



Computational Science:  
Introduction to Finite-Difference Time-Domain  
Windowing & Grid Techniques

## Lecture Outline

- Review of Lecture 11
- Windowing } How to minimize the consequences of limiting the duration of the simulation.
- 2× Grid Technique } How to construct the material arrays for arbitrarily shaped structures.
- Dielectric Averaging } How to minimize the consequences of fitting devices to a Cartesian grid.

# Review

Slide 3

## Maxwell's Equation with Normalized Electric Field

The electric field quantities were normalized

$$\vec{\tilde{E}} = \sqrt{\frac{\epsilon_0}{\mu_0}} \vec{E} = \frac{1}{\eta_0} \vec{E} \quad \vec{\tilde{D}} = \frac{1}{\sqrt{\mu_0 \epsilon_0}} \vec{D} = c_0 \vec{D}$$

Maxwell's equations became

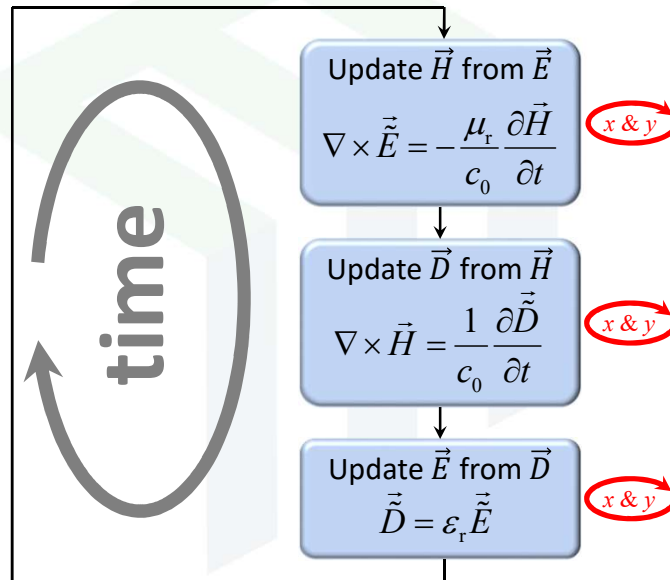
$$\nabla \times \vec{H} = \frac{1}{c_0} \frac{\partial \vec{\tilde{D}}}{\partial t}$$

$$\nabla \times \vec{\tilde{E}} = -\frac{\mu_r}{c_0} \frac{\partial \vec{H}}{\partial t}$$

$$\vec{\tilde{D}} = \epsilon_r \vec{\tilde{E}}$$

Slide 4

## Revised Flow of Maxwell's Equations



## Expanded Maxwell's Equations

These are the final form of Maxwell's equations from which 2D and 3D FDTD methods will be formulated.

$$\frac{\partial \tilde{E}_z}{\partial y} - \frac{\partial \tilde{E}_y}{\partial z} = -\frac{\mu_{xx}}{c_0} \frac{\partial H_x}{\partial t}$$

$$\frac{\partial \tilde{E}_x}{\partial z} - \frac{\partial \tilde{E}_z}{\partial x} = -\frac{\mu_{yy}}{c_0} \frac{\partial H_y}{\partial t}$$

$$\frac{\partial \tilde{E}_y}{\partial x} - \frac{\partial \tilde{E}_x}{\partial y} = -\frac{\mu_{zz}}{c_0} \frac{\partial H_z}{\partial t}$$

$$\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \frac{1}{c_0} \frac{\partial \tilde{D}_x}{\partial t}$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \frac{1}{c_0} \frac{\partial \tilde{D}_y}{\partial t}$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{1}{c_0} \frac{\partial \tilde{D}_z}{\partial t}$$

$$\tilde{D}_x = \epsilon_{xx} \tilde{E}_x$$

$$\tilde{D}_y = \epsilon_{yy} \tilde{E}_y$$

$$\tilde{D}_z = \epsilon_{zz} \tilde{E}_z$$

## Finite-Difference Approximations

$$\begin{aligned} \frac{\partial \tilde{E}_z}{\partial y} - \frac{\partial \tilde{E}_y}{\partial z} &= -\frac{\mu_{xx}}{c_0} \frac{\partial H_x}{\partial t} & \frac{\tilde{E}_z|_l^{i,j,k+1} - \tilde{E}_z|_l^{i,j,k}}{\Delta y} - \frac{\tilde{E}_y|_l^{i,j,k+1} - \tilde{E}_y|_l^{i,j,k}}{\Delta z} &= -\frac{\mu_{xx}|_l^{i,j,k}}{c_0} \frac{H_x|_{l+\frac{\Delta y}{2}}^{i,j,k} - H_x|_{l-\frac{\Delta y}{2}}^{i,j,k}}{\Delta t} \\ \frac{\partial \tilde{E}_x}{\partial z} - \frac{\partial \tilde{E}_z}{\partial x} &= -\frac{\mu_{yy}}{c_0} \frac{\partial H_y}{\partial t} & \frac{\tilde{E}_x|_l^{i,j,k+1} - \tilde{E}_x|_l^{i,j,k}}{\Delta z} - \frac{\tilde{E}_z|_l^{i,j,k+1} - \tilde{E}_z|_l^{i,j,k}}{\Delta x} &= -\frac{\mu_{yy}|_l^{i,j,k}}{c_0} \frac{H_y|_{l+\frac{\Delta z}{2}}^{i,j,k} - H_y|_{l-\frac{\Delta z}{2}}^{i,j,k}}{\Delta t} \\ \frac{\partial \tilde{E}_y}{\partial x} - \frac{\partial \tilde{E}_x}{\partial y} &= -\frac{\mu_{zz}}{c_0} \frac{\partial H_z}{\partial t} & \frac{\tilde{E}_y|_l^{i,j,k+1} - \tilde{E}_y|_l^{i,j,k}}{\Delta x} - \frac{\tilde{E}_x|_l^{i,j,k+1} - \tilde{E}_x|_l^{i,j,k}}{\Delta y} &= -\frac{\mu_{zz}|_l^{i,j,k}}{c_0} \frac{H_z|_{l+\frac{\Delta x}{2}}^{i,j,k} - H_z|_{l-\frac{\Delta x}{2}}^{i,j,k}}{\Delta t} \end{aligned}$$

$$\begin{aligned} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} &= \frac{1}{c_0} \frac{\partial \tilde{D}_x}{\partial t} & \frac{H_z|_{l+\frac{\Delta y}{2}}^{i,j,k} - H_z|_{l+\frac{\Delta y}{2}}^{i,j,k-1}}{\Delta y} - \frac{H_y|_{l+\frac{\Delta z}{2}}^{i,j,k} - H_y|_{l+\frac{\Delta z}{2}}^{i,j,k-1}}{\Delta z} &= \frac{1}{c_0} \frac{\tilde{D}_x|_{l+\Delta t}^{i,j,k} - \tilde{D}_x|_l^{i,j,k}}{\Delta t} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} &= \frac{1}{c_0} \frac{\partial \tilde{D}_y}{\partial t} & \frac{H_x|_{l+\frac{\Delta z}{2}}^{i,j,k} - H_x|_{l+\frac{\Delta z}{2}}^{i,j,k-1}}{\Delta z} - \frac{H_z|_{l+\frac{\Delta x}{2}}^{i,j,k} - H_z|_{l+\frac{\Delta x}{2}}^{i,j,k-1}}{\Delta x} &= \frac{1}{c_0} \frac{\tilde{D}_y|_{l+\Delta t}^{i,j,k} - \tilde{D}_y|_l^{i,j,k}}{\Delta t} \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \frac{1}{c_0} \frac{\partial \tilde{D}_z}{\partial t} & \frac{H_y|_{l+\frac{\Delta x}{2}}^{i,j,k} - H_y|_{l+\frac{\Delta x}{2}}^{i,j,k-1}}{\Delta x} - \frac{H_x|_{l+\frac{\Delta y}{2}}^{i,j,k} - H_x|_{l+\frac{\Delta y}{2}}^{i,j,k-1}}{\Delta y} &= \frac{1}{c_0} \frac{\tilde{D}_z|_{l+\Delta t}^{i,j,k} - \tilde{D}_z|_l^{i,j,k}}{\Delta t} \end{aligned}$$

