CHAPTER 7

Perfectly matched layer absorbing boundary

Because computational storage space is finite, the finite-difference time-domain (FDTD) problem space size is finite and needs to be truncated by special boundary conditions. In the previous chapters we discussed some examples for which the problem space is terminated by perfect electric conductor (PEC) boundaries. However, many applications, such as scattering and radiation problems, require the boundaries simulated as open space. The types of special boundary conditions that simulate electromagnetic waves propagating continuously beyond the computational space are called *absorbing boundary conditions* (ABCs). However, the imperfect truncation of the problem space will create numerical reflections, which will corrupt the computational results in the problem space after certain amounts of simulation time. So far, several various types of ABCs have been developed. However, the perfectly matched layer (PML) introduced by Berenger [15, 16] has been proven to be one of the most robust ABCs [17–20] in comparison with other techniques adopted in the past. PML is a finite-thickness special medium surrounding the computational space based on fictitious constitutive parameters to create a wave-impedance matching condition, which is independent of the angles and frequencies of the wave incident on this boundary. The theory and implementation of the PML boundary condition are illustrated in this chapter.

7.1 Theory of PML

In this section we demonstrate analytically the reflectionless characteristics of PML at the vacuum–PML and PML–PML interfaces [15] in detail.

7.1.1 Theory of PML at the vacuum–PML interface

We provide the analysis of reflection at a vacuum–PML interface in a two-dimensional case. Consider the TE_z polarized plane wave propagating in an arbitrary direction as shown in Figure 7.1. In the given TE_z case E_x, E_y, and H_z are the only field components that exist in

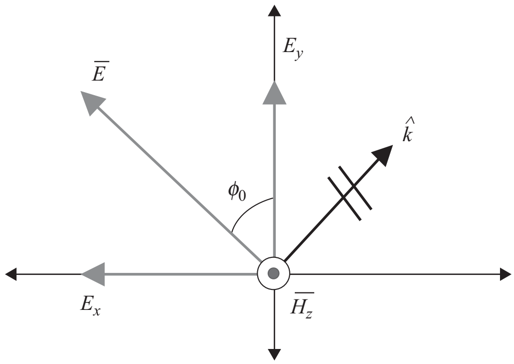


Figure 7.1 The field decomposition of a TE_z polarized plane wave.

the two-dimensional space. These field components can be expressed in the time-harmonic domain as

$$E_x = -E_0 \sin\phi_0 e^{j\omega(t-\alpha x-\beta y)}, \tag{7.1a}$$

$$E_y = E_0 \cos\phi_0 e^{j\omega(t-\alpha x-\beta y)}, \tag{7.1b}$$

$$H_z = H_0 e^{j\omega(t-\alpha x-\beta y)}. \tag{7.1c}$$

Maxwell's equations for a TE_z polarized wave are

$$\varepsilon_0 \frac{\partial E_x}{\partial t} + \sigma^e E_x = \frac{\partial H_z}{\partial y}, \tag{7.2a}$$

$$\varepsilon_0 \frac{\partial E_y}{\partial t} + \sigma^e E_y = -\frac{\partial H_z}{\partial x}, \tag{7.2b}$$

$$\mu_0 \frac{\partial H_z}{\partial t} + \sigma^m H_z = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}. \tag{7.2c}$$

In a TE_z PML medium, H_z can be broken into two artificial components associated with the x and y directions as

$$H_{zx} = H_{zx0} e^{-j\omega\beta y} e^{j\omega(t-\alpha x)}, \tag{7.3a}$$

$$H_{zy} = H_{zy0} e^{-j\omega\alpha x} e^{j\omega(t-\beta y)}, \tag{7.3b}$$

where $H_z = H_{zx} + H_{zy}$. Therefore, a modified set of Maxwell's equations for a TE_z polarized PML medium can be expressed as

$$\varepsilon_0 \frac{\partial E_x}{\partial t} + \sigma_{pey} E_x = \frac{\partial (H_{zx} + H_{zy})}{\partial y}, \tag{7.4a}$$

$$\varepsilon_0 \frac{\partial E_y}{\partial t} + \sigma_{pex} E_y = -\frac{\partial (H_{zx} + H_{zy})}{\partial x}, \tag{7.4b}$$

$$\mu_0 \frac{\partial H_{zx}}{\partial t} + \sigma_{pmx} H_{zx} = -\frac{\partial E_y}{\partial x}, \tag{7.4c}$$

$$\mu_0 \frac{\partial H_{zy}}{\partial t} + \sigma_{pmy} H_{zy} = \frac{\partial E_x}{\partial y}, \tag{7.4d}$$

where σ_{pex}, σ_{pey}, σ_{pmx}, and σ_{pmy} are the introduced fictitious conductivities. With the given conductivities the PML medium described by (7.4) is an anisotropic medium. When $\sigma_{pmx} = \sigma_{pmy} = \sigma^m$, merging (7.4c) and (7.4d) yields (7.2c). Field components E_y and H_{zx} together can represent a wave propagating in the x direction, and field components of E_x and H_{zy} represent a wave propagating in the y direction. Substituting the field equations for the x and y propagating waves in (7.1a), (7.1b), (7.3a), and (7.3b) into the modified Maxwell's equations given, one can obtain

$$\varepsilon_0 E_0 \sin\phi_0 - j\frac{\sigma_{pey}}{\omega} E_0 \sin\phi_0 = \beta(H_{zx0} + H_{zy0}), \tag{7.5a}$$

$$\varepsilon_0 E_0 \cos\phi_0 - j\frac{\sigma_{pex}}{\omega} E_0 \cos\phi_0 = \alpha(H_{zx0} + H_{zy0}), \tag{7.5b}$$

$$\mu_0 H_{zx0} - j\frac{\sigma_{pmx}}{\omega} H_{zx0} = \alpha E_0\, cos\phi_0, \tag{7.5c}$$

$$\mu_0 H_{zy0} - j\frac{\sigma_{pmy}}{\omega} H_{zy0} = \beta E_0 \sin\phi_0. \tag{7.5d}$$

Using (7.5c) and (7.5d) to eliminate magnetic field terms from (7.5a) and (7.5b) yields

$$\varepsilon_0\mu_0\left(1 - j\frac{\sigma_{pey}}{\varepsilon_0\omega}\right)\sin\phi_0 = \beta\left[\frac{\alpha\cos\phi_0}{\left(1 - j(\sigma_{pmx}/\mu_0\omega)\right)} + \frac{\beta\sin\phi_0}{\left(1 - j(\sigma_{pmy}/\mu_0\omega)\right)}\right], \tag{7.6a}$$

$$\varepsilon_0\mu_0\left(1 - j\frac{\sigma_{pex}}{\varepsilon_0\omega}\right)\cos\phi_0 = \alpha\left[\frac{\alpha\cos\phi_0}{\left(1 - j(\sigma_{pmx}/\mu_0\omega)\right)} + \frac{\beta\sin\phi_0}{\left(1 - j(\sigma_{pmy}/\mu_0\omega)\right)}\right]. \tag{7.6b}$$

The unknown constants α and β can be obtained from (7.6a) and (7.6b) as

$$\alpha = \frac{\sqrt{\mu_0\varepsilon_0}}{G}\left(1 - j\frac{\sigma_{pex}}{\omega\varepsilon_0}\right)\cos\phi_0, \tag{7.7a}$$

$$\beta = \frac{\sqrt{\mu_0\varepsilon_0}}{G}\left(1 - j\frac{\sigma_{pey}}{\omega\varepsilon_0}\right)\sin\phi_0, \tag{7.7b}$$

where

$$G = \sqrt{w_x\cos^2\phi_0 + w_y\sin^2\phi_0}, \tag{7.8}$$

and

$$w_x = \frac{1 - j\sigma_{pex}/\omega\varepsilon_0}{1 - j\sigma_{pmx}/\omega\mu_0}, \quad w_y = \frac{1 - j\sigma_{pey}/\omega\varepsilon_0}{1 - j\sigma_{pmy}/\omega\mu_0}, \tag{7.9}$$

Therefore, the generalized field component can be expressed as

$$\psi = \psi_0 e^{j\omega\left(t - \frac{x\cos\phi_0 + y\sin\phi_0}{cG}\right)} e^{-\frac{\sigma_{pex}\cos\phi_0}{\varepsilon_0 cG}x} e^{-\frac{\sigma_{pey}\sin\phi_0}{\varepsilon_0 cG}y}, \tag{7.10}$$

where the first exponential represents the phase of a plane wave and the second and third exponentials govern the decrease in the magnitude of the wave along the x axis and y axis, respectively.

Once α and β are determined by (7.7), the split magnetic field can be determined from (7.5c) and (7.5d) as

$$H_{zx0} = E_0\sqrt{\frac{\varepsilon_0}{\mu_0}}\frac{w_x\cos^2\phi_0}{G}, \tag{7.11a}$$

$$H_{zy0} = E_0\sqrt{\frac{\varepsilon_0}{\mu_0}}\frac{w_y\sin^2\phi_0}{G}. \tag{7.11b}$$

The magnitude of the total magnetic field H_z is then given as

$$H_0 = H_{zx0} + H_{zy0} = E_0\sqrt{\frac{\varepsilon_0}{\mu_0}}G. \tag{7.12}$$

The wave impedance in a TE_z PML medium can be expressed as

$$Z = \frac{E_0}{H_0} = \sqrt{\frac{\mu_0}{\varepsilon_0}}\frac{1}{G}. \tag{7.13}$$

It is important to note that if the conductivity parameters are chosen such that

$$\frac{\sigma_{pex}}{\varepsilon_0} = \frac{\sigma_{pmx}}{\mu_0} \quad \text{and} \quad \frac{\sigma_{pey}}{\varepsilon_0} = \frac{\sigma_{pmy}}{\mu_0}, \tag{7.14}$$

then the term G becomes equal to unity as w_x and w_y becomes equal to unity. Therefore, the wave impedance of this PML medium becomes the same as that of the interior free space. In other words, when the constitutive conditions of (7.14) are satisfied, a TE_z polarized wave can propagate from free space into the PML medium without reflection for all frequencies, and all incident angles as can be concluded from (7.8). One should notice that, when the electric and magnetic losses are assigned to be zero, the field updating equation (7.4) for the PML region becomes that of a vacuum region.

7.1.2 Theory of PML at the PML–PML interface

The reflection of fields at the interface between two different PML media can be analyzed as follows. A TE_z polarized wave of arbitrary incidence traveling from PML layer "1" to PML layer "2" is depicted in Figure 7.2, where the interface is normal to the x axis. The reflection coefficient for an arbitrary incident wave between two lossy media can be expressed as

$$r_p = \frac{Z_2\cos\phi_2 - Z_1\cos\phi_1}{Z_2\cos\phi_2 + Z_1\cos\phi_1}, \tag{7.15}$$

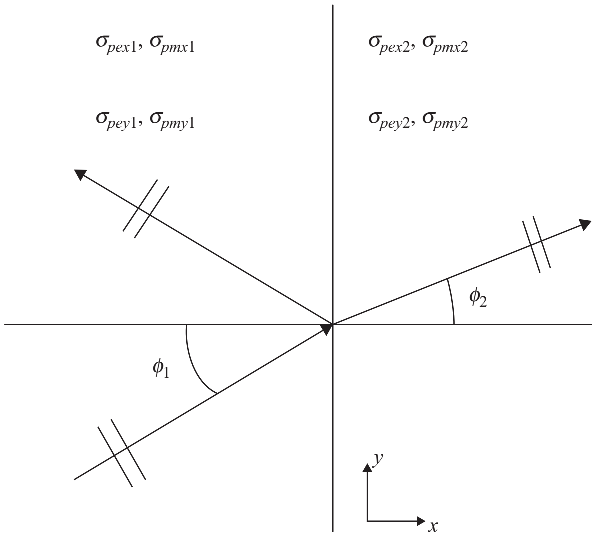


Figure 7.2 The plane wave transition at the interface between two PML media.

where Z_1 and Z_2 are the intrinsic impedances of respective mediums. Applying (7.13), the reflection coefficient r_p becomes

$$r_p = \frac{G_1 \cos\phi_2 - G_2 \cos\phi_1}{G_1 \cos\phi_2 + G_2 \cos\phi_1}. \tag{7.16}$$

The Snell–Descartes law at the interface normal to x of two lossy media can be described as

$$\left(1 - i\frac{\sigma_{y1}}{\varepsilon_0 \omega}\right)\frac{\sin\phi_1}{G_1} = \left(1 - i\frac{\sigma_{y2}}{\varepsilon_0 \omega}\right)\frac{\sin\phi_2}{G_2}. \tag{7.17}$$

When the two media have the same conductivities $\sigma_{pey1} = \sigma_{pey2} = \sigma_{pey}$ and $\sigma_{pmy1} = \sigma_{pmy2} = \sigma_{pmy}$, (7.17) becomes

$$\frac{\sin\phi_1}{G_1} = \frac{\sin\phi_2}{G_2}. \tag{7.18}$$

Moreover, when $(\sigma_{pex1}, \sigma_{pmx1})$, $(\sigma_{pex2}, \sigma_{pmx2})$, and $(\sigma_{pey}, \sigma_{pmy})$ satisfy the matching condition in (7.14), $G_1 = G_2 = 1$. Then (7.18) reduces to $\phi_1 = \phi_2$, and (7.16) reduces to $r_p = 0$. Therefore, theoretically when two PML media satisfy (7.14) and lie at an interface normal to the x axis with the same $(\sigma_{pey}, \sigma_{pmy})$, a wave can transmit through this interface with no reflections, at any angle of incidence and any frequency. When $(\sigma_{pex1}, \sigma_{pmx1}, \sigma_{pey1}, \sigma_{pmy1})$ are assigned to be (0, 0, 0, 0), the PML medium 1 becomes a vacuum. Therefore, when $(\sigma_{pex2}, \sigma_{pmx2})$ satisfies (7.14), the reflection coefficient at this interface is also null, which agrees with the previous vacuum–PML analysis. However, if the two media have the same $(\sigma_{pey}, \sigma_{pmy})$ but do not satisfy (7.14), then the reflection coefficient becomes

$$r_p = \frac{\sin\phi_1 \cos\phi_2 - \sin\phi_2 \cos\phi_1}{\sin\phi_1 \cos\phi_2 + \sin\phi_2 \cos\phi_1}. \tag{7.19}$$

Substituting (7.18) into (7.19), the reflection coefficient of two unmatched PML media becomes

$$r_p = \frac{\sqrt{w_{x1}} - \sqrt{w_{x2}}}{\sqrt{w_{x1}} + \sqrt{w_{x2}}}. \tag{7.20}$$

Equation (7.20) shows that the reflection coefficient for two unmatched PML media is highly dependent on frequency, regardless of the incident angle. When the two PML media follow the reflectionless condition in (7.14), $w_{x1} = w_{x2} = 1$, the reflection coefficient becomes null.

