports(1).is_source_port** is assigned the value ***true*** whereas for the second port **ports(2).is_source_port** is ***false***.

One should notice that the direction of the second sampled current **sampled_currents(2). direction** is toward the negative x direction as ***xn***. The reference currents are defined as flowing into the circuit for the purpose of calculating the network parameters, as can be seen in Figure 6.1.

Listing 6.2 define_output_parameters.m

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

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_voltages = [];
sampled_currents = [];
ports = [];

% figure refresh rate
plotting_step = 100;

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

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

% define sampled voltages
sampled_voltages(1).min_x = 14*dx;
sampled_voltages(1).min_y = 10*dy;
sampled_voltages(1).min_z = 0;
sampled_voltages(1).max_x = 20*dx;
sampled_voltages(1).max_y = 10*dy;
sampled_voltages(1).max_z = 3*dz;
sampled_voltages(1).direction = 'zp';
sampled_voltages(1).display_plot = false;

sampled_voltages(2).min_x = 30*dx;
sampled_voltages(2).min_y = 36*dy;
sampled_voltages(2).min_z = 0.0;
sampled_voltages(2).max_x = 36*dx;
sampled_voltages(2).max_y = 36*dy;
sampled_voltages(2).max_z = 3*dz;
sampled_voltages(2).direction = 'zp';
sampled_voltages(2).display_plot = false;

% define sampled currents
sampled_currents(1).min_x = 14*dx;
sampled_currents(1).min_y = 10*dy;
sampled_currents(1).min_z = 3*dz;
sampled_currents(1).max_x = 20*dx;
sampled_currents(1).max_y = 10*dy;
sampled_currents(1).max_z = 3*dz;
sampled_currents(1).direction = 'yp';
sampled_currents(1).display_plot = false;
```

```
sampled_currents(2).min_x = 30*dx;
sampled_currents(2).min_y = 36*dy;
sampled_currents(2).min_z = 3*dz;
sampled_currents(2).max_x = 36*dx;
sampled_currents(2).max_y = 36*dy;
sampled_currents(2).max_z = 3*dz;
sampled_currents(2).direction = 'yn';
sampled_currents(2).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;

ports(2).sampled_voltage_index = 2;
ports(2).sampled_current_index = 2;
ports(2).impedance = 50;
ports(2).is_source_port = false;
```

Furthermore, one should notice that we can compute only the set of S-parameters (S_{11} and S_{21}) for which the first port is the excitation port. The other set of S-parameters (S_{12} and S_{22}) can be obtained from (S_{11} and S_{21}) due to the symmetry in the structure. If the structure is not symmetric, we should assign the voltage source to the second port location and repeat the simulation. Another approach for obtaining all S-parameters is that, instead of defining a single source, sources can be defined at every port separately, each having internal impedance equal to the port impedance. Then the FDTD simulations can be repeated in a loop a number of times equals the number of ports, where each time one of the ports is set as the excitation port. Each time, the sources associated with the excitation port are activated while other sources are deactivated. The inactive sources will not generate power and serve only as passive terminations. Each time, the S-parameter set for the excitation port as the input can be calculated and stored.

Listing 6.2 illustrates how we can define the ports. The only initialization required for the ports is the determination of the number of ports in ***initialize_output_parameters*** as illustrated in Listing 6.3. After the FDTD time-marching loop is completed, the S-parameters are calculated in ***calculate_frequency_domain_outputs*** after frequency-domain transformation of

Listing 6.3 initialize_output_parameters.m

```
disp('initializing the output parameters');

number_of_sampled_electric_fields  = size(sampled_electric_fields,2);
number_of_sampled_magnetic_fields  = size(sampled_magnetic_fields,2);
number_of_sampled_voltages   = size(sampled_voltages,2);
number_of_sampled_currents   = size(sampled_currents,2);
number_of_ports = size(ports,2);
```

sampled voltages and currents as shown in Listing 6.4. First, the incident and reflected power waves are calculated for every port using (6.1). Then the S-parameters are calculated for the active port using (6.3). The calculated S-parameters are plotted as the last step in ***display_frequency_domain_outputs***, as shown in Listing 6.5.

Listing 6.4 calculate_frequency_domain_outputs.m

