

# FDFD for Calculating Guided Modes

Sometimes exact or approximate analytical solutions exist for waveguides such as rectangular metal waveguides and optical fibers. For waveguides like integrated optical waveguides and photonic crystal waveguides, no analytical solutions exist and numerical solutions are the only option for a designer. When analytical solutions do exist for simple waveguides, usually little can be changed about the waveguide for the solution to still be valid. However, virtually any type of waveguide can be analyzed with numerical techniques. Finite-difference frequency-domain (FDFD) makes calculating guided modes very easy so it is the first implementation to be discussed in this book. Formulation of the method is covered for both rigorous hybrid mode analysis and slab waveguides. Waveguides will be analyzed as an eigenvalue problem so no source will be needed. The effective index method (EIM) is described as an application of slab waveguide analysis to reduce some three-dimensional devices to a two-dimensional model. Last, implementation in MATLAB is discussed and several examples are given for benchmarking codes including a rib waveguide, a slab waveguide, a surface plasmon polariton (SPP), and a microstrip transmission line. The critical concept of convergence is discussed where the resolution of the simulation is increased until numerical error falls below an acceptable threshold.

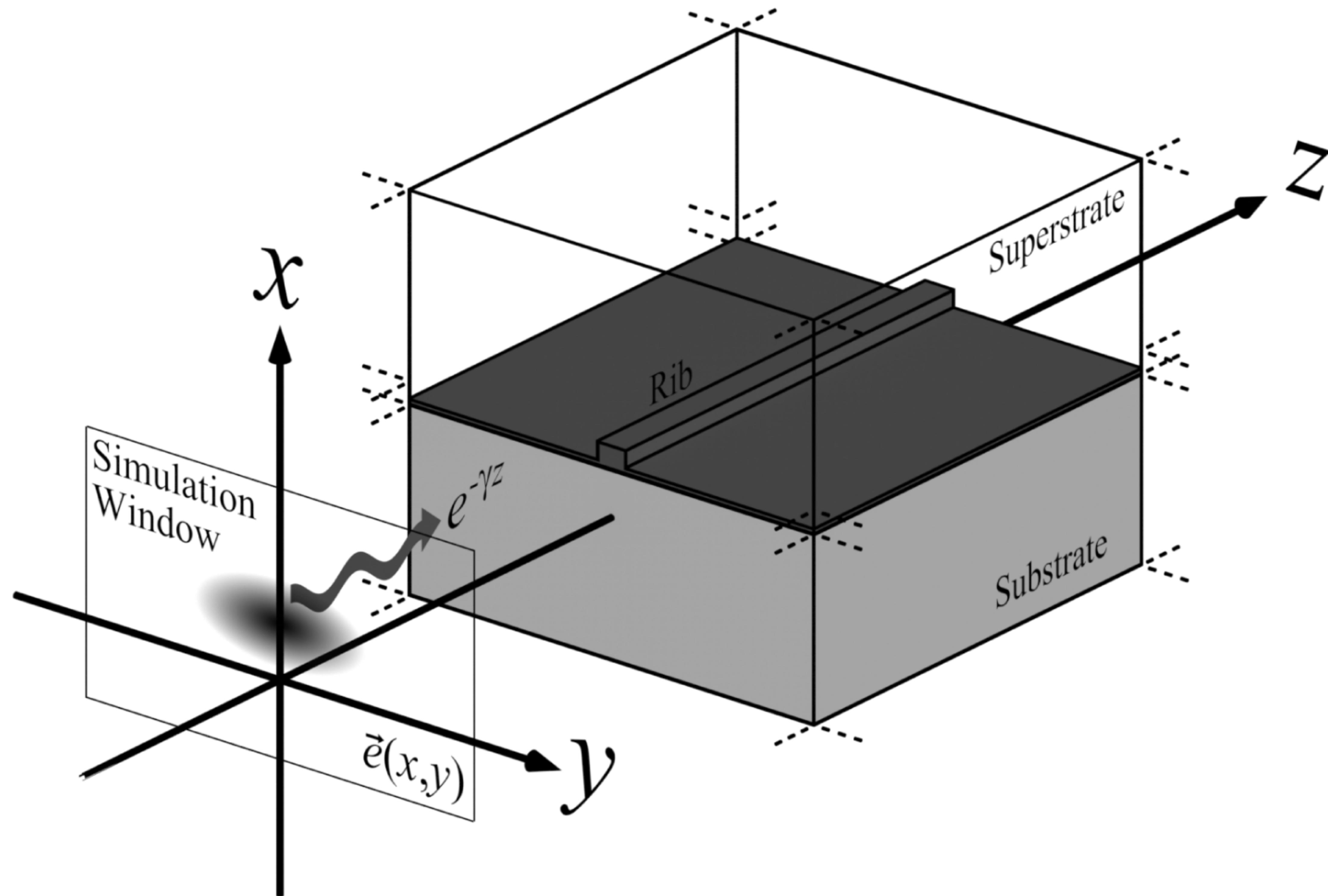
## 6.1 Formulation for Rigorous Hybrid Mode Calculation

Let the geometry for analyzing a channel waveguide be as shown in Figure 6.1. The cross section of the channel waveguide is in the  $xy$  plane while the guided mode propagates in the  $z$ -direction. For hybrid mode analysis, Maxwell's equations will not be separated into two sets of equations. This means that all modes supported by the waveguide will be calculated from the same eigenvalue equation, regardless of how the mode is polarized.

The starting point in FDFD for all waveguide mode calculations is Maxwell's curl equations. The frequency will be known at the start so the free space wavenumber  $k_0$  can be used to normalize the grid coordinates. From Chapter 4, Maxwell's curl equations expanded to

$$\frac{\partial E_z}{\partial y'} - \frac{\partial E_y}{\partial z'} = \mu_{xx} \tilde{H}_x \quad (6.1)$$

$$\frac{\partial E_x}{\partial z'} - \frac{\partial E_z}{\partial x'} = \mu_{yy} \tilde{H}_y \quad (6.2)$$



**Figure 6.1** Geometry of a channel waveguide.

$$\frac{\partial E_y}{\partial x'} - \frac{\partial E_x}{\partial y'} = \mu_{zz} \tilde{H}_z \quad (6.3)$$

$$\frac{\partial \tilde{H}_z}{\partial y'} - \frac{\partial \tilde{H}_y}{\partial z'} = \epsilon_{xx} E_x \quad (6.4)$$

$$\frac{\partial \tilde{H}_x}{\partial z'} - \frac{\partial \tilde{H}_z}{\partial x'} = \epsilon_{yy} E_y \quad (6.5)$$

$$\frac{\partial \tilde{H}_y}{\partial x'} - \frac{\partial \tilde{H}_x}{\partial y'} = \epsilon_{zz} E_z \quad (6.6)$$

where the magnetic field was normalized according to  $\vec{\tilde{H}} = -j\eta_0 \vec{H}$  and the grid coordinates were normalized according to  $x' = k_0 x$ ,  $y' = k_0 y$ , and  $z' = k_0 z$ . As discussed in Chapter 2, all guided modes will have the following form, but here the coordinates and the magnetic field are normalized to be consistent with the formulation.

$$\begin{aligned} \vec{E}(x', y', z') &= \vec{e}(x', y') e^{-\gamma z'/k_0} \\ \vec{\tilde{H}}(x', y', z') &= \vec{h}(x', y') e^{-\gamma z'/k_0} \end{aligned} \quad (6.7)$$

Substituting the form of the solution in (6.7) into (6.1) to (6.6) and simplifying gives

$$\frac{\partial e_z}{\partial y'} + \frac{\gamma}{k_0} e_y = \mu_{xx} \tilde{h}_x \quad (6.8)$$

$$-\frac{\gamma}{k_0} e_x - \frac{\partial e_z}{\partial x'} = \mu_{yy} \tilde{h}_y \quad (6.9)$$

$$\frac{\partial e_y}{\partial x'} - \frac{\partial e_x}{\partial y'} = \mu_{zz} \tilde{h}_z \quad (6.10)$$

$$\frac{\partial \tilde{h}_z}{\partial y'} + \frac{\gamma}{k_0} \tilde{h}_y = \epsilon_{xx} e_x \quad (6.11)$$

$$-\frac{\gamma}{k_0} \tilde{h}_x - \frac{\partial \tilde{h}_z}{\partial x'} = \epsilon_{yy} e_y \quad (6.12)$$

$$\frac{\partial \tilde{h}_y}{\partial x'} - \frac{\partial \tilde{h}_x}{\partial y'} = \epsilon_{zz} e_z \quad (6.13)$$

In these equations, the analysis has reduced to just two dimensions because  $z'$  was eliminated from the analysis. When the field components are made discrete in the  $xy$  plane following the Yee grid scheme, the partial derivatives in each of the above equations can be approximated using finite differences. In addition, observe the ratio  $\gamma/k_0$  that appears in four of the above equations. Let this ratio define the *normalized complex propagation constant* according to  $\tilde{\gamma} = \gamma/k_0$ . The resulting set of discrete equations with the normalized propagation constant is

$$\frac{e_z|_{i,j+1} - e_z|_{i,j}}{\Delta y'} + \tilde{\gamma} e_y|_{i,j} = \mu_{xx}|_{i,j} \tilde{h}_x|_{i,j} \quad (6.14)$$

$$-\tilde{\gamma} e_x|_{i,j} - \frac{e_z|_{i+1,j} - e_z|_{i,j}}{\Delta x'} = \mu_{yy}|_{i,j} \tilde{h}_y|_{i,j} \quad (6.15)$$

$$\frac{e_y|_{i+1,j} - e_y|_{i,j}}{\Delta x'} - \frac{e_x|_{i,j+1} - e_x|_{i,j}}{\Delta y'} = \mu_{zz}|_{i,j} \tilde{h}_z|_{i,j} \quad (6.16)$$

$$\frac{\tilde{h}_z|_{i,j} - \tilde{h}_z|_{i,j-1}}{\Delta y'} + \tilde{\gamma} \tilde{h}_y|_{i,j} = \epsilon_{xx}|_{i,j} e_x|_{i,j} \quad (6.17)$$

$$-\tilde{\gamma} \tilde{h}_x|_{i,j} - \frac{\tilde{h}_z|_{i,j} - \tilde{h}_z|_{i-1,j}}{\Delta x'} = \epsilon_{yy}|_{i,j} e_y|_{i,j} \quad (6.18)$$

$$\frac{\tilde{h}_y|_{i,j} - \tilde{h}_y|_{i-1,j}}{\Delta x'} - \frac{\tilde{h}_x|_{i,j} - \tilde{h}_x|_{i,j-1}}{\Delta y'} = \epsilon_{zz}|_{i,j} e_z|_{i,j} \quad (6.19)$$

Each of these discrete equations is written once for every cell in the grid, and each set of equations is written in matrix form following the procedures described in Chapters 3 and 4. These equations are

$$\mathbf{D}_{y'}^e \mathbf{e}_z + \tilde{\gamma} \mathbf{e}_y = \mu_{xx} \tilde{\mathbf{h}}_x \quad (6.20)$$

$$-\tilde{\gamma} \mathbf{e}_x - \mathbf{D}_{x'}^e \mathbf{e}_z = \mu_{yy} \tilde{\mathbf{h}}_y \quad (6.21)$$

$$\mathbf{D}_{x'}^e \mathbf{e}_y - \mathbf{D}_{y'}^e \mathbf{e}_x = \mu_{zz} \tilde{\mathbf{h}}_z \quad (6.22)$$

$$\mathbf{D}_{y'}^h \tilde{\mathbf{h}}_z + \tilde{\gamma} \tilde{\mathbf{h}}_y = \boldsymbol{\epsilon}_{xx} \mathbf{e}_x \quad (6.23)$$

$$-\tilde{\gamma} \tilde{\mathbf{h}}_x - \mathbf{D}_{x'}^h \tilde{\mathbf{h}}_z = \boldsymbol{\epsilon}_{yy} \mathbf{e}_y \quad (6.24)$$

$$\mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y - \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_x = \boldsymbol{\epsilon}_{zz} \mathbf{e}_z \quad (6.25)$$

Next, (6.22) is solved for  $\tilde{\mathbf{h}}_z$  and (6.25) is solved for  $\mathbf{e}_z$ .

$$\tilde{\mathbf{h}}_z = \mu_{zz}^{-1} (\mathbf{D}_{x'}^e \mathbf{e}_y - \mathbf{D}_{y'}^e \mathbf{e}_x) \quad (6.26)$$

$$\mathbf{e}_z = \boldsymbol{\epsilon}_{zz}^{-1} (\mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y - \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_x) \quad (6.27)$$

It is possible to eliminate  $\tilde{\mathbf{h}}_z$  and  $\mathbf{e}_z$  from the matrix equations by substituting (6.26) into (6.23) and (6.24) and substituting (6.27) into (6.20) and (6.21). This gives the following set of four coupled matrix equations containing only  $\mathbf{e}_x$ ,  $\mathbf{e}_y$ ,  $\tilde{\mathbf{h}}_x$ , and  $\tilde{\mathbf{h}}_y$ .

$$\mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{zz}^{-1} (\mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y - \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_x) + \tilde{\gamma} \mathbf{e}_y = \mu_{xx} \tilde{\mathbf{h}}_x \quad (6.28)$$

$$-\tilde{\gamma} \mathbf{e}_x - \mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} (\mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y - \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_x) = \mu_{yy} \tilde{\mathbf{h}}_y \quad (6.29)$$

$$\mathbf{D}_{y'}^h \mu_{zz}^{-1} (\mathbf{D}_{x'}^e \mathbf{e}_y - \mathbf{D}_{y'}^e \mathbf{e}_x) + \tilde{\gamma} \tilde{\mathbf{h}}_y = \boldsymbol{\epsilon}_{xx} \mathbf{e}_x \quad (6.30)$$

$$-\tilde{\gamma} \tilde{\mathbf{h}}_x - \mathbf{D}_{x'}^h \mu_{zz}^{-1} (\mathbf{D}_{x'}^e \mathbf{e}_y - \mathbf{D}_{y'}^e \mathbf{e}_x) = \boldsymbol{\epsilon}_{yy} \mathbf{e}_y \quad (6.31)$$

It is very important to realize that the longitudinal components  $\tilde{\mathbf{h}}_z$  and  $\mathbf{e}_z$  were not set to zero and are not necessarily equal to zero. Instead, they were just algebraically eliminated from the formulation. Next, (6.28) to (6.31) are simplified, terms are rearranged, and the equations are expressed in a different order.

$$\mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_x - (\mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h + \mu_{yy}) \tilde{\mathbf{h}}_y = \tilde{\gamma} \mathbf{e}_x \quad (6.32)$$

$$(\mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{y'}^h + \mu_{xx}) \tilde{\mathbf{h}}_x - \mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y = \tilde{\gamma} \mathbf{e}_y \quad (6.33)$$

$$\mathbf{D}_{x'}^h \mu_{zz}^{-1} \mathbf{D}_{y'}^e \mathbf{e}_x - (\mathbf{D}_{x'}^h \mu_{zz}^{-1} \mathbf{D}_{x'}^e + \boldsymbol{\epsilon}_{yy}) \mathbf{e}_y = \tilde{\gamma} \tilde{\mathbf{h}}_x \quad (6.34)$$

$$(\mathbf{D}_{y'}^h \mu_{zz}^{-1} \mathbf{D}_{y'}^e + \boldsymbol{\epsilon}_{xx}) \mathbf{e}_x - \mathbf{D}_{y'}^h \mu_{zz}^{-1} \mathbf{D}_{x'}^e \mathbf{e}_y = \tilde{\gamma} \tilde{\mathbf{h}}_y \quad (6.35)$$