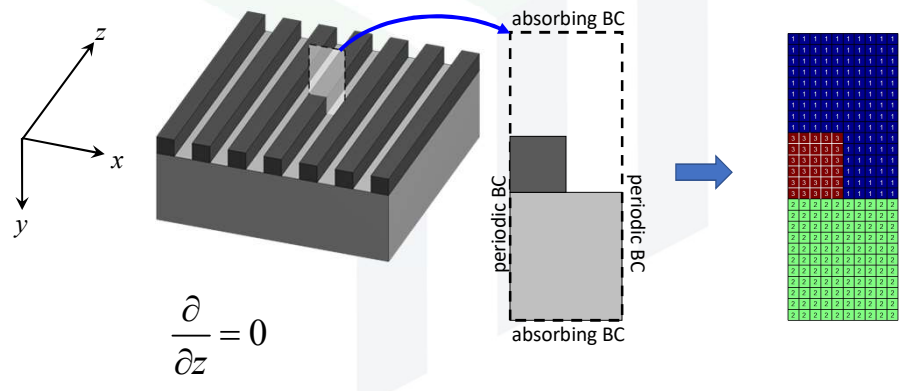
$$\begin{aligned} \tilde{D}_x &= \epsilon_{xx} \tilde{E}_x & \tilde{D}_x|_l^{i,j,k} &= (\epsilon_{xx}|_l^{i,j,k}) \tilde{E}_x|_l^{i,j,k} \\ \tilde{D}_y &= \epsilon_{yy} \tilde{E}_y & \tilde{D}_y|_l^{i,j,k} &= (\epsilon_{yy}|_l^{i,j,k}) \tilde{E}_y|_l^{i,j,k} \\ \tilde{D}_z &= \epsilon_{zz} \tilde{E}_z & \tilde{D}_z|_l^{i,j,k} &= (\epsilon_{zz}|_l^{i,j,k}) \tilde{E}_z|_l^{i,j,k} \end{aligned}$$



Slide 7

## Reduce Problems to Two Dimensions

Sometimes it is possible to describe a physical device using just two dimensions. Doing so dramatically reduces the numerical complexity of the problem and is ALWAYS GOOD PRACTICE.



Slide 8

## Two Distinct Modes

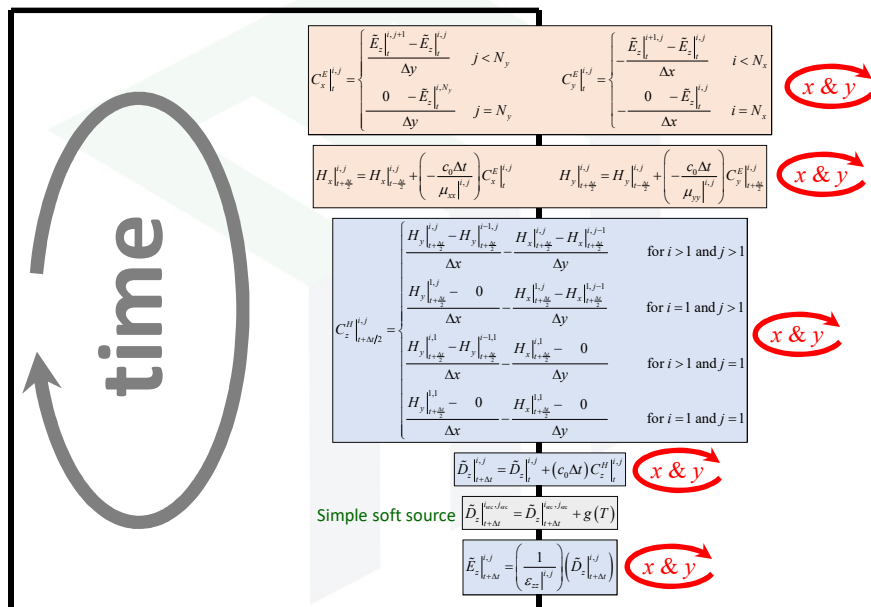
$E_z$  Mode

$$\begin{aligned}
 C_x^E &= \frac{\partial \tilde{E}_z}{\partial y} & C_x^E &= -\frac{\mu_{xx}}{c_0} \frac{\partial H_x}{\partial t} & \rightarrow & C_x^{E|j} = \frac{\tilde{E}_z|^{j+1} - \tilde{E}_z|^{j}}{\Delta y} & C_x^{E|j} &= -\frac{\mu_{xx}|^j}{c_0} \frac{H_x|_{l+\frac{\Delta x}{2}}^{j,j} - H_x|_{l-\frac{\Delta x}{2}}^{j,j}}{\Delta t} \\
 C_y^E &= -\frac{\partial \tilde{E}_z}{\partial x} & C_y^E &= -\frac{\mu_{yy}}{c_0} \frac{\partial H_y}{\partial t} & \rightarrow & C_y^{E|j} = -\frac{\tilde{E}_z|^{j+1} - \tilde{E}_z|^{j}}{\Delta x} & C_y^{E|j} &= -\frac{\mu_{yy}|^j}{c_0} \frac{H_y|_{l+\frac{\Delta y}{2}}^{j,j} - H_y|_{l-\frac{\Delta y}{2}}^{j,j}}{\Delta t} \\
 C_z^H &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} & C_z^H &= \frac{1}{c_0} \frac{\partial \tilde{D}_z}{\partial t} & \rightarrow & C_z^{H|j} = \frac{H_y|_{l+\frac{\Delta x}{2}}^{j,j} - H_y|_{l-\frac{\Delta x}{2}}^{j,j}}{\Delta x} - \frac{H_x|_{l+\frac{\Delta y}{2}}^{j,j} - H_x|_{l-\frac{\Delta y}{2}}^{j,j}}{\Delta y} & C_z^{H|j} &= \frac{1}{c_0} \frac{\tilde{D}_z|_{l+\frac{\Delta z}{2}}^{j,j} - \tilde{D}_z|_{l-\frac{\Delta z}{2}}^{j,j}}{\Delta t} \\
 & & & & \rightarrow & \tilde{D}_z = \epsilon_{zz} \tilde{E}_z & \rightarrow & \tilde{D}_z|_l^{j,j} = (\epsilon_{zz}|^j) \tilde{E}_z|_l^{j,j}
 \end{aligned}$$

$H_z$  Mode

$$\begin{aligned}
 C_z^E &= \frac{\partial \tilde{E}_y}{\partial x} - \frac{\partial \tilde{E}_x}{\partial y} & C_z^E &= -\frac{\mu_{zz}}{c_0} \frac{\partial H_z}{\partial t} & \rightarrow & C_z^{E|j} = \frac{\tilde{E}_y|_l^{j+1} - \tilde{E}_y|_l^{j,j} - \tilde{E}_x|_l^{j,j+1} - \tilde{E}_x|_l^{j,j}}{\Delta x} & C_z^{E|j} &= -\frac{\mu_{zz}|^j}{c_0} \frac{H_z|_{l+\frac{\Delta z}{2}}^{j,j} - H_z|_{l-\frac{\Delta z}{2}}^{j,j}}{\Delta t} \\
 C_x^H &= \frac{\partial H_z}{\partial y} & C_x^H &= \frac{1}{c_0} \frac{\partial \tilde{D}_x}{\partial t} & \rightarrow & C_x^{H|j} = \frac{H_z|_{l+\frac{\Delta y}{2}}^{j,j} - H_z|_{l-\frac{\Delta y}{2}}^{j,j}}{\Delta y} & C_x^{H|j} &= \frac{1}{c_0} \frac{\tilde{D}_x|_{l+\frac{\Delta x}{2}}^{j,j} - \tilde{D}_x|_{l-\frac{\Delta x}{2}}^{j,j}}{\Delta t} \\
 C_y^H &= -\frac{\partial H_z}{\partial x} & C_y^H &= \frac{1}{c_0} \frac{\partial \tilde{D}_y}{\partial t} & \rightarrow & C_y^{H|j} = -\frac{H_z|_{l+\frac{\Delta x}{2}}^{j,j} - H_z|_{l-\frac{\Delta x}{2}}^{j,j}}{\Delta x} & C_y^{H|j} &= \frac{1}{c_0} \frac{\tilde{D}_y|_{l+\frac{\Delta y}{2}}^{j,j} - \tilde{D}_y|_{l-\frac{\Delta y}{2}}^{j,j}}{\Delta t} \\
 & & & & \rightarrow & \tilde{D}_x = \epsilon_{xx} \tilde{E}_x & \rightarrow & \tilde{D}_x|_l^{j,j} = (\epsilon_{xx}|^j) \tilde{E}_x|_l^{j,j} \\
 & & & & \rightarrow & \tilde{D}_y = \epsilon_{yy} \tilde{E}_y & \rightarrow & \tilde{D}_y|_l^{j,j} = (\epsilon_{yy}|^j) \tilde{E}_y|_l^{j,j}
 \end{aligned}$$

