Course Instructor
Dr. Raymond C. Rumpf
Office: A-337
Phone: (915) 747-6958
E-Mail: rcrumpf@utep.edu

Computational Methods in EE

Topic 7c – Numerical Analysis of Slab Waveguides

EE 4386/5301 Computational Methods in EE

Outline

Computational Methods in EE

- Slab Waveguides
- Formulation
- Solution
- Implementation in MATLAB

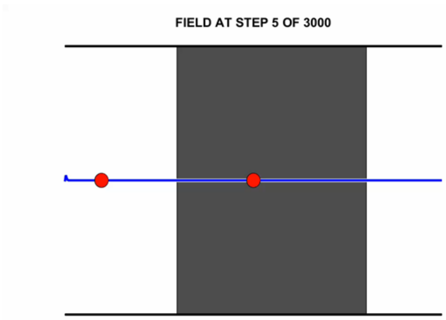
Computational Methods in EE

Slab Waveguides

Refractive Index n

Computational Methods in EE

Light travels at different speeds when it is inside different materials.



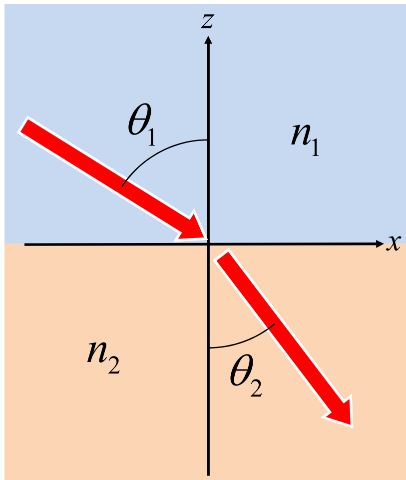
Frequency is constant.
Speed changes.
Wavelength changes.

The factor by which light slows down is called the *refractive index*.

$$n = \frac{c}{v}$$

Snell's Law

Computational Methods in EE

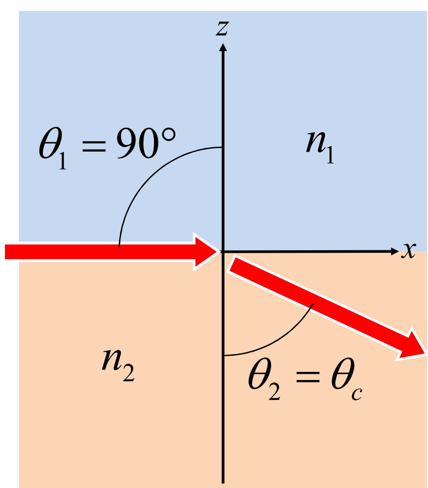


Snell's law quantifies the angles of light rays at an interface.

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$

Critical Angle

Computational Methods in EE



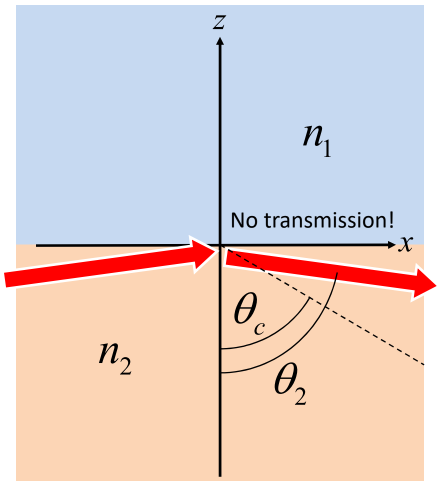
There exists a special angle, the *critical angle*, where the ray in the low-index medium is at 90°.

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$
$$n_1 \sin 90° = n_2 \sin\theta_c$$
$$n_1 = n_2 \sin\theta_c$$
$$\sin\theta_c = n_1/n_2$$
$$\theta_c = \sin^{-1}(n_1/n_2)$$

$$\theta_c = \sin^{-1}(n_1/n_2)$$
$$\text{where } n_2 > n_1$$

Total Internal Reflection (TIR)

Computational Methods in EE



When a light ray is incident onto an interface at an angle greater than the critical angle, the light completely reflects and no light is transmitted.