Equations (6.32) and (6.33) can be combined into a single block matrix equation as well as (6.34) and (6.35). These two block matrix equations are

$$\mathbf{P} \begin{bmatrix} \tilde{\mathbf{h}}_x \\ \tilde{\mathbf{h}}_y \end{bmatrix} = \tilde{\gamma} \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} \quad (6.36)$$

$$\mathbf{Q} \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} = \tilde{\gamma} \begin{bmatrix} \tilde{\mathbf{h}}_x \\ \tilde{\mathbf{h}}_y \end{bmatrix} \quad (6.37)$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{y'}^h & -(\mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h + \boldsymbol{\mu}_{yy}) \\ \mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{y'}^h + \boldsymbol{\mu}_{xx} & -\mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h \end{bmatrix} \quad (6.38)$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D}_{x'}^h \boldsymbol{\mu}_{zz}^{-1} \mathbf{D}_{y'}^e & -(\mathbf{D}_{x'}^h \boldsymbol{\mu}_{zz}^{-1} \mathbf{D}_{x'}^e + \boldsymbol{\epsilon}_{yy}) \\ \mathbf{D}_{y'}^h \boldsymbol{\mu}_{zz}^{-1} \mathbf{D}_{y'}^e + \boldsymbol{\epsilon}_{xx} & -\mathbf{D}_{y'}^h \boldsymbol{\mu}_{zz}^{-1} \mathbf{D}_{x'}^e \end{bmatrix} \quad (6.39)$$

To derive an eigenvalue problem in terms of just the electric field terms  $\mathbf{e}_x$  and  $\mathbf{e}_y$ , first (6.37) is solved for the magnetic field terms  $\tilde{\mathbf{h}}_x$  and  $\tilde{\mathbf{h}}_y$ .

$$\begin{bmatrix} \tilde{\mathbf{h}}_x \\ \tilde{\mathbf{h}}_y \end{bmatrix} = \frac{1}{\tilde{\gamma}} \mathbf{Q} \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} \quad (6.40)$$

Second, (6.40) is substituted into (6.36) to eliminate the magnetic field terms  $\tilde{\mathbf{h}}_x$  and  $\tilde{\mathbf{h}}_y$ . This gives a matrix wave equation in the form of a standard eigenvalue problem  $\mathbf{Av} = \lambda v$ .

$$\boldsymbol{\Omega}^2 \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} = \tilde{\gamma}^2 \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{bmatrix} \quad (6.41)$$

$$\boldsymbol{\Omega}^2 = \mathbf{PQ} \quad (6.42)$$

This general “PQ” form of the eigenvalue problem arises in other methods like the method of lines and rigorous coupled-wave analysis [1]. Solving eigenvalue problems is a huge and involved topic. Fortunately, MATLAB makes this very simple, and (6.41) is solved as simple as  $[V, D] = \text{eigs}(\text{OMEGASQ})$  where OMEGASQ is a sparse matrix defined in (6.42). When (6.41) is solved as an eigenvalue problem, two matrices are calculated. The eigenvector matrix  $V$  contains the electric field components of the modes along its columns. That is, the  $m$ th column of  $V$  is  $\vec{e}_m(x, y)$  that contains  $e_{x,m}(x, y)$  and  $e_{y,m}(x, y)$ . The eigenvalue matrix  $D$  contains the eigenvalues  $\tilde{\gamma}^2$  along its diagonal. In this case, the eigenvalues are the squares of the normalized complex propagation constants for the guided modes because (6.41) is arranged such that  $\tilde{\gamma}^2$

is the eigenvalue. Eigenvectors and eigenvalues always come in pairs. The  $m$ th column in the eigenvector matrix must always be kept with the  $m$ th eigenvalue. Given the eigenvalue  $\tilde{\gamma}_m^2$  of the  $m$ th guided mode, the complex propagation constant  $\gamma_m$ , attenuation coefficient  $\alpha_m$ , phase constant  $\beta_m$ , and effective refractive index  $n_{m,\text{eff}}$  are calculated as follows.

$$\gamma_m = k_0 \sqrt{\tilde{\gamma}_m^2} \quad (6.43)$$

$$\alpha_m = \operatorname{Re}[\gamma_m] \quad (6.44)$$

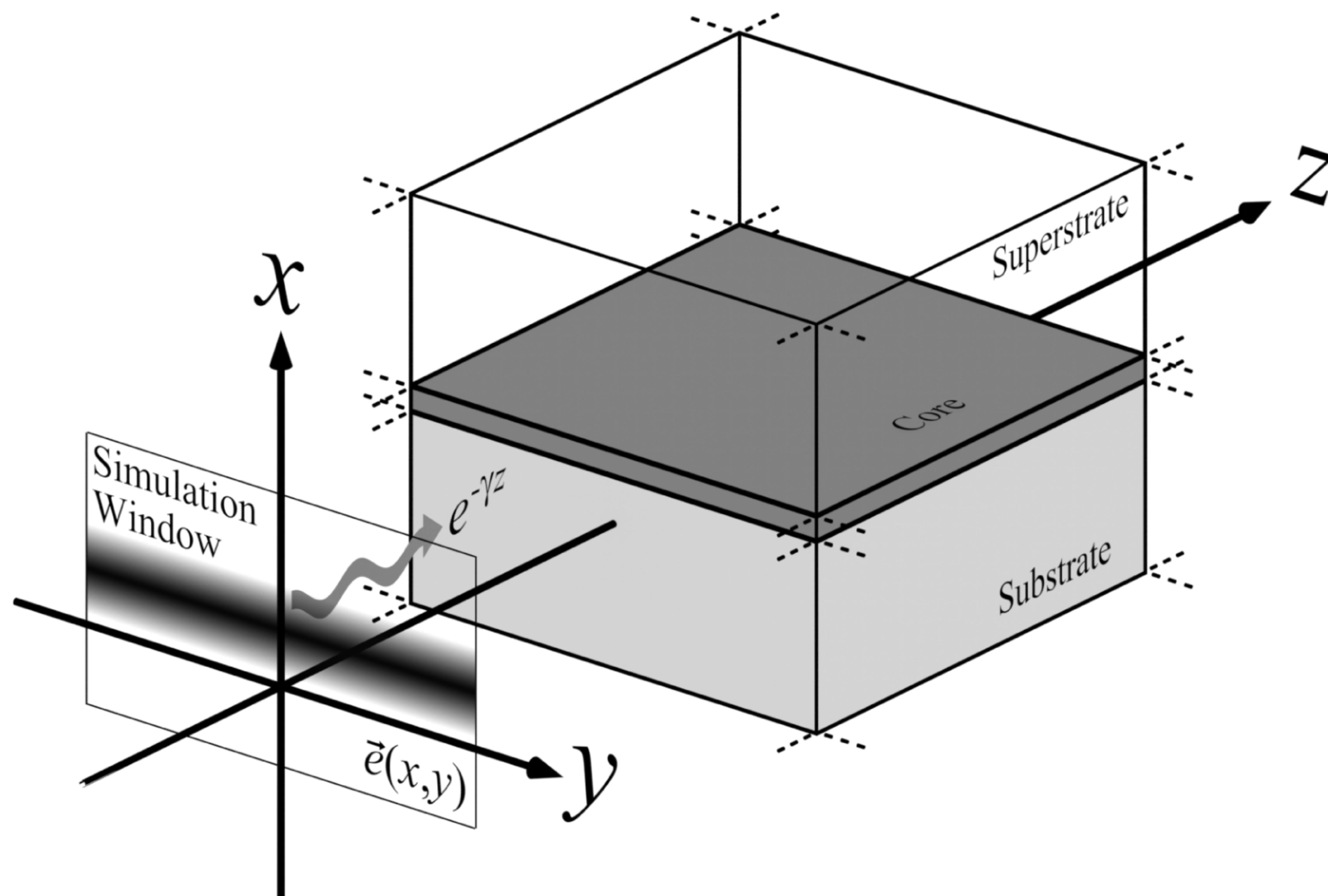
$$\beta_m = \operatorname{Im}[\gamma_m] \quad (6.45)$$

$$n_{m,\text{eff}} = \frac{\gamma_m}{jk_0} \quad (6.46)$$

If needed, the magnetic field components  $\tilde{h}_x(x, y)$  and  $\tilde{h}_y(x, y)$  can be calculated from the solution using (6.40). The longitudinal components  $\tilde{h}_z(x, y)$  and  $e_z(x, y)$  can be calculated using (6.26) and (6.27), respectively.

## 6.2 Formulation for Rigorous Slab Waveguide Mode Calculation

Let the geometry for analyzing a slab waveguide be as shown in Figure 6.2. Let the cross section of the slab waveguide be in the  $x$ -direction while the guided mode propagates in the  $z$ -direction. In this configuration, the slab waveguide and the



**Figure 6.2** Geometry for a slab waveguide.

guided mode itself will be uniform and unchanging in the  $y$ -direction. This means that any derivative with respect to  $y$  must be equal to zero because nothing changes in the  $y$ -direction. Under this condition,  $D_y^e = D_y^h = 0$  and (6.20) to (6.25) reduce to

$$\tilde{\gamma} \mathbf{e}_y = \mu_{xx} \tilde{\mathbf{h}}_x \quad (6.47)$$

$$-\tilde{\gamma} \mathbf{e}_x - D_x^e \mathbf{e}_z = \mu_{yy} \tilde{\mathbf{h}}_y \quad (6.48)$$

$$D_x^e \mathbf{e}_y = \mu_{zz} \tilde{\mathbf{h}}_z \quad (6.49)$$

$$\tilde{\gamma} \tilde{\mathbf{h}}_y = \epsilon_{xx} \mathbf{e}_x \quad (6.50)$$

$$-\tilde{\gamma} \tilde{\mathbf{h}}_x - D_z^h \tilde{\mathbf{h}}_z = \epsilon_{yy} \mathbf{e}_y \quad (6.51)$$

$$D_z^h \tilde{\mathbf{h}}_y = \epsilon_{zz} \mathbf{e}_z \quad (6.52)$$

### 6.2.1 Formulation of E Mode Slab Waveguide Analysis

Observe that (6.47) to (6.52) have decoupled into two independent sets of three equations. The first set will be called the E mode because its analysis will be reduced to one equation in terms of the single electric field quantity  $\mathbf{e}_y$ . The equations for the E mode are given by (6.47), (6.49), and (6.51) and contain only  $\mathbf{e}_y$ ,  $\tilde{\mathbf{h}}_x$ , and  $\tilde{\mathbf{h}}_z$ . The missing components are  $\mathbf{e}_x = \mathbf{e}_z = \tilde{\mathbf{h}}_y = 0$ . For convenience, these three equations are repeated below.

$$-\tilde{\gamma} \tilde{\mathbf{h}}_x - D_z^h \tilde{\mathbf{h}}_z = \epsilon_{yy} \mathbf{e}_y \quad (6.53)$$

$$\tilde{\gamma} \mathbf{e}_y = \mu_{xx} \tilde{\mathbf{h}}_x \quad (6.54)$$

$$D_x^e \mathbf{e}_y = \mu_{zz} \tilde{\mathbf{h}}_z \quad (6.55)$$

The first step to derive the matrix wave equation in the form of an eigenvalue problem is to solve (6.54) for  $\tilde{\mathbf{h}}_x$  and solve (6.55) for  $\tilde{\mathbf{h}}_z$ . This gives

$$\tilde{\mathbf{h}}_x = \tilde{\gamma} \mu_{xx}^{-1} \mathbf{e}_y \quad (6.56)$$

$$\tilde{\mathbf{h}}_z = \mu_{zz}^{-1} D_x^e \mathbf{e}_y \quad (6.57)$$

The second step is to substitute the above expressions into (6.53) to eliminate the terms  $\tilde{\mathbf{h}}_x$  and  $\tilde{\mathbf{h}}_z$  and then rearrange the equation into the form of a generalized eigenvalue problem  $\mathbf{Av} = \lambda \mathbf{Bv}$ . This is solved in the same manner as for hybrid mode analysis and is given by

$$-(D_z^h \mu_{zz}^{-1} D_x^e + \epsilon_{yy}) \mathbf{e}_y = \tilde{\gamma}^2 \mu_{xx}^{-1} \mathbf{e}_y \quad (6.58)$$

When the permeability is close to vacuum,  $\mu_{xx} \approx I$  and (6.58) reduces to a standard eigenvalue problem  $\mathbf{Av} = \lambda v$ . After the eigenvalue problem is solved, the guided mode parameters are calculated the same as for hybrid mode analysis using (6.43) to (6.46). That is because the eigenvalue problem for slab waveguide analysis has the same term for the eigenvalue.

### 6.2.2 Formulation of H Mode Slab Waveguide Analysis

The second set of equations that comes from (6.47) to (6.52) will be called the H mode because its analysis will be reduced to one equation in terms of the single magnetic field quantity  $\tilde{\mathbf{h}}_y$ . The equations for the H mode are given by (6.48), (6.50), and (6.52). The equations contain only  $\tilde{\mathbf{h}}_y$ ,  $\mathbf{e}_x$ , and  $\mathbf{e}_z$  so for the H mode  $\tilde{\mathbf{h}}_x = \tilde{\mathbf{h}}_z = \mathbf{e}_y = 0$ . For convenience, these three equations are repeated below.

$$-\tilde{\gamma}\mathbf{e}_x - \mathbf{D}_{x'}^e \mathbf{e}_z = \mu_{yy} \tilde{\mathbf{h}}_y \quad (6.59)$$

$$\tilde{\gamma} \tilde{\mathbf{h}}_y = \boldsymbol{\epsilon}_{xx} \mathbf{e}_x \quad (6.60)$$

$$\mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y = \boldsymbol{\epsilon}_{zz} \mathbf{e}_z \quad (6.61)$$

The first step to derive the matrix wave equation in the form of an eigenvalue problem is to solve (6.60) for  $\mathbf{e}_x$  and solve (6.61) for  $\mathbf{e}_z$ .

$$\mathbf{e}_x = \tilde{\gamma} \boldsymbol{\epsilon}_{xx}^{-1} \tilde{\mathbf{h}}_y \quad (6.62)$$

$$\mathbf{e}_z = \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h \tilde{\mathbf{h}}_y \quad (6.63)$$

The second step is to substitute the above expressions into (6.59) to eliminate the terms  $\mathbf{e}_x$  and  $\mathbf{e}_z$  and then rearrange the equation into the form of a generalized eigenvalue problem  $\mathbf{Av} = \lambda \mathbf{Bv}$ . This is solved in the same manner as for hybrid mode analysis and is given by

$$-(\mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{zz}^{-1} \mathbf{D}_{x'}^h + \mu_{yy}) \tilde{\mathbf{h}}_y = \tilde{\gamma}^2 \boldsymbol{\epsilon}_{xx}^{-1} \tilde{\mathbf{h}}_y \quad (6.64)$$

Unlike E mode analysis, the B matrix is permittivity which will likely never be equal to that of a vacuum so (6.64) will rarely reduce to a standard eigenvalue problem. After the eigenvalue problem is solved, the guided mode parameters are calculated the same as for hybrid mode analysis using (6.43) to (6.46). That is because the eigenvalue problem for slab waveguide analysis has the same eigenvalue.

