EE 5303

Electromagnetic Analysis Using Finite-Difference Time-Domain

Lecture #21

Grating Simulation Walkthrough

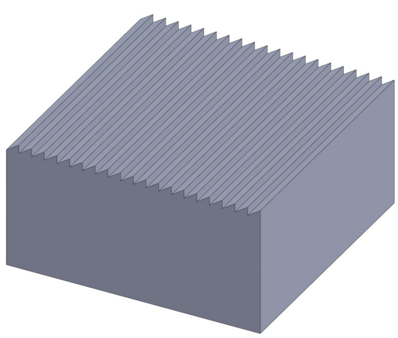
Lecture Outline

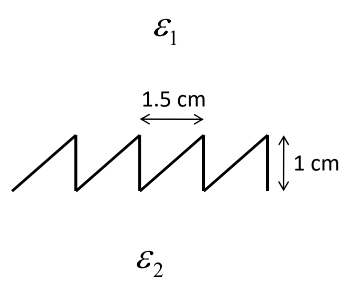
- Walkthrough
  - Step 0 – Problem Definition
  - Step 1 – Define the problem in MATLAB
  - Step 2 – Compute grid
  - Step 3 – Build device on grid
  - Step 4 – Compute source
  - Step 5 – Initialize Fourier transforms
  - Step 6 – Compute the PML
  - Step 7 – Compute update coefficients
  - Step 8 – Initialize FDTD data arrays
  - Step 9 – Main FDTD loop
  - Step 10 – Compute reflectance and transmittance
  - Step 11 – Produce professional looking results
- Results
- What could possibly go wrong?

Walkthrough

Step 0: Problem Definition

Suppose we wish to simulate transmission and reflection from the following saw-tooth grating.





What device are you modeling?	-- a sawtooth grating
What is its geometry?	-- see above
What materials is it made from?	-- ε_r=9.0
What do you wish to learn?	-- diffraction efficiency of spatial harmonics at 10 GHz

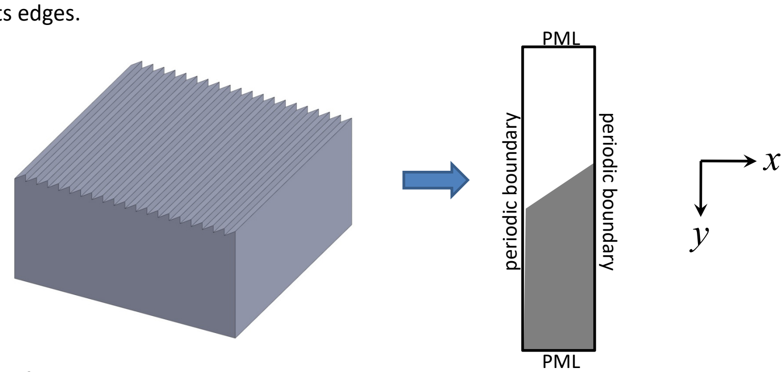
Step 0: Define Problem – *assumptions for 2D*

Assumption #1: Infinite substrate

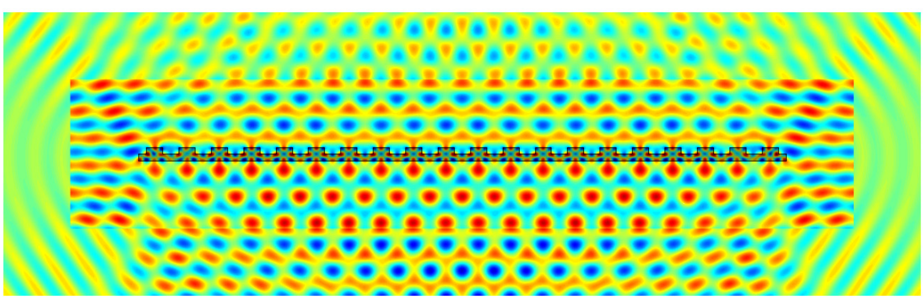
Due to the thickness of the substrate compared to the grating, we can reduce the size of the grid in the vertical dimension by assumption an infinite substrate. This is common practice in photonics because the substrates can be millions of times thicker than the grating.

Assumption #2: Infinitely periodic

We can dramatically reduces the size of the grid in the horizontal direction by assuming the device is infinitely periodic. This is a good assumption when the device is used away from its edges.



Step 0: Define Problem – *validity of periodic BC*



The field in a finite periodic device is very nearly periodic away from the edges. For this device, the infinitely periodic approximation does not predict the field in the outer four (or so) unit cells.

Step 1: Dashboard – *setup MATLAB*

Initialize MATLAB

```
% Lecture22_sawtooth.m

% INITIALIZE MATLAB
close all;
clc;
clear all;

% UNITS
meters      = 1;
centimeters = 1e-2 * meters;
millimeters = 1e-3 * meters;
inches      = 2.54 * centimeters;
feet        = 12 * inches;
seconds     = 1;
hertz       = 1/seconds;
kilohertz   = 1e3 * hertz;
megahertz   = 1e6 * hertz;
gigahertz   = 1e9 * hertz;

% CONSTANTS
e0 = 8.85418782e-12;
u0 = 1.25663706e-6;
N0 = sqrt(u0/e0);
c0 = 299792458 * meters/seconds;
```

Dashboard

```
% SOURCE PARAMETERS
NFREQ = 500;
fmax  = 15 * gigahertz;
FREQ  = linspace(5,15,NFREQ) * gigahertz;

f0    = 10 * gigahertz;
lam0  = c0/f0;

% GRATING PARAMETERS
L = 1.5 * centimeters;
d = 1.0 * centimeters;

er1 = 1.0;
er2 = 9.0;

% GRID PARAMETERS
nmax = sqrt(max([er1 er2]));
NRES = 10;
NPML = [0 0 20 20];
BUF  = 0.5*lam0 * [1 1];
```

Note: nothing is "hard coded" after this*!*

Lecture 21 Slide 7

Step 2: Compute Grid – *grid resolution*

Compute Initial Grid Resolution

$$N_\lambda = 10$$

$$n_{\max} = \sqrt{\varepsilon_2} = 3.0$$

$$\lambda_{\min} = \frac{c_0}{f_{\max}} = 2.0 \text{ cm}$$

$$\Delta x' = \frac{\lambda_{\min}}{n_{\max} N_\lambda} = 666\ \mu\text{m}$$

$$\Delta y' = \frac{\lambda_{\min}}{n_{\max} N_\lambda} = 666\ \mu\text{m}$$

```
% COMPUTE INITIAL GRID RESOLUTION
lam0_min = c0/max(FREQ);
dx       = lam0_min/nmax/NRES;
dy       = lam0_min/nmax/NRES;
```

Snap Grid to Critical Dimensions

$$N'_x = \frac{\Lambda}{\Delta x'} = 22.51 \text{ cells} \xrightarrow[\text{make odd}]{\text{round up}} N_x = 2\,\text{ceil}\left(\frac{N'_x}{2}\right) + 1 = 25$$

$$\Delta x = \frac{\Lambda}{N_x} = 0.6000 \text{ mm}$$

$$N'_y = \frac{d}{\Delta y'} = 15.01 \text{ cells} \xrightarrow{\text{round up}} N_y = 16$$

$$\Delta y = \frac{d}{N_y} = 0.625 \text{ mm}$$

```
% SNAP GRID TO CRITIAL DIMENSIONS
Nx = 2*ceil(L/dx/2) + 1;
dx = L/Nx;
Ny = ceil(d/dy);
dy = d/Ny;
```

Note: Λ and L are the same parameter.