## FDTD Algorithm for $E_z$ Mode

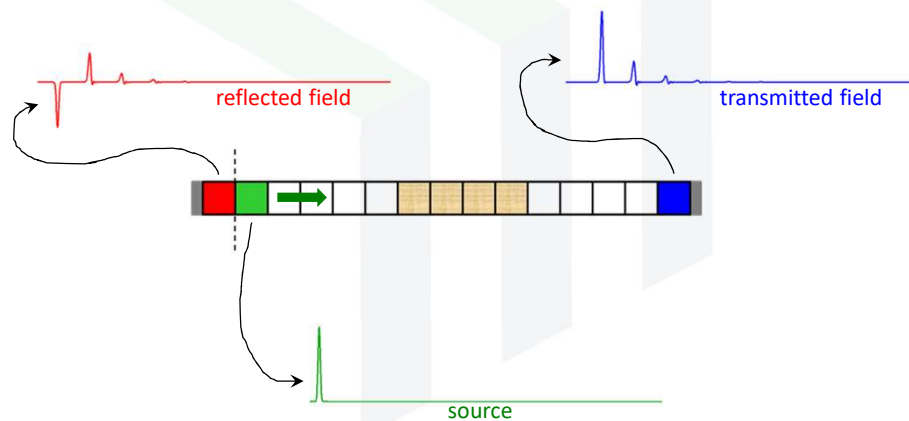


# Windowing

Slide 11

## Time Response from FDTD

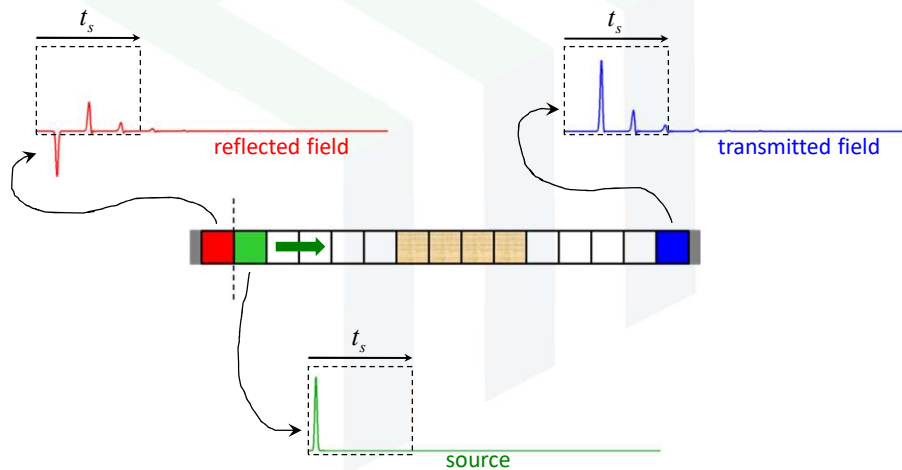
FDTD is a time-domain method so it is the transient response of a device that is characterized. A Fourier transform must be used to calculate frequency-domain, or steady-state, quantities.



Slide 12

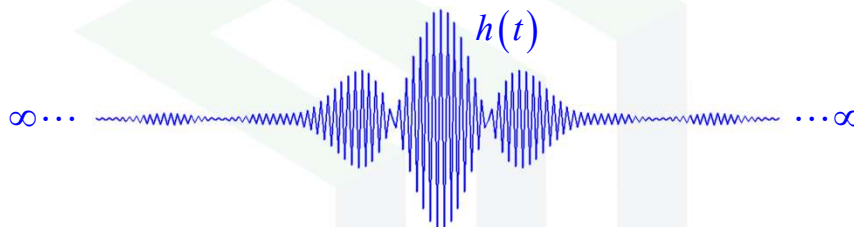
## Duration of the Simulation (i.e. STEPS)

What happens when the simulation is stopped at time  $t_s$  before the simulation is “finished?”

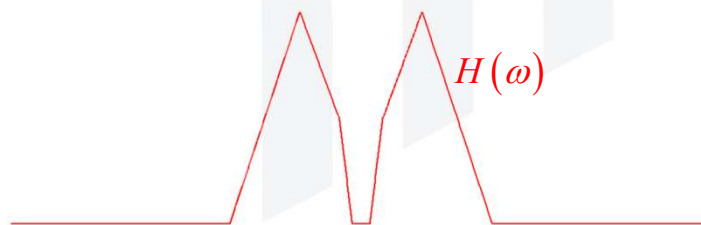


## Example Impulse Response

FDTD produces the “impulse response”  $h(t)$  of a device. In principle, the impulse response extends to infinity.

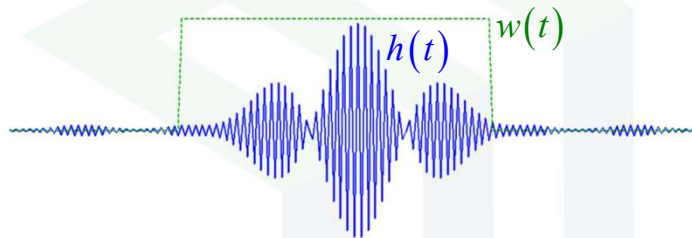


The spectrum of the infinite impulse response may be something like this...

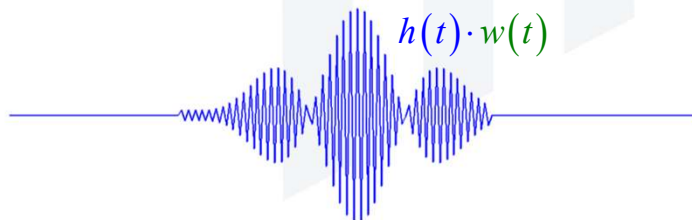


## Mathematical Description of Finite Duration

When a simulation is stopped at time  $t_s$ , it is just like multiplying the infinite impulse response by a pulse function  $w(t)$  with duration  $t_s$ .



This is the resulting “infinite” impulse response when the simulation time is truncated.



## A Property of the Fourier Transform

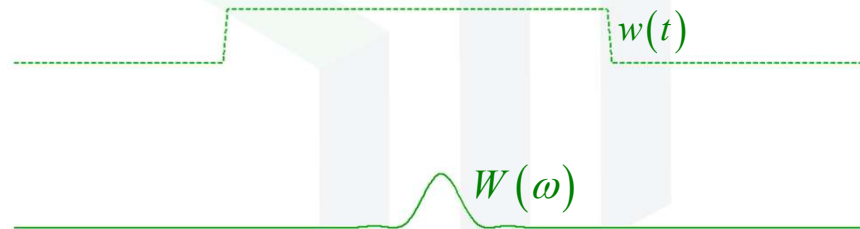
The Fourier transform of the product of two functions is the convolution of the Fourier transform of the two functions.

In terms of our functions  $h(t)$  and  $w(t)$ , this is

$$F\{h(t) \cdot w(t)\} = F\{h(t)\} \otimes F\{w(t)\} = H(\omega) \otimes W(\omega)$$

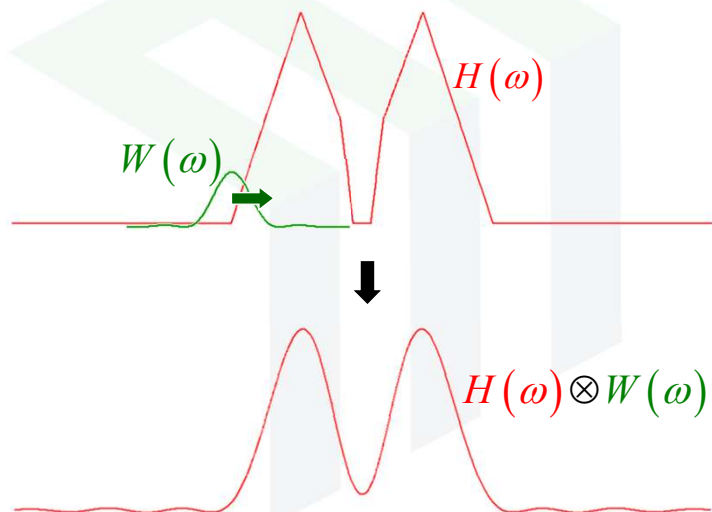
## Fourier Transform of the Window Function

The window function and its Fourier transform are:

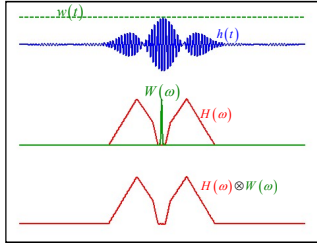


## $H(\omega)$ is “Blurred” by the Window Function

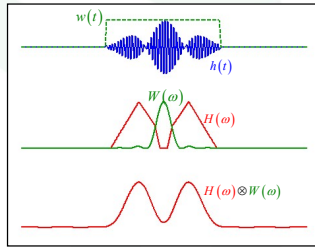
Every point  $H(\omega)$  is essentially blurred by the window function  $W(\omega)$  due to the convolution.