### 6.2.3 Formulations for Slab Waveguides in Other Orientations

It is very useful to derive the matrix eigenvalue problems where the slab waveguide is oriented along different axes. This is particularly useful when the slab waveguide analysis is being used to calculate sources for waveguide simulations or when the EIM is being used to reduce three-dimensional problems down to two dimensions. For slab modes propagating in the  $+x$ -direction in a slab waveguide, the uniform

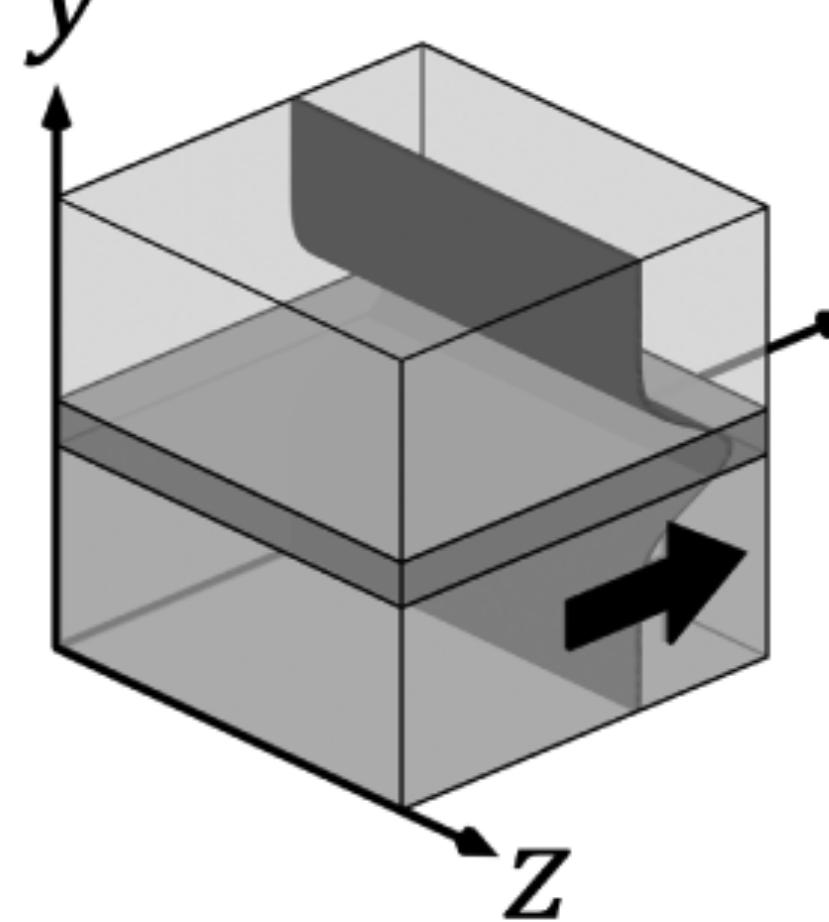
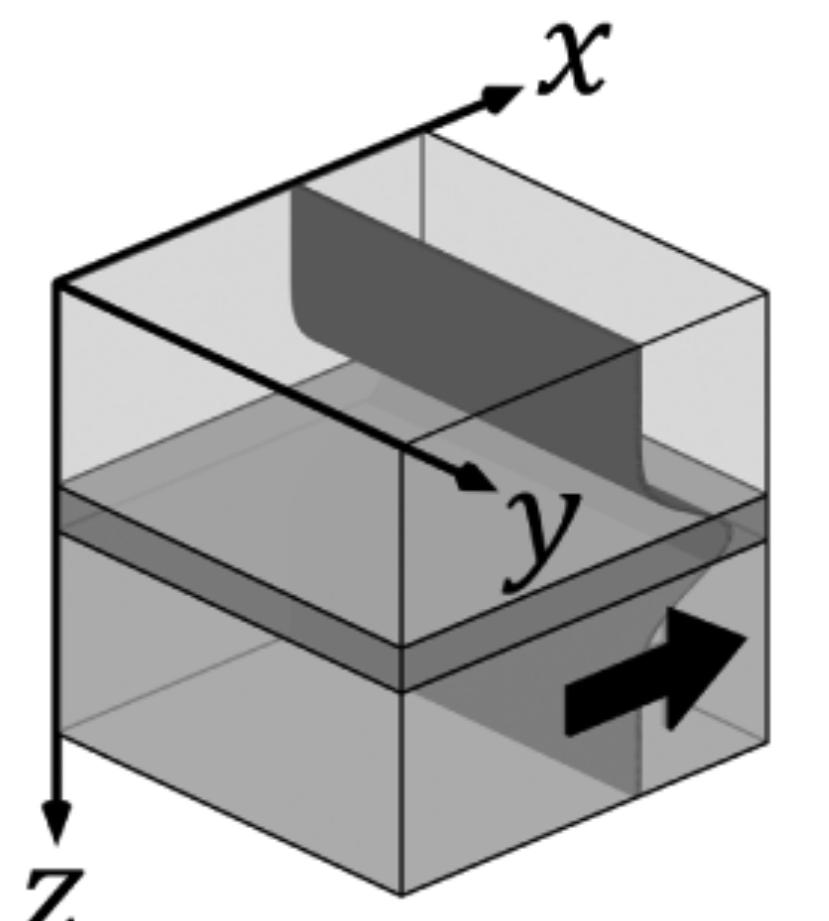
	<b>E Mode</b> $-(\mathbf{D}_{y'}^h \boldsymbol{\mu}_{xx}^{-1} \mathbf{D}_{y'}^e + \boldsymbol{\epsilon}_{zz}) \mathbf{e}_z = \tilde{\gamma}^2 \boldsymbol{\mu}_{yy}^{-1} \mathbf{e}_z$ $\tilde{\mathbf{h}}_x = \boldsymbol{\mu}_{xx}^{-1} \mathbf{D}_{y'}^e \mathbf{e}_z$ $\tilde{\mathbf{h}}_y = \tilde{\gamma} \boldsymbol{\mu}_{yy}^{-1} \mathbf{e}_z$	<b>H Mode</b> $-(\mathbf{D}_{y'}^e \boldsymbol{\epsilon}_{xx}^{-1} \mathbf{D}_{y'}^h + \boldsymbol{\mu}_{zz}) \tilde{\mathbf{h}}_z = \tilde{\gamma}^2 \boldsymbol{\epsilon}_{yy}^{-1} \tilde{\mathbf{h}}_z$ $\mathbf{e}_x = \boldsymbol{\epsilon}_{xx}^{-1} \mathbf{D}_{y'}^h \tilde{\mathbf{h}}_z$ $\mathbf{e}_y = \tilde{\gamma} \boldsymbol{\epsilon}_{yy}^{-1} \tilde{\mathbf{h}}_z$
	<b>E Mode</b> $-(\mathbf{D}_{z'}^h \boldsymbol{\mu}_{xx}^{-1} \mathbf{D}_{z'}^e + \boldsymbol{\epsilon}_{yy}) \mathbf{e}_y = \tilde{\gamma}^2 \boldsymbol{\mu}_{zz}^{-1} \mathbf{e}_y$ $\tilde{\mathbf{h}}_x = -\boldsymbol{\mu}_{xx}^{-1} \mathbf{D}_{z'}^e \mathbf{e}_y$ $\tilde{\mathbf{h}}_z = -\tilde{\gamma} \boldsymbol{\mu}_{zz}^{-1} \mathbf{e}_y$	<b>H Mode</b> $-(\mathbf{D}_{z'}^e \boldsymbol{\epsilon}_{xx}^{-1} \mathbf{D}_{z'}^h + \boldsymbol{\mu}_{yy}) \tilde{\mathbf{h}}_y = \tilde{\gamma}^2 \boldsymbol{\epsilon}_{zz}^{-1} \tilde{\mathbf{h}}_y$ $\mathbf{e}_x = -\boldsymbol{\epsilon}_{xx}^{-1} \mathbf{D}_{z'}^h \tilde{\mathbf{h}}_y$ $\mathbf{e}_z = -\tilde{\gamma} \boldsymbol{\epsilon}_{zz}^{-1} \tilde{\mathbf{h}}_y$

Figure 6.3 Slab waveguide analysis for modes propagating in the  $+x$ -direction.

direction can be either in the  $y$ -direction or the  $z$ -direction. Each choice leads to a different formulation for both the E and H modes and these are summarized in Figure 6.3.

For slab modes propagating in the  $+y$ -direction in a slab waveguide, the uniform direction can be either in the  $x$ -direction or the  $z$ -direction. The formulations for both the E and H modes for this case are summarized in Figure 6.4.

For slab modes propagating in the  $+z$ -direction in a slab waveguide, the uniform direction can be either in the  $x$ -direction or the  $y$ -direction. The formulations for both the E and H modes for this case are summarized in Figure 6.5.

#### 6.2.4 The Effective Index Method

Many times, large and complicated three-dimensional simulations can be reduced to much simpler and more numerically efficient two-dimensional simulations quite accurately. An excellent example is the optical integrated circuit (OIC) for a ring