Lecture 21 Slide 8

Step 2: Compute Grid – *grid size*

Determine Physical Size

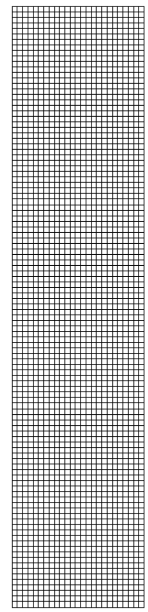
$$S_x = \Lambda = 1.5 \text{ cm}$$
$$S_y = d + \text{BUF}(1) + \text{BUF}(2) = 4.0 \text{ cm}$$

Calculate Number of Cells

$$N_x = \frac{S_x}{\Delta x} = 25 \quad \text{We already know this because } S_x = \Lambda = 1.5 \text{ cm.}$$

$$N_y = \text{NPML}(3) + 3 + \frac{S_y}{\Delta y} + 2 + \text{NPML}(4) = 109$$
$$S_y = N_y \cdot \Delta y = 6.8125 \text{ cm}$$

```
% COMPUTE GRID SIZE
Sx = L;
Sy = sum(BUF) + d;
Ny = ceil(Sy/dy) + NPML(3) + NPML(4) + 5;
Sy = Ny*dy;
```



Step 3: Build Device – *initialize grid*

Initialize Materials to Free Space

$$\mu_{xx}(x,y) = 1.0$$
$$\mu_{yy}(x,y) = 1.0$$
$$\varepsilon_{zz}(x,y) = \varepsilon_{r,1}$$

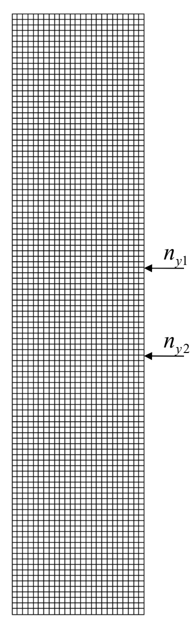
```
% INITIALIZE MATERIALS TO FREE SPACE
URxx = ones(Nx,Ny);
URyy = ones(Nx,Ny);
ERzz = er1 * ones(Nx,Ny);
```

Compute Position Indices

$$n_{y1} = \text{NPML}(3) + 3 + \text{round}\left[\frac{\text{BUF}(1)}{\Delta y}\right]$$

$$n_{y2} = n_{y1} + \text{round}\left[\frac{d}{\Delta y}\right] - 1$$

```
% COMPUTE POSITION INDICES
ny1 = NPML(3) + 3 + round(BUF(1)/dy);
ny2 = ny1 + round(d/dy) - 1;
```



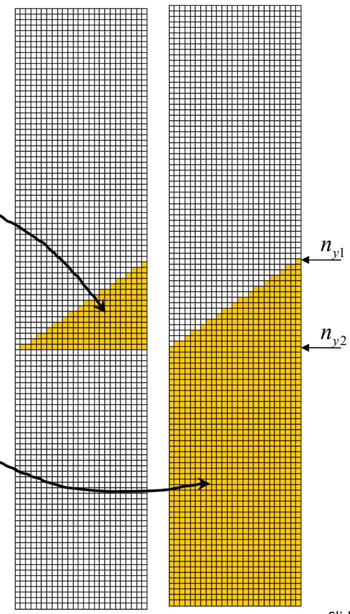
Step 3: Build Device – *add materials*

Add Sawtooth

```
% ADD GRATING
for ny = ny1 : ny2
    f   = (ny - ny1 + 1)/(ny2 - ny1 + 2);
    nx  = round(f*Nx);
    nx2 = Nx;
    nx1 = nx2 - nx + 1;
    ERzz(nx1:nx2,ny) = er2;
end
```

Add Infinite Substrate

```
% ADD INFINITE SUBSTRATE
ERzz(:,ny2+1:Ny) = er2;
```



Step 4: Compute Source – *Gaussian parameters*

Compute Stable Time Step

$$\Delta_{\min} = \min\left[\Delta x, \Delta y\right]$$
$$\Delta t = \Delta_{\min} / \left(2c_0\right)$$

```
% COMPUTE STABLE TIME STEP
dmin = min([dx dy]);
dt   = dmin/(2*c0);
```

Compute Source Position

$$n_{y,\mathrm{src}} = \mathrm{NPML}\left(3\right) + 2$$

```
% SOURCE POSITION
ny_src = NPML(3) + 2;
```

Compute Source Parameters

$$\tau = \frac{0.5}{f_{\max}} \qquad \delta t = \frac{n_{\mathrm{src}}\Delta y}{2c_0} + \frac{\Delta t}{2}$$
$$t_0 \cong 6\tau \qquad A = \sqrt{\varepsilon_{r,\mathrm{src}} / \mu_{r,\mathrm{src}}}$$

```
% COMPUTE TIME PARAMETERS
tau  = 0.5/fmax;
t0   = 6*tau;
A    = sqrt(ERzz(1,ny_src)/URyy(1,ny_src));
delt = 0.5*dy/c0 + dt/2;
```

Step 4: Compute Source – *source functions*

FDTD

Compute Number of Iterations

$$\tau_{\text{prop}} = \frac{n_{\max} S_y}{c_0}$$

$$\tau_{\text{sim}} \cong 2t_0 + 10\tau_{\text{prop}}$$

$$\text{STEPS} = \text{ceil}\left[\frac{\tau_{\text{sim}}}{\Delta t}\right]$$

```
% TIME STEPS
proptime = nmax*Sy/c0;
simtime  = 2*t0 + 10*proptime;
STEPS    = ceil(simtime/dt);
```

Compute Source Functions

$$\tilde{E}_{z,\text{src}} = \exp\left[-\left(\frac{t-t_0}{\tau}\right)^2\right]$$

$$H_{x,\text{src}} = A\exp\left[-\left(\frac{t-t_0+\delta t}{\tau}\right)^2\right]$$

```
% COMPUTE GAUSSIAN SOURCES
ta     = [0:STEPS-1]*dt;
Ez_src = exp(-((ta-t0)/tau).^2);
Hx_src = A*exp(-((ta-t0+delt)/tau).^2);
```


Step 5: Initialize Fourier Transforms

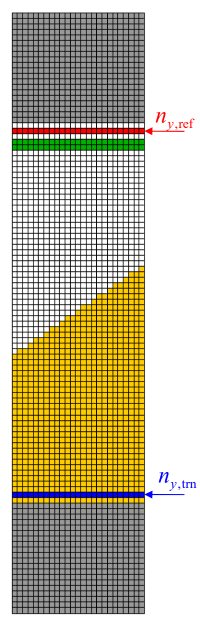
FDTD

Compute Kernels

$$K(f) = \exp(-j2\pi \cdot \Delta t \cdot f)$$

$$K_0 = \exp(-j2\pi \cdot \Delta t \cdot f_0)$$

```
% COMPUTE FOURIER TRANSFORM KERNELS
K  = exp(-1i*2*pi*dt*FREQ);
K0 = exp(-1i*2*pi*dt*f0);
```