This is called *total internal reflection* (TIR).

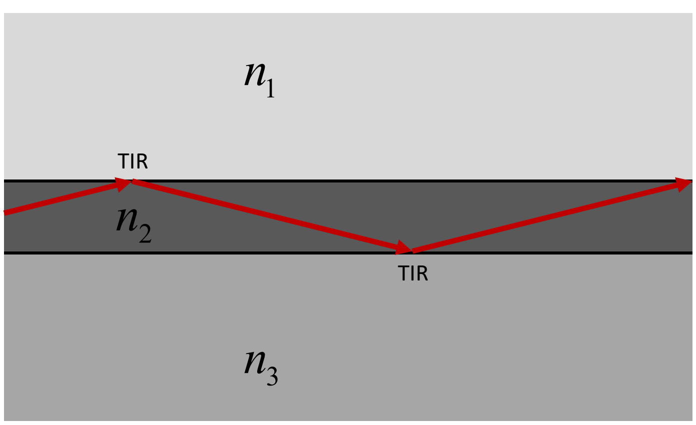
$$\theta_2 > \theta_c$$

Slab Waveguide Analysis — Slide 7

The Slab Waveguide

Computational Methods in EE

If we "sandwich" a slab of high-index material between two materials with lower refractive index, we form a slab waveguide.

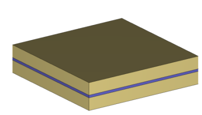


<u>Conditions</u>

$n_2 > n_1$

and

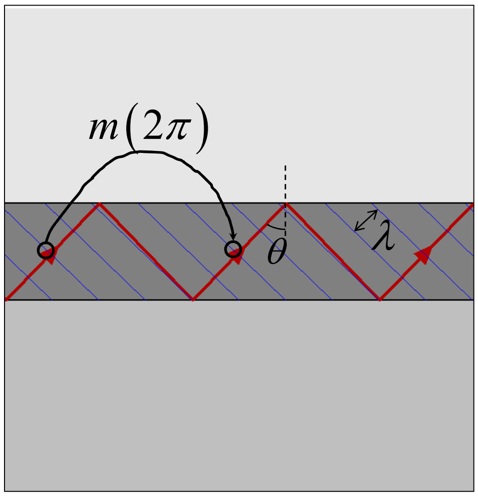
$n_2 > n_3$



Slab Waveguide Analysis — Slide 8

Ray Tracing Picture

Computational Methods in EE



The round trip phase of a ray must be an integer multiple of 2π. Otherwise the wave will interfere with itself and escape from the slab.

Because of this, only certain angles are allowed to propagate in the waveguide.

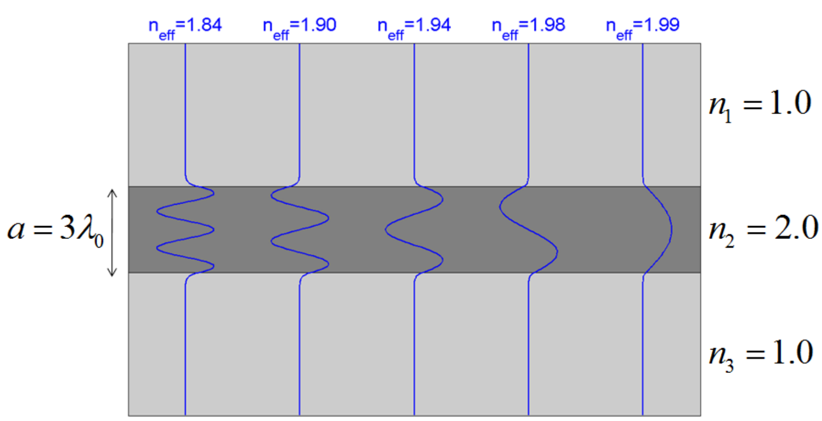
This is the origin of discrete modes in a waveguide.