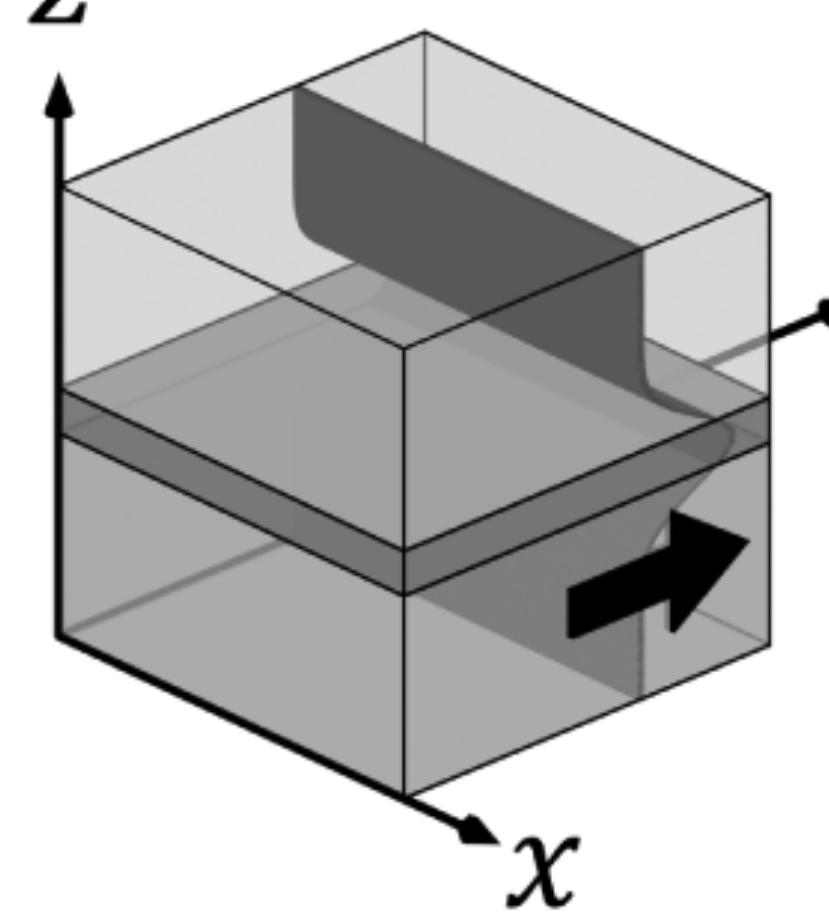
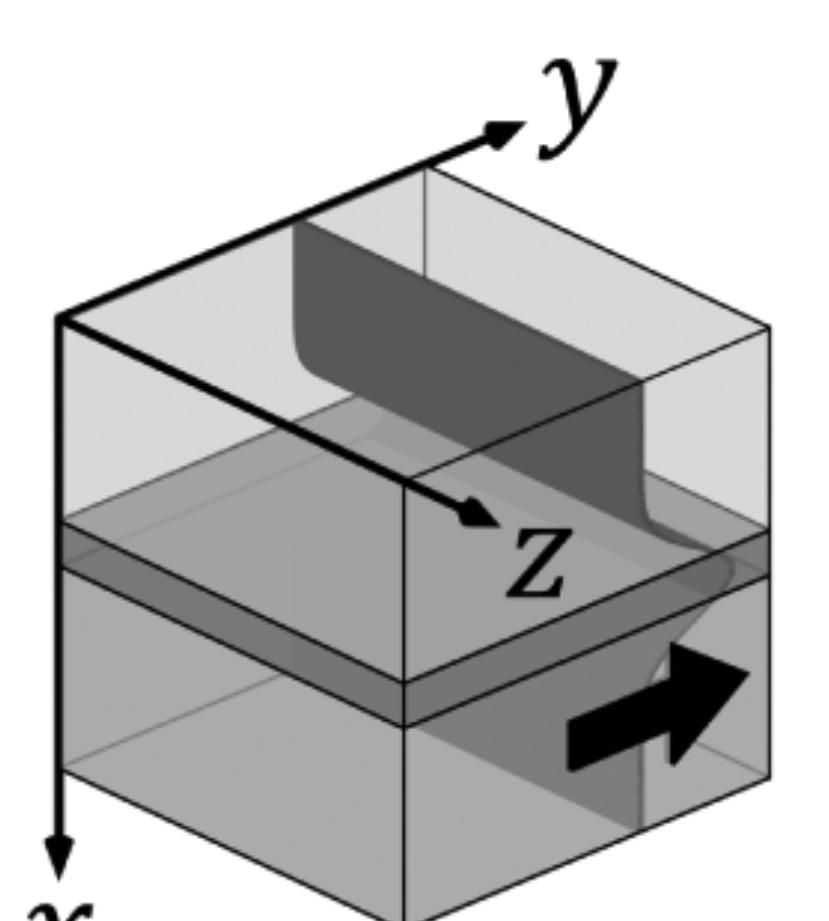
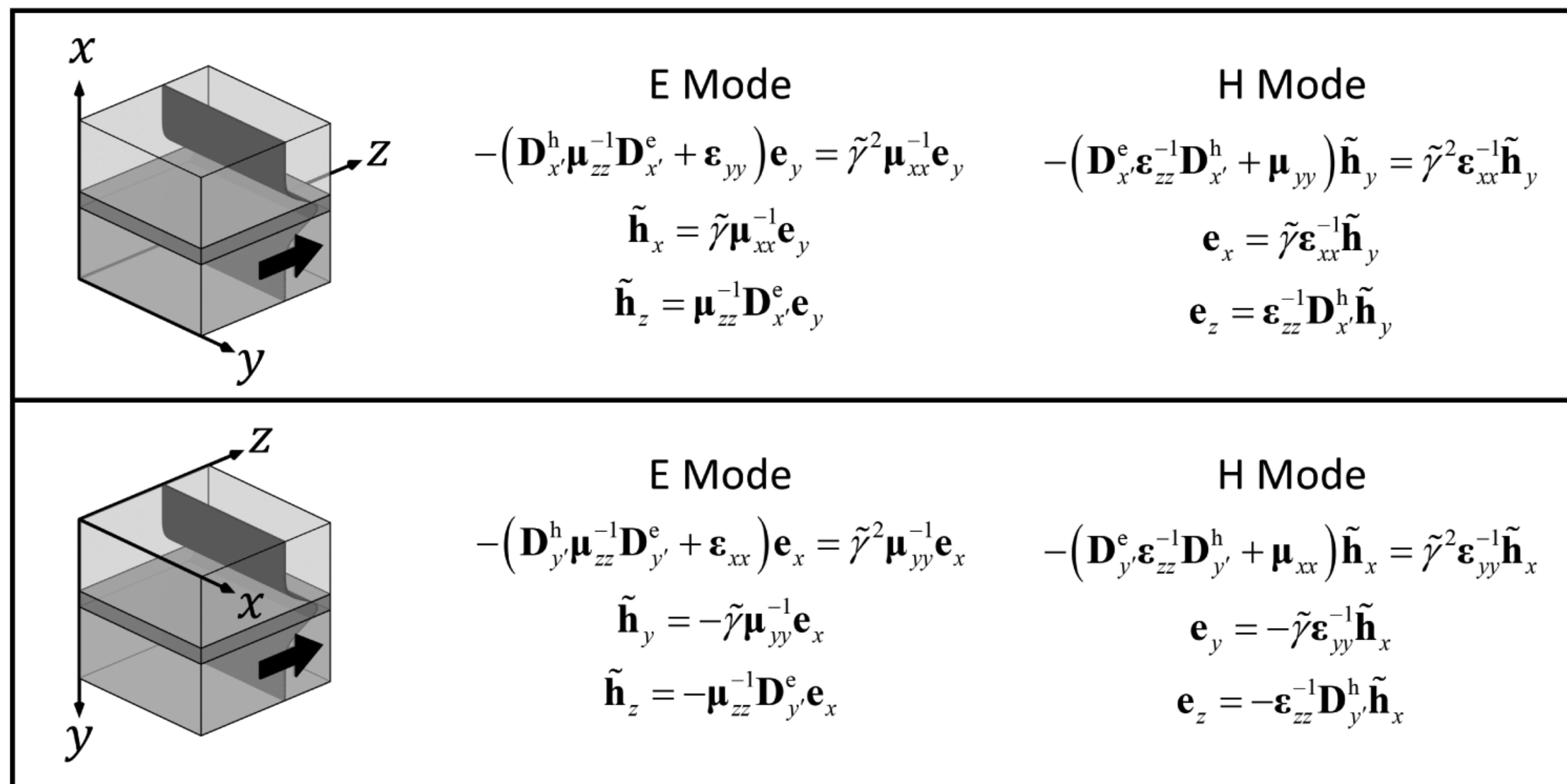
	<b>E Mode</b> $-(\mathbf{D}_{z'}^h \boldsymbol{\mu}_{yy}^{-1} \mathbf{D}_{z'}^e + \boldsymbol{\epsilon}_{xx}) \mathbf{e}_x = \tilde{\gamma}^2 \boldsymbol{\mu}_{zz}^{-1} \mathbf{e}_x$ $\tilde{\mathbf{h}}_y = \boldsymbol{\mu}_{yy}^{-1} \mathbf{D}_{z'}^e \mathbf{e}_x$ $\tilde{\mathbf{h}}_z = \tilde{\gamma} \boldsymbol{\mu}_{zz}^{-1} \mathbf{e}_x$	<b>H Mode</b> $-(\mathbf{D}_{z'}^e \boldsymbol{\epsilon}_{yy}^{-1} \mathbf{D}_{z'}^h + \boldsymbol{\mu}_{xx}) \tilde{\mathbf{h}}_x = \tilde{\gamma}^2 \boldsymbol{\epsilon}_{zz}^{-1} \tilde{\mathbf{h}}_x$ $\mathbf{e}_y = \boldsymbol{\epsilon}_{yy}^{-1} \mathbf{D}_{z'}^h \tilde{\mathbf{h}}_x$ $\mathbf{e}_z = \tilde{\gamma} \boldsymbol{\epsilon}_{zz}^{-1} \tilde{\mathbf{h}}_x$
	<b>E Mode</b> $-(\mathbf{D}_{x'}^h \boldsymbol{\mu}_{yy}^{-1} \mathbf{D}_{x'}^e + \boldsymbol{\epsilon}_{zz}) \mathbf{e}_z = \tilde{\gamma}^2 \boldsymbol{\mu}_{xx}^{-1} \mathbf{e}_z$ $\tilde{\mathbf{h}}_x = -\tilde{\gamma} \boldsymbol{\mu}_{xx}^{-1} \mathbf{e}_z$ $\tilde{\mathbf{h}}_y = -\boldsymbol{\mu}_{yy}^{-1} \mathbf{D}_{x'}^e \mathbf{e}_z$	<b>H Mode</b> $-(\mathbf{D}_{x'}^e \boldsymbol{\epsilon}_{yy}^{-1} \mathbf{D}_{x'}^h + \boldsymbol{\mu}_{zz}) \tilde{\mathbf{h}}_z = \tilde{\gamma}^2 \boldsymbol{\epsilon}_{xx}^{-1} \tilde{\mathbf{h}}_z$ $\mathbf{e}_x = -\tilde{\gamma} \boldsymbol{\epsilon}_{xx}^{-1} \tilde{\mathbf{h}}_z$ $\mathbf{e}_y = -\boldsymbol{\mu}_{yy}^{-1} \mathbf{D}_{x'}^h \tilde{\mathbf{h}}_z$

Figure 6.4 Slab waveguide analysis for modes propagating in the  $+y$ -direction.

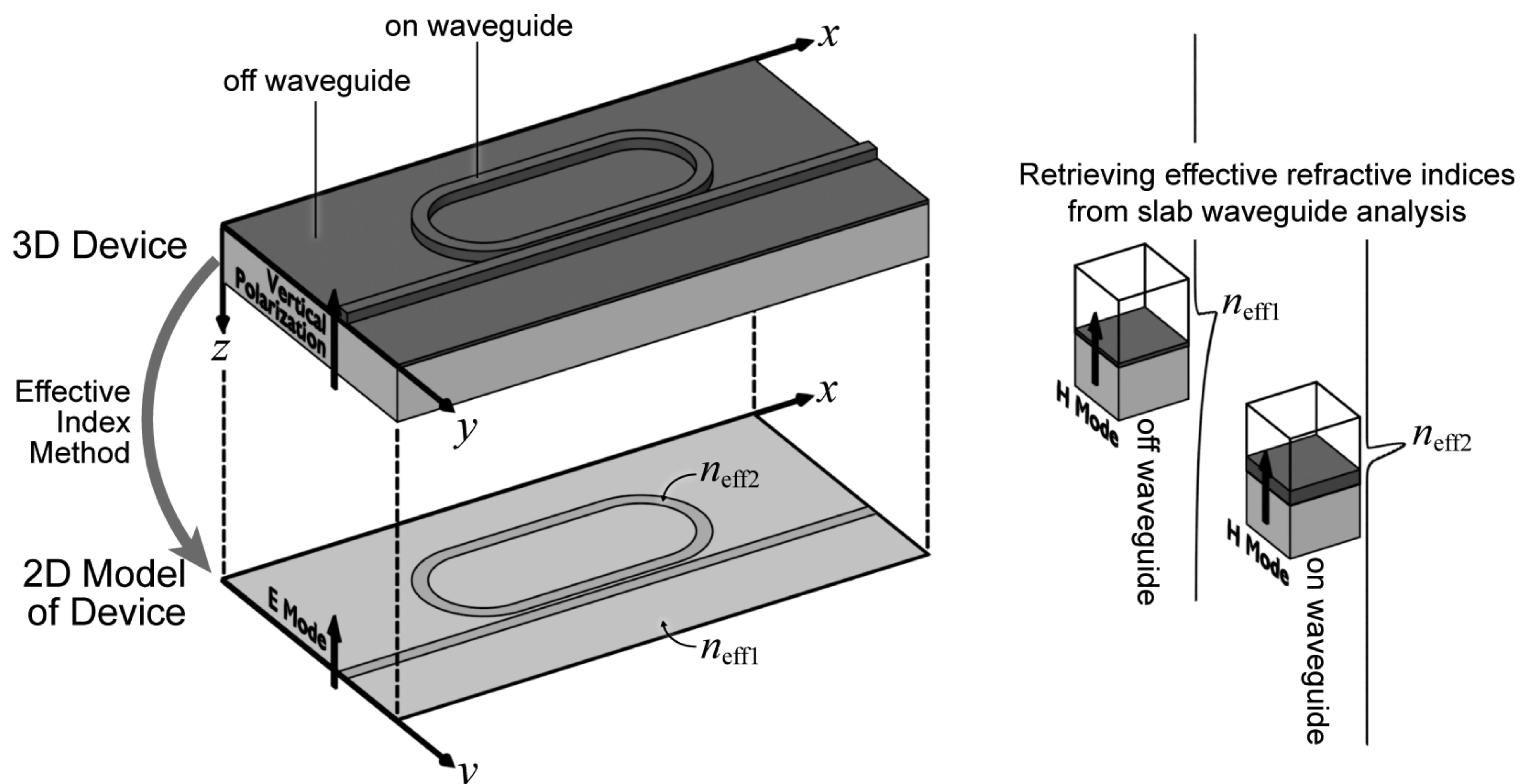


**Figure 6.5** Slab waveguide analysis for modes propagating in the  $+z$ -direction.

resonator depicted in Figure 6.6. This OIC leads to a very large and three-dimensional simulation that would be very computationally intensive to perform rigorously. Instead, the OIC can be reduced to a two-dimensional representation using the EIM [2]. To do this, the circuit is interpreted as being composed of two regions. The first region is away from the rib waveguide and is composed of the substrate, a thin high-index film on the surface, and the air above. The second region comes from on the rib waveguide and is composed of the substrate, the thin high-index film, a high-index rib on top of the film, and the air above. Both regions are analyzed as a slab waveguide to determine the effective refractive index of the fundamental mode. Recall from Chapter 2 that the fundamental mode is the guided mode with the lowest cutoff frequency. The effective refractive index of the off-waveguide region is  $n_{\text{eff}1}$  and the effective refractive index of the on-waveguide region is  $n_{\text{eff}2}$ . These analyses and the resulting fundamental mode calculated for each region are illustrated on the right of Figure 6.6.

Given the two effective refractive indices, a model of the circuit can be constructed in two dimensions as illustrated at the bottom left of Figure 6.6. This two-dimensional representation can be thought of as sort of a top view of the three-dimensional OIC and is constructed from the two effective refractive indices. While this two-dimensional model is not a rigorous representation of the three-dimensional OIC, it is orders of magnitude less computationally intensive while maintaining very good accuracy [2]. In some cases, the regions chosen may not support a guided mode or not even be a slab waveguide at all. In this case, any reasonable estimate of the average refractive index can still provide good results that match well with a three-dimensional simulation.

For best accuracy, it is important to carefully consider whether E or H mode analysis should be performed to ensure the correct polarization is used throughout the entire analysis. For example, the OIC depicted in Figure 6.6 shows the rib waveguide is to be illuminated with a vertically polarized mode. To be consistent, the slab waveguide analysis should be H mode to allow the electric field to be in the  $z$ -direction. The E mode would place the electric field solely in the  $x$ -direction, so the



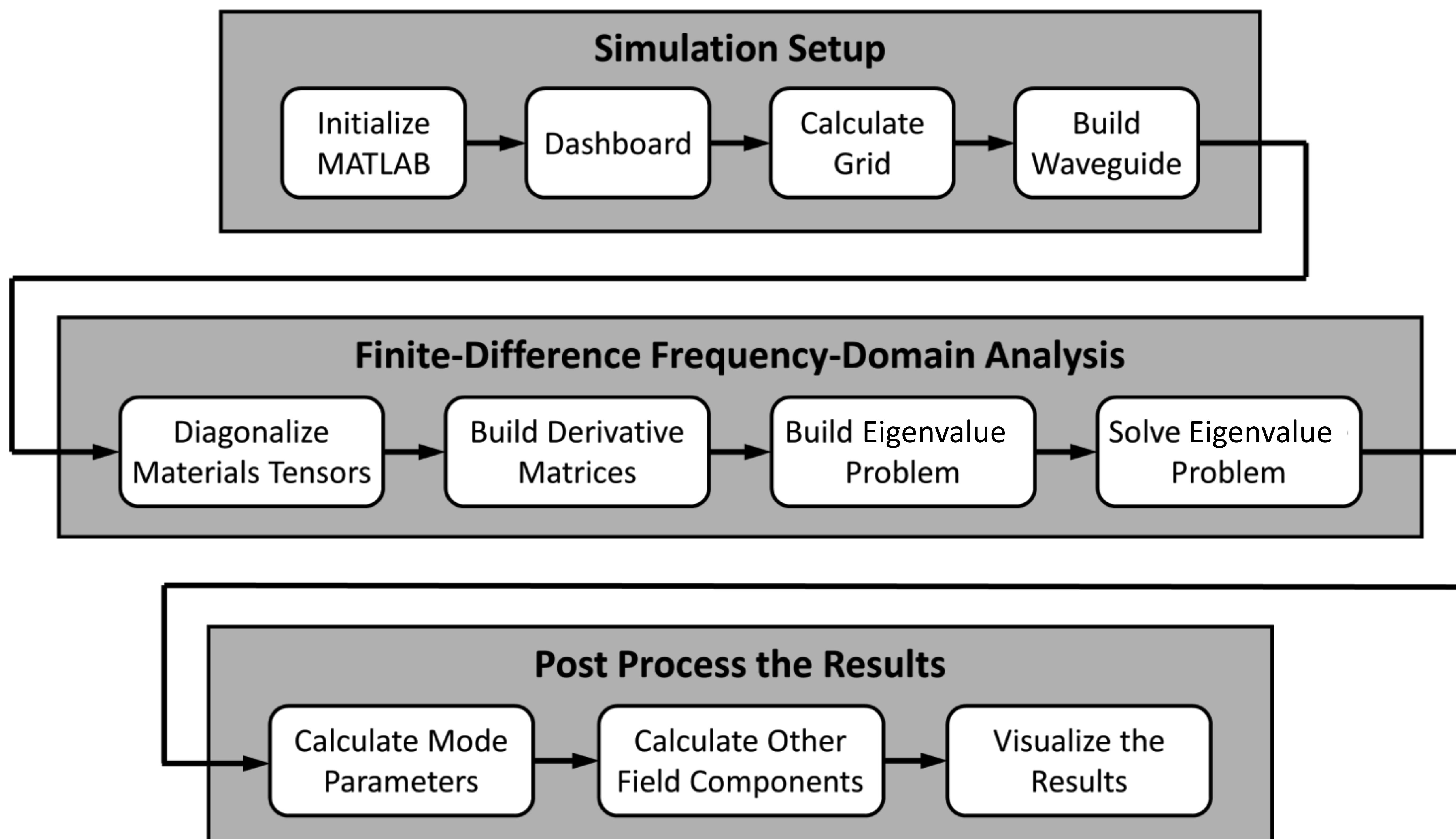
**Figure 6.6** EIM used to reduce a three-dimensional OIC to a two-dimensional representation.