```
% calculation of S-parameters
% calculate incident and reflected power waves
for ind=1:number_of_ports
    svi = ports(ind).sampled_voltage_index;
    sci = ports(ind).sampled_current_index;
    Z = ports(ind).impedance;
    V = sampled_voltages(svi).frequency_domain_value;
    I = sampled_currents(sci).frequency_domain_value;
    ports(ind).a = 0.5*(V+Z.*I)./sqrt(real(Z));
    ports(ind).b = 0.5*(V-conj(Z).*I)./sqrt(real(Z));
    ports(ind).frequencies = frequency_array;
end

% calculate the S-parameters
for ind=1:number_of_ports
    if ports(ind).is_source_port == true
        for oind=1:number_of_ports
            ports(ind).S(oind).values = ports(oind).b ./ ports(ind).a;
        end
    end
end
```

Listing 6.5 display_frequency_domain_outputs.m

```
% figures for S-parameters
for ind=1:number_of_ports
    if ports(ind).is_source_port == true
        frequencies = ports(ind).frequencies*1e-9;
        for oind=1:number_of_ports
            S = ports(ind).S(oind).values;
            Sdb = 20 * log10(abs(S));
            Sphase = angle(S)*180/pi;
            figure;
            subplot(2,1,1);
            plot(frequencies, Sdb,'b-','linewidth',1.5);
            title(['S' num2str(oind) num2str(ind)],'fontsize',12);
            xlabel('frequency (GHz)','fontsize',12);
            ylabel('magnitude (dB)','fontsize',12);
            grid on;
            subplot(2,1,2);
            plot(frequencies, Sphase,'r-','linewidth',1.5);
            xlabel('frequency (GHz)','fontsize',12);
            ylabel('phase (degrees)','fontsize',12);
            grid on;
            drawnow;
        end
    end
end
```

The sample circuit in Figure 6.1 is run for 20,000 time steps, and the S-parameters of the circuit are obtained at the reference planes 10 cells away from the terminations where the ports are defined. Although the reference planes for the S-parameters are defined away from the terminations, it is possible to define the planes at the terminations. It is sufficient to place the sampled voltages and currents on the terminations.

