Computational Science:
Introduction to Finite-Difference Time-Domain

Implementation of One-Dimensional FDTD

Lecture Outline

- Review of Lecture 5
- Sequence of Code Development
- FDTD Implementation
  - Numerical boundary conditions
  - Grid resolution
  - Courant stability condition
  - Perfect 1D boundary condition
  - Sources
  - Total number of iterations
  - Revised FDTD Algorithm

Review of Lecture #5

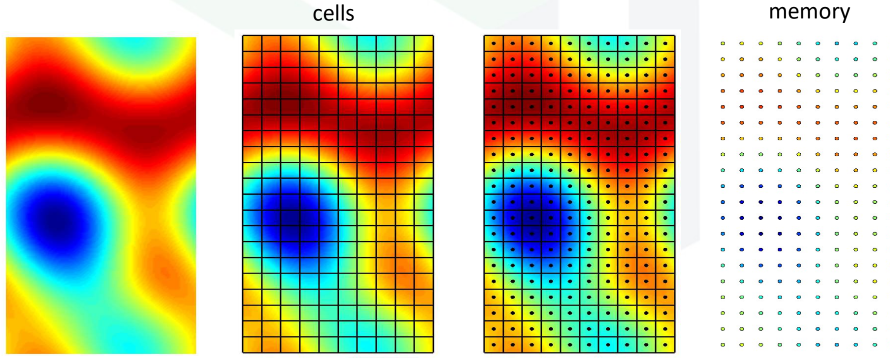
Representing Functions on a Grid

Example physical (continuous) 2D function

A grid is constructed by dividing space into discrete cells

Function is known only at discrete points

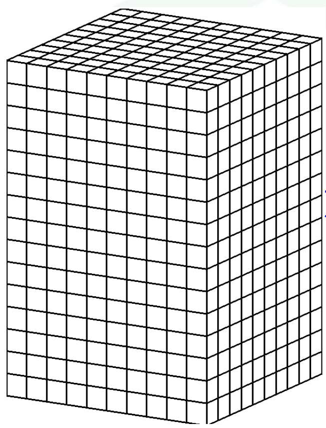
Representation of what is actually stored in memory



EMPossible

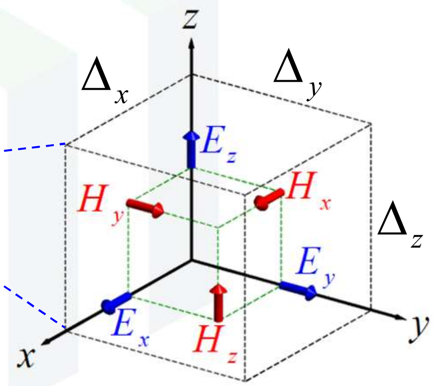
3D Grids

A three-dimensional grid looks like this:



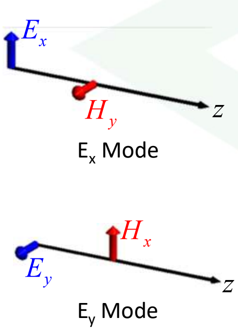
$N_x = 10, N_y = 10, N_z = 15$

A unit cell from the grid looks like this:

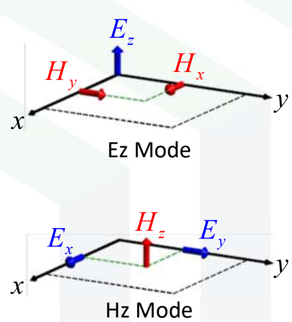


Yee Cell for 1D, 2D, and 3D Grids

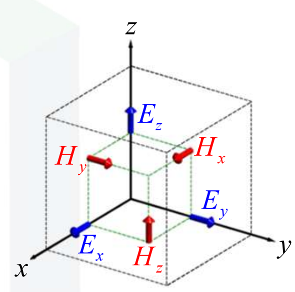
1D Yee Grid



2D Yee Grids



3D Yee Grid



<u>Benefits</u>

- Implicitly satisfies divergence equations
- Naturally handles physical boundary conditions
- Elegant approximation of the curl equations using finite-differences

<u>Consequences</u>

- Field components are in physically different locations
- Field components may reside in different materials even if they are in the same unit cell
- Field components will be out of phase

Formulation of Update Equations (1 of 4)

Normalize the magnetic field,

$$\vec{\tilde{H}} = \sqrt{\frac{\mu_0}{\varepsilon_0}}\vec{H} \quad \Rightarrow \quad \nabla \times \vec{E} = -\frac{[\mu_r]}{c_0}\frac{\partial \vec{\tilde{H}}}{\partial t} \qquad \nabla \times \vec{\tilde{H}} = \frac{[\varepsilon_r]}{c_0}\frac{\partial \vec{E}}{\partial t}$$

Assume linear, isotropic, and non-dispersive materials and expand the curl equations.

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{\mu_{xx}}{c_0}\frac{\partial \tilde{H}_x}{\partial t} \qquad \frac{\partial \tilde{H}_z}{\partial y} - \frac{\partial \tilde{H}_y}{\partial z} = \frac{\varepsilon_{xx}}{c_0}\frac{\partial E_x}{\partial t}$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{\mu_{yy}}{c_0}\frac{\partial \tilde{H}_y}{\partial t} \qquad \frac{\partial \tilde{H}_x}{\partial z} - \frac{\partial \tilde{H}_z}{\partial x} = \frac{\varepsilon_{yy}}{c_0}\frac{\partial E_y}{\partial t}$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{\mu_{zz}}{c_0}\frac{\partial \tilde{H}_z}{\partial t} \qquad \frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} = \frac{\varepsilon_{zz}}{c_0}\frac{\partial E_z}{\partial t}$$

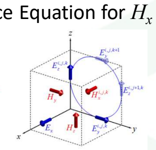
EMPossible

Slide 7

Formulation of Update Equations (2 of 4)

Finite-Difference Equation for H_x

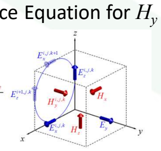
$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{\mu_{xx}}{c_0}\frac{\partial \tilde{H}_x}{\partial t}$$



$$\frac{E_z\big|_t^{i,j+1,k} - E_z\big|_t^{i,j,k}}{\Delta y} - \frac{E_y\big|_t^{i,j,k+1} - E_y\big|_t^{i,j,k}}{\Delta z} = -\frac{\mu_{xx}\big|^{i,j,k}}{c_0}\frac{\tilde{H}_x\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_x\big|_{t-\frac{\Delta t}{2}}^{i,j,k}}{\Delta t}$$

Finite-Difference Equation for H_y

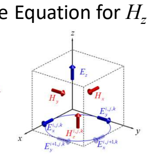
$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{\mu_{yy}}{c_0}\frac{\partial \tilde{H}_y}{\partial t}$$



$$\frac{E_x\big|_t^{i,j,k+1} - E_x\big|_t^{i,j,k}}{\Delta z} - \frac{E_z\big|_t^{i+1,j,k} - E_z\big|_t^{i,j,k}}{\Delta x} = -\frac{\mu_{yy}\big|^{i,j,k}}{c_0}\frac{\tilde{H}_y\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_y\big|_{t-\frac{\Delta t}{2}}^{i,j,k}}{\Delta t}$$