$$\beta = k_0 n_{\text{eff}} = k_0 n \sin\theta$$

Slab Waveguide Analysis Slide 9

Rigorous Analysis

Computational Methods in EE



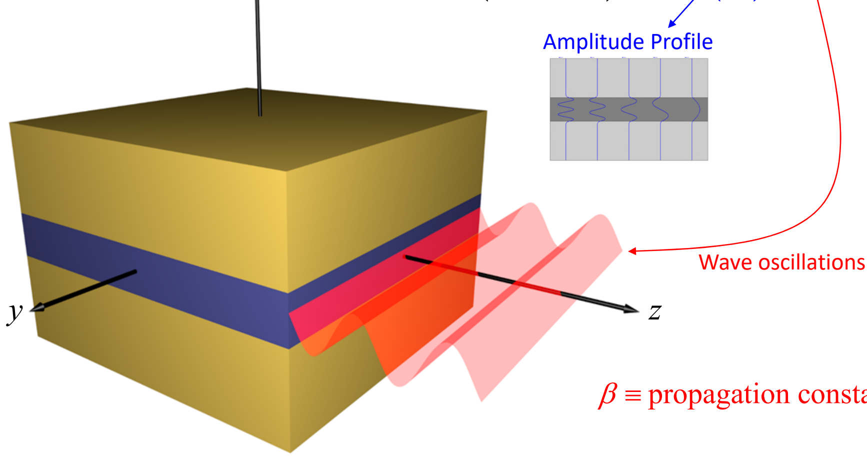
$$\beta = k_0 n_{\text{eff}} = k_0 n \sin\theta$$

Slab Waveguide Analysis Slide 10

Mathematical Form of Solution

Computational Methods in EE

$$\vec{E}(x,y,z)=\vec{A}(x)e^{-j\beta z}$$

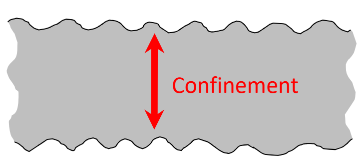


$\beta \equiv$ propagation constant

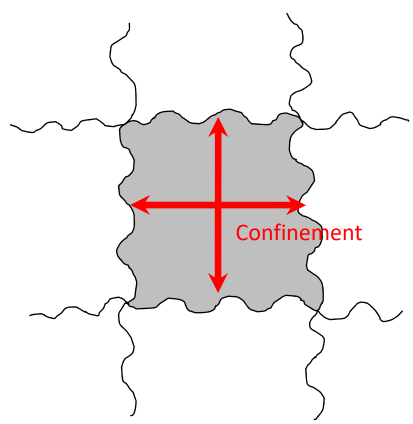
Slab Vs. Channel Waveguides

Computational Methods in EE

Slab waveguides confine energy in only one transverse direction.



Channel waveguides confine energy in both transverse directions.



Computational Methods in EE

Formulation

What is Formulation?

Computational Methods in EE

Formulation is the initial analytical work we do before implementing a computer code.

Usually we start with the governing equation(s) and end with the matrix equation to be solved.

Governing Equations

$$\nabla \times \vec{E} = -j\omega\mu\vec{H}$$
$$\nabla \times \vec{H} = j\omega\varepsilon\vec{E}$$

Matrix Equation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

Governing Equations

Computational Methods in EE

Since this is an electromagnetic problem, we start with Maxwell's curl equations.

$$\nabla \times \vec{E} = -j\omega\mu\vec{H}$$
$$\nabla \times \vec{H} = j\omega\varepsilon\vec{E}$$

Vector Curl

The curl of a vector is a measure of the vector field's tendency to circulate about an axis. The curl quantity is directly along this axis and the magnitude measures the strength of the circulation.

$$\nabla \times \vec{A} = \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right)\hat{a}_x + \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right)\hat{a}_y + \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)\hat{a}_z$$

Expand Governing Equations (1 of 2)

Computational Methods in EE

If we expand the first equation into its vector components, we get

$$\nabla \times \vec{E} = -j\omega\mu\vec{H}$$

$$\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}\right)\hat{a}_x + \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x}\right)\hat{a}_y + \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right)\hat{a}_z = -j\omega\mu\left(H_x\hat{a}_x + H_y\hat{a}_y + H_z\hat{a}_z\right)$$