Figure 6.3(a) shows the calculated S_{11} up to 20 GHz, and Figure 6.3(b) shows the calculated S_{21}. It can be seen that the circuit acts as a low-pass filter where the pass band is up to 5.5 GHz. Comparing the results in Figure 6.3 with the ones published in [14] it is evident that there are differences between the sets of results: some spikes appear in Figure 6.3. This is

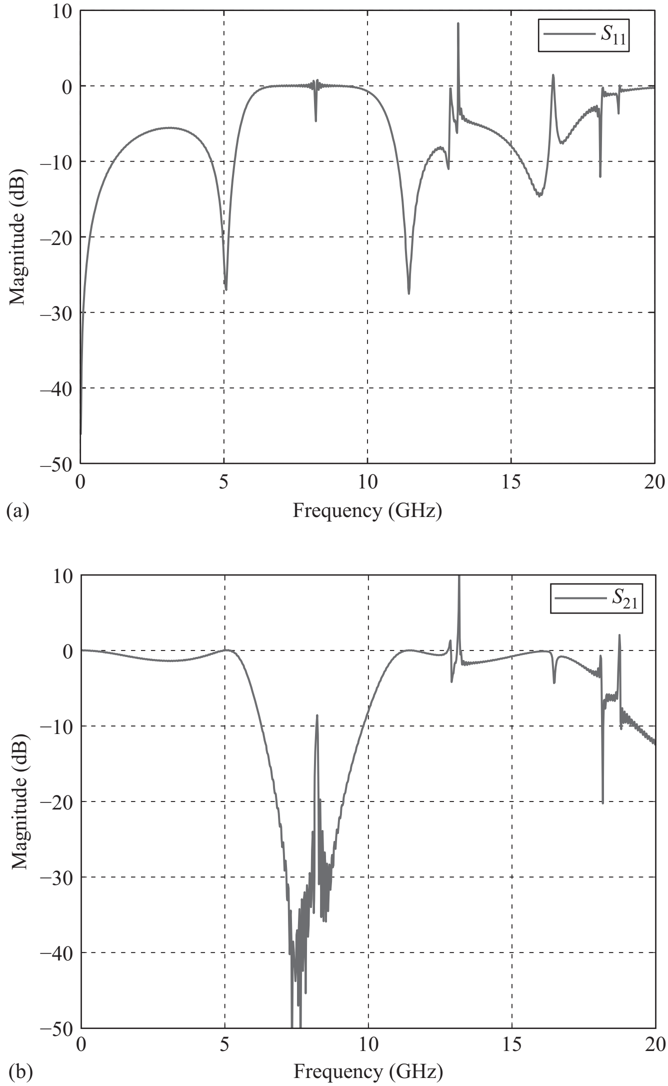

Figure 6.3 S-parameters of the microstrip low-pass filter: (a) S_{11} and (b) S_{21}.

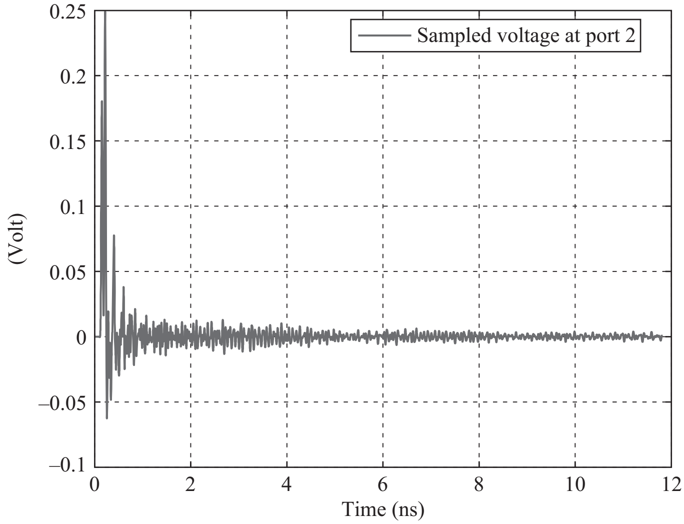

Figure 6.4 Sampled voltage at the second port of the microstrip low-pass filter.

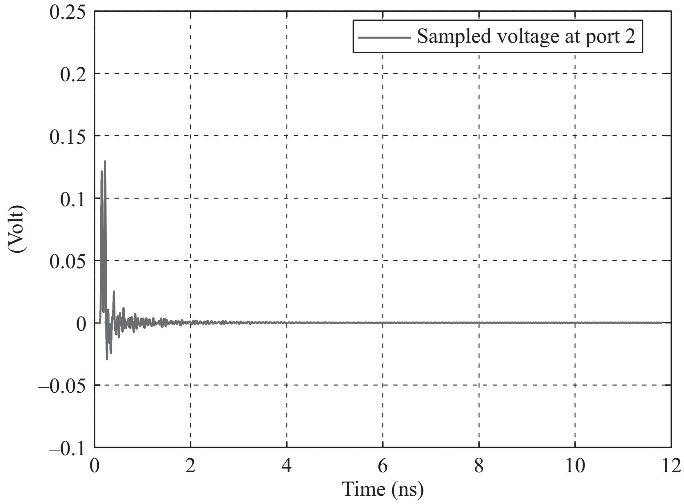

Figure 6.5 Sampled voltage at the second port of the microstrip low-pass filter with $\sigma^e = 0.2$.

because of the PEC boundaries, which make the FDTD simulation space into a cavity that resonates at certain frequencies. Figure 6.4 shows the sampled voltage captured at the second port of this microstrip filter. One can notice that, although the simulation is performed in 20,000 time steps, which is a large number of time steps for this problem, the transient response still did not damp out. This behavior exposes itself as spikes (numerical errors) in the S-parameter plots at different frequencies as shown in Figure 6.3. If the filter has slight loss, the attenuation will improve the numerical errors, as the time-domain response will be diminishing (as shown in Figure 6.5) for the filter lossy dielectric substrate. Therefore, the

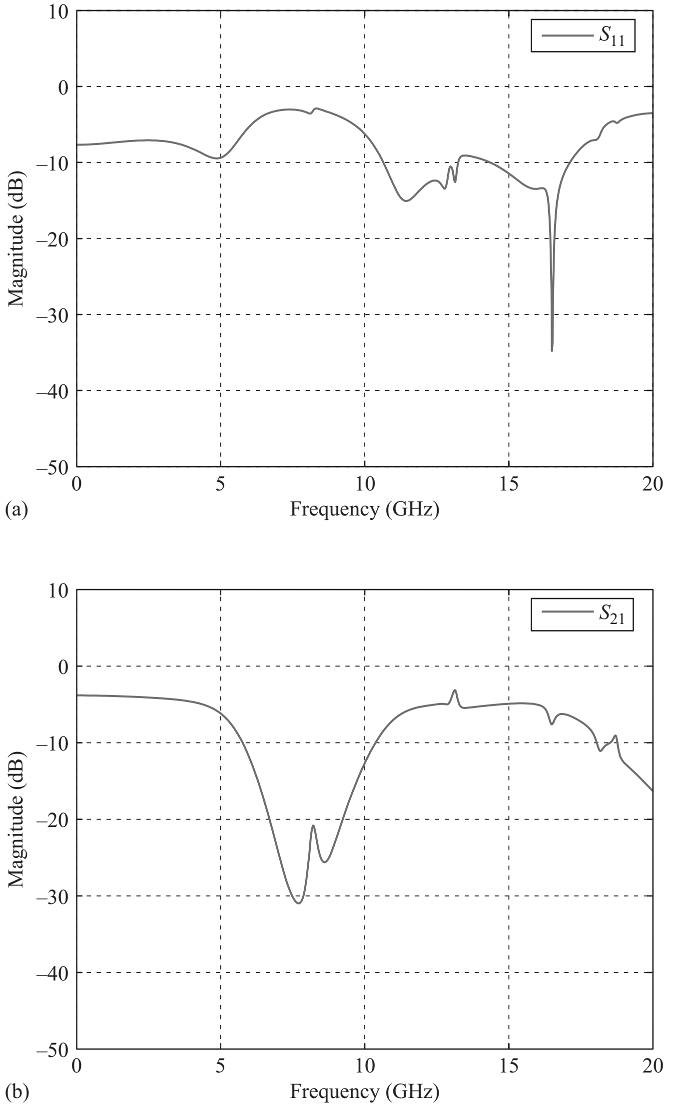

Figure 6.6 S-parameters of the microstrip low-pass filter with $\sigma^e = 0.2$: (a) S_{11} and (b) S_{21}.

truncation at a diminishing time-domain waveform will not cause significant errors even though the cavity modes of the PEC closed box still exist. This is clearly shown in Figure 6.6 (a) and 6.6(b). If the simulation had been performed such that boundaries simulate open space, the simulation time would take less and the S-parameters results would be clear of these spikes. Chapters 7 and 8 discuss algorithms that simulate open boundaries for FDTD simulations.