E mode is not a correct choice. The two-dimensional simulation of the OIC should be E mode to place the electric field perpendicular to the plane of the OIC. Following a similar line of reasoning, if the rib waveguide mode were horizontally polarized, the slab waveguide analyses would need to be E mode while the two-dimensional simulation of the OIC would need to be H mode. Be careful when applying the EIM because identifying the correct polarizations can be tricky!

## 6.3 Implementation of Waveguide Mode Calculations

Calculating guided modes using FDFD consists of three major steps, as illustrated in Figure 6.7. Step 1 calculates everything that is needed for the FDFD analysis and starts by initializing MATLAB. It is followed by the dashboard where all of the parameters are defined that control the simulation. The program then moves on to calculating the grid, which includes the number of cells and the resolution. With the grid calculated, the waveguide is built onto the grid producing the tensor arrays  $\mathbf{ER}_{xx}$ ,  $\mathbf{ER}_{yy}$ ,  $\mathbf{ER}_{zz}$ ,  $\mathbf{UR}_{xx}$ ,  $\mathbf{UR}_{yy}$ , and  $\mathbf{UR}_{zz}$  used to build the eigenvalue problem in the FDFD method.

With the simulation setup, the code goes on to perform the actual FDFD analysis. Everything in this section of code works toward solving the eigenvalue problem. It starts by reshaping the materials arrays into diagonal matrices and then building the derivative matrices. From here, the eigenvalue problem is constructed. For hybrid mode analysis, this entails calculating  $\mathbf{P}$ ,  $\mathbf{Q}$ , and then  $\mathbf{\Omega}^2$  using (6.38), (6.39), and (6.42), respectively. For slab waveguide analysis, this entails calculating the  $\mathbf{A}$  and  $\mathbf{B}$  matrices in (6.58) for the E mode or (6.64) for the H mode. When the eigenvalue problem is constructed, it is solved using the MATLAB function `eigs()` to calculate the eigenvectors and eigenvalues, referred to collectively as the eigenmodes. It is



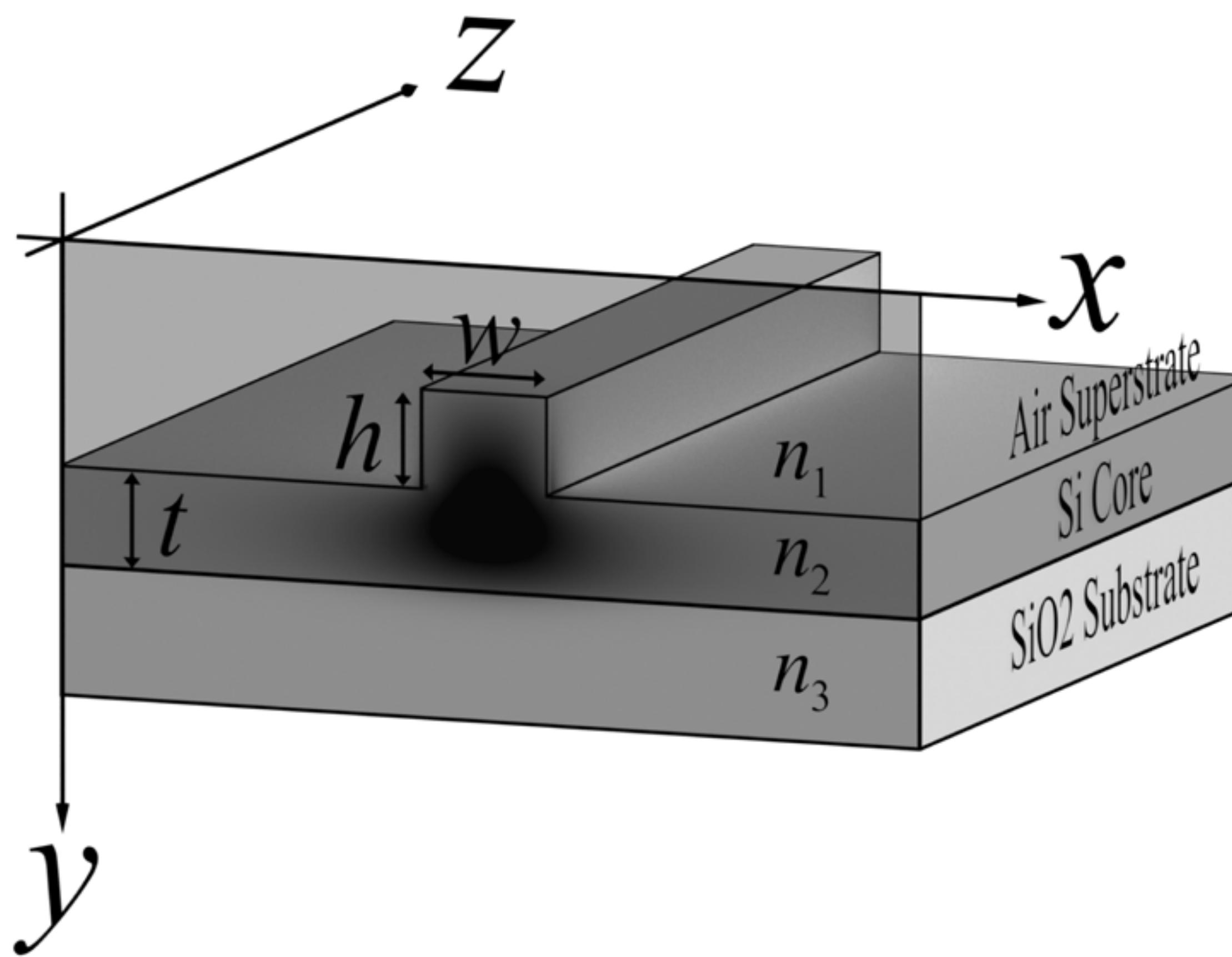
**Figure 6.7** Block diagram of FDFD analysis of waveguides.

less efficient and usually unnecessary to calculate all possible eigenmodes. Instead, it is best to calculate only a small set of eigenmodes. If a good guess can be made for the eigenvalue  $\tilde{\gamma}_0^2$  of the fundamental mode, it is possible to calculate only a small set of eigenmodes with eigenvalues closest to  $\tilde{\gamma}_0^2$ . The small set will be the lowest-order eigenmodes. For dielectric waveguides, the majority of power in the fundamental mode typically resides in the core. For this reason, a good guess for the eigenvalue of the fundamental mode is  $\tilde{\gamma}_0^2 = -n_{\text{core}}^2$ , where  $n_{\text{core}}$  is the refractive index of the core. For other waveguides, some experimentation may be needed to determine a good guess for the eigenvalue of the fundamental mode.

Last, the eigenmodes are analyzed and postprocessed to extract meaning from the analysis. Oftentimes, this starts by calculating the various mode parameters like the complex propagation constant  $\gamma$ , attenuation coefficient  $\alpha$ , phase constant  $\beta$ , and/or the effective refractive index  $n_{\text{eff}}$ . Each guided mode will have its own set of parameters and they can be calculated from the eigenvalue  $\tilde{\gamma}_m^2$  using (6.43) to (6.46). The field terms are extracted from the columns of the eigenvector matrix. If needed, the other field components can be calculated. A typical guided mode calculation will end by visualizing the fields of the guided modes.

### 6.3.1 MATLAB Implementation of Rib Waveguide Analysis

An excellent application for rigorous hybrid mode analysis for waveguides is that of a silicon-on-insulator (SOI) rib waveguide that is commonly used in integrated optics [3–6]. The waveguide has an inhomogeneous dielectric so it does not strictly support TE and TM modes. The geometry and coordinate setup for a typical SOI rib waveguide is shown in Figure 6.8. Let propagation be in the  $z$ -direction and the cross section of the waveguide fall in the  $xy$  plane. The figure shows the waveguide,



**Figure 6.8** Geometry and coordinate setup for Sol rib waveguide. The dimensions are  $w = 0.8 \mu\text{m}$ ,  $h = 0.6 \mu\text{m}$ , and  $t = 0.6 \mu\text{m}$ . The refractive indices are  $n_1 = 1.0$ ,  $n_2 = 3.5$ , and  $n_3 = 1.5$ . The waveguide is analyzed at free space wavelength  $\lambda_0 = 1.55 \mu\text{m}$ .

the coordinate axes, dimensions, refractive indices, and the amplitude profile of the fundamental mode supported by the rib waveguide.

The analysis will be performed at the free space wavelength  $\lambda_0 = 1.55 \mu\text{m}$ , as this is a standard wavelength for telecommunications [7]. The region above the waveguide is called the superstrate and is air in this example. Air will be assigned a refractive index of  $n_1 = 1.0$ . The middle layer and rib are made of silicon (Si) which has a refractive index close to  $n_2 = 3.5$ . The region below the waveguide is called the substrate and is silicon dioxide ( $\text{SiO}_2$ ) in this example. The substrate will be assigned a refractive index of  $n_3 = 1.5$ . The width of the rib is  $w = 0.8 \mu\text{m}$ , the height of the rib is  $h = 0.6 \mu\text{m}$ , and the thickness of the silicon layer away from the rib is  $t = 0.6 \mu\text{m}$ .

The MATLAB code to analyze this rib waveguide can be downloaded at <https://empossible.net/fdfdbook/> and is called `Chapter6_ribwaveguide.m`. The code begins with the header from lines 1 to 11. The first line is the name of the MATLAB file. In this case, the file is saved as “`Chapter 6_ribwaveguide.m`.” The first task performed is initializing MATLAB. This consists of `close all` to close any figure windows that may happen to be open, `clc` to clear the command window, and the most important `clear all` that clears all variables from memory. Next, the units used in the simulation are defined. For photonic waveguide analysis, `micrometers` is set to 1. Any other units, such as nanometers, should be defined relative to `micrometers`.

The next section of the code from lines 12 to 33 is the dashboard. The dashboard contains all of the parameters that define and control the simulation. Parameters to be defined inside of the dashboard include the free space wavelength, waveguide parameters such as dimensions and refractive indices, and grid parameters that control the size and resolution of the grid.

The FDFD analysis here calculates the set of modes that the waveguide can support so the analysis does not require a source. However, the free space wavelength `lam0` for the analysis must still be defined. The parameter `lam0` is the first thing defined in this dashboard on line 17. The second set of parameters defines everything that is needed about the waveguide in order to calculate the modes it supports. This includes the refractive index of the superstrate `rib_n1`, refractive index of the core `rib_n2`, refractive index of the substrate `rib_n3`, height of the rib

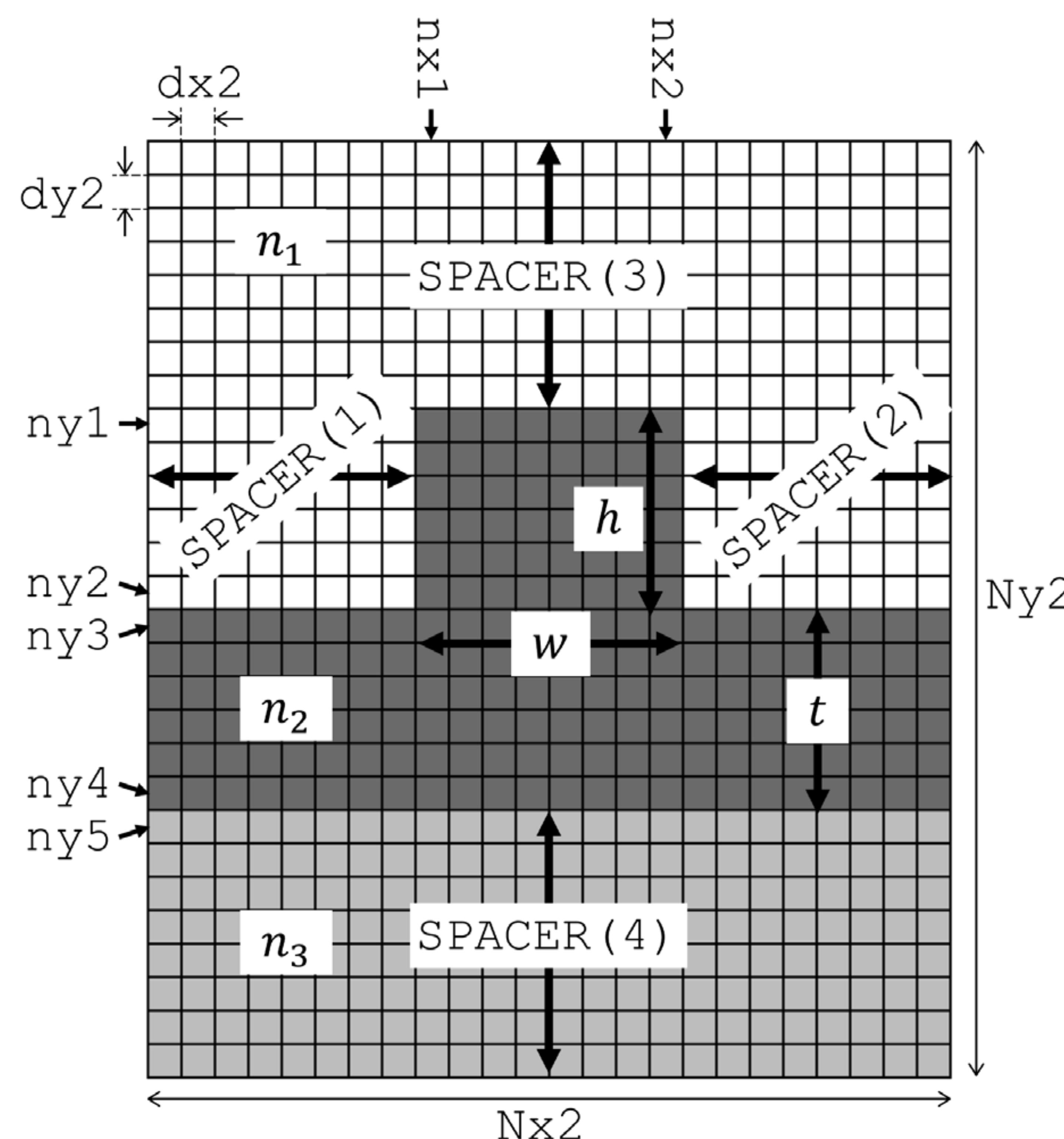
`rib_h`, thickness of the silicon layer away from the rib `rib_t`, and width of the rib `rib_w`. These parameters are as consistent as possible with those shown in Figure 6.8. The reason for the `rib_` prefix is to avoid using common variable names that may accidentally get overwritten with different information later in the code. The third set of parameters defines everything that is needed to calculate the grid. The first parameter `nmax` is the largest refractive index that will be assigned to any cells on the grid. This is needed in order to determine the smallest wavelength the analysis will be required to resolve,  $\lambda_{\min} = \lambda_0/n_{\max}$ . The second parameter is `NRES`. This is the number of points that will be used to resolve the shortest wavelength  $\lambda_{\min}$ . Higher values will improve accuracy by resolving the problem with more points, but the calculations will require more memory and take more time to run. Values in the range of 10 to 40 are typical, but it is necessary to test for convergence to be sure that sufficient resolution is being used. Last, it is necessary to put some space around the rib waveguide so that the guided mode decays to zero before reaching the boundary. This allows simple Dirichlet boundary conditions to be used. Periodic boundary conditions cannot be used since the guided mode is not periodic. `SPACER` is an array containing four numbers that define how much space there will be from the edge of the waveguide to the four grid boundaries. The first number in `SPACER` is the amount of space on the left of the waveguide, the second number is the amount of space on the right, the third number is the amount of space above, and the fourth number is the amount of space below. In the code, the air superstrate will occupy the entire spacer region above the waveguide. The substrate with refractive index  $n_3$  will occupy the entire spacer region below the waveguide. The last parameter defined in the dashboard is the number of modes to calculate `NMODES`. If the grid has 200 by 100 points, 20,000 modes will be calculated unless specified otherwise. Calculation time is greatly improved when only a subset is calculated. In this case, three modes will be calculated. If the number of supported modes is to be determined, `NMODES` will need to be increased until the additional modes are clearly not guided modes. Identifying which modes are guided and which are not will be discussed near the end of this section.