The vector components on each side must be equal.

$$x\text{-component:}\quad \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -j\omega\mu H_x$$

$$y\text{-component:}\quad \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -j\omega\mu H_y$$

$$z\text{-component:}\quad \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -j\omega\mu H_z$$

Expand Governing Equations (2 of 2)

Computational Methods in EE

We now have

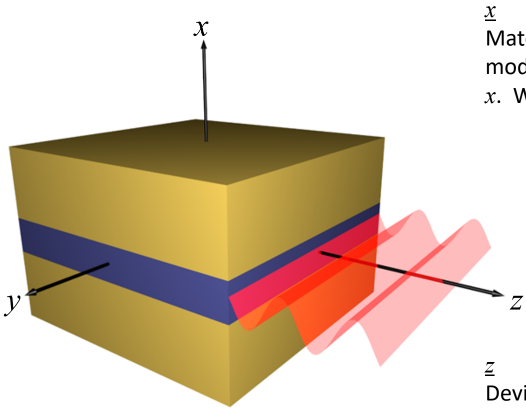
$$\nabla \times \vec{E} = -j\omega\mu\vec{H} \rightarrow \begin{cases} \dfrac{\partial E_z}{\partial y} - \dfrac{\partial E_y}{\partial z} = -j\omega\mu H_x \\ \dfrac{\partial E_x}{\partial z} - \dfrac{\partial E_z}{\partial x} = -j\omega\mu H_y \\ \dfrac{\partial E_y}{\partial x} - \dfrac{\partial E_x}{\partial y} = -j\omega\mu H_z \end{cases}$$

$$\nabla \times \vec{H} = j\omega\varepsilon\vec{E} \rightarrow \begin{cases} \dfrac{\partial H_z}{\partial y} - \dfrac{\partial H_y}{\partial z} = j\omega\varepsilon E_x \\ \dfrac{\partial H_x}{\partial z} - \dfrac{\partial H_z}{\partial x} = j\omega\varepsilon E_y \\ \dfrac{\partial H_y}{\partial x} - \dfrac{\partial H_x}{\partial y} = j\omega\varepsilon E_z \end{cases}$$

How to Reduce Dimensions

Computational Methods in EE

It is always good practice to minimize the number of dimensions utilized in a numerical analysis.



$\underline{x}$
Material changes as a function of x. The mode profile will change as a function of x. We must retain this dimension.

$\underline{y}$
Device is uniform. Wave does not propagate in this direction. Mode profile is uniform.

$$\frac{\partial}{\partial y} = 0$$

$\underline{z}$
Device is uniform. Wave propagates in this direction so wave phase is increasing.

$$\frac{\partial}{\partial z} = -j\beta$$

Apply $\partial/\partial y = 0$

Computational Methods in EE

Since nothing is changing in the y direction, any derivative with respect to y must be zero.

$$\cancel{\frac{\partial E_z}{\partial y}} - \frac{\partial E_y}{\partial z} = -j\omega\mu H_x \qquad \Rightarrow \qquad -\frac{\partial E_y}{\partial z} = -j\omega\mu H_x$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -j\omega\mu H_y \qquad \Rightarrow \qquad \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -j\omega\mu H_y$$

$$\frac{\partial E_y}{\partial x} - \cancel{\frac{\partial E_x}{\partial y}} = -j\omega\mu H_z \qquad \Rightarrow \qquad \frac{\partial E_y}{\partial x} = -j\omega\mu H_z$$

$$\cancel{\frac{\partial H_z}{\partial y}} - \frac{\partial H_y}{\partial z} = j\omega\varepsilon E_x \qquad \Rightarrow \qquad -\frac{\partial H_y}{\partial z} = j\omega\varepsilon E_x$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = j\omega\varepsilon E_y \qquad \Rightarrow \qquad \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = j\omega\varepsilon E_y$$

$$\frac{\partial H_y}{\partial x} - \cancel{\frac{\partial H_x}{\partial y}} = j\omega\varepsilon E_z \qquad \Rightarrow \qquad \frac{\partial H_y}{\partial x} = j\omega\varepsilon E_z$$