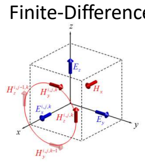
Finite-Difference Equation for H_z

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{\mu_{zz}}{c_0}\frac{\partial \tilde{H}_z}{\partial t}$$



$$\frac{E_y\big|_t^{i+1,j,k} - E_y\big|_t^{i,j,k}}{\Delta x} - \frac{E_x\big|_t^{i,j+1,k} - E_x\big|_t^{i,j,k}}{\Delta y} = -\frac{\mu_{zz}\big|^{i,j,k}}{c_0}\frac{\tilde{H}_z\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_z\big|_{t-\frac{\Delta t}{2}}^{i,j,k}}{\Delta t}$$

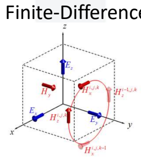
Finite-Difference Equation for E_x



$$\frac{\partial \tilde{H}_z}{\partial y} - \frac{\partial \tilde{H}_y}{\partial z} = \frac{\varepsilon_{xx}}{c_0}\frac{\partial E_x}{\partial t}$$

$$\frac{\tilde{H}_z\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_z\big|_{t+\frac{\Delta t}{2}}^{i,j-1,k}}{\Delta y} - \frac{\tilde{H}_y\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_y\big|_{t+\frac{\Delta t}{2}}^{i,j,k-1}}{\Delta z} = \frac{\varepsilon_{xx}\big|^{i,j,k}}{c_0}\frac{E_x\big|_{t+\Delta t}^{i,j,k} - E_x\big|_t^{i,j,k}}{\Delta t}$$

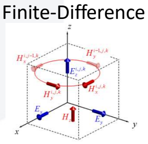
Finite-Difference Equation for E_y



$$\frac{\partial \tilde{H}_x}{\partial z} - \frac{\partial \tilde{H}_z}{\partial x} = \frac{\varepsilon_{yy}}{c_0}\frac{\partial E_y}{\partial t}$$

$$\frac{\tilde{H}_x\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_x\big|_{t+\frac{\Delta t}{2}}^{i,j,k-1}}{\Delta z} - \frac{\tilde{H}_z\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_z\big|_{t+\frac{\Delta t}{2}}^{i-1,j,k}}{\Delta x} = \frac{\varepsilon_{yy}\big|^{i,j,k}}{c_0}\frac{E_y\big|_{t+\Delta t}^{i,j,k} - E_y\big|_t^{i,j,k}}{\Delta t}$$

Finite-Difference Equation for E_z



$$\frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} = \frac{\varepsilon_{zz}}{c_0}\frac{\partial E_z}{\partial t}$$

$$\frac{\tilde{H}_y\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_y\big|_{t+\frac{\Delta t}{2}}^{i-1,j,k}}{\Delta x} - \frac{\tilde{H}_x\big|_{t+\frac{\Delta t}{2}}^{i,j,k} - \tilde{H}_x\big|_{t+\frac{\Delta t}{2}}^{i,j-1,k}}{\Delta y} = \frac{\varepsilon_{zz}\big|^{i,j,k}}{c_0}\frac{E_z\big|_{t+\Delta t}^{i,j,k} - E_z\big|_t^{i,j,k}}{\Delta t}$$

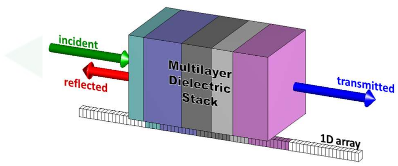
EMPossible

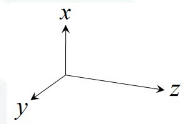
Slide 8

Formulation of Update Equations (3 of 4)

Let the problem be uniform in the x and y directions.

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

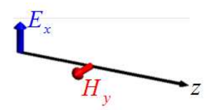




Maxwell's equations separates into two sets of equations.

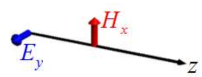
$$\frac{\partial E_x}{\partial z}=-\frac{\mu_{yy}}{c_0}\frac{\partial \tilde{H}_y}{\partial t} \qquad \frac{E_x^{k+1}\big|_t-E_x^k\big|_t}{\Delta z}=-\frac{\mu_{yy}^k}{c_0}\frac{\tilde{H}_y^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_y^k\big|_{t-\frac{\Delta t}{2}}}{\Delta t}$$

$$-\frac{\partial \tilde{H}_y}{\partial z}=\frac{\varepsilon_{xx}}{c_0}\frac{\partial E_x}{\partial t} \qquad -\frac{\tilde{H}_y^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_y^{k-1}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}=\frac{\varepsilon_{xx}^k}{c_0}\frac{E_x^k\big|_{t+\Delta t}-E_x^k\big|_t}{\Delta t}$$



$$\frac{\partial E_y}{\partial z}=\frac{\mu_{xx}}{c_0}\frac{\partial \tilde{H}_x}{\partial t} \qquad \frac{\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_x^{k-1}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}=\frac{\varepsilon_{yy}^k}{c_0}\frac{E_y^k\big|_{t+\Delta t}-E_y^k\big|_t}{\Delta t}$$

$$\frac{\partial \tilde{H}_x}{\partial z}=\frac{\varepsilon_{yy}}{c_0}\frac{\partial E_y}{\partial t} \qquad \frac{E_y^{k+1}\big|_t-E_y^k\big|_t}{\Delta z}=\frac{\mu_{xx}^k}{c_0}\frac{\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_x^k\big|_{t-\frac{\Delta t}{2}}}{\Delta t}$$



EMPossible

Slide 9

Formulation of Update Equations (4 of 4)

Derive update equations by solving the finite-difference equations for the future time value.

$$\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}=\tilde{H}_x^k\big|_{t-\frac{\Delta t}{2}}+m_{Hx}^k\left(\frac{E_y^{k+1}\big|_t-E_y^k\big|_t}{\Delta z}\right) \qquad m_{Ey}^k=\frac{c_0\Delta t}{\varepsilon_{yy}^k}$$

$$E_y^k\big|_{t+\Delta t}=E_y^k\big|_t+m_{Ey}^k\left(\frac{\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_x^{k-1}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}\right) \qquad m_{Hx}^k=\frac{c_0\Delta t}{\mu_{xx}^k}$$

Includes:
- Basic FDTD engine

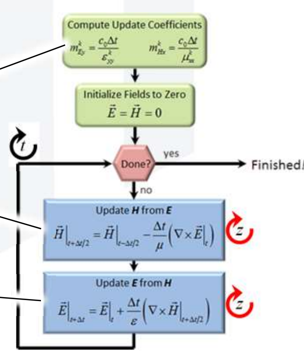
Excludes:
- Dirichlet BC's
- Calculate source parameters
- Simple soft source
- Perfectly absorbing BC's
- TF/SF source
- Fourier transforms
- Reflectance/Transmittance
- Calculate grid parameters
- Incorporate device

Arrive at the following FDTD algorithm.

$$m_{Ey}^k=\frac{c_0\Delta t}{\varepsilon_{yy}^k} \qquad m_{Hx}^k=\frac{c_0\Delta t}{\mu_{xx}^k}$$