Initialize Steady-State Fields

```
% INITIALIZE STEADY-STATE FIELDS
Eref = zeros(Nx,NFREQ);
Etrn = zeros(Nx,NFREQ);
SRC  = zeros(1,NFREQ);

Eref0 = zeros(Nx,1);
Etrn0 = zeros(Nx,1);
ssSRC = 0;
```

Define Position of Record Planes

```
% POSITION OF RECORD PLANES
ny_ref = NPML(3) + 1;
ny_trn = Ny - NPML(4);
```

Compute Refractive Indices in Record Planes

```
% COMPUTE REFRACTIVE INDICES IN RECORD PLANES
nref = sqrt(ERzz(1,ny_ref)*URxx(1,ny_ref));
ntrn = sqrt(ERzz(1,ny_trn)*URxx(1,ny_trn));
```

Step 6: Compute the PML

Compute Size of 2× Grid

```
% NUMBER OF POINTS ON 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
```

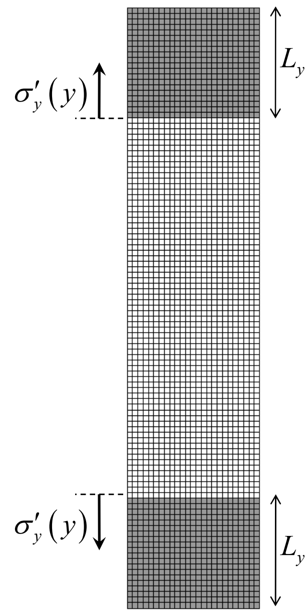
Compute σ_x

There is no PML at the *x*-axis boundary for this problem.

Compute σ_y

$$\sigma_y'(y) = \frac{\varepsilon_0}{2\Delta t} \cdot \left(\frac{y}{L_y}\right)^3$$

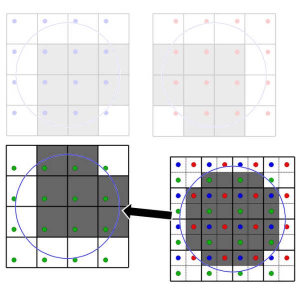
```
% COMPUTE SIG PML PARAMETERS
sigx = zeros(Nx2,Ny2);
sigy = zeros(Nx2,Ny2);
for ny = 1 : 2*NPML(3)
    ny1 = 2*NPML(3) - ny + 1;
    sigy(:,ny1) = (0.5*e0/dt)*(ny/2/NPML(3))^3;
end
for ny = 1 : 2*NPML(4)
    ny1 = Ny2 - 2*NPML(4) + ny;
    sigy(:,ny1) = (0.5*e0/dt)*(ny/2/NPML(4))^3;
end
```



Step 7: Compute Update Coefficients -- H_x

Compute Update Coefficients for Hx

Extract the PML parameters from the 2× grid.



```
% COMPTUE HX UPDATE COEFFICIENTS
sigHx = sigx(1:2:Nx2,2:2:Ny2);
sigHy = sigy(1:2:Nx2,2:2:Ny2);
mHx0  = (1/dt) + sigHy/(2*e0);
mHx1  = ((1/dt) - sigHy/(2*e0))./mHx0;
mHx2  = - c0./URxx./mHx0;
mHx3  = - (c0*dt/e0) * sigHx./URxx ./ mHx0;
```

Compute the update coefficients

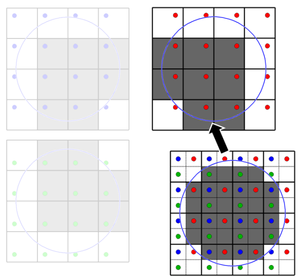
$$m_{Hx0}\big|^{i,j} = \frac{1}{\Delta t} + \left(\frac{\sigma_y^H\big|^{i,j}}{2\varepsilon_0}\right) \qquad m_{Hx1}\big|^{i,j} = \frac{1}{m_{Hx0}\big|^{i,j}}\left[\frac{1}{\Delta t} - \left(\frac{\sigma_y^H\big|^{i,j}}{2\varepsilon_0}\right)\right]$$

$$m_{Hx2}\big|^{i,j} = -\frac{1}{m_{Hx0}\big|^{i,j}} \frac{c_0}{\mu_{xx}\big|^{i,j}} \qquad m_{Hx3}\big|^{i,j} = -\frac{1}{m_{Hx0}\big|^{i,j}} \frac{c_0 \Delta t}{\varepsilon_0} \frac{\sigma_x^H\big|^{i,j}}{\mu_{xx}\big|^{i,j}}$$

Step 7: Compute Update Coefficients -- H_y

Compute Update Coefficients for Hy

Extract the PML parameters from the 2× grid.



```
% COMPUTE HY UPDATE COEFFICIENTS
sigHx = sigx(2:2:Nx2,1:2:Ny2);
sigHy = sigy(2:2:Nx2,1:2:Ny2);
mHy0  = (1/dt) + sigHx/(2*e0);
mHy1  = ((1/dt) - sigHx/(2*e0))./mHy0;
mHy2  = - c0./URyy./mHy0;
mHy3  = - (c0*dt/e0) * sigHy./URyy ./ mHy0;
```

Compute the update coefficients

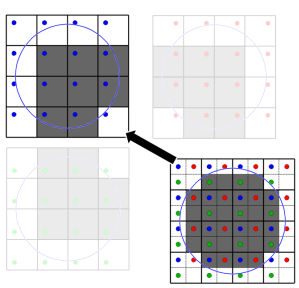
$$m_{Hy0}\Big|^{i,j} = \frac{1}{\Delta t} + \left(\frac{\sigma_x^H\big|^{i,j}}{2\varepsilon_0}\right) \qquad m_{Hy1}\Big|^{i,j} = \frac{1}{m_{Hy0}\big|^{i,j}}\left[\frac{1}{\Delta t} - \left(\frac{\sigma_x^H\big|^{i,j}}{2\varepsilon_0}\right)\right]$$

$$m_{Hy2}\Big|^{i,j} = -\frac{1}{m_{Hy0}\big|^{i,j}}\frac{c_0}{\mu_{yy}\big|^{i,j}} \qquad m_{Hy3}\Big|^{i,j} = -\frac{1}{m_{Hy0}\big|^{i,j}}\frac{c_0\Delta t}{\varepsilon_0}\frac{\sigma_y^H\big|^{i,j}}{\mu_{yy}\big|^{i,j}}$$

Step 7: Compute Update Coefficients -- D_z

Compute Update Coefficients for Dz

Extract the PML parameters from the 2× grid.



```
% COMPUTE DZ UPDATE COEFFICIENTS
sigDx = sigx(1:2:Nx2,1:2:Ny2);
sigDy = sigy(1:2:Nx2,1:2:Ny2);
mDz0  = (1/dt) + (sigDx + sigDy)/(2*e0) ...
        + sigDx.*sigDy*(dt/4/e0^2);
mDz1  = (1/dt) - (sigDx + sigDy)/(2*e0) ...
        - sigDx.*sigDy*(dt/4/e0^2);
mDz1  = mDz1 ./ mDz0;
mDz2  = c0./mDz0;
mDz4  = - (dt/e0^2)*sigDx.*sigDy./mDz0;
```