Slab Waveguide Analysis — Slide 19

Two Distinct Mode Types

Computational Methods in EE

Our revised governing equations have separated into two distinct mode types.

$$-\frac{\partial E_y}{\partial z} = -j\omega\mu H_x$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -j\omega\mu H_y$$

$$\frac{\partial E_y}{\partial x} = -j\omega\mu H_z$$

$$-\frac{\partial H_y}{\partial z} = j\omega\varepsilon E_x$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = j\omega\varepsilon E_y$$

$$\frac{\partial H_y}{\partial x} = j\omega\varepsilon E_z$$

We will analyze the Ey mode

Mode Type 1 – Ey Mode

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = j\omega\varepsilon E_y$$

$$-\frac{\partial E_y}{\partial z} = -j\omega\mu H_x$$

$$\frac{\partial E_y}{\partial x} = -j\omega\mu H_z$$

Mode Type 2 – Hy Mode

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -j\omega\mu H_y$$

$$-\frac{\partial H_y}{\partial z} = j\omega\varepsilon E_x$$

$$\frac{\partial H_y}{\partial x} = j\omega\varepsilon E_z$$

Slab Waveguide Analysis — Slide 20

What About $\partial/\partial z$?

Computational Methods in EE

Our waveguide mode has the following mathematical form

$$\vec{E}(x,y,z)=\vec{A}(x)e^{-j\beta z}$$

Let's calculate the partial derivative with respect to z and see what happens.

$$\frac{\partial}{\partial z}\vec{E}(x,y,z)=\frac{\partial}{\partial z}\left[\vec{A}(x)e^{-j\beta z}\right]=\vec{A}(x)\frac{\partial}{\partial z}e^{-j\beta z}+\cancel{e^{-j\beta z}\frac{\partial}{\partial z}\vec{A}(x)}$$
$$=-j\beta\underbrace{\vec{A}(x)e^{-j\beta z}}_{\vec{E}(x,y,z)}=-j\beta\vec{E}(x,y,z)$$

We conclude that

$$\frac{\partial}{\partial z}=-j\beta$$

1D Governing Equations

Computational Methods in EE

The equations for the Ey mode were

$$\frac{\partial}{\partial z}H_x-\frac{\partial H_z}{\partial x}=j\omega\varepsilon E_y$$
$$-\frac{\partial}{\partial z}E_y=-j\omega\mu H_x$$
$$\frac{\partial E_y}{\partial x}=-j\omega\mu H_z$$

We simply replace $\partial/\partial z$ with $-j\beta$.

$$-j\beta H_x-\frac{dH_z}{dx}=j\omega\varepsilon E_y$$
$$j\beta E_y=-j\omega\mu H_x$$
$$\frac{dE_y}{dx}=-j\omega\mu H_z$$

Normalize

Computational Methods in EE

Before converting our equations to matrix form, we should normalize the spatial coordinate x to put it in terms of wavelength in some manner.

$$\tilde{x} \overset{???}{=} \frac{x}{\lambda_0}$$

It will be mathematically convenient to normalize by multiplying x by the free space wave number k_0.

$$\tilde{x} = k_0 x \qquad k_0 = \frac{2\pi}{\lambda_0} = \frac{\omega\sqrt{\mu\varepsilon}}{n}$$

Normalized Equations

Computational Methods in EE

Applying our normalization, our equations become

$$-jn_{\text{eff}} H_x - \frac{dH_z}{d\tilde{x}} = j\sqrt{\frac{\varepsilon_0}{\mu_0}}\varepsilon_r E_y$$

$$jn_{\text{eff}} E_y = -j\sqrt{\frac{\mu_0}{\varepsilon_0}}\mu_r H_x$$

$$\frac{dE_y}{d\tilde{x}} = -j\sqrt{\frac{\mu_0}{\varepsilon_0}}\mu_r H_z$$

Last, at optical frequencies, the permeability is negligible so $\mu_r = 1$.