Loop over z $\quad \tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}=\tilde{H}_x^k\big|_{t-\frac{\Delta t}{2}}+m_{Hx}^k\left(\frac{E_y^{k+1}\big|_t-E_y^k\big|_t}{\Delta z}\right)$

Loop over z $\quad E_y^k\big|_{t+\Delta t}=E_y^k\big|_t+m_{Ey}^k\left(\frac{\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}}-\tilde{H}_x^{k-1}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}\right)$

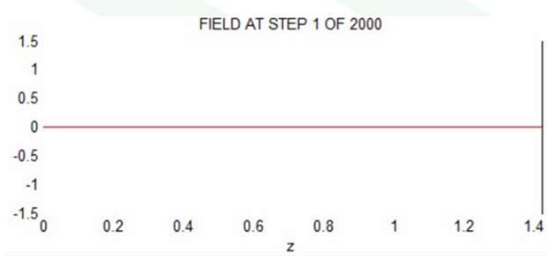


EMPossible

Slide 10

Sequence of Code Development

Step 1 – Basic FDTD Algorithm

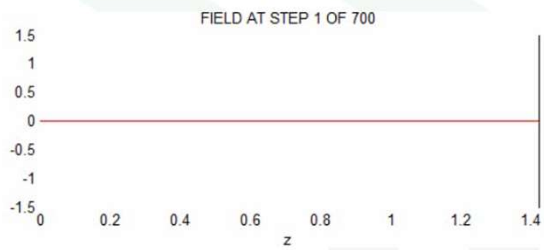


- Basic update equations

EMPossible

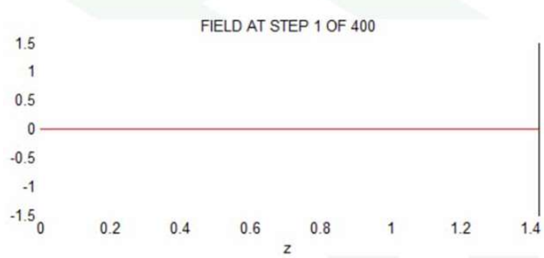
Step 2 – Add Simple Soft Source

- Basic update equations
- Add a soft source

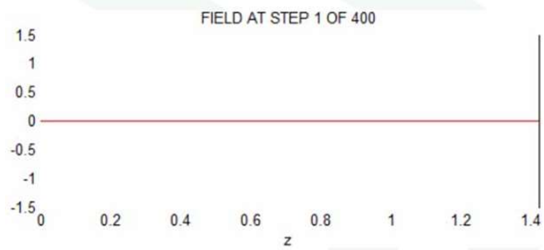


Step 3 – Add Absorbing Boundary

- Basic update equations
- Add a soft source
- Add perfect boundary condition



Step 4 – Add TF/SF

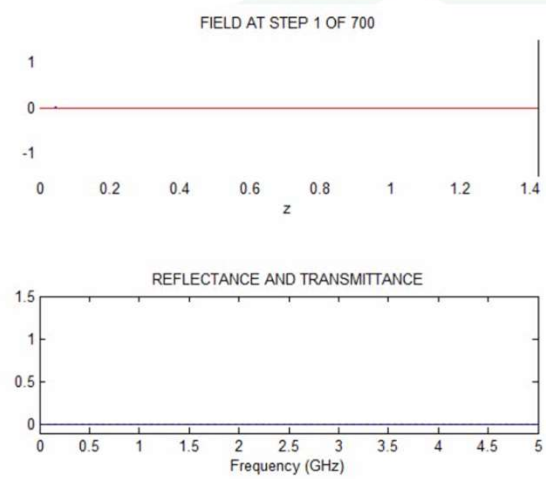


- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source

EMPossible

Slide 15

Step 5 – Move Source & Add T/R

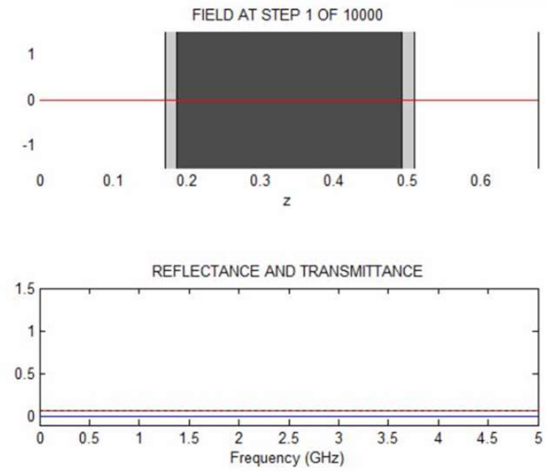


- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source
- Move position of source
- Calculate transmittance and reflectance

EMPossible

Slide 16

Step 6 – Add Device (Complete Algorithm)



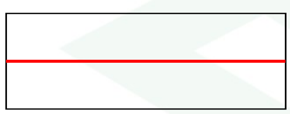
- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source
- Move position of source
- Calculate transmittance and reflectance
- Add a real device

EMPossible

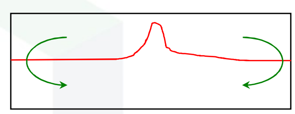
Slide 17

Summary of Code Development Sequence

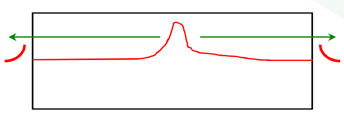
Step 1 – Implement basic FDTD algorithm



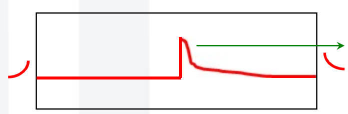
Step 2 – Add the source



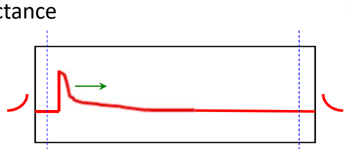
Step 3 – Add absorbing boundary



Step 4 – Add "one-way" source



Step 5 – Calculate transmittance and reflectance



Step 6 – Add a device

EMPossible

Slide 18

Numerical Boundary Conditions

A Problem at the Boundary of the Grid

The update equation must be implemented for every point in the grid.

$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k \left(\frac{E_y^{k+1}\Big|_t - E_y^k\Big|_t}{\Delta z} \right)$$

What do we do at $k = N_z$?

$E_y^{N_z+1}$ does not exist.

$$E_y^k\Big|_{t+\Delta t} = E_y^k\Big|_t + m_{Ey}^k \left(\frac{\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z} \right)$$

What do we do at $k = 1$?

$\tilde{H}_x^0$ does not exist.

EMPossible

Dirichlet Boundary Condition

One easy thing to do is assume the fields outside the grid are zero. This is called a *Dirichlet boundary condition*.

To incorporate the Dirichlet boundary condition, modify the update equations as follows.

$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \begin{cases} \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k\left(\dfrac{E_y^{k+1}\Big|_t - E_y^k\Big|_t}{\Delta z}\right) & k < N_z \\ \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^{N_z}\left(\dfrac{0 - E_y^{N_z}\Big|_t}{\Delta z}\right) & k = N_z \end{cases}$$

$$E_y^k\Big|_{t+\Delta t} = \begin{cases} E_y^1\Big|_t + m_{Ey}^1\left(\dfrac{\tilde{H}_x^1\Big|_{t+\frac{\Delta t}{2}} - 0}{\Delta z}\right) & k = 1 \\ E_y^k\Big|_t + m_{Ey}^k\left(\dfrac{\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z}\right) & k > 1 \end{cases}$$

Equations → MATLAB Code

DO NOT use ‘if’ statements to implement boundary conditions.