The analysis can be applied to two PML media lying at the interface normal to the y axis as well. The Snell–Descartes law related to this interface is

$$\left(1 - i\frac{\sigma_{x1}}{\varepsilon_0\omega}\right)\frac{\sin\phi_1}{G_1} = \left(1 - i\frac{\sigma_{x2}}{\varepsilon_0\omega}\right)\frac{\sin\phi_2}{G_2}. \tag{7.21}$$

If the two media have the same conductivities such that $\sigma_{pex1} = \sigma_{pex2} = \sigma_{pex}$ and $\sigma_{pmx1} = \sigma_{pmx2} = \sigma_{pmx}$, (7.21) reduces to (7.18). Similarly, if $(\sigma_{pey1}, \sigma_{pmy1})$, $(\sigma_{pey2}, \sigma_{pmy2})$, and $(\sigma_{pex}, \sigma_{pmx})$ satisfy the matching condition in (7.14), then $G_1 = G_2 = 1$. Equation (7.21) then reduces to $\phi_1 = \phi_2$, and the reflection coefficient at this interface is $r_p = 0$. To match the vacuum–PML interface normal to the y axis, the reflectionless condition can be achieved when $(\sigma_{pey2}, \sigma_{pmy2})$ of the PML medium satisfies (7.14).

Based on the previous discussion, if a two-dimensional FDTD problem space is attached with an adequate thickness of PML media as shown in Figure 7.3, the outgoing waves will be absorbed without any undesired numerical reflections. The PML regions must be assigned appropriate conductivity values satisfying the matching condition (7.14); the positive and

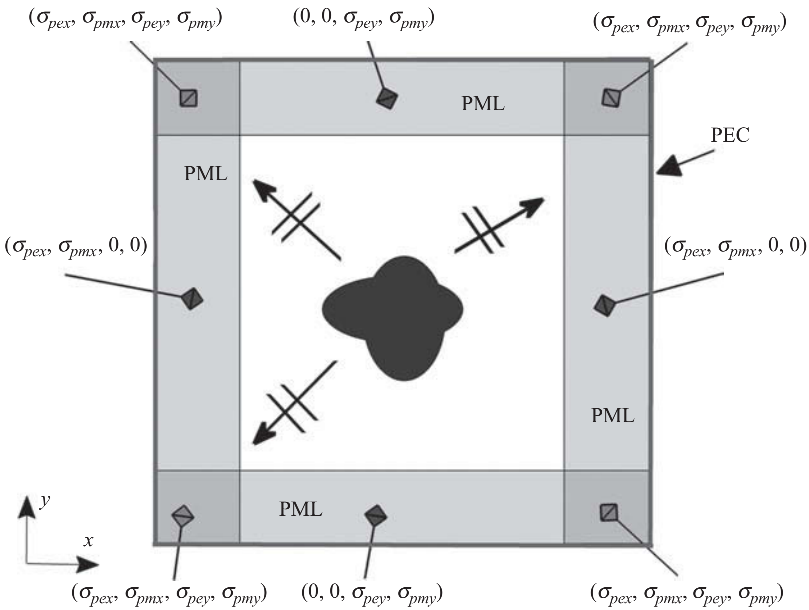


Figure 7.3 The loss distributions in two-dimensional PML regions.

negative x boundaries of the PML regions have nonzero σ_{pex}, and σ_{pmx}, whereas the positive and negative y boundaries of the PML regions have nonzero σ_{pey}, and σ_{pmy} values. The coexistence of nonzero values of σ_{pex}, σ_{pmx}, σ_{pey}, and σ_{pmy} is required at the four corner PML overlapping regions. Using a similar analysis, the conditions of (7.14) can be applied to a TM_z polarized wave to travel from free space to PML and from PML to PML without reflection [21]. Using the same impedance matching condition in (7.14), the modified Maxwell's equations for two-dimensional TM_z PML updating equations are obtained as

$$\varepsilon_0 \frac{\partial E_{zx}}{\partial t} + \sigma_{pex} E_{zx} = \frac{\partial H_y}{\partial x}, \tag{7.22a}$$

$$\varepsilon_0 \frac{\partial E_{zy}}{\partial t} + \sigma_{pey} E_{zy} = -\frac{\partial H_x}{\partial y}, \tag{7.22b}$$

$$\mu_0 \frac{\partial H_x}{\partial t} + \sigma_{pmy} H_x = -\frac{\partial \left(E_{zx} + E_{zy}\right)}{\partial y}, \tag{7.22c}$$

$$\mu_0 \frac{\partial H_y}{\partial t} + \sigma_{pmx} H_y = \frac{\partial \left(E_{zx} + E_{zy}\right)}{\partial x}. \tag{7.22d}$$

The finite difference approximation schemes can be applied to the modified Maxwell's equations (7.4) and (7.22) to obtain the field-updating equations for the PML regions in the two-dimensional FDTD problem space.

7.2 PML equations for three-dimensional problem space

For a three-dimensional problem space, each field component of the electric and magnetic fields is broken into two field components similar to the two-dimensional case. Therefore, the modified Maxwell's equations have 12 field components instead of the original six components. These modified split electric field equations presented in [16] are

$$\varepsilon_0 \frac{\partial E_{xy}}{\partial t} + \sigma_{pey} E_{xy} = \frac{\partial \left(H_{zx} + H_{zy}\right)}{\partial y}, \tag{7.23a}$$

$$\varepsilon_0 \frac{\partial E_{xz}}{\partial t} + \sigma_{pez} E_{xz} = -\frac{\partial \left(H_{yx} + H_{yz}\right)}{\partial z}, \tag{7.23b}$$

$$\varepsilon_0 \frac{\partial E_{yx}}{\partial t} + \sigma_{pex} E_{yx} = -\frac{\partial \left(H_{zx} + H_{zy}\right)}{\partial x}, \tag{7.23c}$$

$$\varepsilon_0 \frac{\partial E_{yz}}{\partial t} + \sigma_{pez} E_{yz} = \frac{\partial \left(H_{xy} + H_{xz}\right)}{\partial z}, \tag{7.23d}$$

$$\varepsilon_0 \frac{\partial E_{zx}}{\partial t} + \sigma_{pex} E_{zx} = \frac{\partial \left(H_{yx} + H_{yz}\right)}{\partial x}, \tag{7.23e}$$

$$\varepsilon_0 \frac{\partial E_{zy}}{\partial t} + \sigma_{pey} E_{zy} = -\frac{\partial \left(H_{xy} + H_{xz}\right)}{\partial y}, \tag{7.23f}$$

whereas the modified Maxwell's split magnetic field equations are

$$\mu_0 \frac{\partial H_{xy}}{\partial t} + \sigma_{pmy} H_{xy} = -\frac{\partial (E_{zx} + E_{zy})}{\partial y}, \tag{7.24a}$$

$$\mu_0 \frac{\partial H_{xz}}{\partial t} + \sigma_{pmz} H_{xz} = \frac{\partial (E_{yx} + E_{yz})}{\partial z}, \tag{7.24b}$$

$$\mu_0 \frac{\partial H_{yz}}{\partial t} + \sigma_{pmz} H_{yz} = -\frac{\partial (E_{xy} + E_{xz})}{\partial z}, \tag{7.24c}$$

$$\mu_0 \frac{\partial H_{yx}}{\partial t} + \sigma_{pmx} H_{yx} = \frac{\partial (E_{zx} + E_{zy})}{\partial x}, \tag{7.24d}$$

$$\mu_0 \frac{\partial H_{zy}}{\partial t} + \sigma_{pmy} H_{zy} = \frac{\partial (E_{xy} + E_{xz})}{\partial y}, \tag{7.24e}$$

$$\mu_0 \frac{\partial H_{zx}}{\partial t} + \sigma_{pmx} H_{zx} = -\frac{\partial (E_{yx} + E_{yz})}{\partial x}. \tag{7.24f}$$

Then the matching condition for a three-dimensional PML is given by

$$\frac{\sigma_{pex}}{\varepsilon_0} = \frac{\sigma_{pmx}}{\mu_0}, \quad \frac{\sigma_{pey}}{\varepsilon_0} = \frac{\sigma_{pey}}{\mu_0}, \quad \text{and} \quad \frac{\sigma_{pez}}{\varepsilon_0} = \frac{\sigma_{pmz}}{\mu_0}. \tag{7.25}$$

If a three-dimensional FDTD problem space is attached with adequate thickness of PML media as shown in Figure 7.4, the outgoing waves will be absorbed without any undesired numerical reflections. The PML regions must be assigned appropriate conductivity values satisfying the matching condition (7.25); the positive and negative x boundaries of PML regions have nonzero σ_{pex} and σ_{pmx}, the positive and negative y boundaries of PML regions have nonzero σ_{pey} and σ_{pmy}, and the positive and negative z boundaries of PML regions have nonzero σ_{pez} and σ_{pmz} values as illustrated in Figure 7.4. The coexistence of nonzero values of σ_{pex}, σ_{pmx}, σ_{pey}, σ_{pmy}, σ_{pez}, and σ_{pmz} is required at the PML overlapping regions.

Finally, applying the finite difference schemes to the modified Maxwell's equations (7.23) and (7.24), one can obtain the FDTD field updating equations for the three-dimensional PML regions.

7.3 PML loss functions

As discussed in the previous sections, PML regions can be formed as the boundaries of an FDTD problem space where specific conductivities are assigned such that the outgoing waves penetrate without reflection and attenuate while traveling in the PML medium. The PML medium is governed by the modified Maxwell's equations (7.4), (7.22), (7.23), and (7.24), which can be used to obtain the updating equations for the field components in the PML regions. Furthermore, the outer boundaries of the PML regions are terminated by PEC walls. When a finite-thickness PML medium backed by a PEC wall is adopted, an incident plane wave may not be totally attenuated within the PML region, and small reflections to the interior domain from the PEC back wall may occur. For a finite-width PML medium where

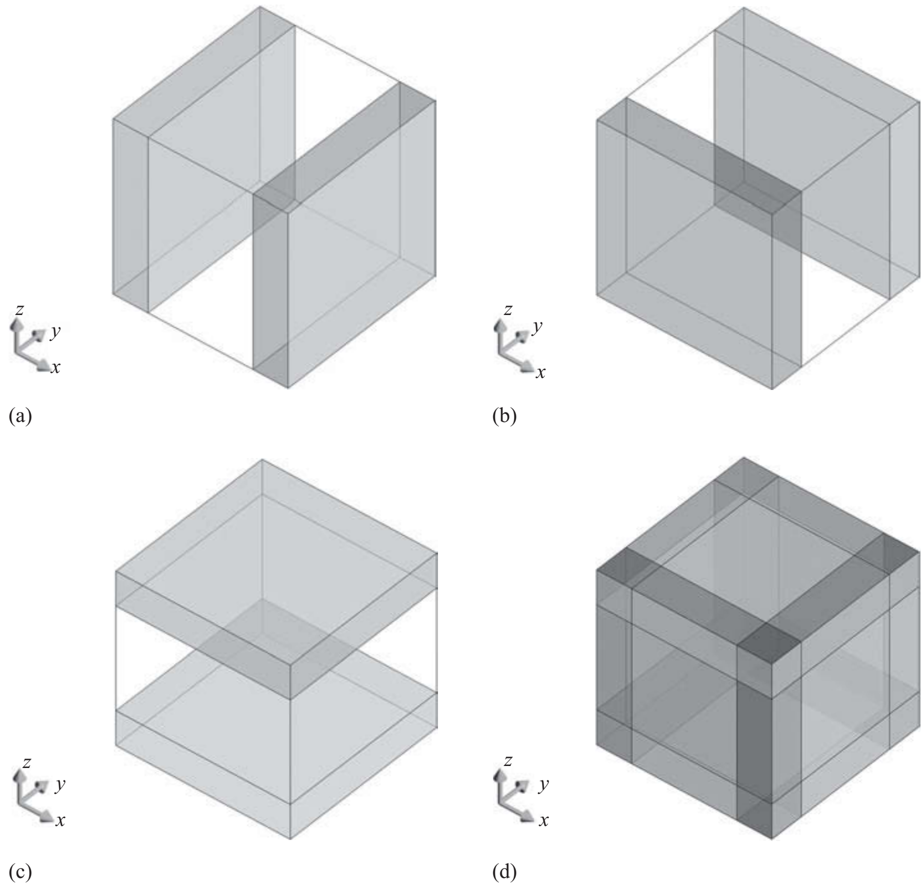


Figure 7.4 Nonzero regions of PML conductivities for a three-dimensional FDTD simulation domain: (a) nonzero σ_{pex} and σ_{pmx}; (b) nonzero σ_{pey} and σ_{pmy}; (c) nonzero σ_{pez} and σ_{pmz}; and (d) overlapping PML regions.

the conductivity distribution is uniform, there is an apparent reflection coefficient, which is expressed as

$$R(\phi_0) = e^{-2\frac{\sigma \cos \phi_0}{\varepsilon_0 c}\delta}, \tag{7.26}$$

where σ is the conductivity of the medium. Here the exponential term is the attenuation factor of the field magnitudes of the plane waves as shown in (7.10), and δ is the thickness of the PML medium. The factor 2 in the exponent is due to the travel distance, which is twice the distance between the vacuum–PML interface and the PEC backing. If ϕ_0 is 0, (7.26) is the reflection coefficient for a finite-thickness PML medium at normal incidence. If ϕ_0 is $\pi/2$, the incident plane wave is grazing to the PML medium and is attenuated by the perpendicular PML medium. From (7.26), the effectiveness of a finite-width PML is dependent on the losses within the PML medium. In addition, (7.26) can be used not only to predict the ideal performance of a finite-width PML medium but also to compute the appropriate loss distribution based on the loss profiles described in the following paragraphs.

As presented in [15], significant reflections were observed when constant uniform losses are assigned throughout the PML media, which is a result of the discrete approximation of fields and material parameters at the domain–PML interfaces and sharp variation of conductivity profiles. This mismatch problem can be tempered using a spatially gradually increasing conductivity distribution, which is zero at the domain–PML interface and tends to be a maximum conductivity $\sigma_{\max}$ at the end of the PML region. In [18] two major types of mathematical functions are proposed as the conductivity distributions or loss profiles: power and geometrically increasing functions. The power-increasing function is defined as

$$\sigma(\rho) = \sigma_{\max}\left(\frac{\rho}{\delta}\right)^{n_{pml}}, \tag{7.27a}$$

$$\sigma_{\max} = -\frac{(n_{pml}+1)\varepsilon_0 c \ln(R(0))}{2\Delta s N}, \tag{7.27b}$$

where ρ is the distance from the computational domain–PML interface to the position of the field component, and δ is the thickness of the PML cells. The parameter N is the number of PML cells, Δs is the cell size used for a PML cell, and $R(0)$ is the reflection coefficient of the finite-width PML medium at normal incidence. The distribution function is linear for $n_{pml} = 1$ and parabolic for $n_{pml} = 2$. To determine the conductivity profile using (7.27a) the parameters $R(0)$ and n_{pml} must be predefined. These parameters are used to determine $\sigma_{\max}$ using (7.27b), which is then used in the calculation of $\sigma(\rho)$. Usually n_{pml} takes a value such as 2, 3, or 4 and $R(0)$ takes a very small value such as 10^{-8} for a satisfactory PML performance.