$$-jn_{\text{eff}} H_x - \frac{dH_z}{d\tilde{x}} = j\sqrt{\frac{\varepsilon_0}{\mu_0}}\varepsilon_r E_y$$

$$jn_{\text{eff}} E_y = -j\sqrt{\frac{\mu_0}{\varepsilon_0}} H_x$$

$$\frac{dE_y}{d\tilde{x}} = -j\sqrt{\frac{\mu_0}{\varepsilon_0}} H_z$$

Final Governing Equation

Computational Methods in EE

We solve the last two equations for H_x and H_z.

$$jn_{\text{eff}}E_y = -j\sqrt{\frac{\mu_0}{\varepsilon_0}}H_x \quad \rightarrow \quad H_x = -n_{\text{eff}}\sqrt{\frac{\varepsilon_0}{\mu_0}}E_y$$

$$\frac{dE_y}{d\tilde{x}} = -j\sqrt{\frac{\mu_0}{\varepsilon_0}}H_z \quad \rightarrow \quad H_z = j\sqrt{\frac{\varepsilon_0}{\mu_0}}\frac{dE_y}{d\tilde{x}}$$

These are substituted into the first equation to get a single equation containing only E_y. This is why it was called the E_y mode.

$$-jn_{\text{eff}}H_x - \frac{dH_z}{d\tilde{x}} = j\sqrt{\frac{\varepsilon_0}{\mu_0}}\varepsilon_r E_y$$

$$-jn_{\text{eff}}\left(-n_{\text{eff}}\sqrt{\frac{\varepsilon_0}{\mu_0}}E_y\right) - \frac{d}{d\tilde{x}}\left(j\sqrt{\frac{\varepsilon_0}{\mu_0}}\frac{dE_y}{d\tilde{x}}\right) = j\sqrt{\frac{\varepsilon_0}{\mu_0}}\varepsilon_r E_y$$

$$n_{\text{eff}}^2 E_y - \frac{d^2E_y}{d\tilde{x}^2} = \varepsilon_r E_y \longrightarrow \frac{d^2E_y}{d\tilde{x}^2} + \varepsilon_r E_y = n_{\text{eff}}^2 E_y$$

Eigen-Value Problem

Computational Methods in EE

For optical problems, we like to put everything in terms of refractive index. We have $\varepsilon_r = n^2$.

$$\frac{d^2E_y}{d\tilde{x}^2} + n^2 E_y = n_{\text{eff}}^2 E_y$$

Our governing equation can be rearranged to the form of a standard eigen-value problem.

$$\boxed{\left[\frac{d^2}{d\tilde{x}^2} + n^2(x)\right]E_y(x) = n_{\text{eff}}^2 E_y(x)}$$

Matrix Form

Computational Methods in EE

We go term-by-term to write the equation in matrix form.

$$\left[\frac{d^2}{d\tilde{x}^2} + n^2(x)\right] E_y(x) = n_{\text{eff}}^2 E_y(x)$$

$$\left(\mathbf{D}_{\tilde{x}}^2 + \mathbf{n}^2\right)\mathbf{e}_y = n_{\text{eff}}^2 \mathbf{e}_y$$

or

$$\underbrace{\left(\mathbf{D}_{\tilde{x}}^2 + \boldsymbol{\varepsilon}\right)}_{\text{Eigen Matrix}}\mathbf{e}_y = \underbrace{n_{\text{eff}}^2}_{\text{Eigen Value}} \mathbf{e}_y$$

Computational Methods in EE

Solution

Solving the Eigen-Value Problem

Computational Methods in EE

$$\left(\mathbf{D}_{\tilde{x}}^2+\boldsymbol{\varepsilon}\right)\mathbf{e}_y=n_{\text{eff}}^2\mathbf{e}_y \quad \rightarrow \quad \begin{aligned}\mathbf{V}&\equiv \text{Eigen-vector matrix}\\ \boldsymbol{\lambda}&\equiv \text{Eigen-value matrix}\end{aligned}$$