Compute the update coefficients

$$m_{Dz0}\Big|^{i,j} = \frac{1}{\Delta t} + \frac{\sigma_x^D\big|^{i,j} + \sigma_y^D\big|^{i,j}}{2\varepsilon_0} + \frac{\left(\sigma_x^D\big|^{i,j}\right)\left(\sigma_y^D\big|^{i,j}\right)\Delta t}{4\varepsilon_0^2} \qquad m_{Dz1}\Big|^{i,j} = \frac{1}{m_{Dz0}\big|^{i,j}}\left[\frac{1}{\Delta t} - \frac{\sigma_x^D\big|^{i,j} + \sigma_y^D\big|^{i,j}}{2\varepsilon_0} - \frac{\left(\sigma_x^D\big|^{i,j}\right)\left(\sigma_y^D\big|^{i,j}\right)\Delta t}{4\varepsilon_0^2}\right]$$

$$m_{Dz2}\Big|^{i,j} = \frac{c_0}{m_{Dz0}\big|^{i,j}} \qquad m_{Dz4}\Big|^{i,j} = -\frac{1}{m_{Dz0}\big|^{i,j}}\frac{\Delta t}{\varepsilon_0^2}\left(\sigma_x^D\big|^{i,j}\right)\left(\sigma_y^D\big|^{i,j}\right)$$

Step 7: Compute Update Coefficients -- E_z

Compute Update Coefficients for Ez

$$m_{Ez1}\Big|^{i,j} = \frac{1}{\varepsilon_{zz}\big|^{i,j}}$$

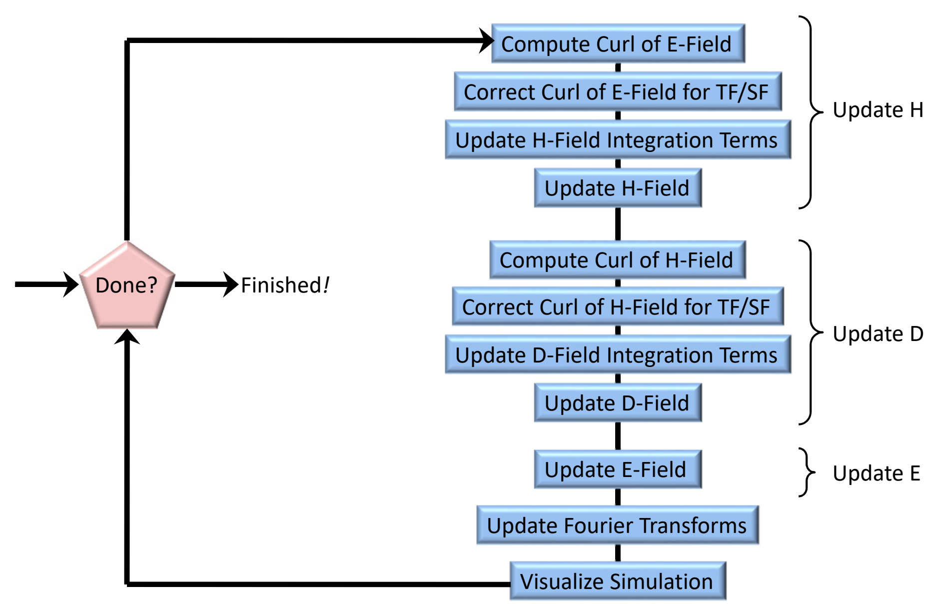
```
% COMPUTE EZ UPDATE COEFFICIENT
mEz1   = 1./ERzz;
```


Step 8: Initialize FDTD Data Arrays

Initialize FDTD Data Arrays

```
% INITIALIZE FIELDS
Hx = zeros(Nx,Ny);
Hy = zeros(Nx,Ny);
Dz = zeros(Nx,Ny);
Ez = zeros(Nx,Ny);

% INITIALIZE CURL
CEx = zeros(Nx,Ny);
CEy = zeros(Nx,Ny);
CHz = zeros(Nx,Ny);

% INITIALIZE INTEGRATION TERMS
ICEx = zeros(Nx,Ny);
ICEy = zeros(Nx,Ny);
IDz  = zeros(Nx,Ny);
```

Step 9: Run FDTD (Main Loop)



Step 9: Main Loop – *Curl of E*

Compute the *x*-Component of the Curl of E

$$C_x^E\Big|_t^{i,j} = \begin{cases} \dfrac{\tilde{E}_z\big|_t^{i,j+1} - \tilde{E}_z\big|_t^{i,j}}{\Delta y} & \text{for } j < N_y \\ \dfrac{\tilde{E}_z\big|_t^{i,1} - \tilde{E}_z\big|_t^{i,N_y}}{\Delta y} & \text{for } j = N_y \end{cases}$$

```
% Compute CEx
for ny = 1 : Ny-1
    for nx = 1 : Nx
        CEx(nx,ny) = (Ez(nx,ny+1) - Ez(nx,ny))/dy;
    end
end
for nx = 1 : Nx
    CEx(nx,Ny) = (Ez(nx,1) - Ez(nx,Ny))/dy;
end
```

Compute the *y*-Component of the Curl of E

$$C_y^E\Big|_t^{i,j} = \begin{cases} -\dfrac{\tilde{E}_z\big|_t^{i+1,j} - \tilde{E}_z\big|_t^{i,j}}{\Delta x} & \text{for } i < N_x \\ -\dfrac{\tilde{E}_z\big|_t^{1,j} - \tilde{E}_z\big|_t^{N_x,j}}{\Delta x} & \text{for } i = N_x \end{cases}$$

```
% Compute CEy
for nx = 1 : Nx-1
    for ny = 1 : Ny
        CEy(nx,ny) = - (Ez(nx+1,ny) - Ez(nx,ny))/dx;
    end
end
for ny = 1 : Ny
    CEy(Nx,ny) = - (Ez(1,ny) - Ez(Nx,ny))/dx;
end
```

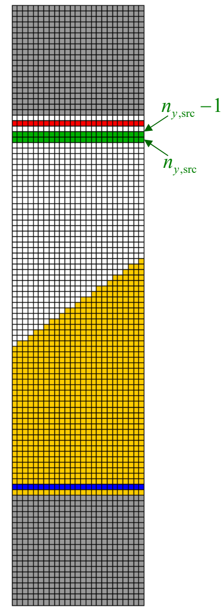
Step 9: Main Loop – *Correct Curl-E for TF/SF*

Correct CEx for TF/SF Framework

$$\underbrace{C_x^E\Big|_t^{i,j_{\mathrm{src}}-1} = \frac{\tilde{E}_z\Big|_t^{i,j_{\mathrm{src}}} - \tilde{E}_z\Big|_t^{i,j_{\mathrm{src}}-1}}{\Delta y}}_{\text{We already calculated this.}} - \frac{1}{\Delta y} E_z^{\mathrm{src}}\Big|_t^{i,j_{\mathrm{src}}}$$

We just need to incorporate this correction term for all values of *i*.

```
% TF/SF
CEx(:,ny_src-1) = CEx(:,ny_src-1) - Ez_src(T)/dy;
```



Step 9: Main Loop – *Update H*

Update Integration Terms

```
% Update Integration Terms
ICEx = ICEx + CEx;
ICEy = ICEy + CEy;
```

Update Hx and Hy

```
% Update Hx and Hy
Hx = mHx1.*Hx + mHx2.*CEx + mHx3.*ICEx;
Hy = mHy1.*Hy + mHy2.*CEy + mHy3.*ICEy;
```