Update Equations

$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k\left(E_y^{k+1}\Big|_t - E_y^k\Big|_t\right) \qquad k < N_z$$

$$\tilde{H}_x^{N_z}\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^{N_z}\left(0 - E_y^{N_z}\Big|_t\right) \qquad k = N_z$$

$$E_y^1\Big|_{t+\Delta t} = E_y^1\Big|_t + m_{Ey}^1\left(\tilde{H}_x^1\Big|_{t+\frac{\Delta t}{2}} - 0\right) \qquad k = 1$$

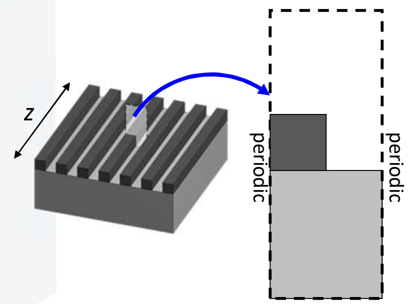
$$E_y^k\Big|_{t+\Delta t} = E_y^k\Big|_t + m_{Ey}^k\left(\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\Big|_{t+\frac{\Delta t}{2}}\right) \qquad k > 1$$

```
%
% MAIN FDTD LOOP
%
for T = 1 : STEPS

    % Update H from E (Dirichlet Boundary Conditions)
    for nz = 1 : Nz-1
        Hx(nz) = Hx(nz) + mHx(nz)*(Ey(nz+1) - Ey(nz))/dz;
    end
    Hx(Nz) = Hx(Nz) + mHx(Nz)*(0 - Ey(Nz))/dz;

    % Update E from H (Dirichlet Boundary Conditions)
    Ey(1) = Ey(1) + mEy(1)*(Hx(1) - 0)/dz;
    for nz = 2 : Nz
        Ey(nz) = Ey(nz) + mEy(nz)*(Hx(nz) - Hx(nz-1))/dz;
    end

end
```

Periodic Boundary Condition

Some devices are periodic along a particular direction. When this is the case, the field is also periodic.

To incorporate a periodic boundary condition, modify the update equations as follows.



$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \begin{cases} \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k\left(\dfrac{E_y^{k+1}\big|_t - E_y^k\big|_t}{\Delta z}\right) & k < N_z \\ \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^{N_z}\left(\dfrac{E_y^{1}\big|_t - E_y^{N_z}\big|_t}{\Delta z}\right) & k = N_z \end{cases}$$

$$E_y^k\Big|_{t+\Delta t} = \begin{cases} E_y^1\Big|_t + m_{Ey}^1\left(\dfrac{\tilde{H}_x^1\big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{N_z}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}\right) & k = 1 \\ E_y^k\Big|_t + m_{Ey}^k\left(\dfrac{\tilde{H}_x^k\big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\big|_{t+\frac{\Delta t}{2}}}{\Delta z}\right) & k > 1 \end{cases}$$

EMPossible

Slide 23

Grid Resolution

Slide 24

Consideration #1: Wavelength

The grid resolution must be sufficient to resolve the shortest wavelength.

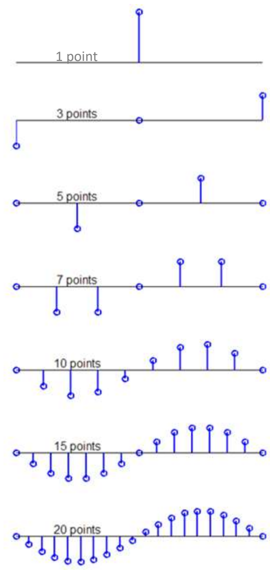
First, determine the smallest wavelength:

$$\lambda_{\min} = \frac{c_0}{f_{\max} n_{\max}}$$

$n_{\max}$ is the largest refractive index found anywhere in the grid. $f_{\max}$ is the maximum frequency in your simulation.

Second, resolve the shortest wavelength with at least 10 cells.

$$\Delta_\lambda \approx \frac{\lambda_{\min}}{N_\lambda} \qquad N_\lambda \geq 10$$



N_λ	Comments
10 to 20	Low contrast dielectrics
20 to 30	High contrast dielectrics
40 to 60	Most metallic structures
100 to 200	Plasmonic devices

EMPossible

Slide 25

Consideration #2: Mechanical Features

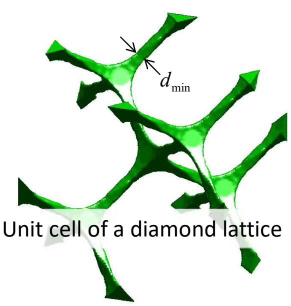
The grid resolution must be sufficient to resolve the smallest mechanical features of the device.

First, determine the smallest feature:

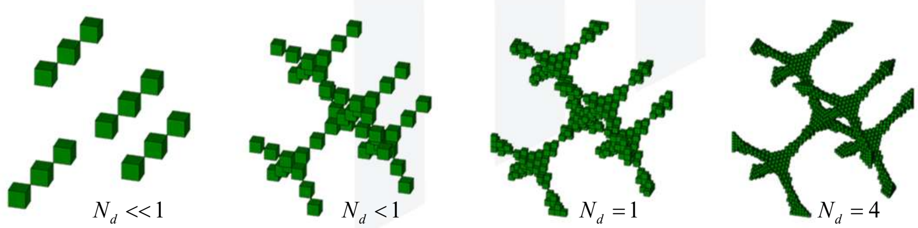
$$d_{\min}$$

Second, resolve the smallest feature with at least 1 to 4 cells.

$$\Delta_d \approx \frac{d_{\min}}{N_d} \qquad N_d \geq 1$$



Unit cell of a diamond lattice



EMPossible

Slide 26

Calculating the Initial Grid Resolution

Must resolve the minimum wavelength

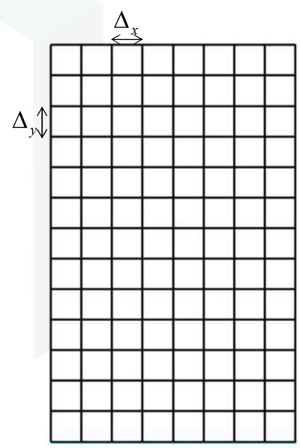
$$\Delta_\lambda = \frac{\lambda_{\min}}{N_\lambda} \qquad N_\lambda \geq 10$$

Must resolve the minimum structural dimension

$$\Delta_d = \frac{d_{\min}}{N_d} \qquad N_d \geq 1$$

Set the initial grid resolution to the smallest number computed above

$$\Delta_x = \Delta_y = \min\left[\Delta_\lambda, \Delta_d\right]$$



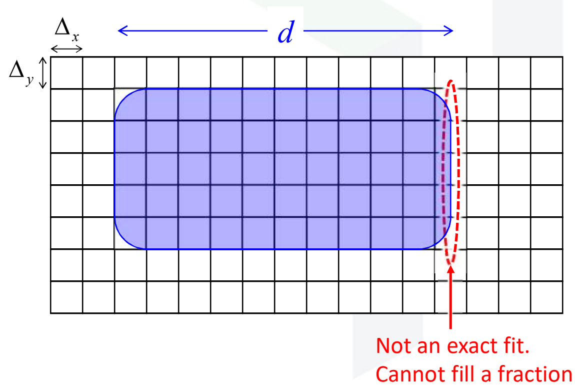
EMPossible Slide 27

Resolving Critical Dimensions (1 of 3)

We have not yet considered the actual dimensions of the device we wish to simulate.