The next section of code from lines 35 to 66 calculates the grid for the waveguide analysis. This includes the number of cells on the grid, `Nx` and `, as well as the grid resolution parameters, dx and dy. The first thing in this section is to calculate preliminary values for the grid resolution parameters. The parameters dx and dy will be adjusted in the second step. Lines 40 and 41 set the preliminary value for both dx and dy equal to the minimum wavelength  $\lambda_{\min}/n_{\max}$  divided by NRES. This calculation does not consider the dimensions of the waveguide at all so it is highly likely that the dimensions will not match the grid perfectly. In fact, at this point in the code, the width of the rib is 36.13 cells wide. Since it is not possible to fill in a fraction of a cell, the simulation will not accurately represent the width of the rib. Lines 43 to 47 adjust dx and dy so that the most critical dimensions of the waveguide are represented exactly by an integer number of cells on the grid. This practice greatly improves the convergence rate of the simulation so it will be possible to achieve high accuracy with a minimum number of cells on the grid. To do this, a critical dimension is chosen for each axis. For the  $x$ -direction, the critical dimension is chosen to be rib_w because it was the only dimension to choose. Given`

the critical dimension, the number of cells representing that dimension is calculated and rounded up to the nearest integer according to  $nx = \text{ceil}(\text{rib\_w}/dx)$ . The resolution parameter is then adjusted by recalculating it as the critical dimension divided by the number of cells just calculated according to  $dx = \text{rib\_w}/nx$ . When done correctly, `rib_w` will be exactly equal to  $nx*dx$ . For the  $y$ -direction, it is not clear whether `rib_t` or `rib_h` is the critical dimension so `rib_t` was chosen. The number of cells representing this dimension is calculated and rounded up to the nearest integer according to  $ny = \text{ceil}(\text{rib\_t}/dy)$ . The resolution parameter is then adjusted by recalculating it as the critical dimension divided by the number of cells just calculated according to  $dy = \text{rib\_t}/ny$ . When done correctly, `rib_t` will exactly equal  $ny*dy$ . Some experimentation with the analysis can give clues about what may be the most important dimension to resolve exactly. After this step, both `dx` and `dy` will usually be slightly smaller than they were initially calculated. Next, the size of the grid is calculated on lines 49 to 56. First, the physical width of the grid `Sx` is calculated on line 50 as the leftmost SPACER region plus the width of the rib plus the rightmost SPACER region. From this, the numerical size of the grid, or the number of grid cells wide, is calculated on line 51 by dividing the physical width by the cell size and rounding up to the nearest integer. Due to the rounding operation, the physical size of the grid may be slightly off so it is recalculated on line 52 as the number of cells wide multiplied by the cell size. The same actions are performed for the vertical size of the grid from lines 54 to 56, but the size of the waveguide is the height of the rib plus the thickness of the silicon layer. At this point, the grid is calculated. If needed, the  $2\times$  grid parameters are calculated followed by the axis vectors to be used for graphics and visualization.

The next section of code from lines 68 to 98 builds the rib waveguide onto the Yee grid. This is done by building the rib waveguide onto the  $2\times$  grid and then extracting the permittivity and permeability tensor arrays on the Yee grid from the  $2\times$  grid. Before anything is added to the  $2\times$  grid, the permittivity and permeability arrays `ER2` and `UR2` on the  $2\times$  grid are initialized to air on lines 73 and 74. Observe that the refractive index is squared because it is relative permittivity that must be assigned to points on the grid, not the refractive index. The two are related through  $\epsilon_r = n^2$ . Forgetting to square the refractive index is one of the most common mistakes made in photonic FDFD simulations.

The second task is to calculate the array indices of where the structures of the waveguide begin and end on the  $2\times$  grid. This is implemented on lines 76 to 84. The overall grid strategy for this is illustrated in Figure 6.9 showing the spacer regions around the rib waveguide, the refractive indices, and the array indices. It is highly recommended to draw a figure like this before building anything onto a grid. Observe that `nx1` and `nx2` are the array indices for where the rib part of the waveguide begins and ends in the  $x$ -direction. The array index `nx1` is calculated as one plus the number the cells wide of the leftmost spacer region. The +1 is included so that the left side of the device is outside of the left spacer region. There is no need to get `nx1` exact to anything. What is important is that `nx2` is located correctly relative to `nx1`. The array index `nx2` is calculated as `nx1` plus the number of cells wide for the rib `round(rib_w/dx)` rounded to the nearest integer minus one. As described in Chapter 1, the minus one is required so that the dimension is not one cell greater



**Figure 6.9** Grid strategy for modeling a rib waveguide on the  $2\times$  grid.

than intended. While one cell may seem small and insignificant, being one cell off can greatly affect the convergence rate and accuracy of some devices. Similarly, the vertical array indices  $ny_1$ ,  $ny_2$ ,  $ny_3$ ,  $ny_4$ , and  $ny_5$  are calculated.

From here, it is an easy task to assign permittivity values to the  $2\times$  grid. The rib is added first on line 87, followed by the silicon layer on line 88, followed by the substrate on line 89. There is no need to assign values for the superstrate because the array  $ER2$  was initialized with the relative permittivity of the superstrate. Observe on line 89 the array is filled from  $ny_5$  to  $Ny_2$  in the  $y$ -direction because  $Ny_2$  is the array index for the bottom of the grid. The last task from lines 91 to 98 is to extract the tensors arrays on the Yee grid from the  $2\times$  arrays.  $ER_{xx}$ ,  $ER_{yy}$ , and  $ER_{zz}$  are extracted from  $ER2$  while  $UR_{xx}$ ,  $UR_{yy}$ , and  $UR_{zz}$  are extracted from  $UR2$ .

With the grid setup and waveguide constructed onto the Yee grid, it is time to perform the FDFD analysis of the waveguide. This section of code from lines 100 to 141 is generic, and the same code can be used to calculate the hybrid modes of any waveguide. Before the matrices for the eigenvalue problem can be calculated, the tensor arrays must be converted to diagonal matrices. For  $ER_{xx}$ , this is done by reshaping the array to a column vector via  $ER_{xx}(:)$ , declaring it as a sparse matrix via  $\text{sparse}(ER_{xx}(:))$ , and then inserting it as a diagonal into a square matrix via  $\text{diag}(\text{sparse}(ER_{xx}(:)))$ . This last step creates a sparse matrix by default. It is very important not to reverse the order of the  $\text{sparse}()$  and  $\text{diag}()$  commands as this will temporarily create a full matrix that will consume a lot of memory and take a long time to process if not crash the computer. The same operation is done for all the material tensor arrays to convert them to diagonal matrices. In this code, the

same variable names are used so the two-dimensional arrays are overwritten by the diagonal matrices. Next, the derivative matrices are constructed on lines 112 to 117 by calling the function `yeeder2d()` described in Chapter 4. `NS` is an array containing the size of the grid `Nx` and `, RES is an array containing the grid resolution parameters dx and dy, and BC is an array containing numbers that define the boundary conditions to be used. For waveguide analysis, it is common to use Dirichlet boundary conditions so BC is set to [0 0]. The formulation used normalized grid coordinates so the input argument RES is multiplied by the free space wavenumber k0 to pass normalized grid resolution terms to yeeder2d().`

Given the diagonal materials matrices and the derivative matrices, the matrices for the eigenvalue problem are calculated on lines 119 to 123. This is done by first calculating the intermediate matrices `P` and `Q` and then calculating the matrix for the eigenvalue problem as `P*Q`. Rather than waste memory storing the `P*Q` matrix, the product is used as the input argument to `eigs()` on line 127. The function `eigs()` is built into MATLAB to solve eigenvalue problems of sparse matrices. There are many different ways this function can be used and many different options for using it. For this example, three input arguments are passed to the function. The first input argument is the matrix `P*Q`. The second input argument is the number of modes to calculate from the eigenvalue problem. The parameter `NMODES` was defined in the dashboard to be 3, so three modes will be calculated that have eigenvalues closest to the value given as the third input argument. Thus, the third input argument needs to be set to the best guess for the eigenvalues of the guided modes. Recall from Chapter 2 for dielectric waveguides that the effective refractive index is close to the average refractive index calculated over the area of the guided mode. Given that the modes reside mostly inside of the silicon material,  $n_2$  is a decent estimate of the effective refractive index of the guided modes. However, the eigenvalues are not the effective refractive index. From (6.43) and (6.46), the eigenvalue that would correspond to an effective index of  $n_2$  is

$$\tilde{\gamma}^2 \approx -n_2^2 \quad (6.65)$$

The function `eigs()` will then return two matrices that are called `Exy` and `D2` in the MATLAB code. If there are  $M$  total cells on the grid, the eigenvector matrix `Exy` will have  $2M$  rows and `NMODES` number of columns. Each column in `Exy` contains the  $\mathbf{e}_x$  and  $\mathbf{e}_y$  column vectors from (6.41). The matrix `D2` will be  $NMODES \times NMODES$  and contain the eigenvalues along its center diagonal. Eigenvectors and eigenvalues come in pairs, so the  $m$ th column in `Exy` corresponds to the eigenvalue in the  $m$ th diagonal position of `D2`. For three modes, these matrices have the form below.

$$\mathbf{E}_{xy} = \begin{bmatrix} \left[ \begin{array}{c} \mathbf{e}_{x,1} \\ \mathbf{e}_{y,1} \end{array} \right] & \left[ \begin{array}{c} \mathbf{e}_{x,2} \\ \mathbf{e}_{y,2} \end{array} \right] & \left[ \begin{array}{c} \mathbf{e}_{x,3} \\ \mathbf{e}_{y,3} \end{array} \right] \end{bmatrix} \quad \mathbf{D}^2 = \begin{bmatrix} \tilde{\gamma}_1^2 & 0 & 0 \\ 0 & \tilde{\gamma}_2^2 & 0 \\ 0 & 0 & \tilde{\gamma}_3^2 \end{bmatrix} \quad (6.66)$$

The square of the normalized complex propagation constant has no meaning. For this reason, it is a common practice to first calculate the square root of the

eigenvalue matrix to calculate the normalized complex propagation constants of the modes. This happens on line 128.

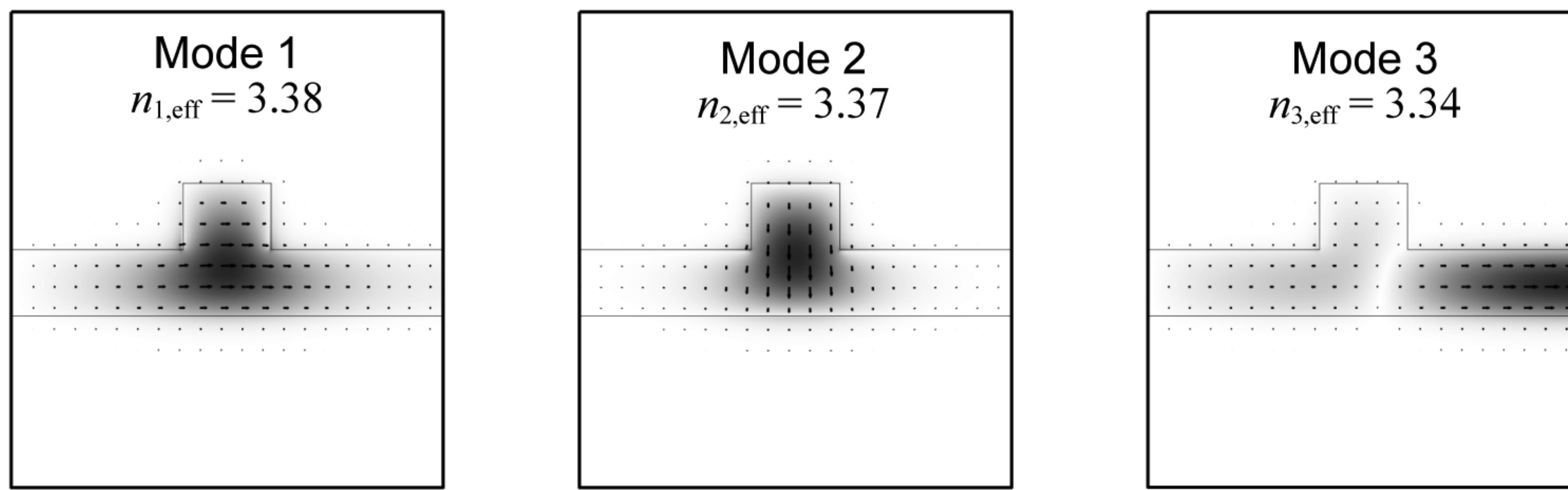
$$\mathbf{D} = \sqrt{\mathbf{D}^2} = \begin{bmatrix} \tilde{\gamma}_1 & 0 & 0 \\ 0 & \tilde{\gamma}_2 & 0 \\ 0 & 0 & \tilde{\gamma}_3 \end{bmatrix} \quad (6.67)$$

For photonics applications, it is usually the effective refractive index of the guided modes that is of interest. These are readily calculated from the normalized complex propagation constants simply by multiplying them by  $-j$ . At the same time, the effective refractive indices are stored in a one-dimensional array `NEFF` instead of a diagonal matrix by extracting the diagonal from the eigenvalue matrix and then multiplying by  $-j$ . This happens on line 128.