$$\mathbf{V}=\left[\begin{bmatrix} e_y^{(1)}(1)\\ e_y^{(1)}(2)\\ e_y^{(1)}(3)\\ \vdots\\ e_y^{(1)}(N_x-1)\\ e_y^{(1)}(N_x)\end{bmatrix} \begin{bmatrix} e_y^{(2)}(1)\\ e_y^{(2)}(2)\\ e_y^{(2)}(3)\\ \vdots\\ e_y^{(2)}(N_x-1)\\ e_y^{(2)}(N_x)\end{bmatrix} \cdots \begin{bmatrix} e_y^{(M)}(1)\\ e_y^{(M)}(2)\\ e_y^{(M)}(3)\\ \vdots\\ e_y^{(M)}(N_x-1)\\ e_y^{(M)}(N_x)\end{bmatrix}\right]$$

M = # modes
Usually $M = N_x$

$$\mathbf{D}=\begin{bmatrix}\left(n_{\text{eff}}^{(1)}\right)^2 & & & \\ & \left(n_{\text{eff}}^{(2)}\right)^2 & & \\ & & \ddots & \\ & & & \left(n_{\text{eff}}^{(M)}\right)^2\end{bmatrix}$$

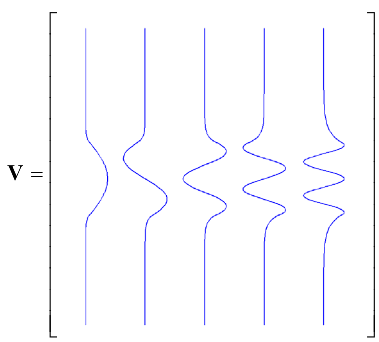
Eigen-vectors and eige-values come in pairs.

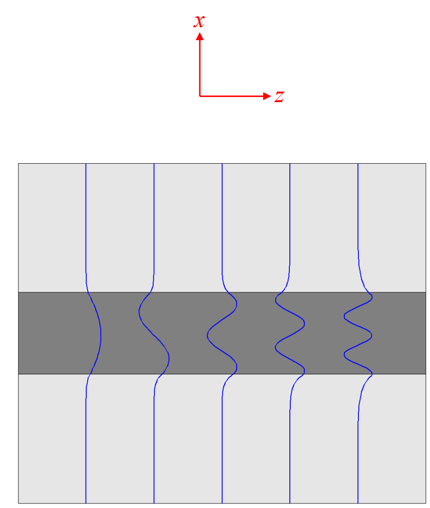
Do not mix up their pairing!

Visualizing the Solution

Computational Methods in EE

The columns of the eigen-vector matrix are pictures of the modes.





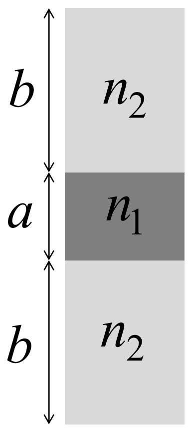
The eigen-values are the effective refractive indices of the modes.

Implementation in MATLAB

Implementation Outline

1. Initialize MATLAB
2. Dashboard (materials, dimensions, etc.)
3. Calculate Grid
4. Build Device on Grid
5. Perform Finite-Difference Analysis
6. Visualize the Results

Dashboard



How big should we make b?

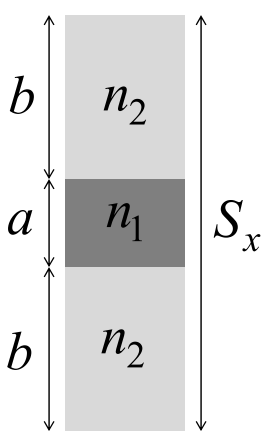
→ Enough to allow the mode to decay to zero before reaching the boundary.

What grid resolution should we use?