This means we likely cannot resolve the exact dimensions of a device.

Suppose we wish to place a device of length d onto a grid.

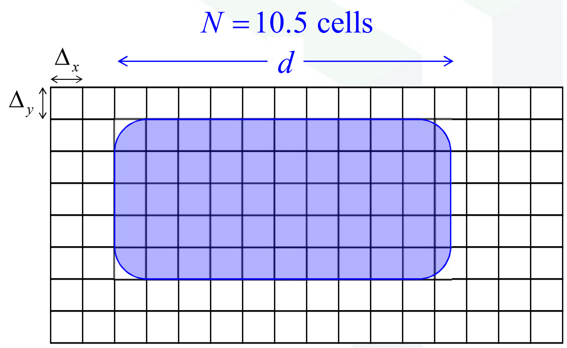


EMPossible Slide 28

Resolving Critical Dimensions (2 of 3)

To fix this, first calculate how many cells N are needed to resolve the most important dimension. In this case, let this be d.

$$N = \frac{d}{\Delta_x}$$



Second, we determine how many cells we wish to exactly resolve d. We do this by rounding N up to the nearest integer.

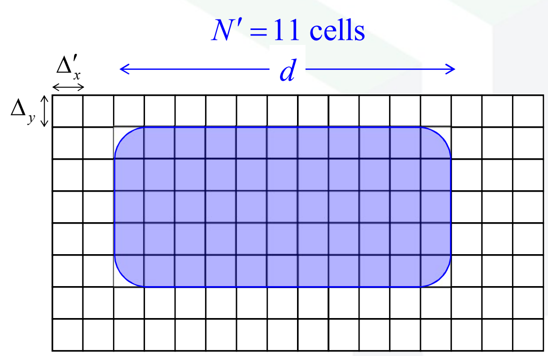
$$N' = \text{ceil}\left[\frac{d}{\Delta_x}\right]$$

$N' = 11$ cells

Resolving Critical Dimensions (3 of 3)

Third, adjust the value of Δ_x to represent the dimension d exactly.

$$\Delta'_x = \frac{d}{N'}$$



Call this step "snapping" the grid to a critical dimension.

Unfortunately, using a uniform grid, this can only be done for one critical dimension per axis.

"Snap" Grid to Critical Dimensions

Decide what dimensions along each axis are critical

→ Typically this is a lattice constant or grating period along x
→ Typically this is a layer thickness along y

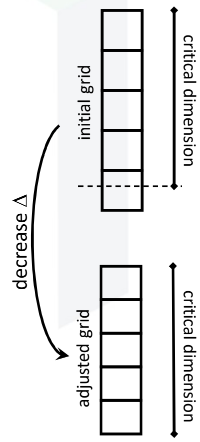
$$d_x \quad \text{and} \quad d_y$$

Compute how many grid cells comprise d_x and d_y and round UP

$$M_x = \text{ceil}\left(d_x/\Delta_x\right)$$
$$M_y = \text{ceil}\left(d_y/\Delta_y\right)$$

Adjust grid resolution to fit these dimensions in grid EXACTLY

$$\Delta_x = d_x/M_x$$
$$\Delta_y = d_y/M_y$$



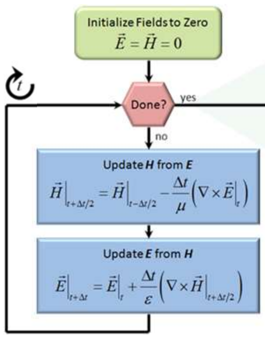
EMPossible

Slide 31

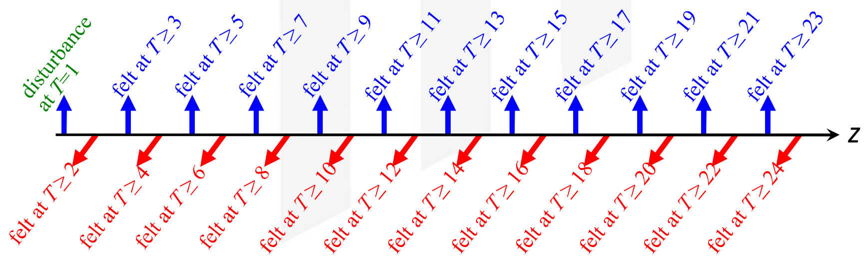
Courant Stability Condition

Slide 32

Numerical Propagation Through Grid



During a single iteration, a disturbance in the electric field at one point can only be felt by the immediately adjacent magnetic fields. It takes at least two time steps before that disturbance is felt by an adjacent electric field. This is simply due to how the update equations are implemented during a single iteration.



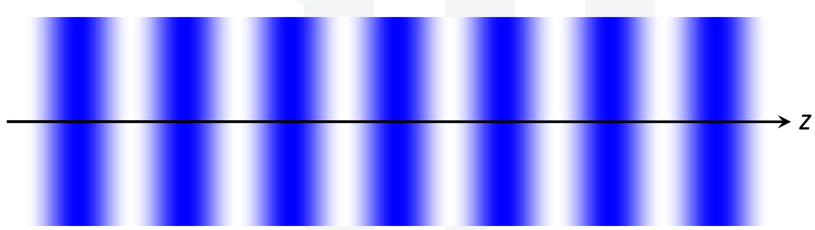
EMPossible

Slide 33

Physical Propagation Through Space

Electromagnetic waves in vacuum propagate at the speed of light.
Inside a material, they propagate at a reduced speed.

$$v = \frac{c_0}{n} \qquad c_0 = 299792458 \text{ m/s} \qquad n \equiv \text{refractive index}$$



EMPossible

Slide 34

A Limit on Δt

Over a time duration of one time step Δt, an electromagnetic disturbance will travel:

Numerical distance covered in one time step: Δz

Physical distance covered in one time step: $c_0 \Delta t / n$

Because of the numerical algorithm, it is not possible for a disturbance to travel farther than one unit cell in a single time step.

We need to make sure that a physical wave would not propagate farther than a single unit cell in one time step.

$$\frac{c_0 \Delta t}{n} < \Delta z$$

This places an upper limit on the time step.

$$\Delta t < \frac{n \Delta z}{c_0}$$

n should be set to the smallest refractive index found anywhere in the grid. Usually this is just made to be 1.0 and dropped from the equation.

EMPossible Slide 35

The Courant Stability Condition

Refractive index is greater than or equal to one, so our condition on Δt can be written more simply as:

$$\Delta t < \frac{\Delta z}{c_0}$$

Sort of a worst case.
n=1 produces the fastest possible physical wave.