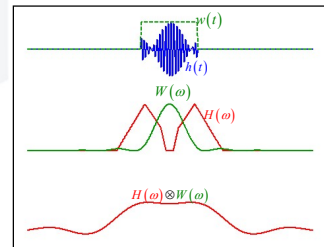
## Severity Trend for Windowing



- Long simulation time
- Wide window
- Narrow window spectrum
- Less "blurring"
- High frequency resolution



- Moderate simulation time
- Moderate window
- Moderate window spectrum
- More "blurring"
- Reduced frequency resolution



- Short simulation time
- Narrow window
- Wide window spectrum
- Much "blurring"
- Poor frequency resolution

## Sampling and Windowing in FDTD

Given the time step  $\Delta t$ , the Nyquist sampling theorem quantifies the highest frequency that can be resolved.

$$f_{\max} = \frac{1}{2\Delta t}$$

Note: In practice, you should set  $\Delta t = \frac{1}{N_t f_{\max}}$   $N_t \geq 10$

Given the FDTD simulation runs for STEPS number of iterations, the frequency resolution is

$$\Delta f = \frac{2f_{\max}}{\text{STEPS}} = \frac{1}{\Delta t \cdot \text{STEPS}}$$

Therefore, to resolve the frequency response down to a resolution of  $\Delta f$ , the number of iterations required is

$$\text{STEPS} \geq \frac{1}{\Delta t \cdot \Delta f}$$

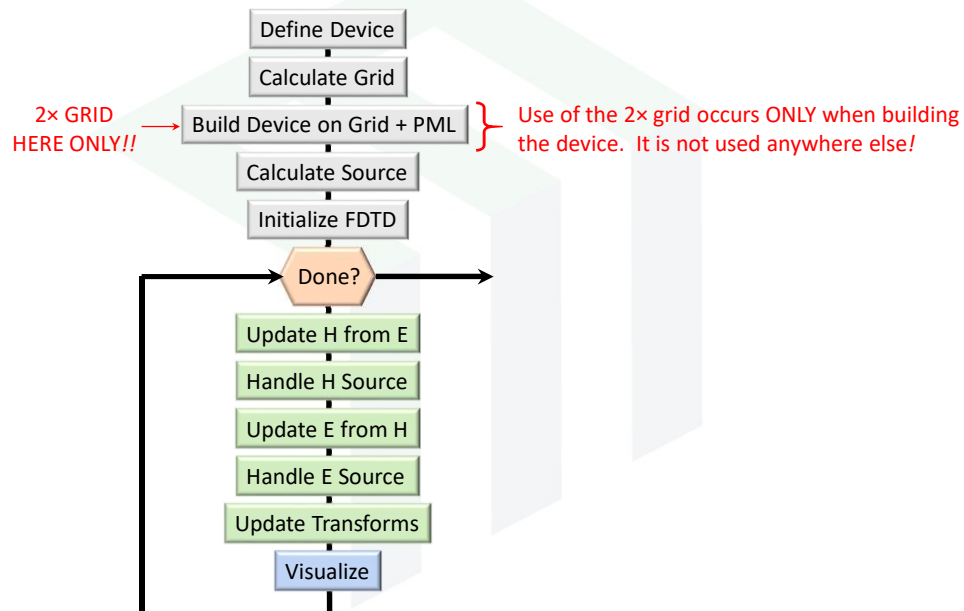
Notes:

1. This is related to the uncertainty principle.
2. You can calculate kernels for very finely spaced frequency points, but the actual frequency resolution is still limited by the windowing effect. Your data will be "blurred."

# 2× Grid Technique

Slide 21

## What is the 2× Grid Technique? (1 of 3)

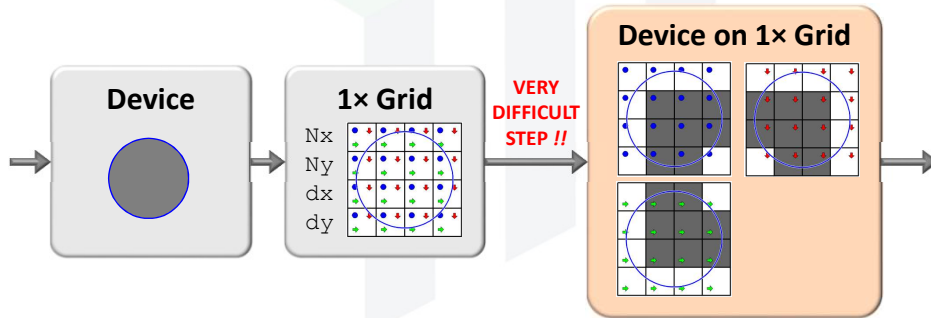


Slide 22

## What is the 2x Grid Technique? (2 of 3)

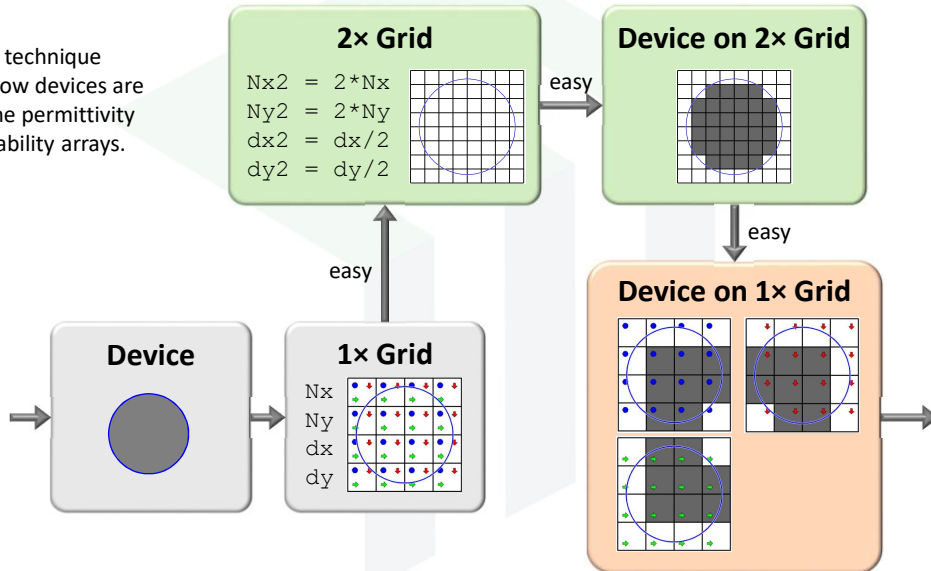
This is the traditional approach for building devices on a Yee grid.

It is very tedious and cumbersome to determine which field components reside in which material.



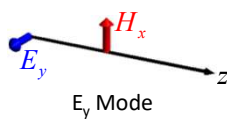
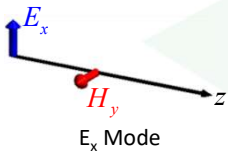
## What is the 2x Grid Technique? (3 of 3)

The 2x grid technique simplifies how devices are built into the permittivity and permeability arrays.

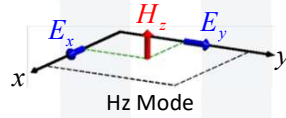
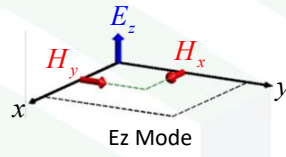


# Recall the Yee Grid

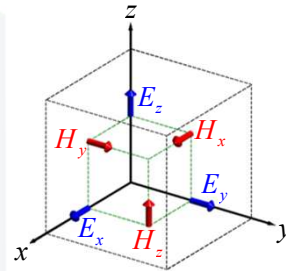
## 1D Yee Grid



## 2D Yee Grids



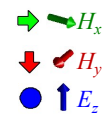
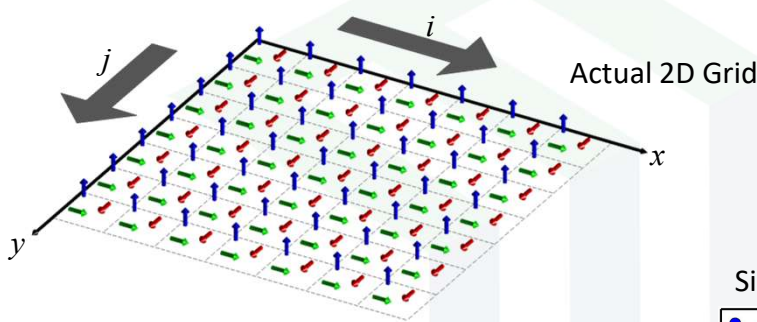
## 3D Yee Grid



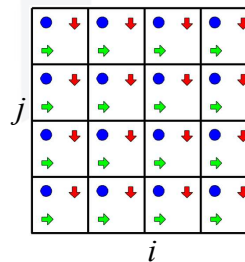
Define  $ER_{xx}$  at the same points as  $E_x$ .  
 Define  $ER_{yy}$  at the same points as  $E_y$ .  
 Define  $ER_{zz}$  at the same points as  $E_z$ .