6.3 Simulation examples

6.3.1 Quarter-wave transformer

In this example the simulation of a microstrip quarter-wave transformer is presented. The geometry and the dimensions of the circuit are shown in Figure 6.7. The circuit is constructed on a substrate having 1 mm thickness and 4.6 dielectric constant. The index of the material type of the microstrip substrate is 4. A voltage source with internal resistance 50 Ω is connected to a 50 Ω microstrip line having 1.8 mm width and 4 mm length. This line is matched to a 100 Ω line through a 70.7 Ω line having 1 mm width and 10 mm length at 4 GHz. The 100 Ω line has 0.4 mm width and 4 mm length and is terminated by a 100 Ω resistor. The FDTD problem space is composed of cubic cells with sides each measuring 0.2 mm. The boundaries of the problem space are PEC on all sides; however, to suppress the cavity resonances another material type is defined as an absorber. The index of the material type of the absorber is 5. The relative permittivity and permeability of this absorber is 1, the electric conductivity is 1, and the magnetic conductivity is 142,130, which is the square of the intrinsic impedance of free space. The air gap between the objects and boundaries is zero on all sides. The bottom boundary serves as the ground of the microstrip circuit, while the other five sides are surrounded by the absorber material. The definition of the relevant problem space parameters and geometry are shown in Listing 6.6. The definition of the voltage source and the resistor are shown in Listing 6.7. In the previous example, the sampled voltages and sampled currents are defined on the microstrip lines away from the voltage

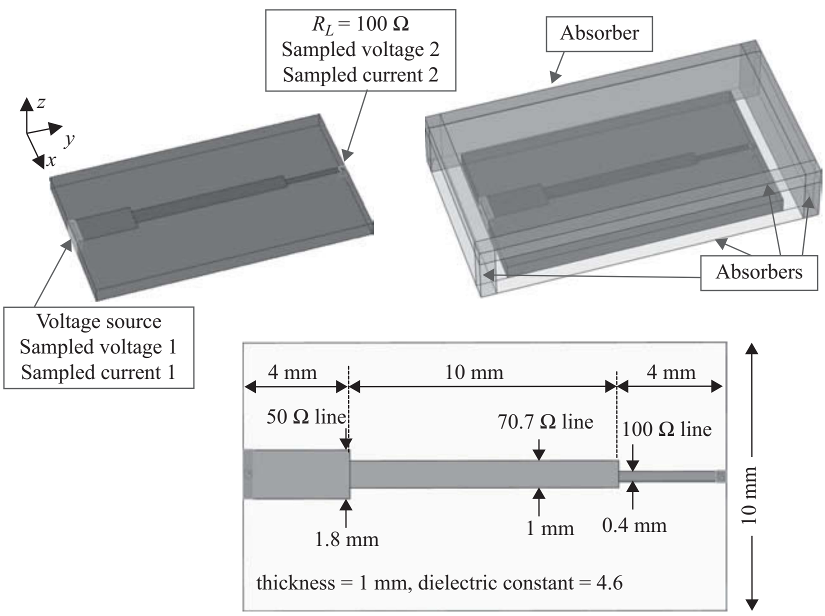

Figure 6.7 The geometry and dimensions of a microstrip quarter-wave transformer matching a 100 Ω line to 50 Ω line.