For 2D or 3D grids, the condition can be generalized as

The Courant stability condition

$$\Delta t < \frac{1}{c_0 \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}}$$

EMPossible Slide 36

Practical Implementation of the Stability Condition

The stability condition will be most restrictive along the shortest dimension of the grid unit cell.

$$\Delta_{\text{min}} = \min\left[\Delta x, \Delta y, \Delta z\right]$$

A good equation to ensure stability and accuracy on any grid is then

$$\Delta t < \frac{\Delta_{\text{min}}}{2c_0}$$

Note: A factor of 0.5 was included here as a safety margin.

This can be generalized to account for special cases.

$$\Delta t < \frac{n_{\text{min}}\Delta_{\text{min}}}{2c_0}$$

1. Your grid is filled with dielectric and travels slower everywhere.
2. Your model includes dispersive materials with refractive index less than one.

EMPossible

Slide 37

Time Step for Our 1D Grid

$$\Delta t = \frac{n_{\text{bc}}\Delta z}{2c_0}$$

n_{bc} = refractive index at the grid boundaries.

This means a wave will travel the distance of one grid cell in exactly two time steps.

It is a necessary condition for the perfect boundary condition we will soon implement.

This implies that we cannot have different materials at the two boundaries using this boundary condition.

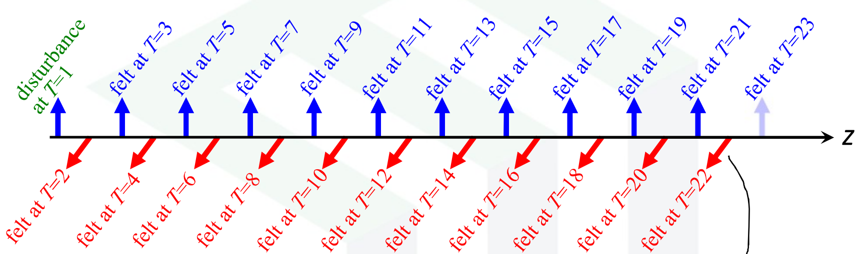
EMPossible

Slide 38

Perfect 1D Boundary Condition

Slide 39

The Problem



The finite-difference equation here requires knowledge of the electric field outside of the grid.

$$\tilde{H}_x^{N_z}\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k \left(\frac{E_y^{N_z+1}\Big|_t - E_y^{N_z}\Big|_t}{\Delta z} \right)$$

Slide 40

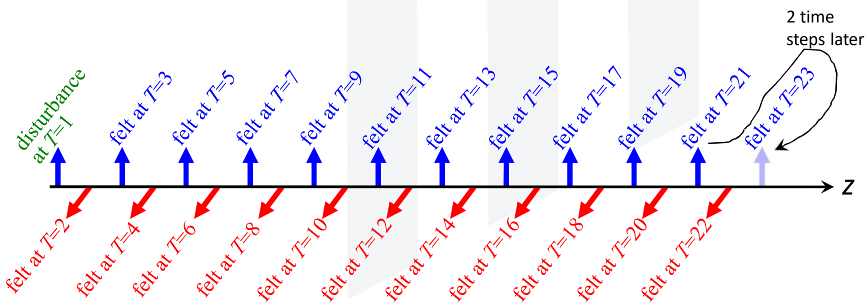
Implementing the Perfect Boundary Condition

If and only if…

- the field is only travelling outward at the boundaries,
- the materials at the boundaries are linear, homogeneous, isotropic and non-dispersive,
- The refractive index at both boundaries is n_{bc},
- $\Delta t = n_{\mathrm{bc}} \Delta z / (2c_0)$ exactly.

Then…

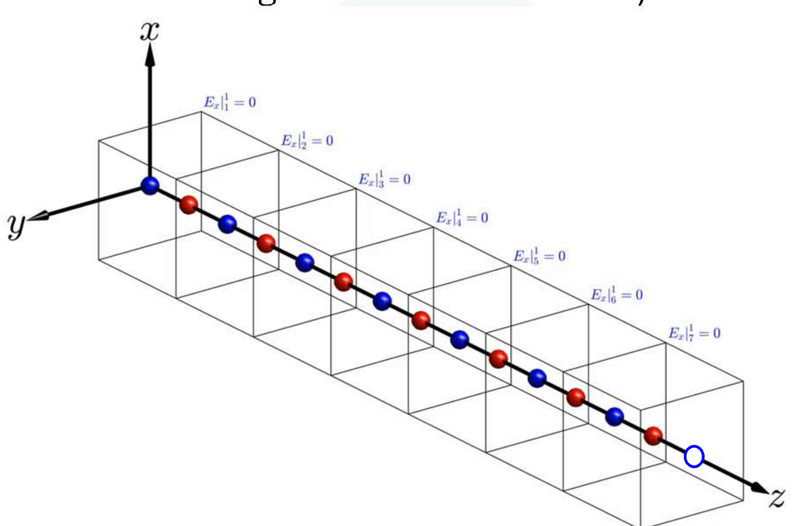
$$E_y^{N_z+1}\Big|_t = E_y^{N_z}\Big|_{t-2\Delta t}$$



EMPossible

Slide 41

Visualizing the Perfect Boundary Condition



$$E_x\Big|_8^n = E_x\Big|_7^{n-2}$$

EMPossible

Slide 42

Summary of the 1D Perfect Boundary Condition

Conditions

- Waves at the boundaries are only travelling outward,
- Materials at the boundaries are linear, homogeneous, isotropic and non-dispersive,
- The refractive index is the same at both boundaries and is n_{bc},
- Time step is chosen so physical waves travel 1 cell in exactly two time steps $\Delta t = n_{bc}\Delta z/(2c_0)$.

Implementation at z-Low Boundary

At the z-low boundary, we need only modify the E-field update equation.

$$h_2 = h_1 \qquad h_1 = \tilde{H}_x^1 \qquad E_y^1\Big|_{t+\Delta t} = E_y^1\Big|_t + m_{Ey}^k\left(\frac{\tilde{H}_x^1\Big|_{t+\frac{\Delta t}{2}} - h_2}{\Delta z}\right)$$

Implementation at z-High Boundary

At the z-high boundary, we need only modify the H-field update equation.

$$e_2 = e_1 \qquad e_1 = E_y^{N_z} \qquad \tilde{H}_x^{N_z}\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k\left(\frac{e_2 - E_y^{N_z}\Big|_t}{\Delta z}\right)$$

EMPossible

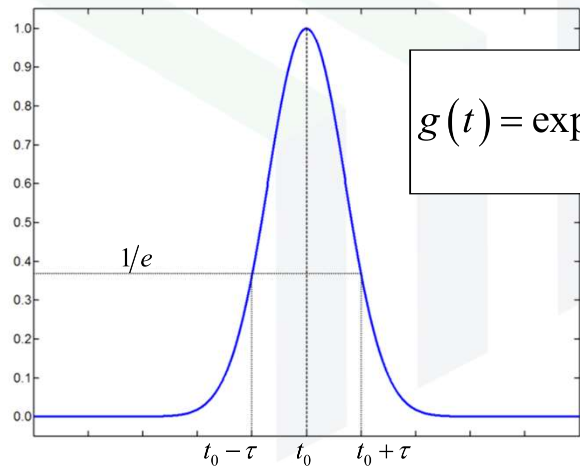
Slide 43

Sources

Slide 44

The Gaussian Pulse

Typically short pulses are used as sources in FDTD. This approximates an impulse function so that a broad range of frequencies can be simulated at the same time in the same simulation.