Define  $UR_{xx}$  at the same points as  $H_x$ .  
 Define  $UR_{yy}$  at the same points as  $H_y$ .  
 Define  $UR_{zz}$  at the same points as  $H_z$ .

# 4x4 Yee Grid for the $E_z$ Mode



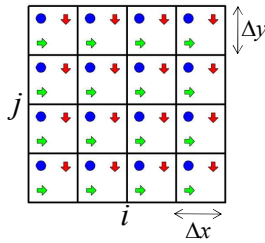
## Simplified Grid



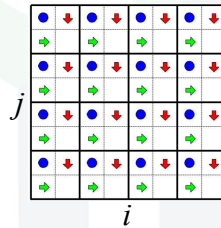
- The field components are physically positioned at the edges of the cell.
- The simplified representation shows the fields inside the cells to convey more clearly which cell they are in.

# The 2x Grid

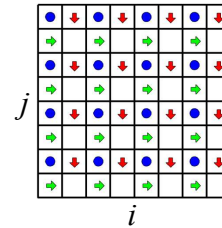
The Conventional 1x Grid



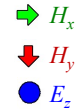
Due to the staggered nature of the Yee grid, we are effectively getting twice the resolution.



It now makes sense to talk about a grid that is at twice the resolution, the "2x grid."



The 2x grid concept is useful because we can create devices (or PMLs) on the 2x grid without having to think about where the different field components are located. In a second step, we can easily pull off the values from the 2x grid where they exist for a particular field component.



# 2x Grid Technique (1 of 9)

We define our ordinary "1x" grid as usual.

The output of this step is the number of cells in the grid  $N_x$  and  $N_y$  and the size of the cells in the grid  $dx$  and  $dy$ .

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```

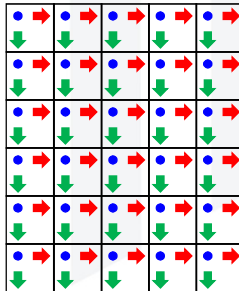


## 2x Grid Technique (2 of 9)

Recall how the various functions overlay onto the grid.

Functions assigned to the same grid cell are in physically different positions and may reside in different materials as a result.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```



## 2x Grid Technique (3 of 9)

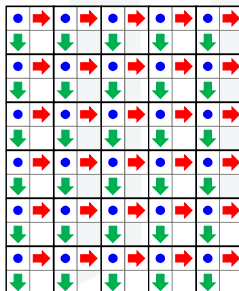
It is like we are getting twice the resolution due to the staggering of the functions.

In order to sort out what values go where, we construct a "2x" grid at twice the resolution of the original grid.

The 2x grid occupies the same physical amount of space as the original grid.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;
```

```
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;
```



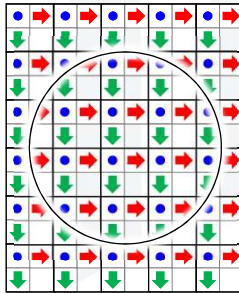
## 2× Grid Technique (4 of 9)

Let's say we wish to construct a cylinder of radius 2 on our grid.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
```



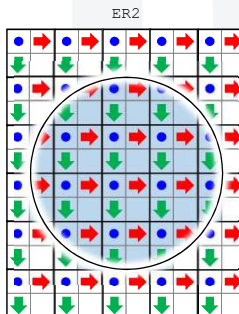
## 2× Grid Technique (5 of 9)

We start by building our object on the 2× grid, ignoring anything about or original grid for now.

```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

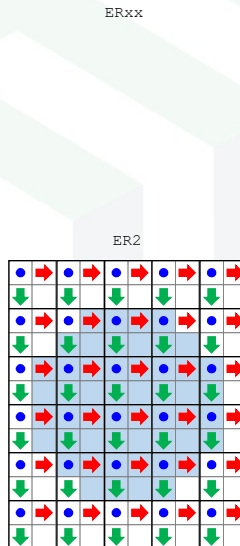
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CIRCLE
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;
```



## 2x Grid Technique (6 of 9)

Given the object on the 2x grid, we extract ER<sub>xx</sub> by grabbing values from ER<sub>2</sub> that correspond to the locations of ER<sub>xx</sub>.



```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

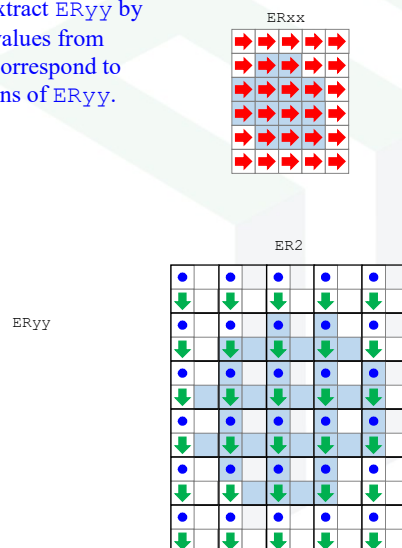
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
```

## 2x Grid Technique (7 of 9)

We then extract ER<sub>yy</sub> by grabbing values from ER<sub>2</sub> that correspond to the locations of ER<sub>yy</sub>.



```
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

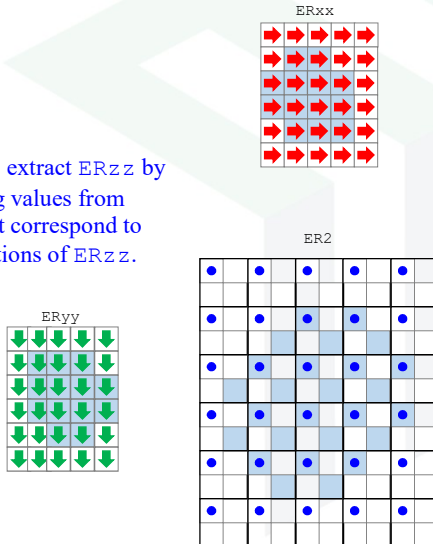
% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
ERyy = ER2(1:2:Nx2,2:2:Ny2);
```

## 2x Grid Technique (8 of 9)

ERzz

Last, we extract ERzz by grabbing values from ER2 that correspond to the locations of ERzz.



```

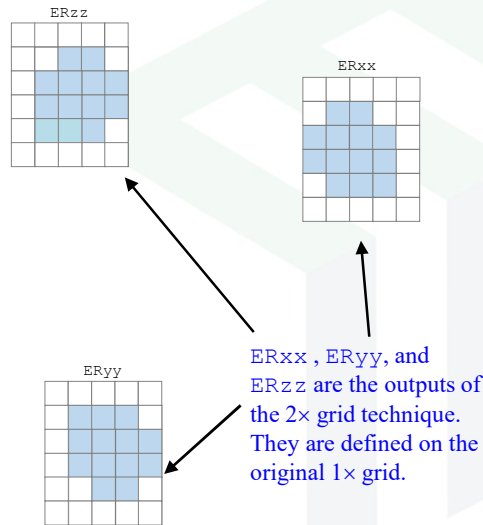
% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
Eryy = ER2(1:2:Nx2,2:2:Ny2);
ERzz = ER2(1:2:Nx2,1:2:Ny2);
    
```

## 2x Grid Technique (8 of 9)



ERxx, Eryy, and ERzz are the outputs of the 2x grid technique. They are defined on the original 1x grid.

```

% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

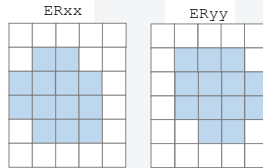
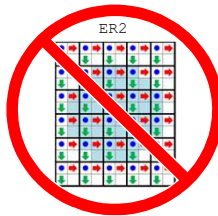
% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
Eryy = ER2(1:2:Nx2,2:2:Ny2);
ERzz = ER2(1:2:Nx2,1:2:Ny2);
    
```

## 2x Grid Technique (9 of 9)

After building ERxx, Eryy and ERzz, the 2x grid is no longer used anywhere.