At this point, the finite-difference analysis of the rib waveguide is complete. Any remaining steps are considered postprocessing. Solving the eigenvalue problem only calculates the electric field components  $\mathbf{e}_x$  and  $\mathbf{e}_y$ , so a common next step is to calculate the other field components  $\mathbf{e}_z$ ,  $\mathbf{h}_x$ ,  $\mathbf{h}_y$ , and  $\mathbf{h}_z$ . The MATLAB code that does this extends from lines 131 to 141. In this code,  $M$  is the total number of points on the grid and is needed in order to extract the  $x$  and  $y$  field components from the eigenvectors. Immediately after calculating  $M$ , the  $x$  and  $y$  components of the electric field are extracted from the eigenvector matrix `Exy` on lines 133 and 134. Afterward, `Ex` and `Ey` will be rectangular matrices because they are essentially the top and bottom halves of the square matrix `Exy`. Next, the eigenvector matrix for the magnetic fields `Hxy` is calculated using (6.40). This matrix contains the  $x$  and  $y$  components of the magnetic fields. These are extracted from `Hxy` as `Hx` and `Hy` and are also rectangular matrices like `Ex` and `Ey`. Last, the  $z$  components of the fields are calculated using (6.26) and (6.27).

Another common thing to be done after finite-difference analysis is visualizing the fields. The MATLAB code to visualize the  $x$  and  $y$  components of the electric fields extends from lines 143 to 173. To make the figure compact, the other field components are not visualized. In most cases, the  $z$  component of the electric field is small and almost all information about the mode profile is contained in just  $x$  and  $y$  components. The code is easily modified to visualize additional field components if that is desired. The visualization code has a loop that iterates through all of the calculated modes and visualizes the  $x$  and  $y$  components horizontally in the figure window. The first step is to extract the column vectors  $\mathbf{e}_x$  and  $\mathbf{e}_y$  from columns of the `Ex` and `Ey` matrices, respectively, and reshape them back to the two-dimensional Yee grid. The second step is to normalize the values in the field arrays so that the maximum value is 1. The third step is to plot `Ex` on the left of the row and `Ey` on the right. The effective refractive index of the mode is added to the title of the plot of `Ex`.

While the MATLAB code may be completed at this point, the analysis of the waveguide is not at all complete until a convergence study has been performed. This will be discussed in more detail later, but good convergence was found for `NRES=30` and spacer regions around  $1.0\lambda_0$ . Using these settings, the grid size was calculated to be `Nx=269` and  `and the total calculation time was around 2.3 seconds on a laptop computer running a 2.30 GHz Intel Core i9-9880H.`



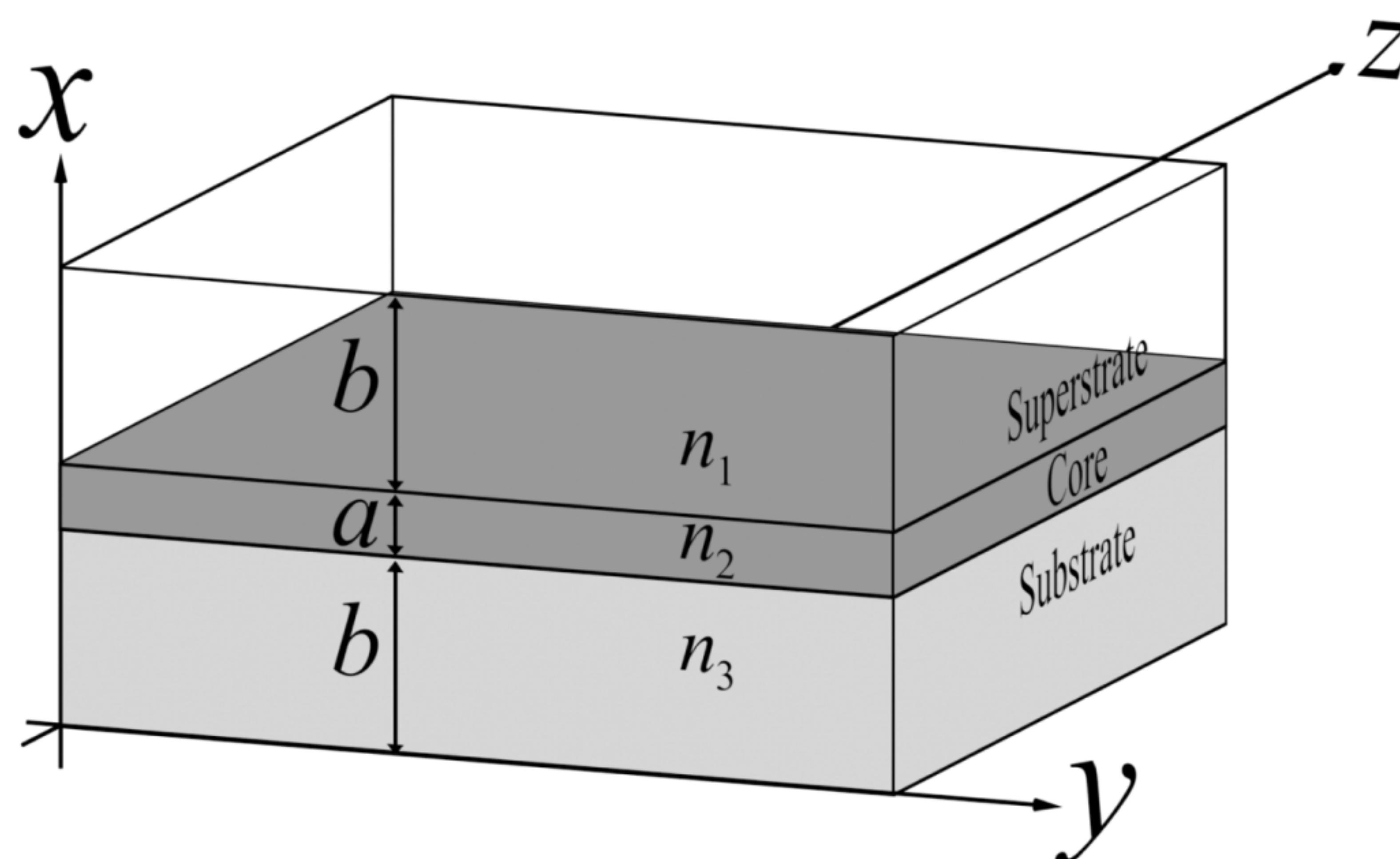
**Figure 6.10** First three eigenmodes from the FDFD analysis of a rib waveguide. Gray shading conveys field amplitude and black arrows convey polarization. The first two modes are guided modes because their power is confined to the rib region. The third mode is not a guided mode because it has power increasing away from the rib region.

The first three eigenmodes calculated from this analysis and the effective refractive index for each are shown in Figure 6.10. In this figure, the gray shading conveys the amplitude of the electric field calculated as  $\sqrt{E_x.^2 + E_y.^2}$ . Simply by inspecting the images of the modes, it is evident that only the first two modes are guided by the rib waveguide. This is evident because the field power of these modes is clearly confined to the vicinity of the rib. The third eigenmode has power increasing away from the rib and reaching a maximum value at the grid boundary. This is not a mode guided by the rib waveguide so it can be concluded that this rib waveguide supports only two modes. It is generally desired to adjust the design of the waveguide so that only a single mode is supported and this finite-difference analysis is a great tool for doing this.

So, what is the meaning of the third eigenmode? Dirichlet boundary conditions were used so it is almost like the rib waveguide is encased inside of a very large metal waveguide. It is not exactly a metal waveguide because the two boundaries where Dirichlet boundary conditions were applied to the magnetic fields make those boundaries a *perfect magnetic conductor* (PMC). At the other two boundaries, Dirichlet boundary conditions were applied to the electric fields so those boundaries have a *perfect electric conductor* (PEC). Only a PEC acts like a true metal. The third eigenmode calculated by this analysis is a mode guided by the larger waveguide, but this does not have any physical meaning related to the rib waveguide and is just ignored.

### 6.3.2 MATLAB Implementation of Slab Waveguide Analysis

Slab waveguide analysis arises frequently enough in photonics and other areas of electromagnetic analysis that the ability to analyze them will prove to be a powerful capability. The effective refractive index of the guided modes can be used to approximate some three-dimensional structures as two-dimensional structures for easier simulation [2]. Slab waveguides are also elements of devices such as guided-mode resonance filters, grating couplers, leaky wave antennas, and more. The geometry and coordinate setup for the basic dielectric slab waveguide is illustrated in Figure 6.11 and is composed of a substrate, core, and superstrate. Without loss



**Figure 6.11** Dielectric slab waveguide where  $n_2 > n_1$  and  $n_2 > n_3$ .

of generality, the propagation direction is set to the  $z$ -direction and the cross section of the waveguide is placed in the  $x$ -direction. The figure shows the waveguide, the coordinate axes, dimensions, and refractive indices. To support guided modes, the core must have a refractive index higher than both the substrate and superstrate. The substrate and superstrate can have different refractive indices, but both must be less than the core refractive index in order to form a waveguide. The thickness  $a$  of the core is the only dimension needed to define the slab waveguide. If  $n_1 \neq n_3$ , there will be some minimum value of  $a$  below which the slab waveguide will not support any guided modes. The dimension  $b$  is provided to ensure the superstrate and substrate regions are large enough to accurately analyze the waveguide. Dirichlet boundary conditions were used at the extreme top and bottom of the grid. These must be sufficiently far away from the core so that the guided modes decay to zero well before reaching the boundaries of the grid. Typically,  $b > 3\lambda_0$ . For this analysis, the top spacer region will be the medium above the core filled with refractive index  $n_1$  and the bottom spacer region will be the medium below the core filled with refractive index  $n_3$ .

The analysis will be performed at the free space wavelength  $\lambda_0 = 1.55 \mu\text{m}$ . The superstrate will be air ( $n_1 = 1.0$ ), the core region will be silicon nitride ( $n_2 = 2.0$ ), and the substrate will be fused silica ( $n_3 = 1.5$ ). The thickness of the slab will be  $a = 1500 \text{ nm}$ . A correct choice for  $b$  will be determined through simulation, but a good guess to start with is three wavelengths  $b = 3\lambda_0$ . The MATLAB code to analyze this slab waveguide can be downloaded at <https://empossible.net/fdfdbbook/> and is called `Chapter6_slabwaveguide.m`. The program starts with a header that is identical to the header for the rib waveguide discussed previously, except that the first line conveys this code was given a different name. The header initializes MATLAB and defines the units that will be used. For this analysis, `micrometers` was set to 1 because it is a photonics simulation with dimensions closest to this scale.

The next section of the code from lines 12 to 32 is the dashboard. The dashboard contains all of the hard-coded numbers that define and control the simulation. First, the free space wavelength `lam0`, followed by whether it is E or H modes to be analyzed. To calculate E modes, the mode parameter should be set to `MODE = 'E.'` To calculate H modes, the mode parameter should be set to `MODE`

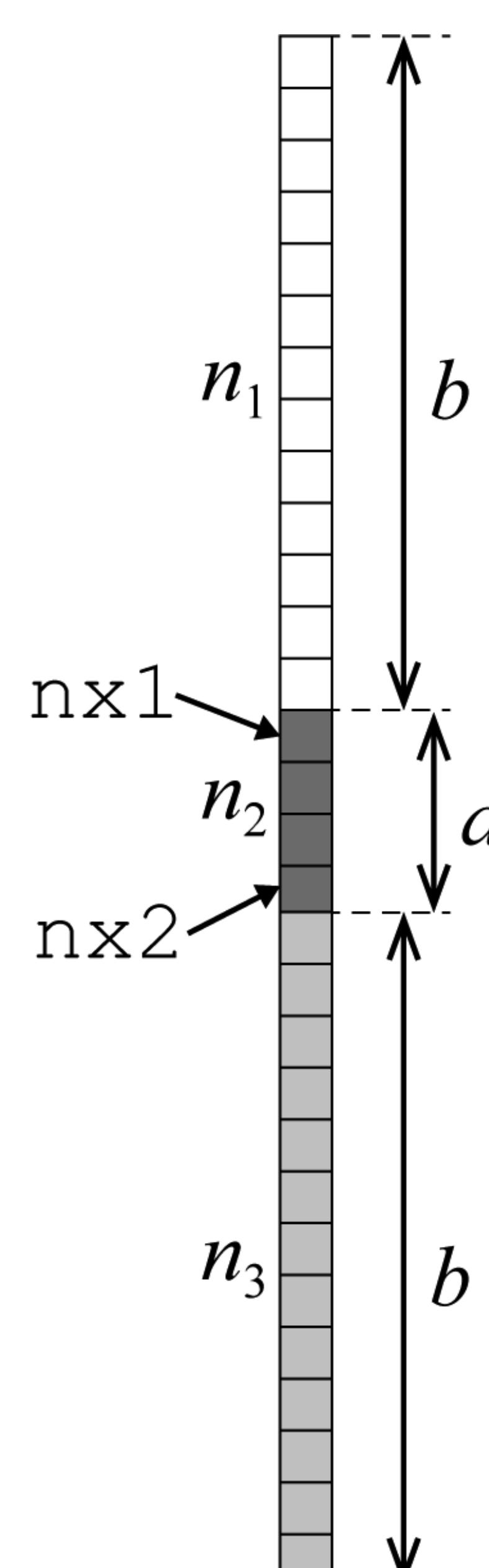
= ‘H.’ The second set of parameters defines everything that is needed about the slab waveguide. This includes the refractive index of the superstrate  $n_1$ , refractive index of the core  $n_2$ , refractive index of the substrate  $n_3$ , and thickness of the core  $a$ . Observe that the parameter  $b$  is not defined here because  $b$  is not a parameter that defines the slab waveguide. Instead, it is a simulation parameter that must be made sufficiently large so that the grid boundaries are far enough away that the modes guided by the slab are calculated accurately. A sufficient amount of space must be placed between the waveguide and the Dirichlet boundary conditions, just like for the rib waveguide analysis. The third set of parameters defines everything that is needed to calculate the grid. The first parameter  $n_{\max}$  is the largest refractive index that will be assigned to any cells on the grid. This is needed in order to determine the smallest wavelength the analysis will be required to resolve,  $\lambda_{\min} = \lambda_0/n_{\max}$ . The second parameter is  $N_{RES}$ . This is the number of points that will be used to resolve the shortest wavelength  $\lambda_{\min}$ . Higher values of  $N_{RES}$  will improve accuracy by resolving the problem with more points, but the calculations will take longer and require more memory to run. Values in the range of 10 to 40 are typical, but it is necessary to test for convergence to be sure that sufficient resolution is being used. Last, the parameter  $b$  defines the amount of space to include outside of the core to ensure the guided modes decay to zero before reaching the boundary of the grid. This allows simple Dirichlet boundary conditions to be used. The last parameter defined in the dashboard is the number of modes to calculate,  $N_{MODES}$ . If the grid has 200 cells, 200 eigenmodes will be calculated unless specified otherwise. It is most efficient to calculate only a subset of the modes because in most simulations the vast majority of eigenmodes are not modes guided by the slab and have no physical meaning related to the slab waveguide. In this case, four modes will be calculated. If the number of supported modes is to be determined,  $N_{MODES}$  will need to be increased until the additional modes are clearly not guided modes.