Listing 6.6 define_geometry.m

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

bricks  = [];
spheres = [];

% define a brick with material type 4
bricks(1).min_x = 0;
bricks(1).min_y = 0;
bricks(1).min_z = 0;
bricks(1).max_x = 10e-3;
bricks(1).max_y = 18e-3;
bricks(1).max_z = 1e-3;
bricks(1).material_type = 4;

% define a brick with material type 2
bricks(2).min_x = 4e-3;
bricks(2).min_y = 0;
bricks(2).min_z = 1e-3;
bricks(2).max_x = 5.8e-3;
bricks(2).max_y = 4e-3;
bricks(2).max_z = 1e-3;
bricks(2).material_type = 2;

% define a brick with material type 2
bricks(3).min_x = 4.4e-3;
bricks(3).min_y = 4e-3;
bricks(3).min_z = 1e-3;
bricks(3).max_x = 5.4e-3;
bricks(3).max_y = 14e-3;
bricks(3).max_z = 1e-3;
bricks(3).material_type = 2;

% define a brick with material type 2
bricks(4).min_x = 4.8e-3;
bricks(4).min_y = 14e-3;
bricks(4).min_z = 1e-3;
bricks(4).max_x = 5.2e-3;
bricks(4).max_y = 18e-3;
bricks(4).max_z = 1e-3;
bricks(4).material_type = 2;

% define absorber for zp side
bricks(5).min_x = -1e-3;
bricks(5).min_y = -2e-3;
bricks(5).min_z = 3e-3;
bricks(5).max_x = 11e-3;
bricks(5).max_y = 20e-3;
bricks(5).max_z = 4e-3;
bricks(5).material_type = 5;
```

```
% define absorber for xn side
bricks(6).min_x = -1e-3;
bricks(6).min_y = -2e-3;
bricks(6).min_z = 0;
bricks(6).max_x = 0;
bricks(6).max_y = 20e-3;
bricks(6).max_z = 4e-3;
bricks(6).material_type = 5;

% define absorber for xp side
bricks(7).min_x = 10e-3;
bricks(7).min_y = -2e-3;
bricks(7).min_z = 0;
bricks(7).max_x = 11e-3;
bricks(7).max_y = 20e-3;
bricks(7).max_z = 4e-3;
bricks(7).material_type = 5;

% define absorber for yn side
bricks(8).min_x = -1e-3;
bricks(8).min_y = -2e-3;
bricks(8).min_z = 0;
bricks(8).max_x = 11e-3;
bricks(8).max_y = -1e-3;
bricks(8).max_z = 4e-3;
bricks(8).material_type = 5;
% define absorber for yp side
bricks(9).min_x = -1e-3;
bricks(9).min_y = 19e-3;
bricks(9).min_z = 0;
bricks(9).max_x = 11e-3;
bricks(9).max_y = 20e-3;
bricks(9).max_z = 4e-3;
bricks(9).material_type = 5;
```