The geometrically increasing distribution for $\sigma(\rho)$ is given by

$$\sigma(\rho) = \sigma_0 g^{\frac{\rho}{\Delta s}}, \tag{7.28a}$$

$$\sigma_0 = -\frac{\varepsilon_0 c \ln(g)}{2\Delta s g^N - 1}\ln(R(0)), \tag{7.28b}$$

where the parameter g is a real number used for a geometrically increasing function.

7.4 FDTD updating equations for PML and MATLAB® implementation

7.4.1 PML updating equations – two-dimensional TE_z case

The PML updating equations can be obtained for the two-dimensional TE_z case by applying the central difference approximation to the derivatives in the modified Maxwell's equations (7.4). After some manipulations one can obtain the two-dimensional TE_z PML updating equations based on the field positioning scheme given in Figure 1.10 as

$$E_x^{n+1}(i,j) = C_{exe}(i,j) \times E_x^n(i,j) + C_{exhz}(i,j) \times \left(H_z^{n+\frac{1}{2}}(i,j) - H_z^{n+\frac{1}{2}}(i,j-1)\right), \tag{7.29}$$

where

$$C_{exe}(i,j) = \frac{2\varepsilon_0 - \Delta t \sigma_{pey}(i,j)}{2\varepsilon_0 + \Delta t \sigma_{pey}(i,j)},$$

$$C_{exhz}(i,j) = \frac{2\Delta t}{\left(2\varepsilon_0 + \Delta t \sigma^{e}_{pey}(i,j)\right)\Delta y}.$$

$$E_y^{n+1}(i,j) = C_{eye}(i,j) \times E_y^{n}(i,j) + C_{eyhz}(i,j) \times \left(H_z^{n+\frac{1}{2}}(i,j) - H_z^{n+\frac{1}{2}}(i-1,j)\right), \quad (7.30)$$

where

$$C_{eye}(i,j) = \frac{2\varepsilon_0 - \Delta t \sigma_{pex}(i,j)}{2\varepsilon_0 + \Delta t \sigma_{pex}(i,j)},$$

$$C_{eyhz}(i,j) = -\frac{2\Delta t}{\left(2\varepsilon_0 + \Delta t \sigma_{pex}(i,j)\right)\Delta x}.$$

$$H_{zx}^{n+\frac{1}{2}}(i,j) = C_{hzxh}(i,j) \times H_{zx}^{n-\frac{1}{2}}(i,j) + C_{hzxey}(i,j) \times \left(E_y^{n}(i+1,j) - E_y^{n}(i,j)\right), \quad (7.31)$$

where

$$C_{hzxh}(i,j) = \frac{2\mu_0 - \Delta t \sigma_{pmx}(i,j)}{2\mu_0 + \Delta t \sigma_{pmx}(i,j)},$$

$$C_{hzxey}(i,j) = -\frac{2\Delta t}{\left(2\mu_0 + \Delta t \sigma_{pmx}(i,j)\right)\Delta x}.$$

$$H_{zy}^{n+\frac{1}{2}}(i,j) = C_{hzyh}(i,j) \times H_{zy}^{n-\frac{1}{2}}(i,j) + C_{hzyex}(i,j) \times \left(E_x^{n}(i,j+1) - E_x^{n}(i,j)\right), \quad (7.32)$$

where

$$C_{hzyh}(i,j) = \frac{2\mu_0 - \Delta t \sigma_{pmy}(i,j)}{2\mu_0 + \Delta t \sigma_{pmy}(i,j)},$$

$$C_{hzyex}(i,j) = \frac{2\Delta t}{\left(2\mu_0 + \Delta t \sigma_{pmy}(i,j)\right)\Delta y}.$$

One should recall that $H_z(i,j) = H_{zx}(i,j) + H_{zy}(i,j)$. As illustrated in Figure 7.3 certain PML conductivities are defined at certain regions. Therefore, each of the equations (7.29)–(7.32) is applied in a two-dimensional problem space where its respective PML conductivity is defined. Figure 7.5 shows the PML regions where the conductivity parameters are nonzero and the respective field components that need to be updated at each region. Figure 7.5(a) shows the regions where σ_{pex} is defined. These regions are denoted as *xn* and *xp*. One can notice from (7.4b) and (7.30) that σ_{pex} appears in the equation where E_y is updated. Therefore, the components of E_y lying in the *xn* and *xp* regions are updated at every time step using (7.30). The other components of E_y that are located in the intermediate region can be updated using the regular non-PML updating equation (1.34). Similarly, σ_{pey} is nonzero in the regions that are denoted as *yn* and *yp* in Figure 7.5(b). The components of E_x lying in the *yn* and *yp*

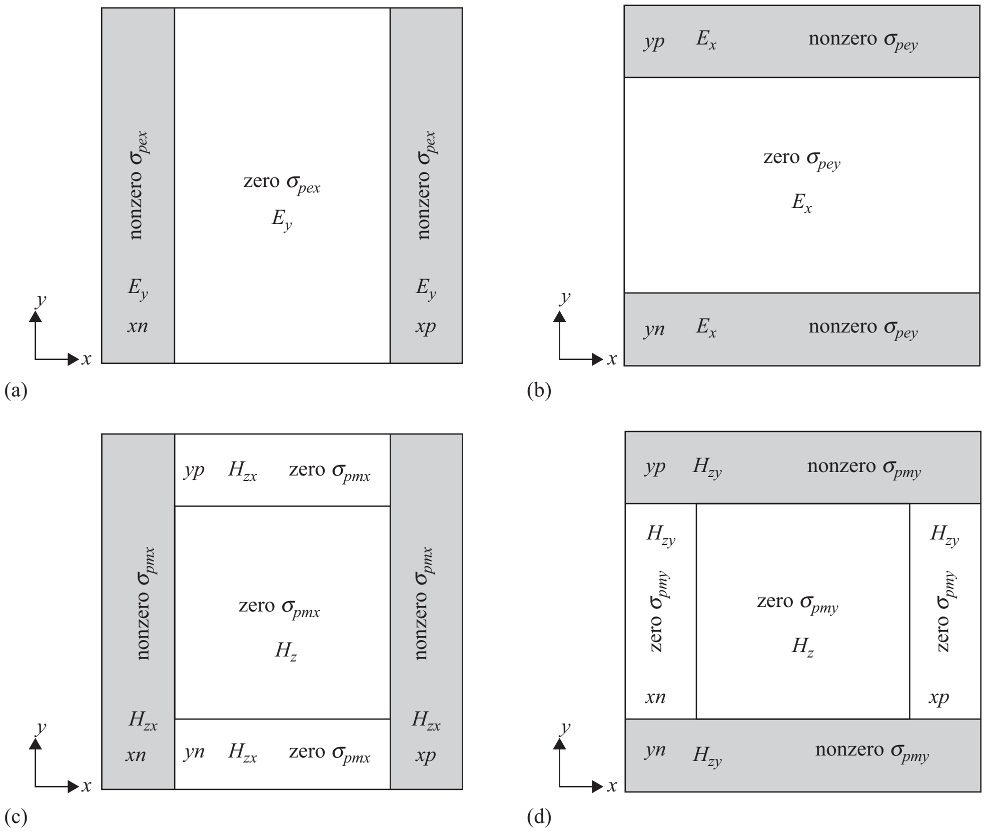


Figure 7.5 Nonzero TE_z regions of PML conductivities: (a) nonzero σ_{pex}; (b) nonzero σ_{pey}; (c) nonzero σ_{pmx}; and (d) nonzero σ_{pmy}.

regions are updated at every time step using (7.29), and the components located in the intermediate region are updated using the regular non-PML updating equation (1.33).

Since the magnetic field H_z is the sum of two split fields H_{zx} and H_{zy} in the PML regions, its update is more complicated. The PML regions and non-PML regions are shown in Figure 7.5(c) and 7.5(d), where the PML regions are denoted as xn, xp, yn, and yp. The magnetic field components of H_z lying in the non-PML region can be updated using the regular updating equation (1.35). However, the components of H_z in the PML regions are not directly calculated; the components of H_{zx} and H_{zy} are calculated in the PML regions using the appropriate updating equations, and then they are summed up to yield H_z in the PML region. The conductivity σ_{pmx} is nonzero in the xn and xp regions as shown in Figure 7.5(c). The components of H_{zx} are calculated in the same regions using (7.31). However, components of H_{zx} need to be calculated in the yn and yp regions as well. Since σ_{pmx} is zero in these regions setting σ_{pmx} zero in (7.31) will yield the required updating equation for H_{zx} in the yn and yp regions as

$$H_{zx}^{n+\frac{1}{2}}(i,j) = C_{hzxh}(i,j) \times H_{zx}^{n-\frac{1}{2}}(i,j) + C_{hzxey}(i,j) \times \left(E_y^n(i+1,j) - E_y^n(i,j)\right), \qquad (7.33)$$

where

$$C_{hzxh}(i,j) = 1, \quad C_{hzxey}(i,j) = -\frac{\Delta t}{\mu_0 \Delta x}.$$

Similarly, the conductivity σ_{pmy} is nonzero in the yn and yp regions as shown in Figure 7.5(d). The components of H_{zy} are calculated in these regions using (7.32). The components of H_{zy} need to be calculated in the xn and xp regions as well. Since σ_{pmy} is zero in these regions, setting σ_{pmy} to zero in (7.32) will yield the required updating equation for H_{zy} in the xn and xp regions as

$$H_{zy}^{n+\frac{1}{2}}(i,j) = C_{hzyh}(i,j) \times H_{zy}^{n-\frac{1}{2}}(i,j) + C_{hzyex}(i,j) \times \left(E_x^n(i,j+1) - E_x^n(i,j)\right), \tag{7.34}$$

where

$$C_{hzyh}(i,j) = 1, \quad C_{hzyex}(i,j) = \frac{\Delta t}{\mu_0 \Delta y}.$$

After all the components of H_{zx} and H_{zy} are updated in the PML regions, they are added to calculate H_z.

7.4.2 PML updating equations – two-dimensional TM_z case

The PML updating equations can be obtained for the two-dimensional TM_z case by applying the central difference approximation to the derivatives in the modified Maxwell's equations (7.22). After some manipulations one can obtain the two-dimensional TM_z PML updating equations based on the field positioning scheme given in Figure 1.11 as

$$E_{zx}^{n+1}(i,j) = C_{ezxe}(i,j) \times E_{zx}^n(i,j) + C_{ezxhy}(i,j) \times \left(H_y^{n+\frac{1}{2}}(i,j) - H_y^{n+\frac{1}{2}}(i-1,j)\right), \tag{7.35}$$

where

$$C_{ezxe}(i,j) = \frac{2\varepsilon_0 - \Delta t \sigma_{pex}(i,j)}{2\varepsilon_0 + \Delta t \sigma_{pex}(i,j)},$$

$$C_{ezxhy}(i,j) = \frac{2\Delta t}{\left(2\varepsilon_0 + \Delta t \sigma_{pex}(i,j)\right)\Delta x}.$$

$$E_{zy}^{n+1}(i,j) = C_{ezye}(i,j) \times E_z^n(i,j) + C_{ezyhx}(i,j) \times \left(H_x^{n+\frac{1}{2}}(i,j) - H_x^{n+\frac{1}{2}}(i,j-1)\right), \tag{7.36}$$

where

$$C_{ezye}(i,j) = \frac{2\varepsilon_0(i,\ j) - \Delta t \sigma_{pey}(i,j)}{2\varepsilon_0(i,j) + \Delta t \sigma_{pey}(i,j)},$$

$$C_{ezyhx}(i,j) = -\frac{2\Delta t}{\left(2\varepsilon_0 + \Delta t \sigma_{pey}(i,j)\right)\Delta y}.$$

$$H_x^{n+\frac{1}{2}}(i,j) = C_{hxh}(i,j) \times H_x^{n-\frac{1}{2}}(i,j) + C_{hxez}(i,j) \times \left(E_z^n(i,j+1) - E_z^n(i,j)\right), \tag{7.37}$$

where

$$C_{hxh}(i,j) = \frac{2\mu_0 - \Delta t \sigma_{pmy}(i,j)}{2\mu_0 + \Delta t \sigma_{pmy}(i,j)},$$

$$C_{hxez}(i,j) = -\frac{2\Delta t}{(2\mu_0 + \Delta t \sigma_{pmy}(i,j))\Delta y}.$$

$$H_y^{n+\frac{1}{2}}(i,j) = C_{hyh}(i,j) \times H_y^{n-\frac{1}{2}}(i,j) + C_{hyez}(i,j) \times \left(E_z^n(i+1,j) - E_z^n(i,j)\right), \quad (7.38)$$

where

$$C_{hyh}(i,j) = \frac{2\mu_0 - \Delta t \sigma_{pmx}(i,j)}{2\mu_0 + \Delta t \sigma_{pmx}(i,j)},$$

$$C_{hyez}(i,j) = \frac{2\Delta t}{(2\mu_0 + \Delta t \sigma_{pmx}(i,j))\Delta x}.$$

One should recall that $E_z(i, j) = E_{zx}(i, j) + E_{zy}(i, j)$. Consider the nonzero PML conductivity regions given in Figure 7.6. Figure 7.6(a) shows that σ_{pmx} is defined in the regions

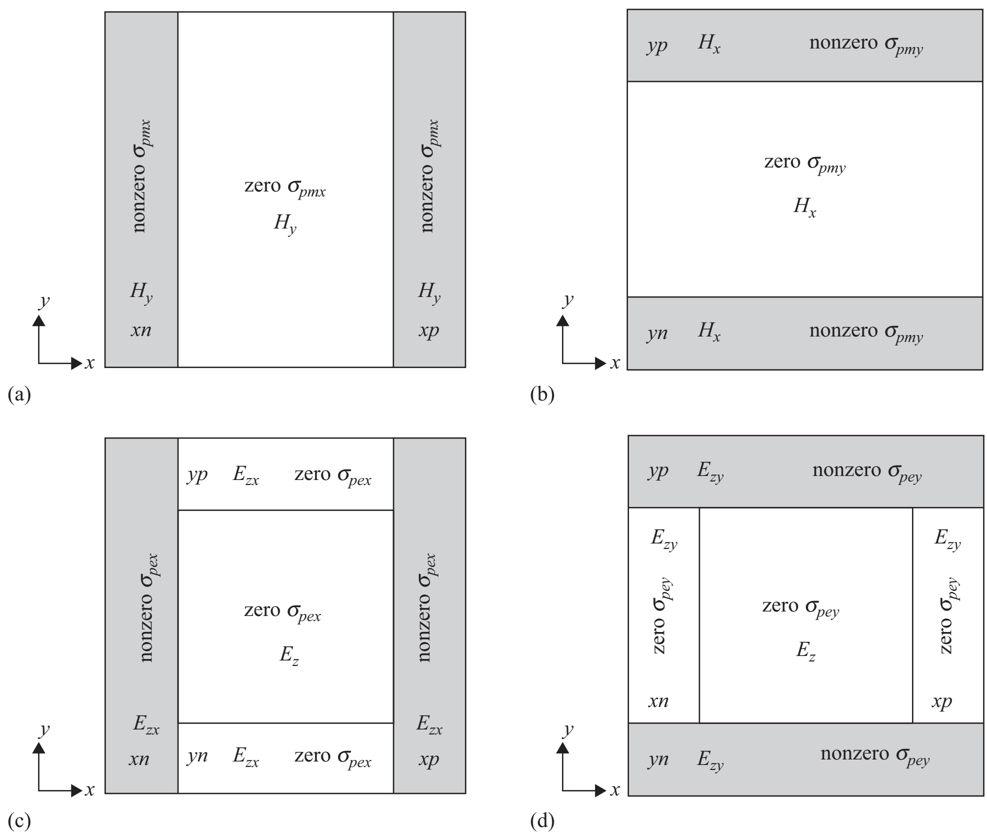


Figure 7.6 Nonzero TM_z regions of PML conductivities: (a) nonzero σ_{pmx}; (b) nonzero σ_{pmy}; (c) nonzero σ_{pex}; and (d) nonzero σ_{pey}.

denoted as xn and xp. Therefore, components of H_y lying in these regions are updated at every time step using the PML updating equation (7.38). The components of H_y lying in the intermediate region are updated using the regular updating equation (1.38). The conductivity σ_{pmy} is nonzero in the yn and yp regions shown in Figure 7.6(b); therefore, the components of H_x lying in the yn and yp regions are updated using the PML updating equation (7.37). The components of H_x lying in the intermediate region are updated using the regular updating equation (1.37).