All of the 2x grid parameters may be deleted at this point because they are no longer needed.



```

% DEFINE GRID
Nx = 5;
Ny = 6;
dx = 1;
dy = 1;

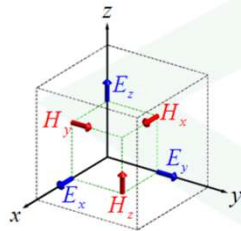
% 2X GRID
Nx2 = 2*Nx;
Ny2 = 2*Ny;
dx2 = dx/2;
dy2 = dy/2;

% CREATE CYLINDER
r = 2;
xa2 = [0:Nx2-1]*dx2;
ya2 = [0:Ny2-1]*dy2;
xa2 = xa2 - mean(xa2);
ya2 = ya2 - mean(ya2);
[Y2,X2] = meshgrid(ya2,xa2);
ER2 = (X2.^2 + Y2.^2) <= r^2;

% EXTRACT 1X GRID PARAMETERS
ERxx = ER2(2:2:Nx2,1:2:Ny2);
Eryy = ER2(1:2:Nx2,2:2:Ny2);
ERzz = ER2(1:2:Nx2,1:2:Ny2);
    
```

**No more 2x grid!!!**

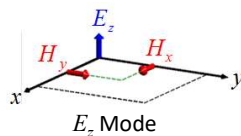
## MATLAB Code for Parsing Onto 1x Grid



```

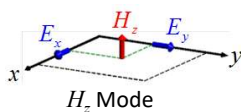
ERxx = ER2(2:2:Nx2,1:2:Ny2,1:2:Nz2);
Eryy = ER2(1:2:Nx2,2:2:Ny2,1:2:Nz2);
ERzz = ER2(1:2:Nx2,1:2:Ny2,2:2:Nz2);

URxx = UR2(1:2:Nx2,2:2:Ny2,2:2:Nz2);
URyy = UR2(2:2:Nx2,1:2:Ny2,2:2:Nz2);
URzz = UR2(2:2:Nx2,2:2:Ny2,1:2:Nz2);
    
```



```

ERzz = ER2(1:2:Nx2,1:2:Ny2);
URxx = UR2(1:2:Nx2,2:2:Ny2);
URyy = UR2(2:2:Nx2,1:2:Ny2);
    
```



```

ERxx = ER2(2:2:Nx2,1:2:Ny2);
Eryy = ER2(1:2:Nx2,2:2:Ny2);
URzz = UR2(2:2:Nx2,2:2:Ny2);
    
```

# Dielectric Averaging

Slide 39

## What is Dielectric Averaging?

Suppose there is a grid, but the size of the device is not an exact integer number of grid cells. What can be done?

Look at a close-up of the cell in question and calculate a weighted average.

$\begin{matrix} \updownarrow 30\% \\ \leftarrow 70\% \\ \epsilon_r = 6.0 \\ \epsilon_r = 1.0 \end{matrix}$

$\approx$

$\epsilon_{r,\text{eff}} = 2.5$

$\epsilon_{r,\text{eff}} = (30\%)(6.0) + (70\%)(1.0) = 2.5$

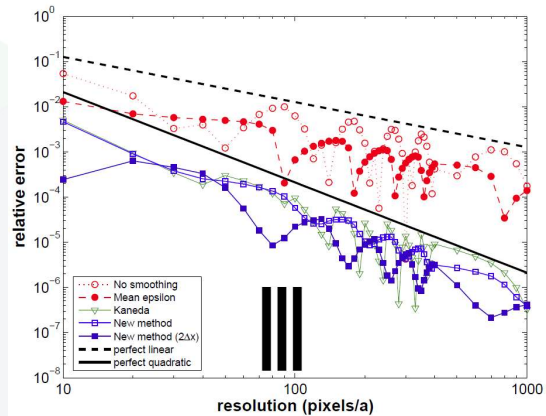
Based on the above, build the device on the grid as follows using that average:

1	1	1	1	6	6	6	6	6	6	6	6	6	6	2.5	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---

Slide 40

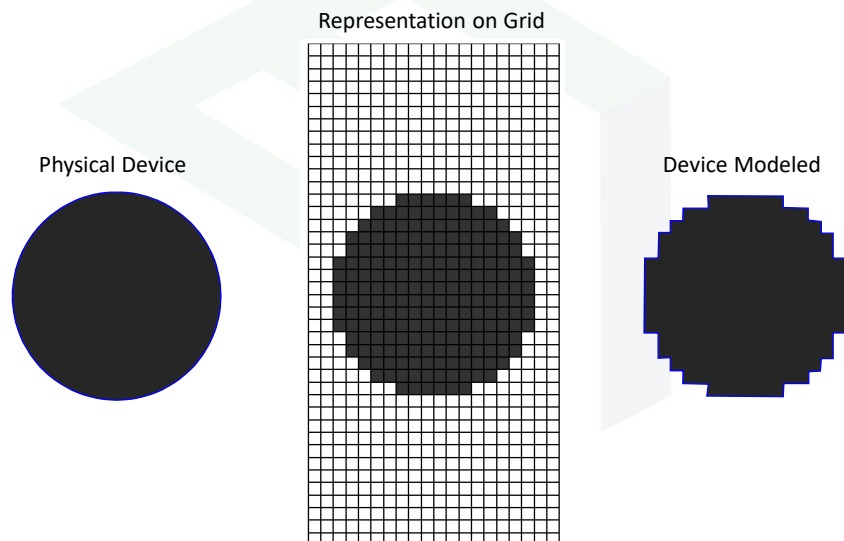
## Reason for Dielectric Averaging

- Dielectric averaging improves the rate of convergence.
- You can get away with coarser grid resolution using dielectric averaging. Your simulations will run faster and be more memory efficient.
  - Perhaps NRES=10 instead of NRES=20.

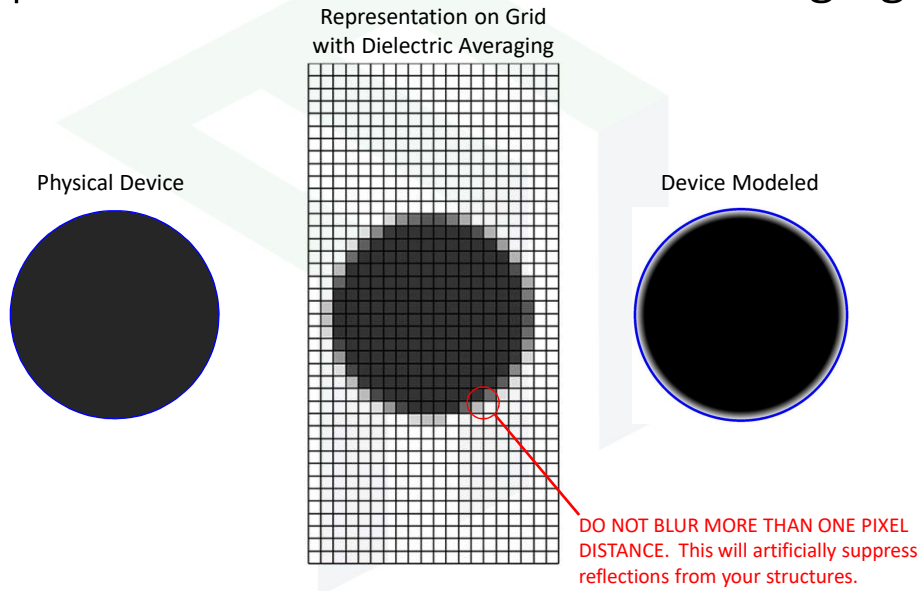


Optics Letters, Vol. 31, Issue 20, pp. 2972-2974 (2006)