Listing 6.7 define_sources_and_lumped_elements.m

```
voltage_sources(1).min_x = 4e-3;
voltage_sources(1).min_y = 0;
voltage_sources(1).min_z = 0;
voltage_sources(1).max_x = 5.8e-3;
voltage_sources(1).max_y = 0.4e-3;
voltage_sources(1).max_z = 1e-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;

resistors(1).min_x = 4.8e-3;
resistors(1).min_y = 17.6e-3;
resistors(1).min_z = 0;
resistors(1).max_x = 5.2e-3;
resistors(1).max_y = 18e-3;
resistors(1).max_z = 1e-3;
resistors(1).direction = 'z';
resistors(1).resistance = 100;
```

source and the terminating resistor. In this example, the sampled voltages and sampled currents are defined on the voltage source and the terminating resistor. Then the sampled voltages and currents are assigned to ports. The definition of the output parameters is shown in Listing 6.8.

Listing 6.8 define_output_parameters.m

```
% frequency domain parameters
frequency_domain.start = 2e7;
frequency_domain.end   = 8e9;
frequency_domain.step  = 2e7;

% define sampled voltages
sampled_voltages(1).min_x = 4e-3;
sampled_voltages(1).min_y = 0;
sampled_voltages(1).min_z = 0;
sampled_voltages(1).max_x = 5.8e-3;
sampled_voltages(1).max_y = 0.4e-3;
sampled_voltages(1).max_z = 1e-3;
sampled_voltages(1).direction = 'zp';
sampled_voltages(1).display_plot = false;

% define sampled voltages
sampled_voltages(2).min_x = 4.8e-3;
sampled_voltages(2).min_y = 17.6e-3;
sampled_voltages(2).min_z = 0;
sampled_voltages(2).max_x = 5.2e-3;
sampled_voltages(2).max_y = 18e-3;
sampled_voltages(2).max_z = 1e-3;
sampled_voltages(2).direction = 'zp';
sampled_voltages(2).display_plot = false;

% define sampled currents
sampled_currents(1).min_x = 4e-3;
sampled_currents(1).min_y = 0;
sampled_currents(1).min_z = 0.4e-3;
sampled_currents(1).max_x = 5.8e-3;
sampled_currents(1).max_y = 0.4e-3;
sampled_currents(1).max_z = 0.6e-3;
sampled_currents(1).direction = 'zp';
sampled_currents(1).display_plot = false;
```

```
% define sampled currents
sampled_currents(2).min_x = 4.8e-3;
sampled_currents(2).min_y = 17.6e-3;
sampled_currents(2).min_z = 0.4e-3;
sampled_currents(2).max_x = 5.2e-3;
sampled_currents(2).max_y = 18e-3;
sampled_currents(2).max_z = 0.6e-3;
sampled_currents(2).direction = 'zp';
sampled_currents(2).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;

ports(2).sampled_voltage_index = 2;
ports(2).sampled_current_index = 2;
ports(2).impedance = 100;
ports(2).is_source_port = false;
```

The FDTD simulation is run for 5,000 time steps, and the S-parameters of the simulation are plotted in Figure 6.8. The plotted S_{11} indicates that a good match has been obtained at 4 GHz. Furthermore, the results are free of spikes, indicating that the absorbers used in the simulation suppressed the cavity resonances. However, one should keep in mind that although the use of absorbers improves the FDTD simulation results, it may introduce some other undesired errors to the simulation results. In the following chapters, more advanced absorbing boundaries are discussed that minimize these errors.