The components of E_z are the sum of two split fields E_{zx} and E_{zy} in all of the PML regions xn, xp, yn, and yp as illustrated in Figure 7.6(c) and 7.6(d). The components of E_z in the intermediate non-PML region are updated using the regular updating equation (1.36). The conductivity σ_{pex} is nonzero in the xn and xp regions as shown in Figure 7.6(c); therefore, components of E_{zx} in these regions are updated using (7.35). Furthermore, the components of E_{zx} in the yn and yp regions need to be calculated as well. Since σ_{pex} is zero in these regions, setting σ_{pex} to zero in (7.35) yields the updating equation for E_{zx} in the yn and yp regions as

$$E_{zx}^{n+1}(i,j) = C_{ezxe}(i,j) \times E_{zx}^{n}(i,j) + C_{ezxhy}(i,j) \times \left(H_y^{n+\frac{1}{2}}(i,j) - H_y^{n+\frac{1}{2}}(i-1,j)\right), \quad (7.39)$$

where

$$C_{ezxe}(i,j) = 1, \quad C_{ezxhy}(i,j) = \frac{\Delta t}{\varepsilon_0 \Delta x}.$$

Similarly, σ_{pey} is nonzero in the yn and yp regions as shown in Figure 7.6(d); therefore, components of E_{zy} in these regions are updated using (7.36). Furthermore, the components of E_{zy} in the xn and xp regions need to be calculated as well. Since σ_{pey} is zero in these regions, setting σ_{pey} to zero in (7.36) yields the updating equation for E_{zy} in the xn and xp regions as

$$E_{zy}^{n+1}(i,j) = C_{ezye}(i,j) \times E_{z}^{n}(i,j) + C_{ezyhx}(i,j) \times \left(H_x^{n+\frac{1}{2}}(i,j) - H_x^{n+\frac{1}{2}}(i,j-1)\right), \quad (7.40)$$

where

$$C_{ezye}(i,j) = 1, \quad C_{ezyhx}(i,j) = -\frac{\Delta t}{\varepsilon_0 \Delta y}.$$

After all the components of E_{zx} and E_{zy} are updated in the PML regions, they are added to calculate E_z.

7.4.3 MATLAB® implementation of the two-dimensional FDTD method with PML

In this section we demonstrate the implementation of a two-dimensional FDTD MATLAB code including PML boundaries. The main routine of the two-dimensional program is named ***fdtd_solve_2d*** and is given in Listing 7.1. The general structure of the two-dimensional FDTD program is the same as the three-dimensional FDTD program; it is composed of problem definition, initialization, and execution sections. Many of the routines and notations of the two-dimensional FDTD program are similar to their three-dimensional FDTD counterparts; thus, the corresponding details are provided only when necessary.

Listing 7.1 fdtd_solve_2d.m

```
% initialize the matlab workspace
 clear all; close all; clc;

% define the problem
define_problem_space_parameters_2d;
define_geometry_2d;
define_sources_2d;
define_output_parameters_2d;

% initialize the problem space and parameters
initialize_fdtd_material_grid_2d;
initialize_fdtd_parameters_and_arrays_2d;
initialize_sources_2d;
initialize_updating_coefficients_2d;
initialize_boundary_conditions_2d;
initialize_output_parameters_2d;
initialize_display_parameters_2d;

% draw the objects in the problem space
draw_objects_2d;

% FDTD time marching loop
run_fdtd_time_marching_loop_2d;

% display simulation results
post_process_and_display_results_2d;
```

7.4.3.1 Definition of the two-dimensional FDTD problem

The types of boundaries surrounding the problem space are defined in the subroutine ***define_problem_space_parameters_2d***, a partial code of which is shown in Listing 7.2. Here if a boundary on one side is defined as PEC, the variable **boundary.type** takes the value "***pec***," whereas for PML it takes the value "***pml***." The parameter **air_buffer_number_of_cells** determines the distance in number of cells between the objects in the problem space and the boundaries, whether PEC or PML. The parameter **pml_number_of_cells** determines the thickness of the PML regions in number of cells. In this implementation the power increasing function (7.27a) is used for the PML conductivity distributions along the thickness of the PML regions. Two additional parameters are required for the PML, the theoretical reflection coefficient ($R(0)$) and the order of PML (n_{pml}), as discussed in Section 7.3. These two parameters are defined as **boundary_pml_R_0** and **boundary.pml_order**, respectively.

In the subroutine ***define_geometry_2d*** two-dimensional geometrical objects such as circles and rectangles can be defined by their coordinates, sizes, and material types.

The sources exciting the two-dimensional problem space are defined in the subroutine ***define_sources_2d***. Unlike the three-dimensional case, in this implementation the impressed current sources are defined as sources explicitly as shown in Listing 7.3.

Listing 7.2 define_problem_space_parameters_2d.m

```
% ==<boundary conditions>========
% Here we define the boundary conditions parameters
% 'pec' : perfect electric conductor
% 'pml' : perfectly matched layer

boundary.type_xn = 'pml';
boundary.air_buffer_number_of_cells_xn = 10;
boundary.pml_number_of_cells_xn = 5;

boundary.type_xp = 'pml';
boundary.air_buffer_number_of_cells_xp = 10;
boundary.pml_number_of_cells_xp = 5;

boundary.type_yn = 'pml';
boundary.air_buffer_number_of_cells_yn = 10;
boundary.pml_number_of_cells_yn = 5;

boundary.type_yp = 'pml';
boundary.air_buffer_number_of_cells_yp = 10;
boundary.pml_number_of_cells_yp = 5;

boundary.pml_order = 2;
boundary.pml_R_0 = 1e-8;
```

Listing 7.3 define_sources_2d.m

```
disp('defining sources');

impressed_J = [];
impressed_M = [];

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

% electric current sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
impressed_J(1).min_x = -0.1e-3;
impressed_J(1).min_y = -0.1e-3;
impressed_J(1).max_x = 0.1e-3;
impressed_J(1).max_y = 0.1e-3;
impressed_J(1).direction = 'zp';
impressed_J(1).magnitude = 1;
impressed_J(1).waveform_type = 'gaussian';
impressed_J(1).waveform_index = 1;

% % magnetic current sources
% % direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
```

```
% impressed_M(1).min_x = -0.1e-3;
% impressed_M(1).min_y = -0.1e-3;
% impressed_M(1).max_x = 0.1e-3;
% impressed_M(1).max_y = 0.1e-3;
% impressed_M(1).direction = 'zp';
% impressed_M(1).magnitude = 1;
% impressed_M(1).waveform_type = 'gaussian';
% impressed_M(1).waveform_index = 1;
```

Outputs of the program are defined in the subroutine ***define_output_parameters_2d***. In this implementation the output parameters are **sampled_electric_fields** and **sampled_magnetic_fields** captured at certain positions.

7.4.3.2 Initialization of the two-dimensional FDTD problem

The steps of the initialization process are shown in Listing 7.1. This process starts with the subroutine ***initialize_fdtd_material_grid_2d***. In this subroutine the first task is the calculation of the dimensions of the two-dimensional problem space and the number of cells *nx* and *ny* in the *x* and *y* dimensions, respectively. If some of the boundaries are defined as PML, the PML regions are also included in the problem space. Therefore, *nx* and *ny* include the PML number of cells as well. Then the material component arrays of the two-dimensional problem space are constructed using the material averaging schemes that were discussed in Section 3.2. Next, while creating the material grid, the PML regions are assigned free-space parameters. At this point the PML regions are treated as if they are free-space regions; however, later PML conductivity parameters are assigned to these regions, and special updating equations are used to update fields in these regions. Furthermore, new parameters are defined and assigned appropriate values as **n_pml_xn**, **n_pml_xp**, **n_pml_yn**, and **n_pml_yp** to hold the numbers of cells for the thickness of the PML regions. Four logical parameters, **is_pml_xn**, **is_pml_xp**, **is_pml_yn**, and **is_pml_yp**, are defined to indicate whether the respective side of the computational boundary is PML or not. Since these parameters are used frequently, shorthand notations are more appropriate to use. Figure 7.7 illustrates a problem space composed of two circles, two rectangles, an impressed current source at the center, and 5 cells thick PML boundaries. The air gap between the objects and the PML is 10 cells.

The subroutine ***initialize_fdtd_parameters_and_arrays_2d*** includes the definition of some parameters such as ε_0, μ_0, c, and Δt that are required for the FDTD calculation and definition and initialization of field arrays **Ex**, **Ey**, **Ez**, **Hx**, **Hy**, and **Hz**. The field arrays are defined for all the problem space including the PML regions.

In the subroutine ***initialize_sources_2d*** the indices indicating the positions of the impressed electric and magnetic currents are determined and stored as the subfields of the respective parameters **impressed_J** and **impressed_M**. Since the impressed currents are used as explicit sources, their source waveforms are also computed. A two-dimensional problem can have two modes of operation as *TE* and *TM*. The mode of operation is determined by the sources. For instance, an impressed electric current J_{iz} excites E_z fields in the problem space, thus giving rise to a TM_z operation. Similarly, impressed magnetic currents M_{ix} and M_{iy} also give rise to TM_z operation. The impressed currents M_{iz}, J_{ix}, and J_{iy} give rise to TE_z operation. Therefore, the mode of operation is determined by the impressed currents as shown in Listing 7.4, which includes the partial code of ***initialize_sources_2d***. Here a

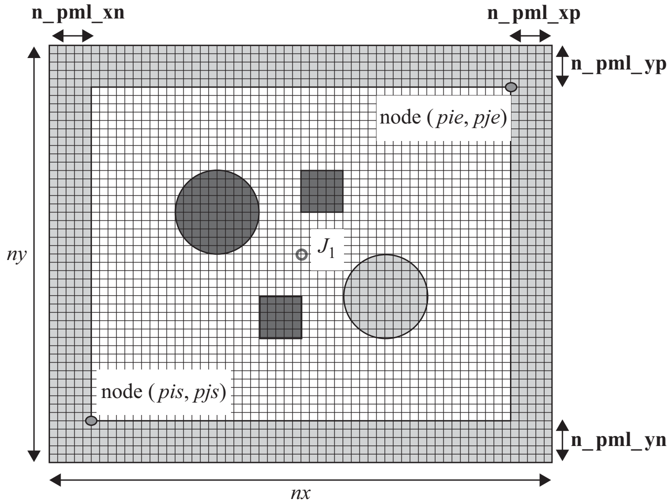


Figure 7.7 A two-dimensional FDTD problem space with PML boundaries.

Listing 7.4 initialize_sources_2d.m

```
% determine if TEz or TMz
is_TEz = false;
is_TMz = false;
for ind = 1:number_of_impressed_J
    switch impressed_J(ind).direction(1)
        case 'x'
            is_TEz = true;
        case 'y'
            is_TEz = true;
        case 'z'
            is_TMz = true;
    end
end
for ind = 1:number_of_impressed_M
    switch impressed_M(ind).direction(1)
        case 'x'
            is_TMz = true;
        case 'y'
            is_TMz = true;
        case 'z'
            is_TEz = true;
    end
end
```

logical parameter **is_TEz** is defined to indicate that the mode of operation is TE_z, whereas another logical parameter **is_TMz** is defined to indicate that the mode of operation is TM_z.

The regular updating coefficients of the two-dimensional FDTD method are calculated in the subroutine ***initialize_updating_coefficients_2d*** based on the updating equations (1.33)–(1.38), including the impressed current coefficients as well.

Some coefficients and field arrays need to be defined and initialized for the application of the PML boundary conditions. The PML initialization process is performed in the subroutine ***initialize_boundary_conditions_2d***, which is shown in Listing 7.5. The indices of the nodes determining the non-PML rectangular region as illustrated in Figure 7.7 are calculated as (*pis*, *pjs*) and (*pie*, *pje*). Then two separate subroutines dedicated to the initialization of the TE_z and TM_z cases are called based on the mode of operation.

The coefficients and fields required for the TE_z PML boundaries are initialized in the subroutine ***initialize_pml_boundary_conditions_2d_TEz***, and partial code for this case is shown in Listing 7.6. Figure 7.8 shows the field distribution for the TE_z case. The field components in the shaded region are updated by the PML updating equations. The field components on the outer boundary are not updated and are kept at zero value during the FDTD iterations since they are simulating the PEC boundaries. The magnetic field components H_{zx} and H_{zy} are defined in the four PML regions shown in Figure 7.5; therefore, the corresponding field arrays **Hzx_xn**, **Hzx_xp**, **Hzx_yn**, **Hzx_yp**, **Hzy_xn**, **Hzy_xp**, **Hzy_yn**, and **Hzy_yp** are initialized in Listing 7.6.

Then for each region the corresponding conductivity parameters and PML updating coefficients are calculated. Listing 7.6 shows the initialization for the *xn* and *yp* regions.