Step 9: Main Loop – *Curl of H*

Compute the *z*-Component of the Curl of H

$$C_z^H\Big|_{t+\frac{\Delta t}{2}}^{i,j} = \begin{cases} \dfrac{H_y\big|_{t+\frac{\Delta t}{2}}^{i,j} - H_y\big|_{t+\frac{\Delta t}{2}}^{i-1,j}}{\Delta x} - \dfrac{H_x\big|_{t+\frac{\Delta t}{2}}^{i,j} - H_x\big|_{t+\frac{\Delta t}{2}}^{i,j-1}}{\Delta y} & \text{for } i>1 \text{ and } j>1 \\ \dfrac{H_y\big|_{t+\frac{\Delta t}{2}}^{1,j} - H_y\big|_{t+\frac{\Delta t}{2}}^{N_x,j}}{\Delta x} - \dfrac{H_x\big|_{t+\frac{\Delta t}{2}}^{1,j} - H_x\big|_{t+\frac{\Delta t}{2}}^{1,j-1}}{\Delta y} & \text{for } i=1 \text{ and } j>1 \\ \dfrac{H_y\big|_{t+\frac{\Delta t}{2}}^{i,1} - H_y\big|_{t+\frac{\Delta t}{2}}^{i-1,1}}{\Delta x} - \dfrac{H_x\big|_{t+\frac{\Delta t}{2}}^{i,1} - H_x\big|_{t+\frac{\Delta t}{2}}^{i,N_y}}{\Delta y} & \text{for } i>1 \text{ and } j=1 \\ \dfrac{H_y\big|_{t+\frac{\Delta t}{2}}^{1,1} - H_y\big|_{t+\frac{\Delta t}{2}}^{N_x,1}}{\Delta x} - \dfrac{H_x\big|_{t+\frac{\Delta t}{2}}^{1,1} - H_x\big|_{t+\frac{\Delta t}{2}}^{1,N_y}}{\Delta y} & \text{for } i=1 \text{ and } j=1 \end{cases}$$

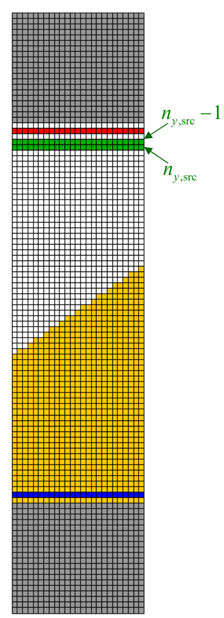
```
% Compute CHz
CHz(1,1) = (Hy(1,1) - Hy(Nx,1))/dx ...
         - (Hx(1,1) - Hx(1,Ny))/dy;
for nx = 2 : Nx
    CHz(nx,1) = (Hy(nx,1) - Hy(nx-1,1))/dx ...
              - (Hx(nx,1) - Hx(nx,Ny))/dy;
end
for ny = 2 : Ny
    CHz(1,ny) = (Hy(1,ny) - Hy(Nx,ny))/dx ...
              - (Hx(1,ny) - Hx(1,ny-1))/dy;
    for nx = 2 : Nx
        CHz(nx,ny) = (Hy(nx,ny) - Hy(nx-1,ny))/dx ...
                   - (Hx(nx,ny) - Hx(nx,ny-1))/dy;
    end
end
```


Step 9: Main Loop – *Correct Curl-H for TF/SF*

Correct CHz for TF/SF Framework

$$C_z^H\Big|_{t+\frac{\Delta t}{2}}^{i,j_{\text{src}}} = \underbrace{\frac{H_y\big|_{t+\frac{\Delta t}{2}}^{i,j_{\text{src}}} - H_y\big|_{t+\frac{\Delta t}{2}}^{i-1,j_{\text{src}}}}{\Delta x} - \frac{H_x\big|_{t+\frac{\Delta t}{2}}^{i,j_{\text{src}}} - H_x\big|_{t+\frac{\Delta t}{2}}^{i,j_{\text{src}}-1}}{\Delta y}}_{\text{We already calculated this.}} + \frac{1}{\Delta y}\tilde{H}_x^{\text{src}}\Big|_{t+\frac{\Delta t}{2}}^{i,j_{\text{src}}-1}$$

We just need to incorporate this correction term for all values of *i*.




```
% TF/SF
CHz(:,ny_src) = CHz(:,ny_src) + Hx_src(T)/dy;
```

Step 9: Main Loop – *Update D and E*

Update Integration Term for D-Field Update

```
% Update Integration Term
IDz = IDz + Dz;
```

Update Dz

```
% Update Dz
Dz = mDz1.*Dz + mDz2.*CHz + mDz4.*IDz;
```

Update Ez

```
% Update Ez
Ez = mEz1.*Dz;
```

Lecture 21 Slide 27

Step 9: Main Loop – *Update Fourier Transforms*

Update the Fourier Transforms for the Frequency Sweep

$$E_{\text{ref}}\left(f\right)\Big|^{i,n_{y,\text{ref}}} \cong \Delta t \sum_{m=1}^{M} \left(e^{-j2\pi f \Delta t}\right)^{m} \cdot E_{z}\Big|_{m}^{i,n_{y,\text{ref}}}$$

$$E_{\text{trn}}\left(f\right)\Big|^{i,n_{y,\text{trn}}} \cong \Delta t \sum_{m=1}^{M} \left(e^{-j2\pi f \Delta t}\right)^{m} \cdot E_{z}\Big|_{m}^{i,n_{y,\text{trn}}}$$

These are recorded at each frequency and at each point across the entire grid in the *x* direction.

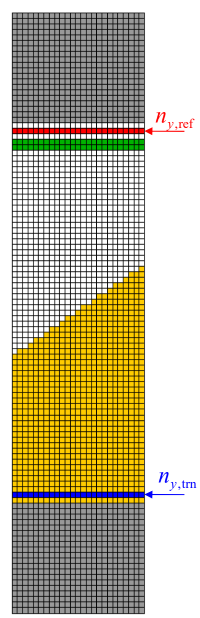
$$E_{\text{src}}\left(f\right) \cong \Delta t \sum_{m=1}^{M} \left(e^{-j2\pi f \Delta t}\right)^{m} \cdot E_{z,\text{src}}\Big|_{m}$$

Only a single value is stored for each frequency.

```
% Update Fourier Transforms
for nfreq = 1 : NFREQ
    Eref(:,nfreq) = Eref(:,nfreq) + (K(nfreq)^T)*Ez(:,ny_ref)*dt;
    Etrn(:,nfreq) = Etrn(:,nfreq) + (K(nfreq)^T)*Ez(:,ny_trn)*dt;
    SRC(nfreq)     = SRC(nfreq) + (K(nfreq)^T)*Ez_src(T)*dt;
end
```

Update the Fourier Transform for the Design Frequency

```
% Update f0 Fourier Transform
Eref0 = Eref0 + (K0^T)*Ez(:,ny_ref)*dt;
Etrn0 = Etrn0 + (K0^T)*Ez(:,ny_trn)*dt;
ssSRC = ssSRC + (K0^T)*Ez_src(T)*dt;
```



Lecture 21 Slide 28

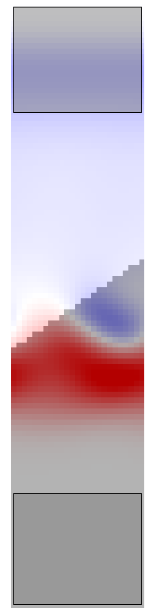
Step 9: Main Loop – *Visualize Simulation*

FDTD

Draw the Field Superimposed on the Materials

```
% Update Graphical Status
if ~mod(T,10)

    % draw field
    OPTS.emax = 0.5;
    draw2d(xa,ya,ERzz,Ez,NPML,OPTS);
    axis equal tight;
    title([num2str(T) ' of ' num2str(STEPS) ]);

    % force MATLAB to draw graphics
    drawnow;

end
```

Graphics are very slow!
This `if` statement updates the visualization only every 10 iterations.