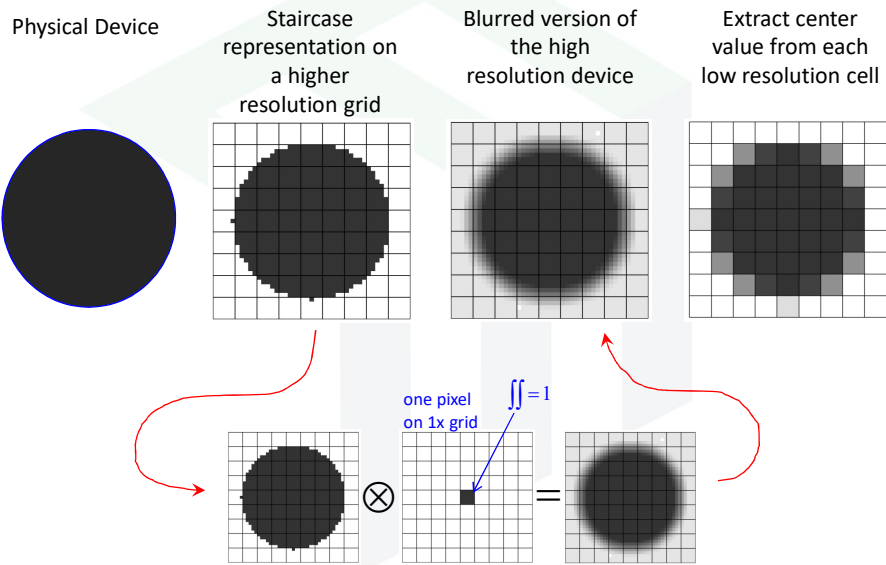
## Typical Staircase Approximation



# Representation with Dielectric Averaging

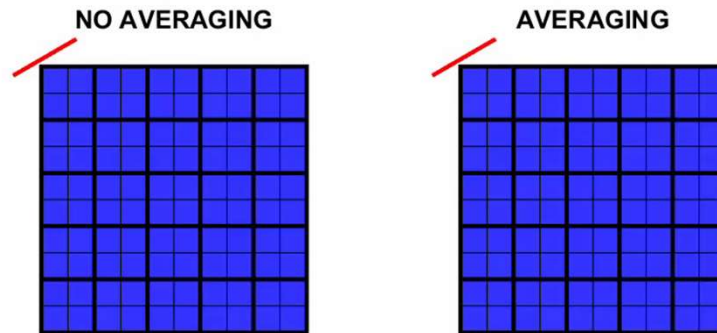


# Procedure to Perform Averaging



Use a convolution to calculate the average dielectric constant in each cell.

## Animation of Dielectric Averaging



## Product of Two Functions

Given the product of two functions that is being approximating numerically.

$$f(z) = a(z) \cdot b(z)$$

Slow convergence is encountered whenever one or both of these functions is discontinuous.

Suppose  $a(z)$  has a discontinuity. Numerical convergence can be significantly improved by “smoothing” the discontinuous function at the discontinuity.

$$f(z) = \langle a(z) \rangle \cdot b(z)$$

## Double Discontinuity (1 of 2)

Now, suppose both  $a(z)$  and  $b(z)$  are discontinuous at the same point. It is very difficult to improve the convergence of this problem.

There exists, however, a special case where  $a(z)$  and  $b(z)$  are discontinuous at the same point, but their product is continuous. That is,  $f(z)$  is continuous.

Rearrange the equation so that only a single discontinuity exists on each side.

$$\frac{1}{a(z)} \cdot f(z) = b(z)$$

## Double Discontinuity (2 of 2)

Improve convergence by smoothing  $1/a(z)$ .

$$\left\langle \frac{1}{a(z)} \right\rangle \cdot f(z) = b(z)$$

Move the  $a(z)$  term to the right hand side to get

$$f(z) = \frac{1}{\left\langle \frac{1}{a(z)} \right\rangle} \cdot b(z)$$

In the double discontinuity case where the product is continuous, smooth the reciprocal of  $a(z)$  and then reciprocate the smoothed function.

## Summary of Smoothing

When all functions are continuous, no smoothing is needed.

$$f(z) = a(z) \cdot b(z)$$

When one of the functions has a step discontinuity, convergence is improved by smoothing that function.

$$f(z) = \langle a(z) \rangle \cdot b(z)$$

When both  $a(z)$  and  $b(z)$  are discontinuous at the same point, but their product is continuous, convergence is improved by smoothing the reciprocal of one of the functions.

$$f(z) = \langle a^{-1}(z) \rangle^{-1} \cdot b(z)$$

## Smoothing and Maxwell's Equations

In Maxwell's equations, there is a product of two functions...

$$\vec{D}(z) = \epsilon_r(z) \cdot \vec{E}(z)$$

The dielectric function is discontinuous at the interface between two materials. Boundary conditions require that

$$E_{1,\parallel} = E_{2,\parallel} \quad \text{Tangential component is continuous across the interface}$$

$$\epsilon_1 E_{1,\perp} = \epsilon_2 E_{2,\perp} \quad \text{Normal component is discontinuous across the interface, but the product of } \epsilon \mathbf{E} \text{ is continuous.}$$

In electromagnetics, smooth the dielectric function differently for the tangential and normal components. This implies that the smoothed dielectric function will be anisotropic and described by a tensor.

## High Level Formulation for Maxwell's Equations

First, decompose the electric field into tangential and normal components at an interface.

$$\epsilon_r(z) \cdot \vec{E}(z) = \epsilon_r(z) \cdot [\vec{E}_{\parallel}(z) + \vec{E}_{\perp}(z)]$$

Second, expand this equation.

$$\epsilon_r(z) \cdot \vec{E}(z) = \epsilon_r(z) \cdot \vec{E}_{\parallel}(z) + \epsilon_r(z) \cdot \vec{E}_{\perp}(z)$$

Third, associate different dielectric functions with the different field components. Before smoothing, they are the same.

$$\epsilon_r(z) \cdot \vec{E}(z) = \epsilon_{\parallel}(z) \vec{E}_{\parallel}(z) + \epsilon_{\perp}(z) \vec{E}_{\perp}(z)$$

Finally, smooth the two dielectric functions differently according to the rules.

$$\epsilon_r(z) \cdot \vec{E}(z) = \langle \epsilon_{\parallel}(z) \rangle \vec{E}_{\parallel}(z) + \langle \epsilon_{\perp}^{-1}(z) \rangle^{-1} \vec{E}_{\perp}(z)$$

## Detailed Formulation (1 of 7)

In Maxwell's equations, we have

$$\vec{D} = [\epsilon] \vec{E}$$

The dielectric tensor is

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}$$

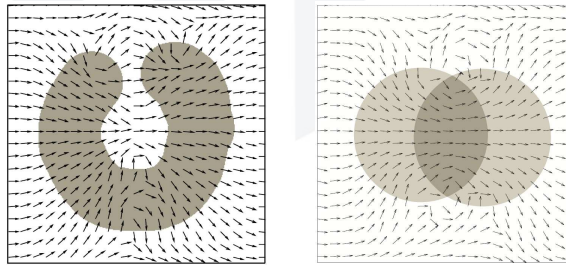
## Detailed Formulation (2 of 7)

The electric field can be written as the sum of parallel and perpendicular polarizations.

$$\vec{E} = \vec{E}_{\parallel} + \vec{E}_{\perp}$$

For an arbitrarily shaped device, these components can vary across the grid. Suppose we could calculate a vector function throughout the grid that is normal to all the interfaces. This called the “normal vector” field.

$$\hat{n}(x, y, z)$$



P. Gotz, T. Schuster, K. Frenner, S. Rafler, W. Osten, “Normal vector method for the RCWA with automated vector field generation,” Opt. Express 16(22), 17295-17301 (2008).

## Detailed Formulation (3 of 7)

The perpendicular component of  $\vec{E}$  can be computed from the normal vector field as follows.

$$\vec{E}_{\perp} = \hat{n}(\hat{n} \cdot \vec{E})$$

It follows that the parallel component of  $\vec{E}$  is

$$\begin{aligned} \vec{E}_{\parallel} &= \vec{E} - \vec{E}_{\perp} \\ &= \vec{E} - \hat{n}(\hat{n} \cdot \vec{E}) \end{aligned}$$

## Detailed Formulation (4 of 7)

In matrix notation, the perpendicular component can be written as

$$\vec{E}_\perp = \hat{n}(\hat{n} \cdot \vec{E}) \quad \rightarrow \quad \begin{bmatrix} E_x^\perp \\ E_y^\perp \\ E_z^\perp \end{bmatrix} = \begin{bmatrix} n_x(n_x E_x + n_y E_y + n_z E_z) \\ n_y(n_x E_x + n_y E_y + n_z E_z) \\ n_z(n_x E_x + n_y E_y + n_z E_z) \end{bmatrix}$$

$$= \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

## Detailed Formulation (5 of 7)

Let

$$[\hat{N}] = \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix}$$

The perpendicular and parallel components are then

$$\vec{E}_\perp = \hat{n}(\hat{n} \cdot \vec{E}) = [\hat{N}] \vec{E}$$

$$\vec{E}_\parallel = \vec{E} - [\hat{N}] \vec{E} = ([I] - [\hat{N}]) \vec{E}$$

## Detailed Formulation (6 of 7)

The constitutive relation can be written in terms of the parallel and perpendicular components of the  $\vec{E}$  field.

$$\begin{aligned}\vec{D} &= [\varepsilon] (\vec{E}_{\parallel} + \vec{E}_{\perp}) \\ &= [\varepsilon] \vec{E}_{\parallel} + [\varepsilon] \vec{E}_{\perp}\end{aligned}$$