→ Convergence

```
% slabdemo.m

% INITIALIZE MATLAB
close all;
clc;
clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% DASHBOARD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FREE SPACE WAVELENGTH
lam0 = 1.0;

% SLAB PARAMETERS
n1 = 2.0;
n2 = 1.0;
a  = 3*lam0;

% GRID
b  = 5*lam0;
dx = lam0/20;

% NUMBER OF MODES TO CALCULATE
M = 5;
```


Build Device on Grid




```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% BUILD WAVEGUIDE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% COMPUTE GRID
Sx = a + 2*b;
Nx = ceil(Sx/dx);
Sx = Nx*dx;

xa = [0.5:Nx-0.5]*dx;
xa = xa - mean(xa);

% COMPUTE START AND STOP INDICES
nx  = round(a/dx);
nx1 = round((Nx - nx)/2);
nx2 = nx1 + nx - 1;

% BUILD N
N             = ones(Nx,1);
N(1:nx-1)     = n2;
N(nx1:nx2)    = n1;
N(nx2+1:Nx)   = n2;
```

Perform Finite-Difference Analysis

Computational Methods in EE

$$\mathbf{D}_{\tilde{x}}^2 = \frac{1}{(k_0 \Delta x)^2}\begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & & \ddots & & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix}$$

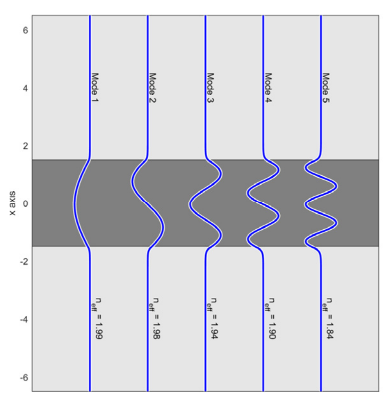
$$\mathbf{N} = \begin{bmatrix} n(1) & & & & \\ & n(2) & & & \\ & & n(3) & & \\ & & & \ddots & \\ & & & & n(N_x) \end{bmatrix}$$

$$\underbrace{\left(\mathbf{D}_{\tilde{x}}^2 + \mathbf{N}^2\right)}_{\mathbf{A}}\mathbf{e}_y = n_{\text{eff}}^2 \mathbf{e}_y$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% PERFORM FD ANALYSIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CALCULATE k0
k0 = 2*pi/lam0;

% BUILD DX2
DX2 = sparse(Nx,Nx);
DX2 = spdiags(+1*ones(Nx,1),-1,DX2);
DX2 = spdiags(-2*ones(Nx,1), 0,DX2);
DX2 = spdiags(+1*ones(Nx,1),+1,DX2);
DX2 = DX2 / (k0*dx)^2;

% MAKE N DIAGONAL
N = diag(sparse(N(:)));

% SOLVE EIGEN-VALUE PROBLEM
A      = DX2 + N^2;
[V,D]  = eig(full(A));
NEFF   = sqrt(diag(D));
```


Visualize the Results

Computational Methods in EE




```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% VISUALIZE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SORT MODES
[~,ind] = sort(real(NEFF),'descend');
V       = V(:,ind);
NEFF    = NEFF(ind);

% OPEN FIGURE WINDOW
figure('Color','w');
hold on;

% DRAW SLAB WAVEGUIDE
x = [0 2*(M+1) 2*(M+1) 0 0];
y = [ -b-a/2 -b-a/2 b+a/2 b+a/2 -b-a/2 ];
fill(x,y,0.9*[1 1 1]);
y = [ -a/2 -a/2 a/2 a/2 -a/2 ];
fill(x,y,0.5*[1 1 1]);

% DRAW AND LABEL MODES
for m = 1 : M
    x0 = 2*m;
    y0 = (a + b)/2;
    x = x0 + 3*V(:,m);
    y = linspace(-b-a/2,b+a/2,Nx);
    line(x,y,'Color','w','LineWidth',4);
    h = line(x,y,'Color','b','LineWidth',2);
    text(x0,y0,['Mode ' num2str(m)],'Rotation',-90,...
         'HorizontalAlignment','center','VerticalAlignment','bottom');
    text(x0,-y0,['n_{eff} = ' num2str(NEFF(m),'%4.2f')],'Rotation',-90,...
         'HorizontalAlignment','center','VerticalAlignment','bottom');
end

% SET GRAPHICS VIEW
hold off;
h2 = get(h,'Parent');
set(h2,'XTick',[]);
axis equal tight;
ylabel('x axis');
```