Lecture 21 Slide 29

Step 10: R & T at f_0 – *Wave vector components*

FDTD

Compute Free Space Wave Number

$$\lambda_0 = c_0 / f_0 \qquad k_0 = 2\pi / \lambda_0$$

Compute Incident Wave Vector

$$k_{x,\mathrm{inc}} = 0$$

$$k_{y,\mathrm{inc}} = k_0 n_{\mathrm{ref}}$$

Assuming normal incidence.

```
% COMPUTE WAVE VECTOR COMPOMENTS
lam0  = c0/f0;
k0    = 2*pi/lam0;
kxinc = 0;
kyinc = k0*nref;
m     = [-floor(Nx/2):floor(Nx/2)]';
kx    = kxinc - 2*pi*m/Sx;
kyR   = sqrt((k0*nref)^2 - kx.^2);
kyT   = sqrt((k0*ntrn)^2 - kx.^2);
```

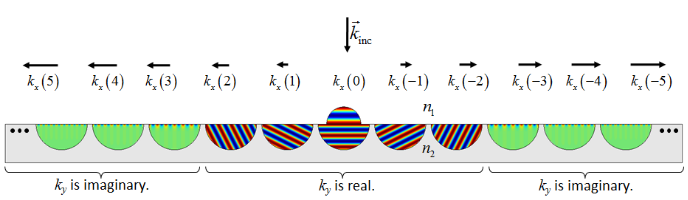
Compute Transverse Wave Vector Expansion

$$k_x(m) = k_{x,\mathrm{inc}} - \frac{2\pi m}{\Lambda_x} \qquad m = -\mathrm{floor}\left(\tfrac{1}{2} N_x\right), \cdots, \mathrm{floor}\left(\tfrac{1}{2} N_x\right)$$

Note: Λ_x and `Sx` are the same parameter.

Compute Longitudinal Components of the Wave Vectors

$$k_{y,\mathrm{ref}}(m) = \sqrt{(k_0 n_{\mathrm{ref}})^2 - k_x^2(m)} \qquad k_{y,\mathrm{trn}}(m) = \sqrt{(k_0 n_{\mathrm{trn}})^2 - k_x^2(m)}$$



Lecture 21 Slide 30

Step 10: R & T at f_0 – *Reflectance*

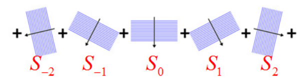
Normalize the Field Amplitude to the Source

$$E_{\text{ref}}\left(x,f_0\right) \cong E_{\text{ref}}\left(x,f_0\right) \div E_{\text{src}}\left(f_0\right)$$

This makes it look like the source at f_0 had an amplitude of 1.

Calculate the Amplitudes of the Spatial Harmonics

$$\text{FFT}\left[E_{\text{ref}}\left(x\right)\right] = \left[S_{-M} \quad \cdots \quad S_{-2} \quad S_{-1} \quad S_0 \quad S_1 \quad S_2 \quad \cdots \quad S_M\right]$$



Calculate the Diffraction Efficiencies of the Spatial Harmonics

$$\text{DE}_{\text{ref}}\left(m\right) = \frac{\left|\vec{S}_{\text{ref}}\left(m\right)\right|^2}{\left|\vec{S}_{\text{inc}}\right|^2} \text{Re}\left[\frac{k_{y,\text{ref}}\left(m\right)}{k_{y,\text{inc}}}\right]$$

Calculate the Overall Reflectance

$$R\left(f_0\right) = \sum_{N_x} \text{DE}_{\text{ref}}\left(m\right)$$

```
% COMPUTE REFLECTANCE
ref  = Eref0 / ssSRC;                      %normalize to source power
ref  = fftshift(fft(ref))/Nx;              %compute spatial harmonics
ref  = real(kyR/kyinc) .* abs(ref).^2;     %compute diffraction efficiencies
REF0 = sum(ref);                           %compute reflectance
```


Step 10: R & T at f_0 – *Transmittance*

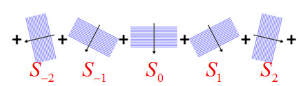
Normalize the Field Amplitude to the Source

$$E_{\text{trn}}\left(x,f_0\right) \cong E_{\text{trn}}\left(x,f_0\right) \div E_{\text{src}}\left(f_0\right)$$

This makes it look like the source at f_0 had an amplitude of 1.

Calculate the Amplitudes of the Spatial Harmonics

$$\text{FFT}\left[E_{\text{trn}}\left(x\right)\right] = \left[S_{-M} \quad \cdots \quad S_{-2} \quad S_{-1} \quad S_0 \quad S_1 \quad S_2 \quad \cdots \quad S_M\right]$$



Calculate the Diffraction Efficiencies of the Spatial Harmonics

$$\text{DE}_{\text{trn}}\left(m\right) = \frac{\left|\vec{S}_{\text{trn}}\left(m\right)\right|^2}{\left|\vec{S}_{\text{inc}}\right|^2} \text{Re}\left[\frac{k_{y,\text{trn}}\left(m\right)}{k_{y,\text{inc}}} \frac{\mu_{r,\text{ref}}}{\mu_{r,\text{trn}}}\right]$$

Calculate the Overall Transmittance

$$T\left(f_0\right) = \sum_{N_x} \text{DE}_{\text{trn}}\left(m\right)$$

```
% COMPUTE TRANSMITTANCE
trn  = Etrn0 / ssSRC;                      %normalize to source power
trn  = fftshift(fft(trn))/Nx;              %compute spatial harmonics
trn  = real(kyT/kyinc) .* abs(trn).^2;     %compute diffraction efficiencies
TRN0 = sum(trn);                           %compute reflectance
```

Step 10: R & T for Frequency Sweep

Initialize the Reflectance and Transmittance Data Arrays

```
% INITIALIZE REFLECTANCE AND TRANSMITTANCE
REF = zeros(1,NFREQ);
TRN = zeros(1,NFREQ);
```

Note: we only need one value per frequency.

Condense the Last Few Slides Inside a Loop Over Frequency

```
% LOOP OVER FREQUENCY
for nfreq = 1 : NFREQ
    % Compute Wave Vector Components
    lam0 = c0/FREQ(nfreq);                %free space wavelength
    k0   = 2*pi/lam0;                     %free space wave number
    kyinc = k0*nref;                      %incident wave vector
    m    = [-floor(Nx/2):floor(Nx/2)]';   %spatial harmonic orders
    kx   =  - 2*pi*m/Sx;                  %wave vector expansion
    kyR = sqrt((k0*nref)^2 - kx.^2);      %ky in reflection region
    kyT = sqrt((k0*ntrn)^2 - kx.^2);      %ky in transmission region

    % Compute Reflectance
    ref = Eref(:,nfreq)/SRC(nfreq);       %normalize to source
    ref = fftshift(fft(ref))/Nx;          %compute spatial harmonics
    ref = real(kyR/kyinc) .* abs(ref).^2; %compute diffraction eff.
    REF(nfreq) = sum(ref);                %compute reflectance

    % Compute Transmittance
    trn = Etrn(:,nfreq)/SRC(nfreq);       %normalize to source
    trn = fftshift(fft(trn))/Nx;          %compute spatial harmonics
    trn = real(kyT/kyinc) .* abs(trn).^2; %compute diffraction eff.
    TRN(nfreq) = sum(trn);                %compute transmittance
end
```


Step 10: R & T at f_0 – *Energy conservation*

Compute the Conservation of Energy

$$R(f_0)+T(f_0)=?$$

If your simulation does not contain loss or gain, this sum should be very close to 100%.