$$g(t)=\exp\left[-\left(\frac{t-t_0}{\tau}\right)^2\right]$$

EMPossible

Slide 45

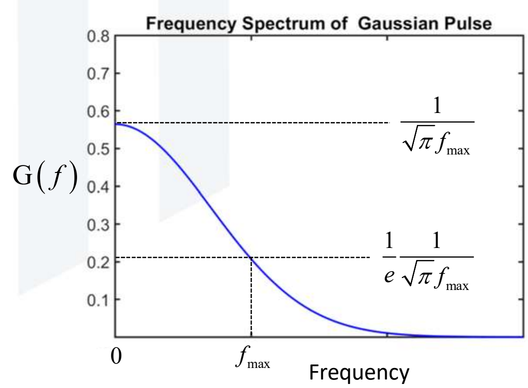
Frequency Content of Gaussian Pulse

The Fourier transform of a Gaussian pulse is another Gaussian function.

$$g(t)=\exp\left(-\frac{t^2}{\tau^2}\right) \quad \Leftrightarrow \quad \mathrm{G}(f)=\frac{1}{\sqrt{\pi}f_{\max}}\exp\left[-\frac{f^2}{f_{\max}^2}\right]$$

The frequency content of the Gaussian pulse extends from DC up to above $f_{\max}$. The frequency $f_{\max}$ is actually the $1/e$ point of the frequency spectrum.

$$f_{\max}=\frac{1}{\pi\tau}$$



EMPossible

Slide 46

Designing the Pulse Source (1 of 2)

Step 1: Determine the maximum frequency of interest in your simulation.

$$f_{\max}$$

Step 2: Compute the pulse duration to have sufficient energy at $f_{\max}$.

$$f_{\max} = \frac{1}{\pi\tau} \quad \rightarrow \quad \tau \leq \frac{1}{\pi f_{\max}} \qquad\qquad \boxed{\tau \cong \frac{0.5}{f_{\max}}}$$

Step 3: You may need to reduce your time step. Your Gaussian pulse should be resolved by at least 10 to 20 time steps.

$$\Delta t = \frac{\tau}{N_t}$$

$$N_t \geq 10$$

Typically, you determine a first Δt based on the Courant stability condition, then determine a second Δt based on resolving the maximum frequency, and finally go with the smallest value of the two Δt's.

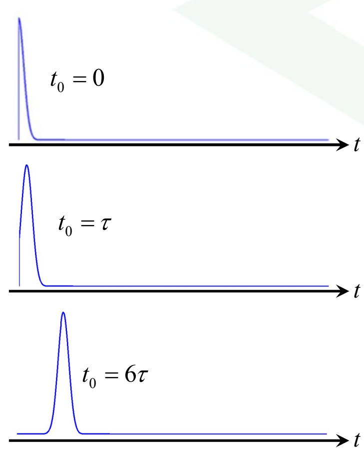
All of this should be satisfied automatically if $\Delta t = n\Delta z/(2c_0)$.

Designing the Pulse Source (2 of 2)

Step 4: Compute the delay time t_0

The pulse source must start at zero and gradually increase. NO STEP FUNCTIONS!!

$$t_0 \geq 3\tau$$

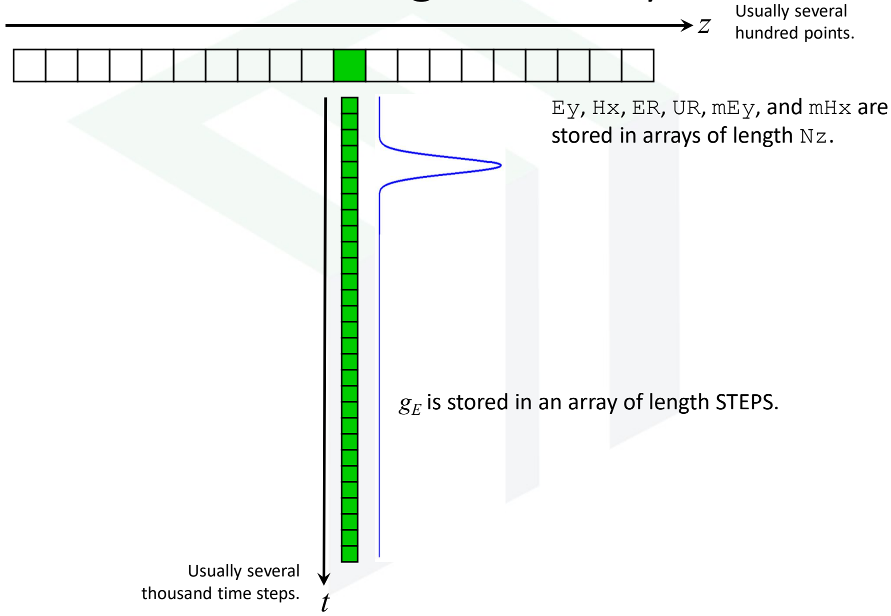


WRONG!!
The step function at the beginning will induce very large field gradients.

STILL WRONG!!
While better, this source still starts with a step function that will produce large field gradients.

CORRECT!!
This source "eases" into the Gaussian source.

Visualizing the Arrays



EMPossible

49

Two Ways to Incorporate a Simple Source

Source #1: Simple Hard Source

The simple hard source is the easiest to implement. After updating the field across the entire grid, one field component at one point on the grid is overwritten with the source. This approach injects power into the model, but the source point behaves like a perfect electric conductor or perfect magnetic conductor and will scatter waves.

$$\left.\tilde{H}_x^k\right|_{t+\frac{\Delta t}{2}} = \left.g_H\right|_k \quad \text{and/or} \quad \left.E_y^k\right|_{t+\Delta t} = \left.g_E\right|_k$$

OVERWRITE
Not usually a practical source.
We won't use it in this course.

Source #2: Simple Soft Source

The simple soft source is better than the hard source because it is transparent to scattered waves passing through it. After updating the field across the entire grid, the source function is added to one field component at one point on the grid. This approach injects power into the model in both directions. It is great for testing boundary conditions.

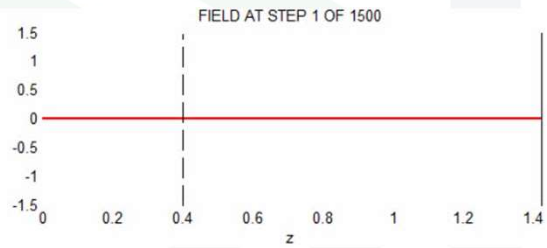
$$\left.\tilde{H}_x^k\right|_{t+\frac{\Delta t}{2}} = \left.\tilde{H}_x^k\right|_{t+\frac{\Delta t}{2}} + \left.g_H\right|_k \quad \text{and/or} \quad \left.E_y^k\right|_{t+\Delta t} = \left.E_y^k\right|_{t+\Delta t} + \left.g_E\right|_k$$

ADD TO
Rarely used.
Use this until we learn TF/SF.