One should take care while calculating the conductivity arrays. The conductivities σ_{pex} and σ_{pey} are associated with the electric fields E_x and E_y, respectively, whereas σ_{pmx} and σ_{pmy} are associated with the magnetic field H_z. Therefore, the conductivity components are located in different positions. As mentioned before, the conductivity $\sigma(\rho)$ is zero at the PML interior-domain interface, and it increases to a maximum value σ_{max} at the end of the PML

Listing 7.5 initialize_boundary_conditions_2d.m

```
disp('initializing boundary conditions');

% determine the boundaries of the non-pml region
pis = n_pml_xn+1;
pie = nx-n_pml_xp+1;
pjs = n_pml_yn+1;
pje = ny-n_pml_yp+1;

if is_any_side_pml
    if is_TEz
        initialize_pml_boundary_conditions_2d_TEz;
    end
    if is_TMz
        initialize_pml_boundary_conditions_2d_TMz;
    end
end
```

Listing 7.6 initialize_pml_boundary_conditions_2d_TEz.m

```
% initializing PML boundary conditions for TEz
disp('initializing PML boundary conditions for TEz');

Hzx_xn = zeros(n_pml_xn,ny);
Hzy_xn = zeros(n_pml_xn,ny-n_pml_yn-n_pml_yp);
Hzx_xp = zeros(n_pml_xp,ny);
Hzy_xp = zeros(n_pml_xp,ny-n_pml_yn-n_pml_yp);
Hzx_yn = zeros(nx-n_pml_xn-n_pml_xp, n_pml_yn);
Hzy_yn = zeros(nx,n_pml_yn);
Hzx_yp = zeros(nx-n_pml_xn-n_pml_xp, n_pml_yp);
Hzy_yp = zeros(nx,n_pml_yp);

pml_order = boundary.pml_order;
R_0 = boundary.pml_R_0;

if is_pml_xn
    sigma_pex_xn = zeros(n_pml_xn,ny);
    sigma_pmx_xn = zeros(n_pml_xn,ny);

    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dx*n_pml_xn);
    rho_e = ([n_pml_xn:-1:1] - 0.75)/n_pml_xn;
    rho_m = ([n_pml_xn:-1:1] - 0.25)/n_pml_xn;
    for ind = 1:n_pml_xn
        sigma_pex_xn(ind,:) = sigma_max * rho_e(ind)^pml_order;
        sigma_pmx_xn(ind,:) = ...
            (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coeffiecients updating Ey
    Ceye_xn = (2*eps_0 - dt*sigma_pex_xn)./(2*eps_0+dt*sigma_pex_xn);
    Ceyhz_xn= -(2*dt/dx)./(2*eps_0 + dt*sigma_pex_xn);

    % Coeffiecients updating Hzx
    Chzxh_xn  = (2*mu_0 - dt*sigma_pmx_xn)./(2*mu_0+dt*sigma_pmx_xn);
    Chzxey_xn = -(2*dt/dx)./(2*mu_0 + dt*sigma_pmx_xn);

    % Coeffiecients updating Hzy
    Chzyh_xn  =  1;
    Chzyex_xn =  dt/(dy*mu_0);
end
if is_pml_yp
    sigma_pey_yp = zeros(nx,n_pml_yp);
    sigma_pmy_yp = zeros(nx,n_pml_yp);

    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dy*n_pml_yp);
    rho_e = ([1:n_pml_yp] - 0.75)/n_pml_yp;
    rho_m = ([1:n_pml_yp] - 0.25)/n_pml_yp;
    for ind = 1:n_pml_yp
        sigma_pey_yp(:,ind) = sigma_max * rho_e(ind)^pml_order;
        sigma_pmy_yp(:,ind) = ...
```

```
            (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coeffiecients updating Ex
    Cexe_yp  = (2*eps_0 - dt*sigma_pey_yp)./(2*eps_0+dt*sigma_pey_yp);
    Cexhz_yp = (2*dt/dy)./(2*eps_0 + dt*sigma_pey_yp);

    % Coeffiecients updating Hzx
    Chzxh_yp  = 1;
    Chzxey_yp = -dt/(dx*mu_0);

    % Coeffiecients updating Hzy
    Chzyh_yp  = (2*mu_0 - dt*sigma_pmy_yp)./(2*mu_0+dt*sigma_pmy_yp);
    Chzyex_yp = (2*dt/dy)./(2*mu_0 + dt*sigma_pmy_yp);
end
```

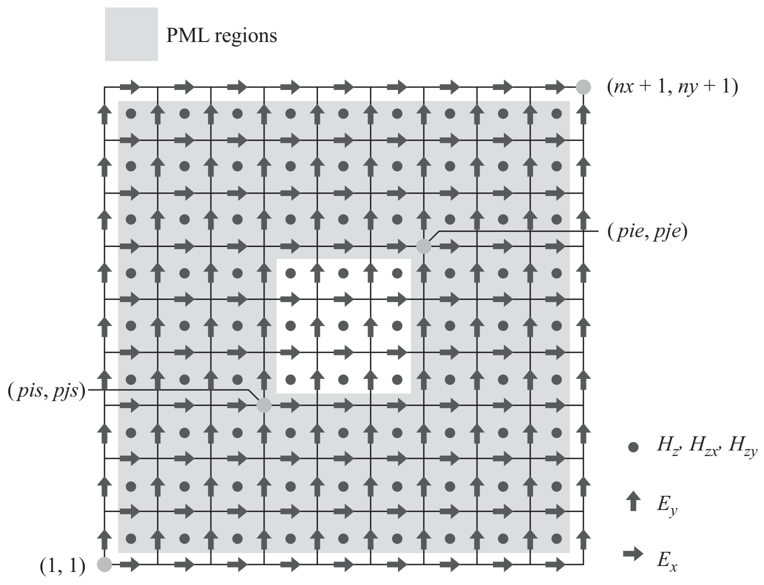


Figure 7.8 TE_z field components in the PML regions.

region. In this implementation the imaginary PML regions are shifted inward by a quarter cell size as shown in Figure 7.8. If the thickness of the PML region is N cells, this shift ensures that N electric field components and N magnetic field components are updated across the PML thickness. For instance, consider the cross-section of a two-dimensional problem space shown in Figure 7.9, which illustrates the field components updated in the *xn* and *xp* regions. The distance of the electric field components E_y from the interior boundary of the PML is denoted as ρ_e whereas for the magnetic field components H_z it is denoted as ρ_m. The distances ρ_e and ρ_m are used in (7.27a) to calculate the values of σ_{pex} and σ_{pmx}, respectively. These values are stored in arrays **sigma_pex_xn** and **sigma_pmx_xn** for the *xn* region as

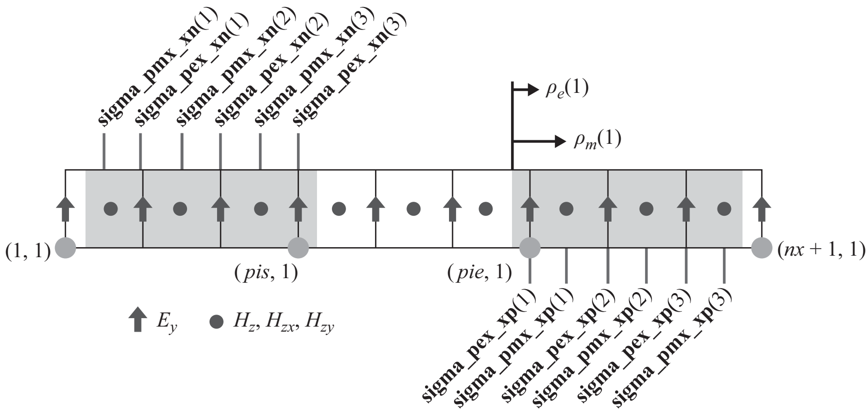


Figure 7.9 Field components updated by PML equations.

shown in Listing 7.6 and in **sigma_pex_xp** and **sigma_pmx_xp** for the *xp* region. Calculation of σ_{pey} and σ_{pmy} follows the same logic. Then these parameters are used to calculate the respective PML updating coefficients in (7.29)–(7.32).

The initialization of the auxiliary split fields and the PML updating coefficients for the two-dimensional TM_z case is performed in the subroutine ***initialize_pml_boundary_conditions_2d_Tmz***, a partial code of which is given for the *xp* and *yn* regions in Listing 7.7. The initialization of the TM_z case follows the same logic as the TE_z case. The field positioning and PML regions are shown in Figure 7.10, while the positions of the field components updated by the PML equations and conductivity positions are shown in Figure 7.11 as a reference.

7.4.3.3 Running the two-dimensional FDTD simulation: the time-marching loop

After the initialization process is completed, the subroutine ***run_fdtd_time_marching_loop_2d*** including the time-marching loop of FDTD procedure is called. The implementation of the FDTD updating loop is shown in Listing 7.8.

During the time-marching loop the first step at every iteration is the update of magnetic field components using the regular updating equations in ***update_magnetic_fields_2d*** as shown in Listing 7.9. In the TE_z case the H_z field components in the intermediate regions of Figure 7.5(c) and 7.5(d) are updated based on (1.35). In the TM_z case the H_x field components in the intermediate region of Figure 7.6(b) and H_y field components in the intermediate region of Figure 7.6(a) are updated based on (1.37) and (1.38), respectively.

Then in ***update_impressed_M*** the impressed current terms appearing in (1.35), (1.37), and (1.38) are added to their respective field terms H_z, H_x, and H_y as shown in Listing 7.10.

The subroutine ***update_magnetic_fields_for_PML_2d*** is used to update the magnetic field components needing special PML updates. As can be followed in Listing 7.11 the TE_z and TM_z cases are treated in separate subroutines.

The TM_z case is implemented in Listing 7.12, where H_x is updated in the *yn* and *yp* regions of Figure 7.6(b) using (7.37). H_y is updated in the *xn* and *xp* regions of Figure 7.6(a) using (7.38).

The TE_z case is implemented in Listing 7.13. H_{zx} is updated in the *xn* and *xp* regions of Figure 7.5(c) using (7.31). H_{zx} is updated in the *yn* and *yp* regions of Figure 7.5(c) using

Listing 7.7 initialize_pml_boundary_conditions_2d_TMz.m

```
% initializing PML boundary conditions for TMz
disp('initializing PML boundary conditions for TMz');

Ezx_xn = zeros(n_pml_xn,nym1);
Ezy_xn = zeros(n_pml_xn,nym1-n_pml_yn-n_pml_yp);
Ezx_xp = zeros(n_pml_xp,nym1);
Ezy_xp = zeros(n_pml_xp,nym1-n_pml_yn-n_pml_yp);
Ezx_yn = zeros(nxm1-n_pml_xn-n_pml_xp, n_pml_yn);
Ezy_yn = zeros(nxm1,n_pml_yn);
Ezx_yp = zeros(nxm1-n_pml_xn-n_pml_xp, n_pml_yp);
Ezy_yp = zeros(nxm1,n_pml_yp);

pml_order = boundary.pml_order;
R_0 = boundary.pml_R_0;

if is_pml_xp
    sigma_pex_xp = zeros(n_pml_xp,nym1);
    sigma_pmx_xp = zeros(n_pml_xp,nym1);

    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dx*n_pml_xp);
    rho_e = ([1:n_pml_xp] - 0.25)/n_pml_xp;
    rho_m = ([1:n_pml_xp] - 0.75)/n_pml_xp;
    for ind = 1:n_pml_xp
        sigma_pex_xp(ind,:) = sigma_max * rho_e(ind)^pml_order;
        sigma_pmx_xp(ind,:) = ...
            (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coeffiecients updating Hy
    Chyh_xp  = (2*mu_0 - dt*sigma_pmx_xp)./(2*mu_0 + dt*sigma_pmx_xp);
    Chyez_xp = (2*dt/dx)./(2*mu_0 + dt*sigma_pmx_xp);

    % Coeffiecients updating Ezx
    Cezxe_xp  = (2*eps_0 - dt*sigma_pex_xp)./(2*eps_0+dt*sigma_pex_xp);
    Cezxhy_xp = (2*dt/dx)./(2*eps_0 + dt*sigma_pex_xp);

    % Coeffiecients updating Ezy
    Cezye_xp  =  1;
    Cezyhx_xp = -dt/(dy*eps_0);
end

if is_pml_yn
    sigma_pey_yn = zeros(nxm1,n_pml_yn);
    sigma_pmy_yn = zeros(nxm1,n_pml_yn);

    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dy*n_pml_yn);
    rho_e = ([n_pml_yn:-1:1] - 0.25)/n_pml_yn;
    rho_m = ([n_pml_yn:-1:1] - 0.75)/n_pml_yn;
    for ind = 1:n_pml_yp
        sigma_pey_yn(:,ind) = sigma_max * rho_e(ind)^pml_order;
```

```
        sigma_pmy_yn(:,ind) = ...
            (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coeffiecients updating Hx
    Chxh_yn = (2*mu_0 - dt*sigma_pmy_yn)./(2*mu_0+dt*sigma_pmy_yn);
    Chxez_yn= -(2*dt/dy)./(2*mu_0 + dt*sigma_pmy_yn);

    % Coeffiecients updating Ezx
    Cezxe_yn  =  1;
    Cezxhy_yn =  dt/(dx*eps_0);

    % Coeffiecients updating Ezy
    Cezye_yn = (2*eps_0 - dt*sigma_pey_yn)./(2*eps_0+dt*sigma_pey_yn);
    Cezyhx_yn= -(2*dt/dy)./(2*eps_0 + dt*sigma_pey_yn);
end
```

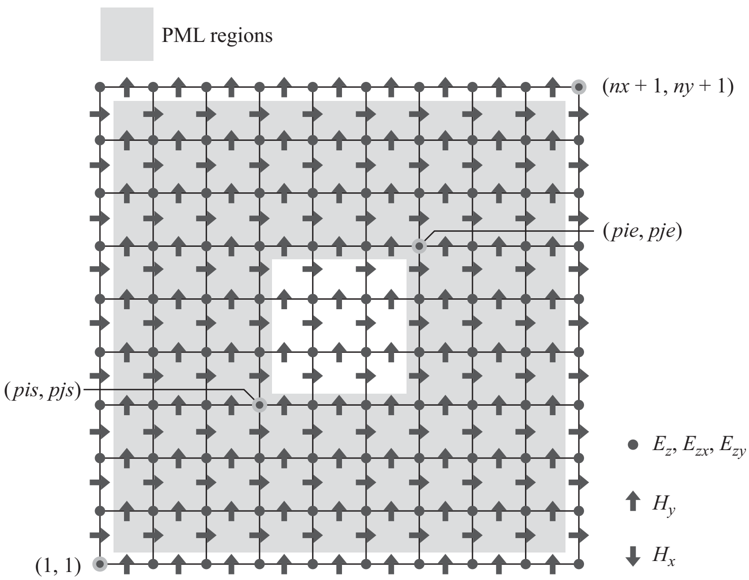


Figure 7.10 TM_z field components in the PML regions.

(7.33). H_{zy} is updated in the *yn* and *yp* regions of Figure 7.5(d) using (7.32). H_{zy} is updated in *xn* and *xp* regions of Figure 7.5(d) using (7.34). After all these updates are completed, the components of H_{zx} and H_{zy}, located at the same positions, are added to obtain H_z at the same positions.