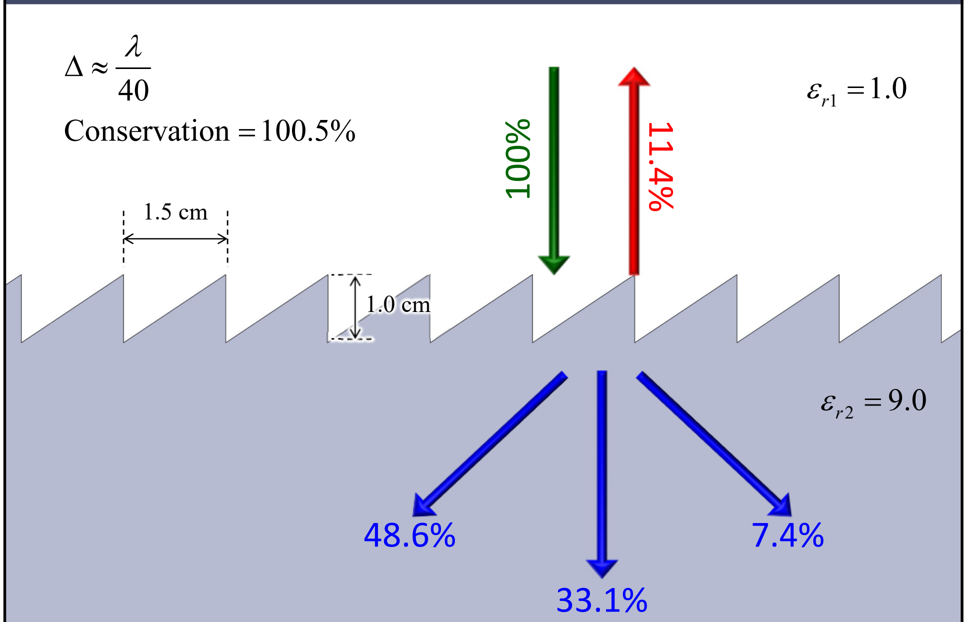
```
% COMPUTE ENERGY CONSERVATION
CON0 = REF0 + TRN0;
CON  = REF  + TRN;

% REPORT RESULTS AT DESIGN FREQUENCY
disp(['Reflectance   = ' num2str(100*REF0,'%4.1f') '%']);
disp(['Transmittance = ' num2str(100*TRN0,'%4.1f') '%']);
disp(['Conservation  = ' num2str(100*CON0,'%4.1f') '%']);
```

Step 11: Produce Professional Looking Results

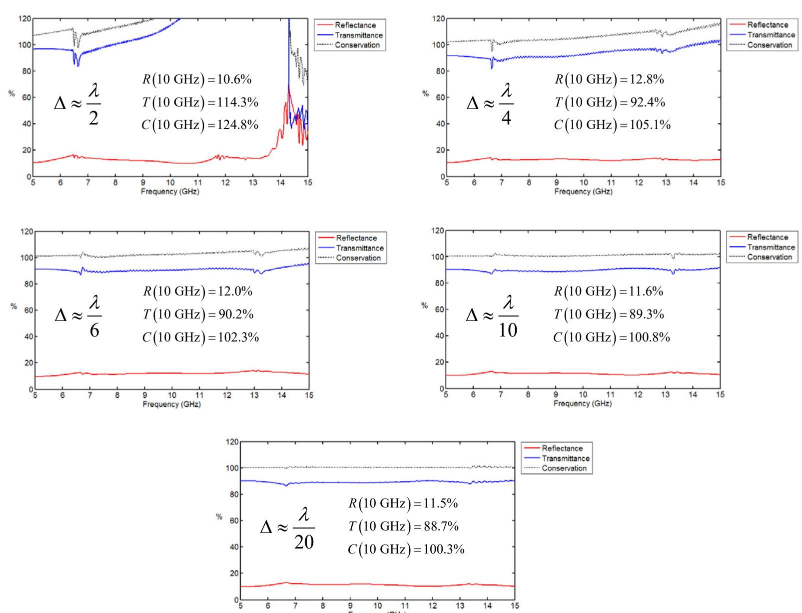
```
% INITIALIZE FIGURE WINDOW
close all;
fig = figure('Color','w');

% PLOT LINEAR REFLECTANCE, TRANSMITTANCE AND ENERGY CONSERVATION
h = plot(FREQ/gigahertz,100*REF,'-r','LineWidth',2);
hold on;
plot(FREQ/gigahertz,100*TRN,'-b','LineWidth',2);
plot(FREQ/gigahertz,100*CON,':k','LineWidth',2);
hold off;

axis([FREQ(1)/gigahertz FREQ(NFREQ)/gigahertz 0 105 ]);
h2 = get(h,'Parent');
set(h2,'FontSize',14,'LineWidth',2);
h = legend('Reflectance','Transmittance','Conservation');
set(h,'Location','NorthEastOutside');

xlabel('Frequency (GHz)');
ylabel('%','Rotation',0,'HorizontalAlignment','right');
```


Results

Diffraction Efficiencies at 10 GHz

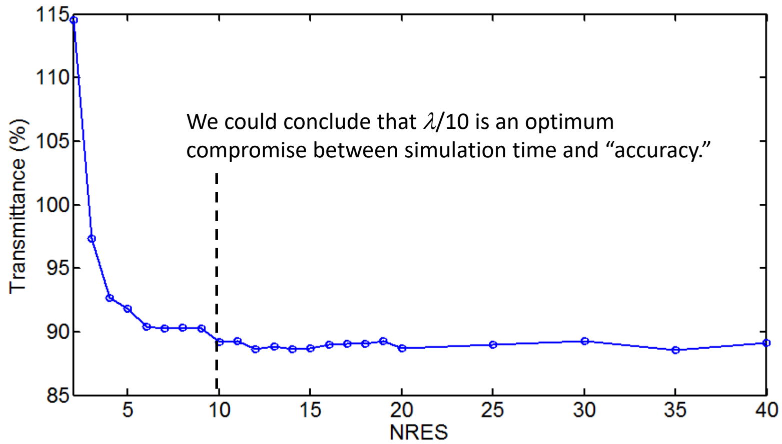


Convergence (1 of 2)



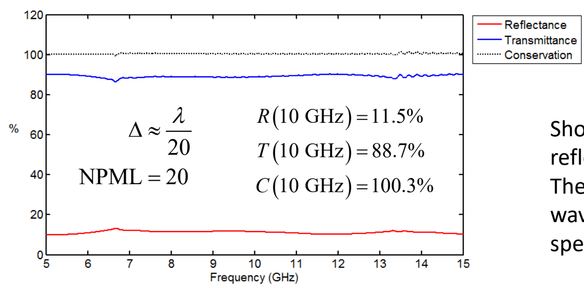
Convergence (2 of 2)

The best way to assess convergence is to plot the desired output parameter as a function of grid resolution for this model.