EMPossible

Slide 50

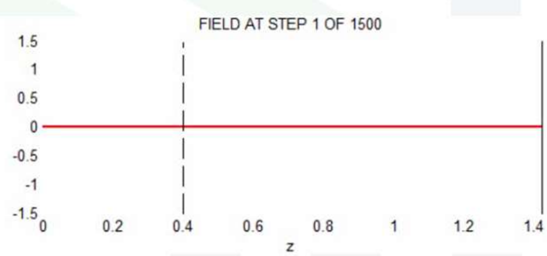
Movie of Source #1 – Simple Hard Source



EMPossible

51

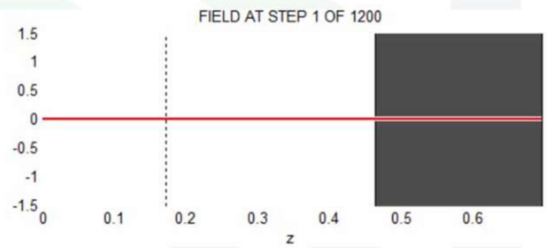
Movie of Source #2 – Simple Soft Source



EMPossible

52

Movie of Source #3 – TF/SF Soft Source



EMPossible

53

Total Number of Iterations

Slide 54

Considerations for Estimating the Total Number of Iterations

The total number of iterations depends heavily on the device being modeled and what properties of it are being calculated.

Device Considerations
1. Highly resonant devices typically require more iterations.
2. Purely scattering devices typically require very few iterations.
3. More iterations are needed the more times the waves bounce around in the grid.

Information Considerations
1. Calculating abrupt spectral shapes requires many iterations.
2. Calculating fine frequency resolution requires many iterations.
3. Calculating only the approximate position of resonances often requires fewer iterations. Great for hunting resonances!

A Rule of Thumb

How long does it take a wave to propagate across the grid (worst case)?

$$\text{time} = \frac{\text{distance}}{\text{velocity}} \quad \rightarrow \quad t_{\text{prop}} = \frac{n_{\max} N_z \Delta z}{c_0}$$

Simulation time T must include the entire pulse of duration τ.

$$T \geq 12\tau$$

Simulation time should allow for ~5 bounces.

$$T \geq 5t_{\text{prop}}$$

A rule-of-thumb for total simulation time is then

$$T = 12\tau + 5t_{\text{prop}}$$

Note: For highly resonant devices, this will NOT be enough time!

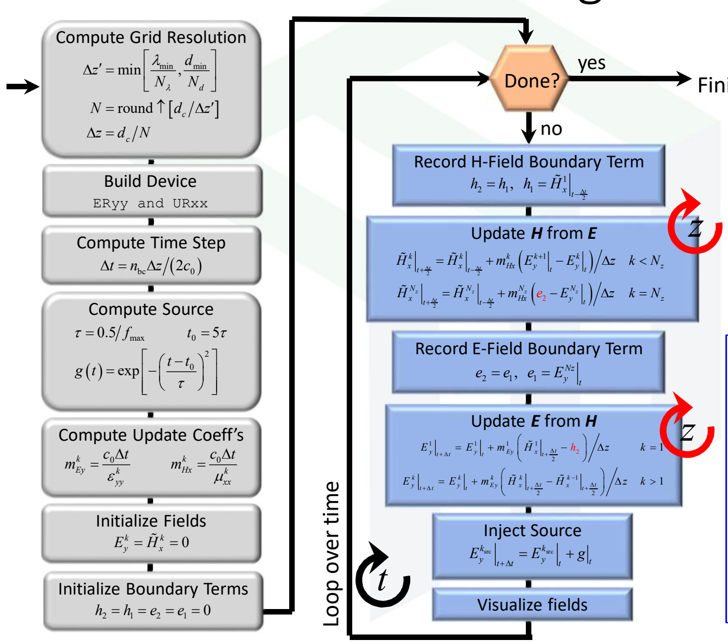
Given the time step Δt, the total number of iterations is then

$$\text{STEPS} = \text{round}\uparrow\left[\frac{T}{\Delta t}\right]$$

This must be an integer quantity.

Revised FDTD Algorithm

Revised FDTD Algorithm



Includes:
- Basic FDTD engine
- Dirichlet BC's
- Calculate source parameters
- Simple soft source
- Perfectly absorbing BC's

Excludes:
- TF/SF source
- Fourier transforms
- Reflectance/transmittance
- Calculate grid parameters
- Incorporate device

EMPossible

Equations → MATLAB Code

Grid Resolution

$$\Delta z' = \min\left[\frac{\lambda_{\min}}{N_\lambda}, \frac{d_{\min}}{N_d}\right]$$

$$N = \text{round}\uparrow\left[d_c/\Delta z'\right]$$
$$\Delta z = d_c/N$$

```
% COMPUTE DEFAULT GRID RESOLUTION
dz1 = min(LAMBDA)/nmax/NRES;
dz2 = dmin/NDRES;
dz  = min(dz1,dz2);

% SNAP GRID TO CRITIAL DIMENSIONS
N  = ceil(dc/dz);
dz = dc/N;
```

Update Equations

$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^k\left(E_y^{k+1}\Big|_t - E_y^k\Big|_t\right) \qquad k < N_z$$

$$\tilde{H}_x^{N_z}\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^{N_z}\Big|_{t-\frac{\Delta t}{2}} + m_{Hx}^{N_z}\left(E_y^{N_z}\Big|_{t-2\Delta t} - E_y^{N_z}\Big|_t\right) \qquad k = N_z$$

$$E_y^k\Big|_{t+\Delta t} = E_y^k\Big|_t + m_{Ey}^k\left(\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\Big|_{t+\frac{\Delta t}{2}}\right) \qquad k > 1$$

$$E_y^1\Big|_{t+\Delta t} = E_y^1\Big|_t + m_{Ey}^1\left(\tilde{H}_x^1\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^1\Big|_{t+\frac{\Delta t}{2}-2\Delta t}\right) \qquad k = 1$$

$$E_y^{k_{\text{src}}}\Big|_{t+\Delta t} = E_y^{k_{\text{src}}}\Big|_t + g\Big|_t$$

```
%
% MAIN FDTD LOOP
%
for T = 1 : STEPS

    % Update H from E (Perfect Boundary Conditions)
    H2=H1; H1=Hx(1);
    for nz = 1 : Nz-1
        Hx(nz) = Hx(nz) + mHx(nz)*(Ey(nz+1) - Ey(nz))/dz;
    end
    Hx(Nz) = Hx(Nz) + mHx(Nz)*(E2 - Ey(Nz))/dz;

    % Update E from H (Perfect Boundary Conditions)
    E2=E1; E1=Ey(Nz);
    Ey(1) = Ey(1) + mEy(1)*(Hx(1) - H2)/dz;
    for nz = 2 : Nz
        Ey(nz) = Ey(nz) + mEy(nz)*(Hx(nz) - Hx(nz-1))/dz;
    end

    % Inject Soft Source
    Ey(nzsrc) = Ey(nzsrc) + g(T);
end
```

EMPossible