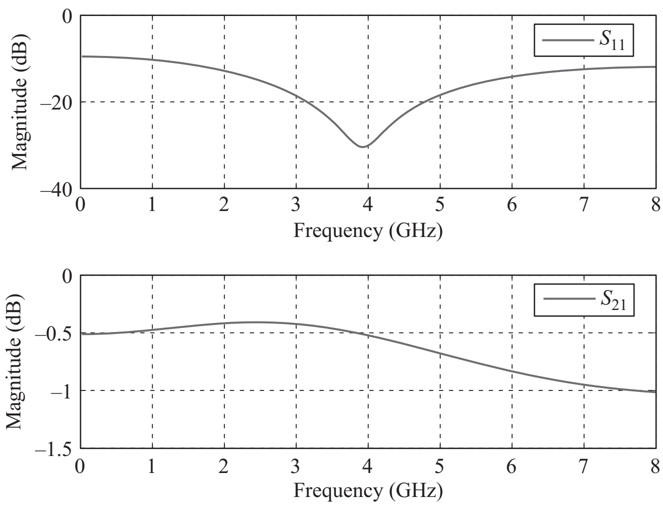

Figure 6.8 The S_{11} and S_{21} of the microstrip quarter-wave transformer circuit.

6.4 Exercises

6.1 Consider the low-pass filter circuit in the example described in Section 6.2. Use the same type of absorber as shown in the example in Section 6.3.1 to terminate the boundaries of the low-pass filter circuit. Use absorbers with 5 cells thickness, and leave at least 5 cells air gap between the circuit and the absorbers in the *xn*, *xp*, *yn*, and *yp* directions. Leave at least 10 cells air gap between the circuit and the absorbers in the *zp* direction. In the *zn* direction the PEC boundary will serve as the ground plane of the circuit. The FDTD problem space is illustrated in Figure 6.9 as a reference. Rerun the simulation for 3,000 time steps, and verify that the spikes in Figure 6.3 due to cavity resonances are suppressed.

6.2 Consider the quarter-wave transformer circuit shown in Example 6.3.1. Redefine the voltage source and the resistor such that voltage source will feed the 100 Ω line and will have 100 Ω internal resistance, while the resistor will terminate the 50 Ω line and will have 50 Ω resistance. Set port 2 as the active port, and run the simulation. Obtain the figures plotting S_{12} and S_{22}, and verify that they are similar to the S_{21} and S_{11} in Figure 6.8, respectively.

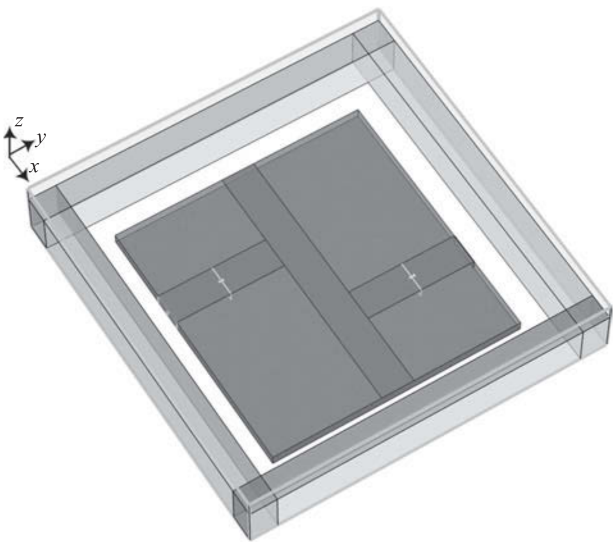

Figure 6.9 The problem space for the low-pass filter with absorbers on the *xn*, *xp*, *yn*, *yp*, and *zp* sides.