The update of the electric field components using the regular updating equations is performed in ***update_electric_fields_2d*** as shown in Listing 7.14. In the TM_z case the E_z field components in the intermediate regions of Figure 7.6(c) and 7.6(d) are updated based on

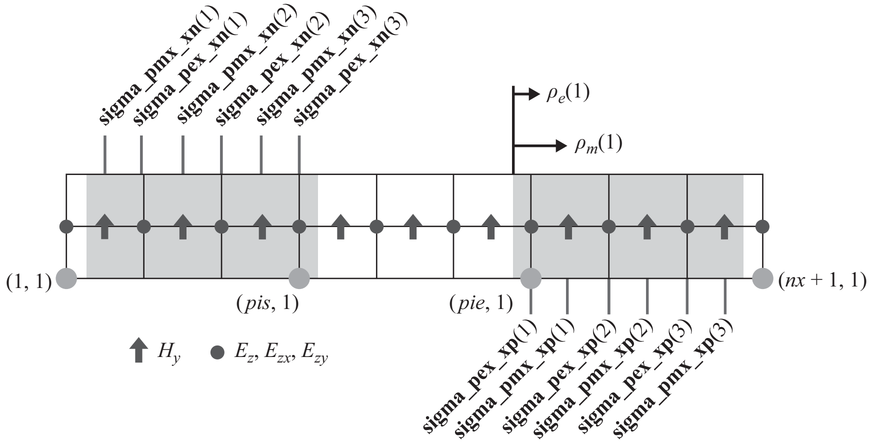


Figure 7.11 Field components updated by PML equations.

Listing 7.8 run_fdtd_time_marching_loop_2d.m

```
disp (['Starting the time marching loop']);
disp(['Total number of time steps : ' ...
    num2str(number_of_time_steps)]);

start_time = cputime;
current_time = 0;

for time_step = 1:number_of_time_steps
    update_magnetic_fields_2d;
    update_impressed_M;
    update_magnetic_fields_for_PML_2d;
    capture_sampled_magnetic_fields_2d;
    update_electric_fields_2d;
    update_impressed_J;
    update_electric_fields_for_PML_2d;
    capture_sampled_electric_fields_2d;
    display_sampled_parameters_2d;
end

end_time = cputime;
total_time_in_minutes = (end_time - start_time)/60;
disp(['Total simulation time is ' ...
    num2str(total_time_in_minutes) ' minutes.']);
```

(1.36). In the TE_z case the E_x field components in the intermediate region of Figure 7.5(b) and E_y field components in the intermediate region of Figure 7.5(a) are updated based on (1.33) and (1.34), respectively.

Then in ***update_impressed_J*** the impressed current terms appearing in (1.36), (1.33), and (1.34) are added to their respective field terms E_z, E_x, and E_y.

Listing 7.9 update_magnetic_fields_2d.m

```
% update magnetic fields

current_time = current_time + dt/2;

% TEz
if is_TEz
Hz(pis:pie-1,pjs:pje-1) = ...
    Chzh(pis:pie-1,pjs:pje-1).* Hz(pis:pie-1,pjs:pje-1) ...
    + Chzex(pis:pie-1,pjs:pje-1) ...
    .* (Ex(pis:pie-1,pjs+1:pje)-Ex(pis:pie-1,pjs:pje-1)) ...
    + Chzey(pis:pie-1,pjs:pje-1) ...
    .*(Ey(pis+1:pie,pjs:pje-1)-Ey(pis:pie-1,pjs:pje-1));
end

% TMz
if is_TMz
    Hx(:,pjs:pje-1) = Chxh(:,pjs:pje-1) .* Hx(:,pjs:pje-1) ...
        + Chxez(:,pjs:pje-1) .* (Ez(:,pjs+1:pje)-Ez(:,pjs:pje-1));

    Hy(pis:pie-1,:) = Chyh(pis:pie-1,:) .* Hy(pis:pie-1,:) ...
        + Chyez(pis:pie-1,:) .* (Ez(pis+1:pie,:)-Ez(pis:pie-1,:));
end
```

Listing 7.10 update_impressed_M.m

```
% updating magnetic field components
% associated with the impressed magnetic currents

 for ind = 1:number_of_impressed_M
    is = impressed_M(ind).is;
    js = impressed_M(ind).js;
    ie = impressed_M(ind).ie;
    je = impressed_M(ind).je;
    switch (impressed_M(ind).direction(1))
    case 'x'
        Hx(is:ie,js:je-1) = Hx(is:ie,js:je-1) ...
        + impressed_M(ind).Chxm * impressed_M(ind).waveform(time_step);
    case 'y'
        Hy(is:ie-1,js:je) = Hy(is:ie-1,js:je) ...
        + impressed_M(ind).Chym * impressed_M(ind).waveform(time_step);
    case 'z'
        Hz(is:ie-1,js:je-1) = Hz(is:ie-1,js:je-1) ...
        + impressed_M(ind).Chzm * impressed_M(ind).waveform(time_step);
    end
end
```

Listing 7.11 update_magnetic_fields_for_PML_2d.m

```
% update magnetic fields at the PML regions
if is_any_side_pml == false
    return;
end
if is_TEz
    update_magnetic_fields_for_PML_2d_TEz;
end
if is_TMz
    update_magnetic_fields_for_PML_2d_TMz;
end
```

Listing 7.12 update_magnetic_fields_for_PML_2d_TMz.m

```
% update magnetic fields at the PML regions
% TMz
if is_pml_xn
    Hy(1:pis-1,2:ny) = Chyh_xn .* Hy(1:pis-1,2:ny) ...
        + Chyez_xn .* (Ez(2:pis,2:ny)-Ez(1:pis-1,2:ny));
end

if is_pml_xp
    Hy(pie:nx,2:ny) = Chyh_xp .* Hy(pie:nx,2:ny) ...
    + Chyez_xp .* (Ez(pie+1:nxp1,2:ny)-Ez(pie:nx,2:ny));
end

if is_pml_yn
    Hx(2:nx,1:pjs-1) = Chxh_yn .* Hx(2:nx,1:pjs-1) ...
        + Chxez_yn.*(Ez(2:nx,2:pjs)-Ez(2:nx,1:pjs-1));
end

if is_pml_yp
    Hx(2:nx,pje:ny) = Chxh_yp .* Hx(2:nx,pje:ny) ...
        + Chxez_yp.*(Ez(2:nx,pje+1:nyp1)-Ez(2:nx,pje:ny));
end
```

The subroutine ***update_electric_fields_for_PML_2d*** is used to update the electric field components needing special PML updates. As can be followed in Listing 7.15 the TE_z and TM_z cases are treated in separate subroutines.

The TE_z case is implemented in Listing 7.16, where E_x is updated in the *yn* and *yp* regions of Figure 7.5(b) using (7.29). E_y is updated in the *xn* and *xp* regions of Figure 7.5(a) using (7.30).

The TM_z case is implemented in Listing 7.17, where E_{zx} is updated in the *xn* and *xp* regions of Figure 7.6(c) using (7.35). E_{zx} is updated in the *yn* and *yp* regions of Figure 7.6(c) using (7.39). E_{zy} is updated in the *yn* and *yp* regions of Figure 7.6(d) using (7.36). E_{zy} is updated in the *xn* and *xp* regions of Figure 7.6(d) using (7.40). After all these updates are completed, the components of E_{zx} and E_{zy}, located at the same positions, are added to obtain E_z at the same positions.

Listing 7.13 update_magnetic_fields_for_PML_2d_TEz.m

```
% update magnetic fields at the PML regions
% TEz
if is_pml_xn
    Hzx_xn = Chzxh_xn .* Hzx_xn+Chzxey_xn.*(Ey(2:pis,:) - Ey(1:pis-1,:));
    Hzy_xn = Chzyh_xn .* Hzy_xn ...
        + Chzyex_xn.*(Ex(1:pis-1,pjs+1:pje)-Ex(1:pis-1,pjs:pje-1));
end
if is_pml_xp
    Hzx_xp = Chzxh_xp .* Hzx_xp ...
        + Chzxey_xp.*(Ey(pie+1:nxp1,:) - Ey(pie:nx,:));
    Hzy_xp = Chzyh_xp .* Hzy_xp ...
        + Chzyex_xp.*(Ex(pie:nx,pjs+1:pje)-Ex(pie:nx,pjs:pje-1));
end
if is_pml_yn
    Hzx_yn = Chzxh_yn .* Hzx_yn ...
        + Chzxey_yn.*(Ey(pis+1:pie,1:pjs-1)-Ey(pis:pie-1,1:pjs-1));
    Hzy_yn = Chzyh_yn .* Hzy_yn ...
        + Chzyex_yn.*(Ex(:,2:pjs)-Ex(:,1:pjs-1));
end
if is_pml_yp
    Hzx_yp = Chzxh_yp .* Hzx_yp ...
        + Chzxey_yp.*(Ey(pis+1:pie,pje:ny)-Ey(pis:pie-1,pje:ny));
    Hzy_yp = Chzyh_yp .* Hzy_yp ...
        + Chzyex_yp.*(Ex(:,pje+1:nyp1)-Ex(:,pje:ny));
end
Hz(1:pis-1,1:pjs-1) = Hzx_xn(:,1:pjs-1) + Hzy_yn(1:pis-1,:);
Hz(1:pis-1,pje:ny)  = Hzx_xn(:,pje:ny) + Hzy_yp(1:pis-1,:);
Hz(pie:nx,1:pjs-1)  = Hzx_xp(:,1:pjs-1) + Hzy_yn(pie:nx,:);
Hz(pie:nx,pje:ny)   = Hzx_xp(:,pje:ny) + Hzy_yp(pie:nx,:);
Hz(1:pis-1,pjs:pje-1) = Hzx_xn(:,pjs:pje-1) + Hzy_xn;
Hz(pie:nx,pjs:pje-1)  = Hzx_xp(:,pjs:pje-1) + Hzy_xp;
Hz(pis:pie-1,1:pjs-1) = Hzx_yn + Hzy_yn(pis:pie-1,:);
Hz(pis:pie-1,pje:ny)  = Hzx_yp + Hzy_yp(pis:pie-1,:);
```

Listing 7.14 update_electric_fields_2d.m

```
current_time = current_time + dt/2;

if is_TEz
    Ex(:,pjs+1:pje-1) = Cexe(:,pjs+1:pje-1).*Ex(:,pjs+1:pje-1) ...
                        + Cexhz(:,pjs+1:pje-1).*...
                        (Hz(:,pjs+1:pje-1)-Hz(:,pjs:pje-2));

    Ey(pis+1:pie-1,:) = Ceye(pis+1:pie-1,:).*Ey(pis+1:pie-1,:) ...
                        + Ceyhz(pis+1:pie-1,:).* ...
                        (Hz(pis+1:pie-1,:)-Hz(pis:pie-2,:));
end
```

```
if is_TMz
    Ez(pis+1:pie-1,pjs+1:pje-1) = ...
        Ceze(pis+1:pie-1,pjs+1:pje-1).*Ez(pis+1:pie-1,pjs+1:pje-1) ...
        + Cezhy(pis+1:pie-1,pjs+1:pje-1) ...
        .* (Hy(pis+1:pie-1,pjs+1:pje-1)-Hy(pis:pie-2,pjs+1:pje-1)) ...
        + Cezhx(pis+1:pie-1,pjs+1:pje-1) ...
        .* (Hx(pis+1:pie-1,pjs+1:pje-1)-Hx(pis+1:pie-1,pjs:pje-2));
end
```

Listing 7.15 update_electric_fields_for_PML_2d.m

```
% update electric fields at the PML regions
% update magnetic fields at the PML regions
if is_any_side_pml == false
    return;
end
if is_TEz
    update_electric_fields_for_PML_2d_TEz;
end
if is_TMz
    update_electric_fields_for_PML_2d_TMz;
end
```

Listing 7.16 update_electric_fields_for_PML_2d_TEz.m

```
% update electric fields at the PML regions
% TEz
if is_pml_xn
    Ey(2:pis,:) = Ceye_xn .* Ey(2:pis,:) ...
        + Ceyhz_xn .* (Hz(2:pis,:)-Hz(1:pis-1,:));
end

if is_pml_xp
    Ey(pie:nx,:) = Ceye_xp .* Ey(pie:nx,:) ...
        + Ceyhz_xp .* (Hz(pie:nx,:)-Hz(pie-1:nx-1,:));
end

if is_pml_yn
    Ex(:,2:pjs) = Cexe_yn .* Ex(:,2:pjs) ...
        + Cexhz_yn .* (Hz(:,2:pjs)-Hz(:,1:pjs-1));
end

if is_pml_yp
    Ex(:,pje:ny) = Cexe_yp .* Ex(:,pje:ny) ...
        + Cexhz_yp .* (Hz(:,pje:ny)-Hz(:,pje-1:ny-1));
end
```

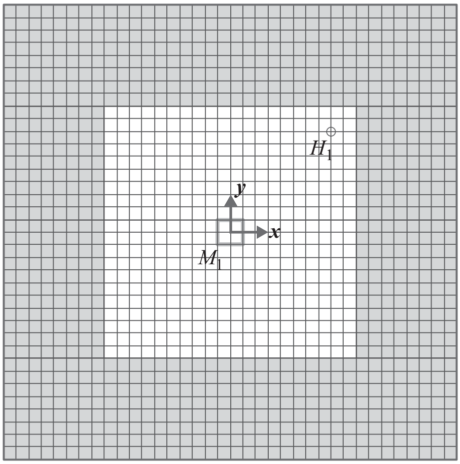
Listing 7.17 update_electric_fields_for_PML_2d_TMz.m

```
% update electric fields at the PML regions
% TMz
if is_pml_xn
    Ezx_xn = Cezxe_xn .* Ezx_xn ...
        + Cezxhy_xn .* (Hy(2:pis,2:ny)-Hy(1:pis-1,2:ny));
    Ezy_xn = Cezye_xn .* Ezy_xn ...
        + Cezyhx_xn .* (Hx(2:pis,pjs+1:pje-1)-Hx(2:pis,pjs:pje-2));
end
if is_pml_xp
    Ezx_xp = Cezxe_xp .* Ezx_xp + Cezxhy_xp.* ...
        (Hy(pie:nx,2:ny)-Hy(pie-1:nx-1,2:ny));
    Ezy_xp = Cezye_xp .* Ezy_xp ...
        + Cezyhx_xp .* (Hx(pie:nx,pjs+1:pje-1)-Hx(pie:nx,pjs:pje-2));
end
if is_pml_yn
    Ezx_yn = Cezxe_yn .* Ezx_yn ...
        + Cezxhy_yn .* (Hy(pis+1:pie-1,2:pjs)-Hy(pis:pie-2,2:pjs));
    Ezy_yn = Cezye_yn .* Ezy_yn ...
        + Cezyhx_yn .* (Hx(2:nx,2:pjs)-Hx(2:nx,1:pjs-1));
end
if is_pml_yp
    Ezx_yp = Cezxe_yp .* Ezx_yp ...
        + Cezxhy_yp .* (Hy(pis+1:pie-1,pje:ny)-Hy(pis:pie-2,pje:ny));
    Ezy_yp = Cezye_yp .* Ezy_yp ...
        + Cezyhx_yp .* (Hx(2:nx,pje:ny)-Hx(2:nx,pje-1:ny-1));
end
Ez(2:pis,2:pjs)   = Ezx_xn(:,1:pjs-1) + Ezy_yn(1:pis-1,:);
Ez(2:pis,pje:ny)  = Ezx_xn(:,pje-1:nym1) + Ezy_yp(1:pis-1,:);
Ez(pie:nx,pje:ny) = Ezx_xp(:,pje-1:nym1) + Ezy_yp(pie-1:nxm1,:);
Ez(pie:nx,2:pjs)  = Ezx_xp(:,1:pjs-1) + Ezy_yn(pie-1:nxm1,:);
Ez(pis+1:pie-1,2:pjs)  = Ezx_yn + Ezy_yn(pis:pie-2,:);
Ez(pis+1:pie-1,pje:ny) = Ezx_yp + Ezy_yp(pis:pie-2,:);
Ez(2:pis,pjs+1:pje-1)  = Ezx_xn(:,pjs:pje-2) + Ezy_xn;
Ez(pie:nx,pjs+1:pje-1) = Ezx_xp(:,pjs:pje-2) + Ezy_xp;
```