Smooth the dielectric functions differently according to our rules for optimum convergence.

$$\begin{aligned}\vec{D} &= [\varepsilon_{\parallel}] \vec{E}_{\parallel} + [\varepsilon_{\perp}] \vec{E}_{\perp} & \llbracket \varepsilon_{\parallel} \rrbracket &= \langle [\varepsilon] \rangle \\ &\downarrow & \llbracket \varepsilon_{\perp} \rrbracket &= \langle [\varepsilon]^{-1} \rangle^{-1} \\ \vec{D} &= \llbracket \varepsilon_{\parallel} \rrbracket \vec{E}_{\parallel} + \llbracket \varepsilon_{\perp} \rrbracket \vec{E}_{\perp}\end{aligned}$$

## Detailed Formulation (7 of 7)

Putting all of this together leads to

$$\begin{aligned}\vec{D} &= \llbracket \varepsilon_{\parallel} \rrbracket \vec{E}_{\parallel} + \llbracket \varepsilon_{\perp} \rrbracket \vec{E}_{\perp} \\ &= \llbracket \varepsilon_{\parallel} \rrbracket ([I] - [\hat{N}]) \vec{E} + \llbracket \varepsilon_{\perp} \rrbracket [\hat{N}] \vec{E} \\ &= \left\{ \llbracket \varepsilon_{\parallel} \rrbracket ([I] - [\hat{N}]) + \llbracket \varepsilon_{\perp} \rrbracket [\hat{N}] \right\} \vec{E}\end{aligned}$$

We can derive an effective dielectric tensor from this equation.

$$\begin{aligned}[\varepsilon_{\text{smooth}}] &= \llbracket \varepsilon_{\parallel} \rrbracket ([I] - [\hat{N}]) + \llbracket \varepsilon_{\perp} \rrbracket [\hat{N}] \\ &= \llbracket \varepsilon_{\parallel} \rrbracket + (\llbracket \varepsilon_{\perp} \rrbracket - \llbracket \varepsilon_{\parallel} \rrbracket) [\hat{N}]\end{aligned}$$

## Summary of Formulation

Given the simulation problem defined by

$$\vec{D} = [\boldsymbol{\varepsilon}] \vec{E}$$

Convergence rate can be improved by smoothing the dielectric function according to

$$[\boldsymbol{\varepsilon}_{\text{smooth}}] = \llbracket \boldsymbol{\varepsilon}_{\parallel} \rrbracket + (\llbracket \boldsymbol{\varepsilon}_{\perp} \rrbracket - \llbracket \boldsymbol{\varepsilon}_{\parallel} \rrbracket) [\hat{N}]$$

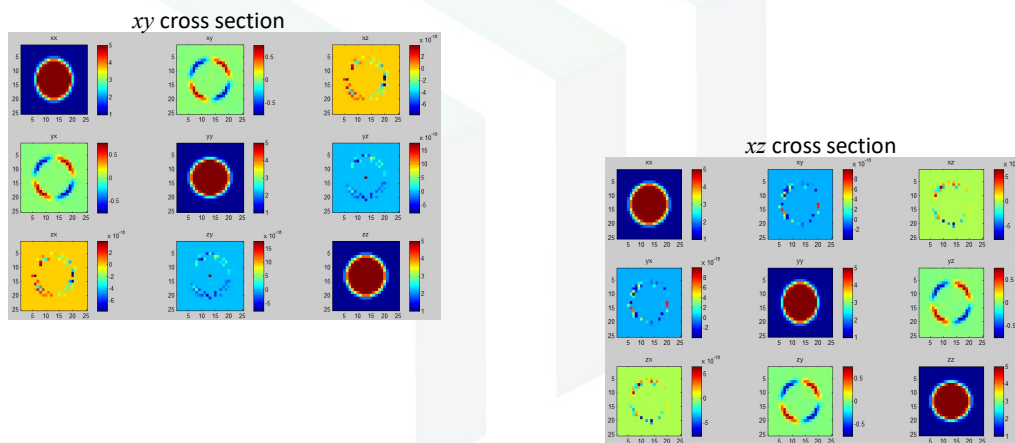
$$[\hat{N}] = \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix}$$

$$\llbracket \boldsymbol{\varepsilon}_{\parallel} \rrbracket = \langle [\boldsymbol{\varepsilon}] \rangle$$

$$\llbracket \boldsymbol{\varepsilon}_{\perp} \rrbracket = \langle [\boldsymbol{\varepsilon}]^{-1} \rangle^{-1}$$

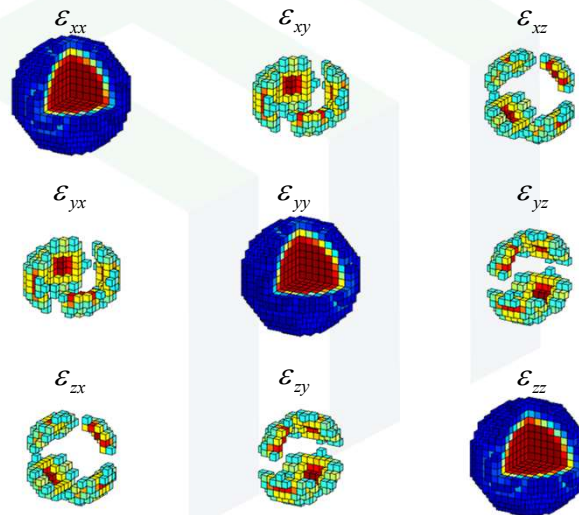
## Dielectric Averaging of a Sphere (1 of 2)

Given a sphere with dielectric constant  $\varepsilon_r=5.0$  in air and in a grid with  $N_x=N_y=N_z=25$  cells, the dielectric tensor after averaging is



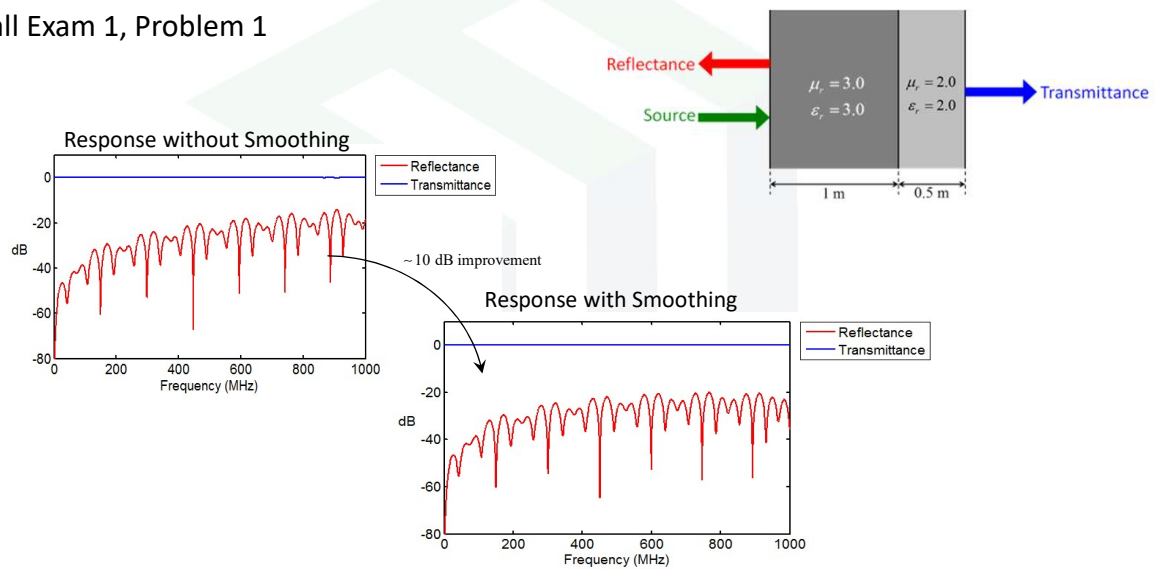
## Dielectric Averaging of a Sphere (2 of 2)

A 3D visualization is:



## Simulation Example

Recall Exam 1, Problem 1



## Comments on Dielectric Averaging

- Even if the original dielectric function is isotropic, the averaged dielectric function is anisotropic.
- Anisotropic averaging requires calculating the normal vector field. This can be difficult, especially for arbitrary structures.
- Convergence still tends to improve even when only isotropic averaging is used. 
$$[\epsilon_{\text{smooth}}] \cong \langle [\epsilon] \rangle$$
- The  $E_z$  mode does not require anisotropic averaging of the dielectric function.
- The  $H_z$  mode does not require anisotropic averaging of the permeability.