The next section of code from lines 35 to 56 calculates the one-dimensional grid that will be used to represent the slab waveguide. The first thing in this section is to calculate the first guess at the grid resolution parameter  $dx$ . This is the first guess because  $dx$  is refined in the second step. The grid resolution  $dx$  is set equal to the minimum wavelength  $\lambda_0/n_{\max}$  divided by  $N_{RES}$ . This calculation does not consider the thickness of the slab waveguide so it is highly unlikely that the dimensions will fit on the grid perfectly. In fact, at this point in the code, the core is 19.35 cells. Since it is not possible to fill in a fraction of a cell, the simulation is not able to accurately represent the thickness of the core. The second step in this section of the code adjusts  $dx$  so that the thickness of the slab is represented exactly by an integer number of cells on the grid. Adjusting  $dx$  in this manner greatly improves the convergence rate of the simulation so it will be possible to achieve high accuracy with a minimum number of points on the grid. To do this, line 42 calculates the number of cells currently representing the thickness of the core and rounds it up to the nearest integer. The resolution parameter  $dx$  is adjusted on line 43 by recalculating it as the slab thickness divided by the number of cells just calculated. After this step,  $dx$  will be slightly smaller than it was initially calculated. Next, the size of the grid is calculated on lines 45 and 46. First, the physical size of the grid  $Sx$  is calculated as two regions of thickness  $b$  and one region of thickness  $a$ . From this, line 47 calculates the total number of grid cells by dividing the physical size by the

cell size and rounding up to the nearest integer. Due to the rounding operation, the physical size of the grid  $S_x$  may be slightly off so line 48 recalculates it as the number of cells wide times the cell size. If needed, lines 54 to 56 calculate the  $2 \times$  grid parameters followed by the axis arrays to be used for graphics and visualization. The  $2 \times$  grid is used here for illustration purposes and consistency, but it is less common to use the technique for slab waveguide analysis.

The next section code from lines 58 to 81 builds the slab waveguide onto the one-dimensional Yee grid. This is done by building the slab waveguide onto the  $2 \times$  grid and then extracting the permittivity and permeability tensor arrays on the Yee grid from the  $2 \times$  grid. Before anything is built onto the  $2 \times$  grid, the permittivity and permeability arrays  $ER2$  and  $UR2$  on the  $2 \times$  grid are initialized to air. The second task is to calculate the array indices  $nx1$  and  $nx2$  of where the core begins and ends on the  $2 \times$  grid, respectively. The overall grid strategy for this is illustrated in Figure 6.12. The array index  $nx1$  is calculated as one plus the number of the cells comprising the superstrate of size  $b$ . There is no need to get  $nx1$  exact to anything. It is most important to calculate  $nx2$  correctly relative to  $nx1$ . The array index  $nx2$  is calculated as  $nx1$  plus the number of cells for the slab rounded to the nearest integer minus one. With the calculated array indices, the slab waveguide is built onto the  $2 \times$  grid on lines 71 to 73. Line 71 adds the superstrate from point 1 up to point  $nx1-1$ . Line 72 adds the core from point  $nx1$  to point  $nx2$ . Line 73 adds the substrate from point  $nx2+1$  all the way down to point  $Nx2$ . Observe that the refractive index is being squared. That is because it is the relative permittivity that is being assigned to points on the grid where  $\epsilon_r = n^2$ . The last task is to extract the tensor arrays on the Yee grid from the  $2 \times$  arrays.  $ERxx$ ,  $ERyy$ , and  $ERzz$  are extracted from  $ER2$  while  $URxx$ ,  $URyy$ , and  $URzz$  are extracted from  $UR2$ .

With the grid setup and slab waveguide built onto the Yee grid, it is time to perform the FDFD analysis. The section of code from lines 83 to 115 is generic, and the same code can be used to calculate the eigenmodes of any slab waveguide. Before the matrices for the eigenvalue problem can be built, the material tensor arrays must



**Figure 6.12** Grid strategy for a slab waveguide.

be converted into diagonal matrices using the same procedure followed in the rib waveguide analysis. This happens on lines 88 to 93. Next, lines 96 to 100 construct the derivative matrices by calling the function `yeeder2d()` described in Chapter 4. `NS` is an array containing the size of the grid. To build derivative matrices for the one-dimensional grid used here, the size of the grid in the  $y$ -direction is set to 1 by making `NS=[Nx 1]`. The resolution array is set to `RES=[dx 1]`, but the specific numerical value given for  $dy$  does not matter since it is not used. The formulation used normalized grid coordinates so the input argument `RES` is multiplied by the free space wavenumber  $k_0$  to pass normalized grid resolution terms to `yeeder2d()`. The boundary condition array is set to `BC=[0 0]`, but the boundary conditions for the  $y$ -axis boundaries also do not matter because the  $y$  derivative matrices are not used in this analysis.

Given the diagonal materials matrices and the derivative matrices, the matrices for the generalized eigenvalue problem  $\mathbf{Ax} = \lambda \mathbf{Bx}$  can be constructed. The matrices  $A$  and  $B$  are constructed differently depending on whether it is the E mode or H mode to be calculated. For the E mode, the matrices are constructed from (6.58). For the H mode, the matrices are constructed from (6.64). Be cautious because it is easy to forget to invert the `URxx` or `ERxx` matrices or including the negative sign when calculating  $B$ , only to get incorrect eigenmodes. The function `eigs()` is built into MATLAB to solve eigenvalue problems of sparse matrices. For this example, four input arguments are passed to the function. The first two input arguments are the  $A$  and  $B$  matrices. The third input argument `NMODES` was defined in the dashboard and is the number of eigenmodes to be calculated. The parameter `NMODES` was defined in the dashboard to be 4, so four modes will be calculated that have eigenvalues closest to the value given as the fourth input argument to `eigs()`. Thus, the fourth input argument is set to the best guess for the eigenvalues of the guided modes. Given that the modes reside mostly inside of the core region,  $n_2$  is a good estimate of the effective refractive index of the guided modes. However, the eigenvalues are not directly the effective refractive index. From (6.43) and (6.46), the eigenvalue that would correspond to an effective index of  $n_2$  is

$$\tilde{\gamma}^2 \approx -n_2^2 \quad (6.68)$$

From here, the function `eigs()` returns two matrices that are called `Fy` and `D2` in the MATLAB code. If there are `Nx` points on the grid, the eigenvector matrix `Fy` will have `Nx` rows and `NMODES` number of columns. Each column in `Fy` contains either the  $\mathbf{e}_y$  column vectors for the E mode or the  $\tilde{\mathbf{h}}_y$  column vectors for the H mode. The eigenvector matrix was given the symbol `F` instead of `E` or `H` because it could represent either electric or magnetic fields. The matrix `D2` will be of size `NMODES×NMODES` and contain the eigenvalues along its diagonal. When four modes are calculated for the E mode, these matrices have the form below.

$$\mathbf{F}_y = \begin{bmatrix} e_{y,1}(1) & e_{y,2}(1) & e_{y,3}(1) & e_{y,4}(1) \\ e_{y,1}(2) & e_{y,2}(2) & e_{y,3}(2) & e_{y,4}(2) \\ \vdots & \vdots & \vdots & \vdots \\ e_{y,1}(N_x) & e_{y,2}(N_x) & e_{y,3}(N_x) & e_{y,4}(N_x) \end{bmatrix} \quad (6.69)$$

$$\mathbf{D}^2 = \begin{bmatrix} \tilde{\gamma}_1^2 & 0 & 0 & 0 \\ 0 & \tilde{\gamma}_2^2 & 0 & 0 \\ 0 & 0 & \tilde{\gamma}_3^2 & 0 \\ 0 & 0 & 0 & \tilde{\gamma}_4^2 \end{bmatrix} \quad (6.70)$$

The square of the normalized complex propagation constant has no meaning so it is common practice to first calculate the square root of the eigenvalue matrix to calculate the normalized complex propagation constants of the modes.

$$\mathbf{D} = \sqrt{\mathbf{D}^2} = \begin{bmatrix} \tilde{\gamma}_1 & 0 & 0 & 0 \\ 0 & \tilde{\gamma}_2 & 0 & 0 \\ 0 & 0 & \tilde{\gamma}_3 & 0 \\ 0 & 0 & 0 & \tilde{\gamma}_4 \end{bmatrix} \quad (6.71)$$

The effective refractive index of the guided modes is readily calculated from the normalized complex propagation constants simply by multiplying by  $-j$ . At the same time, the effective refractive indices are stored in a one-dimensional array NEFF instead of a diagonal matrix by extracting the diagonal from the eigenvalue matrix and then multiplying by  $-j$ .

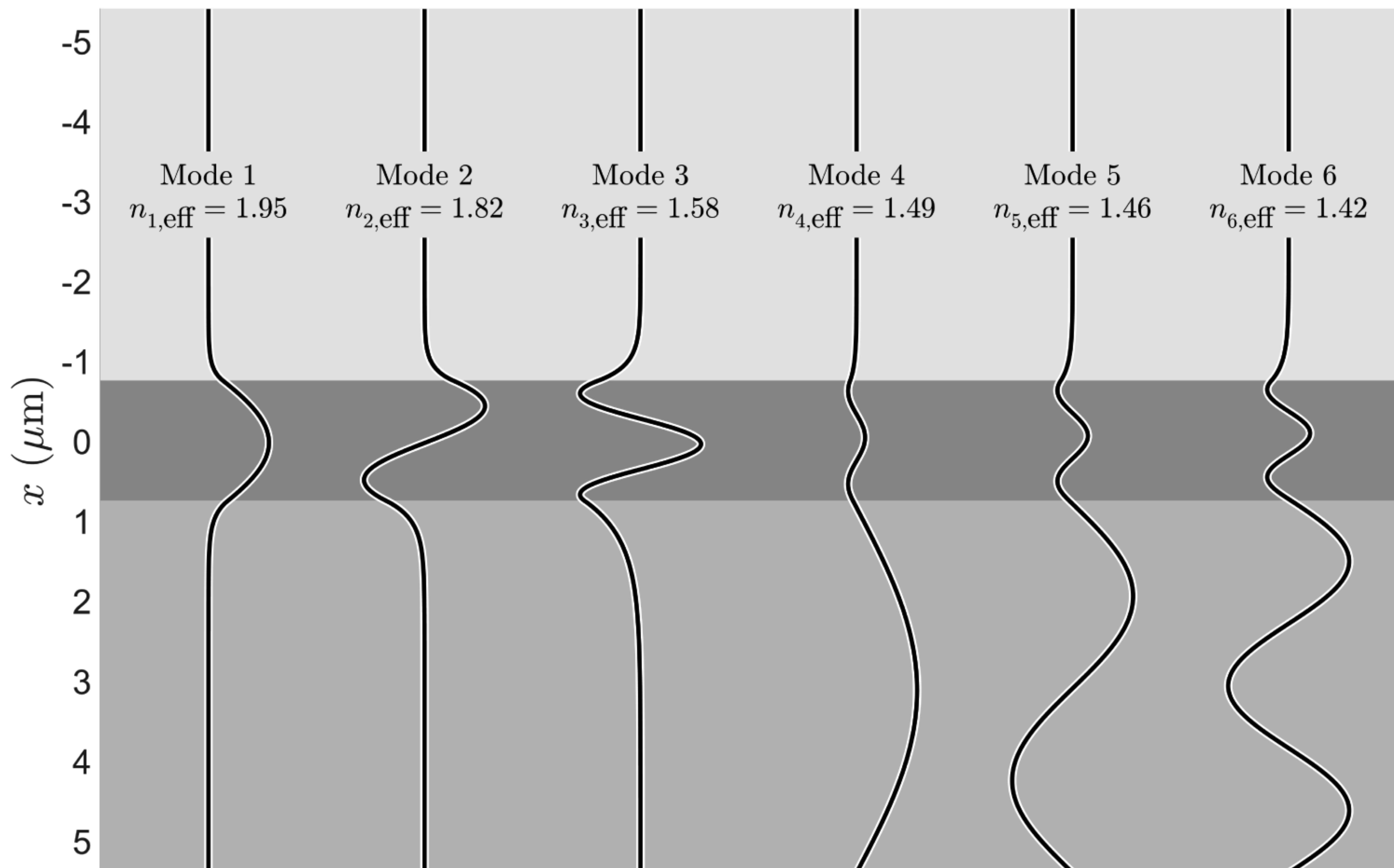
Before the finite-difference analysis of the slab waveguide can be considered complete, a convergence study must be performed. Typically, this is a plot of the effective refractive index of the guided modes as a function of the NRES parameter and as a function of the spacer region parameter  $b$ . Acceptable convergence was found at NRES=20 and  $b=3*1am0$ . Any remaining steps after calculating the modes are considered postprocessing. This may include calculating the field components not directly calculated by solving the eigenvalue problem, visualizing the modes, or something else.

The MATLAB code to visualize the fields in the eigenvector matrix extends from lines 117 to 145. The other field components associated with the modes are not calculated or visualized because almost all information about the mode is contained in just  $e_y$  for the E mode and just  $\tilde{h}_y$  for the H mode. First, the figure window is prepared for visualization by clearing it and declaring a “hold on” statement that allows multiple graphic elements to be superimposed. First, the core of the slab waveguide is drawn to the figure window as a rectangle using MATLAB’s `fill()` function. Two arrays  $x$  and  $y$  are calculated and passed to this function that contain the vertices working around the perimeter of the rectangle. The rectangle is drawn as a light color of gray as  $0.8*[1 1 1]$ . After the core is drawn, the eigenmodes are drawn on top of this so that they can be visualized in relation to the geometry of the slab waveguide. To space the modes more easily, the eigenmodes are normalized so that the maximum value is 1. A `for` loop then draws the modes one at a time. The parameter  $x_0$  is calculated to be the horizontal position of where the mode profile will be plotted. Two lines are drawn to visualize the mode. First is a thick white line and the second is a narrower blue line. This trick gives the line a white glow that will allow the mode profile to stand out regardless of what colors are drawn in the background to represent the waveguide. The lines are then labeled with the effective refractive index of the mode using MATLAB’s `text()` function.

The first six eigenmodes calculated from this analysis and the effective refractive indices are shown in Figure 6.13 for the E mode and Figure 6.14 for the H mode. Inspection of these results shows that only three modes are supported by the slab waveguide for both E and H modes. This is obvious for two reasons. First, the field power for the guided modes is clearly confined to the vicinity of the core. The last three eigenmodes have power increasing away from the core so they are not modes guided by the slab. Second, the effective refractive indices of the unguided modes are less than or equal to the substrate refractive index. Guided modes have the majority of their power in the core, leading to effective refractive indices greater than the refractive indices outside of the core. Like with the rib waveguide analysis, the unguided modes are actually modes guided by the larger waveguide that are artificially formed when using Dirichlet boundary conditions at the grid boundaries. These do not have a physical meaning related to the slab waveguide and should be ignored.

### 6.3.3 Animating the Slab Waveguide Mode

This section demonstrates a fun and educational way to visualize a slab waveguide mode. A GIF animation of a slab waveguide mode propagating through a slab waveguide is generated using the technique described in Chapter 1. Lines 117 to 144 from the original code presented in Section 6.3.2 are replaced with lines 117 to 193 of the alternative code presented in Section 6.3.3. The revised code can be downloaded at <https://empossible.net/fdfdbbook/> and is called `Chapter6_animatedslabmode.m`. Lines 121 to 123 define the name and the number of frames of the GIF animation. Line 126 identifies which mode is to be animated. In this case, the second-order mode is chosen.



**Figure 6.13** First six eigenmodes from E mode analysis of a slab waveguide.