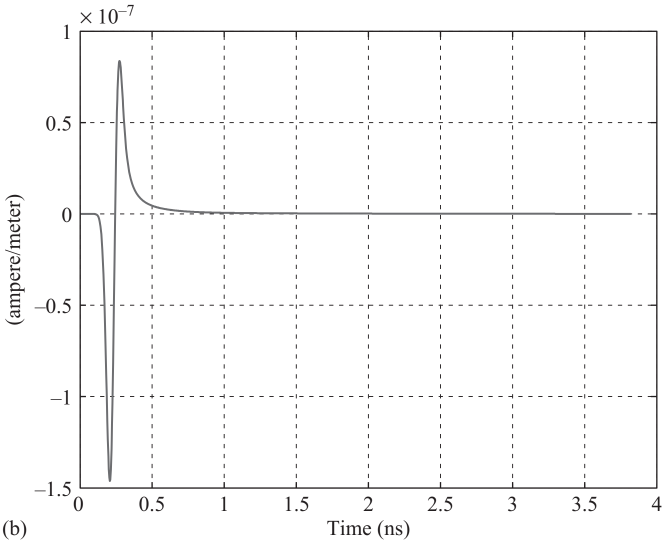
7.5 Simulation examples

7.5.1 Validation of PML performance

In this section, we evaluate the performance of the two-dimensional PML for the TE_z case. A two-dimensional problem is constructed as shown in Figure 7.12(a). The problem space is empty (all free space) and is composed of 36×36 cells with cell size 1 mm on a side. There are eight cell layers of PML on the four sides of the boundaries. The order of the PML parameter n_{pml} is 2, and the theoretical reflection coefficient $R(0)$ is 10^{-8}. The problem space is excited by a z-directed impressed magnetic current as defined in Listing 7.18. The impressed magnetic current is centered at the origin and has a Gaussian waveform. A sampled magnetic field with z component is placed at the position $x = 8$ mm and $y = 8$ mm, two cells away from the upper right corner of the PML boundaries as defined in Listing 7.19. The problem is run for 1,800 time steps. The captured sampled magnetic field H_z is plotted in



(a)



(b)

Figure 7.12 A two-dimensional TE_z FDTD problem terminated by PML boundaries and its simulation results: (a) an empty two-dimensional problem space and (b) sampled H_z in time.

Figure 7.12(b) as a function of time. The captured field includes the effect of the reflected fields from the PML boundaries as well.

To determine how well the PML boundaries simulate the open boundaries, a reference case is constructed as shown in Figure 7.13(a). The cell size, the source, and the output of this problem space are the same as the previous one, but in this case the problem space size is 600×600 cells, and it is terminated by PEC boundaries on four sides. Any fields excited by the source will propagate, will hit the PEC boundaries, and will propagate back to the center. Since the problem size is large, it will take some time until the reflected fields arrive at the

Listing 7.18 define_sources_2d.m

```
disp('defining␣sources');

impressed_J = [];
impressed_M = [];

% define source waveform types and parameters
waveforms.gaussian(1).number_of_cells_per_wavelength = 0;
waveforms.gaussian(2).number_of_cells_per_wavelength = 25;

% magnetic current sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
impressed_M(1).min_x = -1e-3;
impressed_M(1).min_y = -1e-3;
impressed_M(1).max_x = 1e-3;
impressed_M(1).max_y = 1e-3;
impressed_M(1).direction = 'zp';
impressed_M(1).magnitude = 1;
impressed_M(1).waveform_type = 'gaussian';
impressed_M(1).waveform_index = 2;
```

Listing 7.19 define_output_parameters_2d.m

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

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_transient_E_planes = [];
sampled_frequency_E_planes = [];

% figure refresh rate
plotting_step = 10;

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

% define sampled magnetic fields
sampled_magnetic_fields(1).x = 8e-3;
sampled_magnetic_fields(1).y = 8e-3;
sampled_magnetic_fields(1).component = 'z';
```

sampling point. Therefore, the fields captured at the sampling point before any reflected fields arrive are the same as the fields that would be observed if the boundaries are open space. In the given example no reflection is observed in the 1,800 time steps of simulation. Therefore, this case can be considered as a reference case for an open space during the 1,800 time steps. The captured sampled magnetic field is shown in Figure 7.13(b) as a function of time.

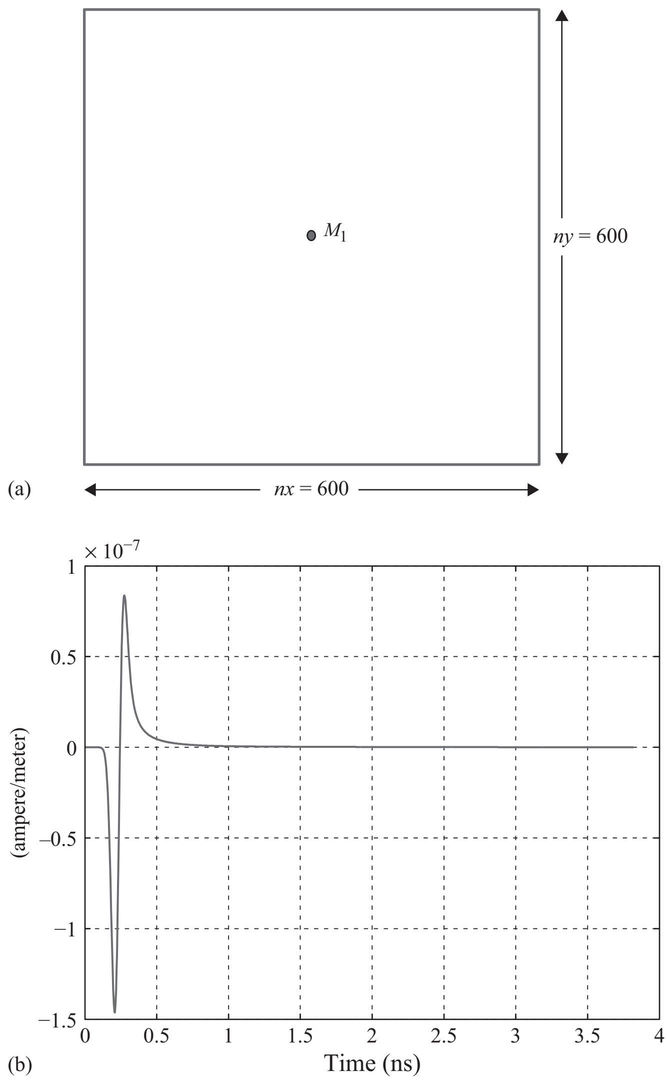


Figure 7.13 A two-dimensional TE_z FDTD problem used as open boundary reference and its simulation results: (a) an empty two-dimensional problem space and (b) sampled H_z in time.

No difference can be seen by looking at the responses of the PML case and the reference case. The difference between the two cases is a measure for the amount of reflection from the PML and can be determined numerically. The difference denoted as $error_t$ is the error as a function of time and calculated by

$$error_t = 20 \times \log_{10}\left(\frac{|H_z^{pml} - H_z^{ref}|}{max\left(|H_z^{ref}|\right)}\right), \tag{7.41}$$

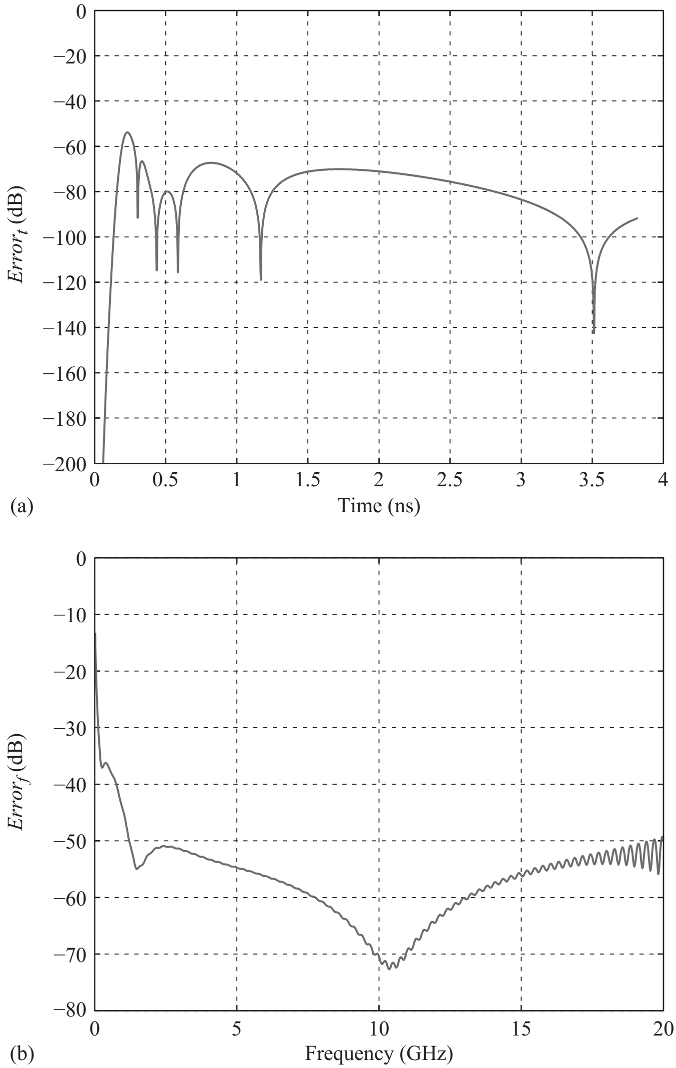


Figure 7.14 Error in time and frequency domains.

where H_z^{pml} is the sampled magnetic field in the PML problem case and H_z^{ref} is the sampled magnetic field in the reference case. The error as a function of time is plotted in Figure 7.14(a). The error in frequency domain as well is obtained as $error_f$ from the difference between the Fourier transforms of the sampled magnetic fields from the PML and reference cases by

$$error_f = 20 \times \log_{10}\left(\frac{|F(H_z^{pml}) - F(H_z^{ref})|}{F\left(|H_z^{ref}|\right)}\right), \tag{7.42}$$

where the operator $F(\cdot)$ denotes the Fourier transform. The error in frequency-domain $error_f$ is plotted in Figure 7.14(b). The errors obtained in this example can further be reduced by using a larger number of cells of PML thickness, a better choice of $R(0)$, and a higher order of PML n_{pml}. One can see in Figure 7.14(b) that the performance of the PML degrades at low frequencies.

7.5.2 Electric field distribution

Since the fields are calculated on a plane by the two-dimensional FDTD program, it is possible to capture and display the electric and magnetic field distributions as a runtime animation while the simulation is running. Furthermore, it is possible to calculate the field distribution as a response of a time-harmonic excitation at predefined frequencies. Then the time-harmonic field distribution can be compared with results obtained from simulation of the same problem using frequency-domain solvers. In this example the two-dimensional FDTD program is used to calculate the electric field distribution in a problem space including a cylinder of circular cross-section with radius 0.2 m, and dielectric constant 4, due to a current line source placed 0.2 m away from the cylinder and excited at 1 GHz frequency. Figure 7.15 illustrates the geometry of the two-dimensional problem space. The problem space is composed of square cells with 5 mm on a side and is terminated by PML boundaries with 8 cells thickness. The air gap between the cylinder and the boundaries is 30 cells in the *xn*, *yn*, and *yp* directions and 80 cells in the *xp* direction. The definition of the geometry is shown in Listing 7.20. The line source is an impressed current density with a sinusoidal waveform as shown in Listing 7.21.

In this example we define two new output types: (1) transient electric field distributions represented with a parameter named **sampled_transient_E_planes**; and (2) electric field distributions calculated at certain frequencies represented with a parameter named **sampled_frequency_E_planes**. The definition of these parameters is shown in Listing 7.22, and the initialization of these parameters are performed in the subroutine ***initialize_output_parameters_2d*** is shown in Listing 7.23.

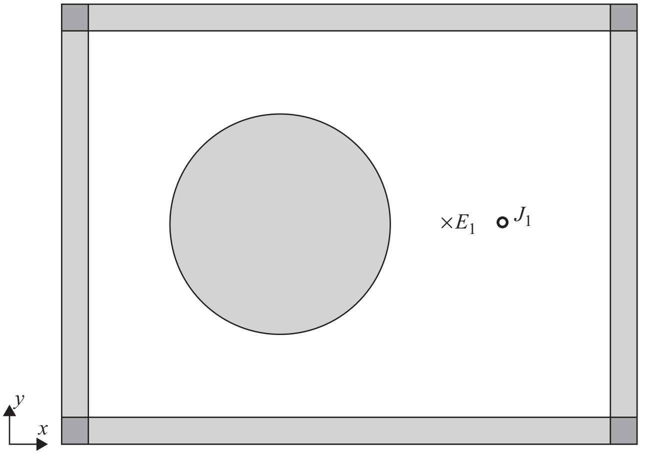


Figure 7.15 A two-dimensional problem space including a cylinder and a line source.

Listing 7.20 define_geometry_2d.m

```
% define a circle
circles(1).center_x = 0.4;
circles(1).center_y = 0.5;
circles(1).radius   = 0.2;
circles(1).material_type = 4;
```

Listing 7.21 define_sources_2d.m

```
waveforms.sinusoidal(1).frequency = 1e9;

% electric current sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
impressed_J(1).min_x = 0.8;
impressed_J(1).min_y = 0.5;
impressed_J(1).max_x = 0.8;
impressed_J(1).max_y = 0.5;
impressed_J(1).direction = 'zp';
impressed_J(1).magnitude = 1;
impressed_J(1).waveform_type = 'sinusoidal';
impressed_J(1).waveform_index = 1;
```

Listing 7.22 define_output_parameters_2d.m

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

sampled_electric_fields = [];
sampled_magnetic_fields = [];
sampled_transient_E_planes = [];
sampled_frequency_E_planes = [];
% define sampled electric field distributions
% component can be 'x', 'y', 'z', or 'm' (magnitude)
% transient
sampled_transient_E_planes(1).component = 'z';

% frequency domain
sampled_frequency_E_planes(1).component = 'z';
sampled_frequency_E_planes(1).frequency = 1e9;
```

The electric fields at node positions are captured and displayed as an animation while the simulation is running in the subroutine ***display_sampled_parameters_2d*** as shown in Listing 7.24.