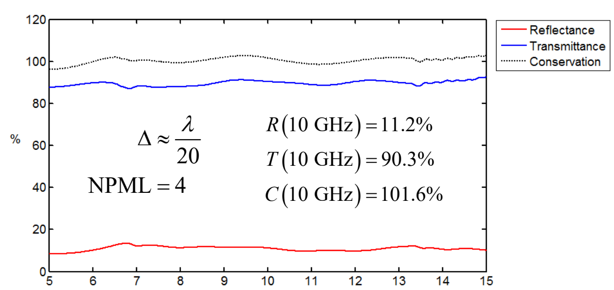


Lecture 21 Slide 39

Effect of Short PML



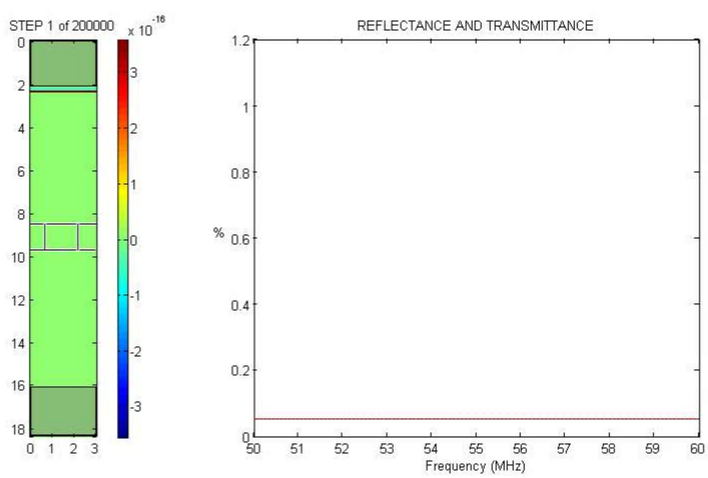
Short PMLs cause non-physical reflections from the boundaries. These reflections cause standing waves that produce "rolling" in the spectral response.



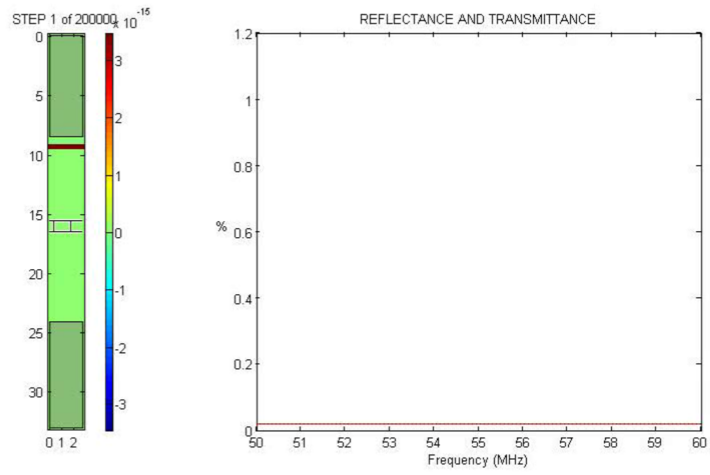
Lecture 21 Slide 40

What Could Possibly Go Wrong?

Good Simulation of a GMR Filter



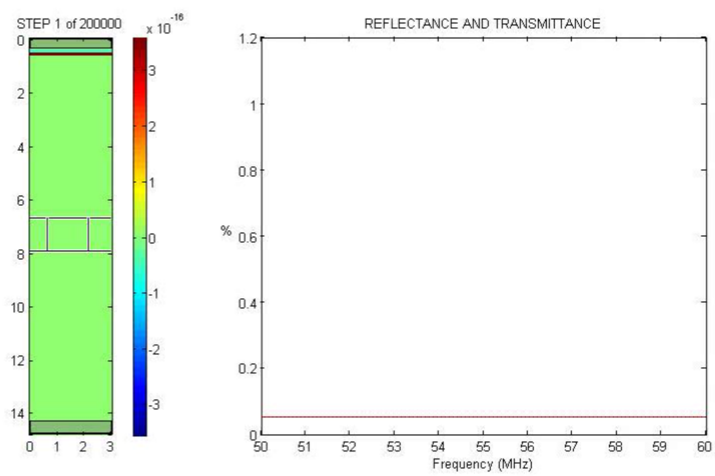
Same Simulation With Poor Grid Resolution



Aside from potentially garbage results, poor grid resolution produces large numerical dispersion that tends to shift spectra to lower frequencies, or longer wavelengths.

Lecture 21 Slide 43

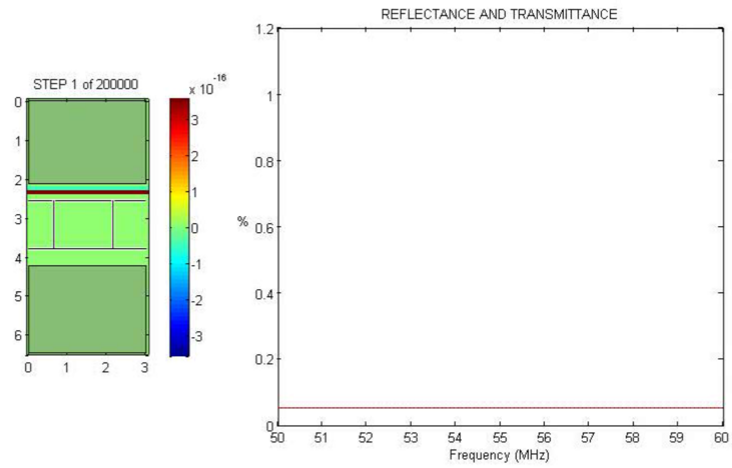
Same Simulation With Small PML



Reflections from boundaries produces a rolling frequency response that also violates conservation.

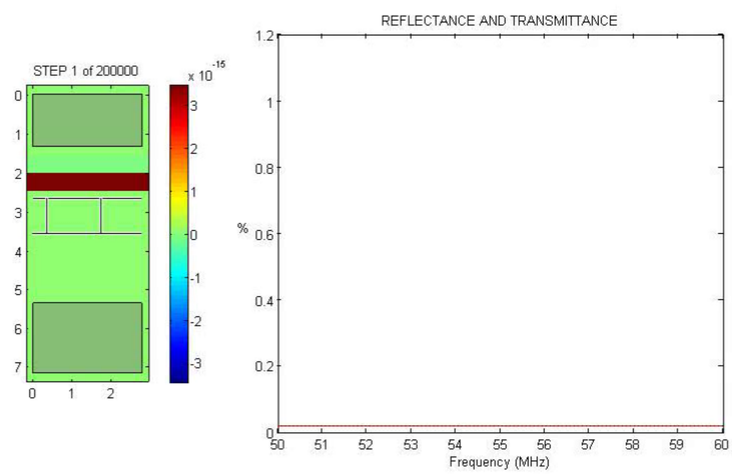
Lecture 21 Slide 44

Same Simulation with Small Spacer Region



Small spacer regions provides an escape path for power by allowing evanescent fields to couple to the PML's. In this simulation, the largest evanescent field occurs due to the guided mode on resonance.

Same Simulation with Everything Wrong!



It is sometimes difficult to make conclusions when multiple things are wrong. It is surprising that this simulation is as good as it is.