The electric fields at node positions are captured and calculated as the *frequency-domain* response at the given frequency in the subroutine ***capture_sampled_electric_fields_2d*** as shown in Listing 7.25. One should notice in the given code that the fields are being captured

Listing 7.23 initialize_output_parameters_2d.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_transient_E_planes=size(sampled_transient_E_planes,2);
number_of_sampled_frequency_E_planes=size(sampled_frequency_E_planes,2);
% initialize sampled transient electric field
for ind=1:number_of_sampled_transient_E_planes
    sampled_transient_E_planes(ind).figure = figure;
end

% initialize sampled time harmonic electric field
for ind=1:number_of_sampled_frequency_E_planes
    sampled_frequency_E_planes(ind).sampled_field = zeros(nxp1,nyp1);
end
xcoor = linspace(fdtd_domain.min_x,fdtd_domain.max_x,nxp1);
ycoor = linspace(fdtd_domain.min_y,fdtd_domain.max_y,nyp1);
```

Listing 7.24 display_sampled_parameters_2d.m

```
% display sampled electric field distribution
for ind=1:number_of_sampled_transient_E_planes
    figure(sampled_transient_E_planes(ind).figure);
    Es = zeros(nxp1, nyp1);
    component = sampled_transient_E_planes(ind).component;
    switch (component)
        case 'x'
            Es(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
        case 'y'
            Es(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
        case 'z'
            Es = Ez;
        case 'm'
            Exs(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
            Eys(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
            Ezs = Ez;
            Es = sqrt(Exs.^2 + Eys.^2 + Ezs.^2);
    end
    imagesc(xcoor,ycoor,Es.');
    axis equal; axis xy; colorbar;
    title(['Electric field <' component '>[' num2str(ind) ']']);
    drawnow;
end
```

after 6,000 time steps. Here it is assumed that the time-domain response of the sinusoidal excitation has reached the steady state after 6,000 time steps. Then the magnitude of the steady fields is captured. Therefore, with the given code it is possible to capture the magnitude of the frequency-domain response and only at the single excitation frequency. The given code cannot

Listing 7.25 capture_sampled_electric_fields_2d.m

```
% capture sampled time harmonic electric fields on a plane
if time_step >6000
    for ind=1:number_of_sampled_frequency_E_planes
        Es = zeros(nxp1, nyp1);
        component = sampled_frequency_E_planes(ind).component;
        switch (component)
            case 'x'
                Es(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
            case 'y'
                Es(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
            case 'z'
                Es = Ez;
            case 'm'
                Exs(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
                Eys(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
                Ezs = Ez;
                Es = sqrt(Exs.^2 + Eys.^2 + Ezs.^2);
        end
        I = find(Es > sampled_frequency_E_planes(ind).sampled_field);
        sampled_frequency_E_planes(ind).sampled_field(I) = Es(I);
    end
end
```

Listing 7.26 display_frequency_domain_outputs_2d.m

```
% display sampled time harmonic electric fields on a plane
for ind=1:number_of_sampled_frequency_E_planes
    figure;
    f = sampled_frequency_E_planes(ind).frequency;
    component = sampled_frequency_E_planes(ind).component;
    Es = abs(sampled_frequency_E_planes(ind).sampled_field);
    Es = Es/max(max(Es));
    imagesc(xcoor,ycoor,Es.');
    axis equal; axis xy; colorbar;
    title(['Electric␣field␣at␣f␣=␣' ...
        num2str(f*1e-9) '␣GHz,␣<' component '>[' num2str(ind) ']']);
    drawnow;
end
```

calculate the *phase* of the response. The given algorithm can further be improved to calculate the phases as well; however, in the following example, a more efficient method based on discrete Fourier transform (DFT) is presented, which can be used to calculate both the magnitude and phase responses concurrently. After the simulation is completed, the calculated magnitude response can be plotted using the code shown in Listing 7.26.

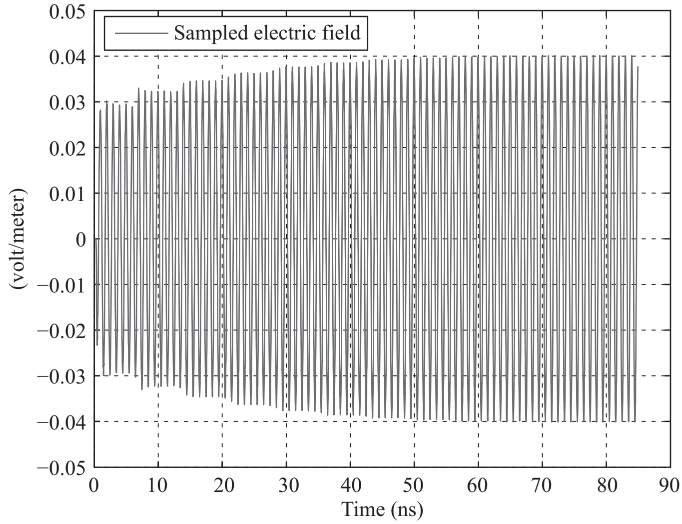


Figure 7.16 Sampled electric field at a point between the cylinder and the line source.

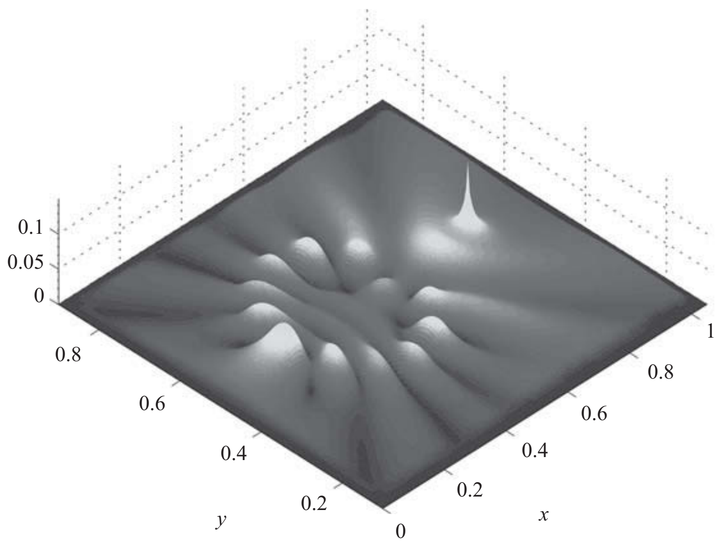


Figure 7.17 Magnitude of electric field distribution calculated by FDTD.

The two-dimensional FDTD program is excited for 8,000 time steps, and the transient electric field is sampled at a point between the cylinder and the line source as shown in Figure 7.15. The sampled electric field is plotted in Figure 7.16, which shows that the simulation has reached the steady state after 50 ns. Furthermore, the *magnitude* of the electric field distribution is captured for 1 GHz as discussed already is shown in Figure 7.17 as a surface plot. The same problem is solved using boundary value solution (BVS) [22], and

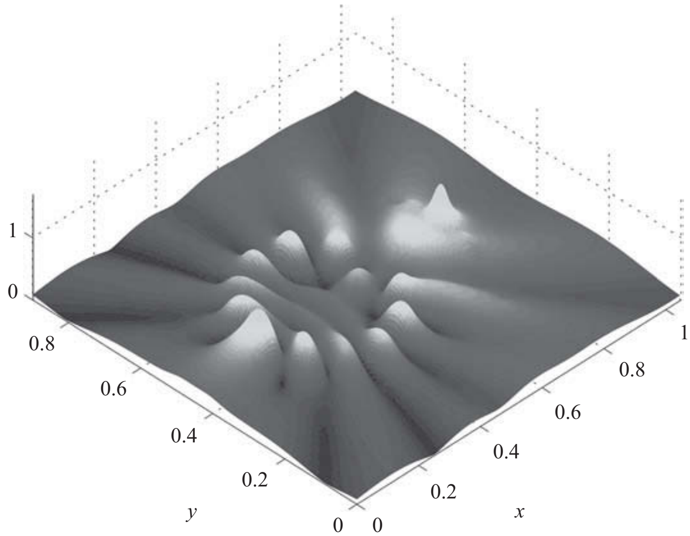


Figure 7.18 Magnitude of electric field distribution calculated by BVS.

the result is shown in Figure 7.18 for comparison. It can be seen that the results agree very well. The levels of the magnitudes are different since these figures are normalized to different values.

7.5.3 Electric field distribution using DFT

The previous example demonstrated how the magnitude of electric field distribution can be calculated as a response of a time-harmonic excitation. As discussed before, it is only possible to obtain results for a single frequency with the given technique. However, if the excitation is a waveform including a spectrum of frequencies, then it should be possible to obtain results for multiple frequencies using DFT. We modify the previous example to be able to calculate field distributions for multiple frequencies.

The output for the field distribution is defined in the subroutine ***define_output_parameters*** as shown in Listing 7.27. One can notice that it is possible to define multiple field distributions with different frequencies. The excitation waveform as well shall include the desired frequencies in its spectrum. An impressed current line source is defined as shown in Listing 7.28. To calculate field distributions for multiple frequencies we have to implement an

Listing 7.27 define_output_parameters_2d.m

```
% frequency domain
sampled_frequency_E_planes(1).component = 'z';
sampled_frequency_E_planes(1).frequency = 1e9;
sampled_frequency_E_planes(2).component = 'z';
sampled_frequency_E_planes(2).frequency = 2e9;
```

Listing 7.28 define_sources_2d.m

```
% define source waveform types and parameters
waveforms.gaussian(1).number_of_cells_per_wavelength = 0;
waveforms.gaussian(2).number_of_cells_per_wavelength = 20;
waveforms.sinusoidal(1).frequency = 1e9;

% magnetic current sources
% direction: 'xp', 'xn', 'yp', 'yn', 'zp', or 'zn'
impressed_J(1).min_x = 0.8;
impressed_J(1).min_y = 0.5;
impressed_J(1).max_x = 0.8;
impressed_J(1).max_y = 0.5;
impressed_J(1).direction = 'zp';
impressed_J(1).magnitude = 1;
impressed_J(1).waveform_type = 'gaussian';
impressed_J(1).waveform_index = 2;
```

Listing 7.29 capture_sampled_electric_fields_2d.m

```
% capture sampled time harmonic electric fields on a plane
for ind=1:number_of_sampled_frequency_E_planes
    w = 2 * pi * sampled_frequency_E_planes(ind).frequency;
    Es = zeros(nxp1, nyp1);
    component = sampled_frequency_E_planes(ind).component;
    switch (component)
        case 'x'
            Es(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
        case 'y'
            Es(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
        case 'z'
            Es = Ez;
        case 'm'
            Exs(2:nx,:) = 0.5 * (Ex(1:nx-1,:) + Ex(2:nx,:));
            Eys(:,2:ny) = 0.5 * (Ey(:,1:ny-1) + Ey(:,2:ny));
            Ezs = Ez;
            Es = sqrt(Exs.^2 + Eys.^2 + Ezs.^2);
    end
    sampled_frequency_E_planes(ind).sampled_field = ...
        sampled_frequency_E_planes(ind).sampled_field ...
        + dt * Es * exp(-j*w*dt*time_step);
end
```

on-the-fly DFT. Therefore, the subroutine ***capture_sampled_electric_fields_2d*** is modified as shown in Listing 7.29.

The FDTD simulation is run, and electric field distributions are calculated for 1 and 2 GHz and are plotted in Figures 7.19 and 7.20, respectively. One can notice that the result obtained at 1 GHz using the on-the-fly DFT technique is the same as the ones shown in Section 7.5.2.

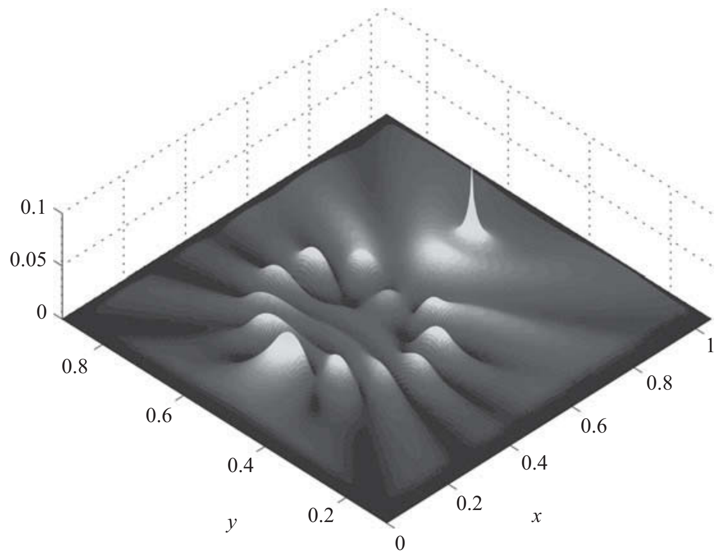


Figure 7.19 Magnitude of electric field distribution calculated by FDTD using DFT at 1 GHz.

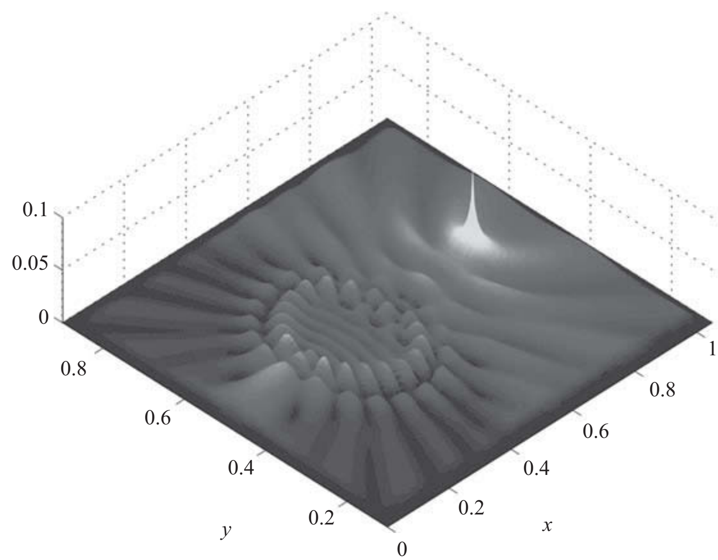


Figure 7.20 Magnitude of electric field distribution calculated by FDTD using DFT at 2 GHz.

7.6 Exercises

7.1 The performance of the two-dimensional PML for the TE_z case is evaluated in Section 7.5.1. Follow the same procedure, and evaluate the performance of the two-dimensional PML case for the TM_z case. You can use the same parameters as the given example. Notice that you need to use an electric current source to excite the TM_z mode, and you can sample E_z at a point close to the PML boundaries.

7.2 In Sections 7.5.2 and 7.5.3 new code sections are added to the two-dimensional FDTD program to add the functionality of displaying the electric field distributions. Follow the same procedure, and add new code sections to the program such that the program will display magnetic field distributions as animations while the simulation is running and it will display the magnitude of the magnetic field distribution as the response of time-harmonic excitations at a number of frequencies.