Lecture #11

Formulation of 2D-FDTD Without a PML

Lecture Outline

• Code development sequence for 2D
• Maxwell’s equations
• Finite-difference approximations
• Reduction to two dimensions
• Update equations
• Boundary Conditions
• Revised FDTD Algorithm for 2D Simulations
Code Development Sequence

Step 1 – Basic Update Equations

The basic update equations are implemented along with simple Dirichlet boundary conditions.
Step 2 – Incorporate Periodic Boundaries

Periodic boundary conditions are incorporated so that a wave leaving the grid reenters the grid at the other side.

Step 3 – Incorporate a PML

The perfectly matched layer (PML) absorbing boundary condition is incorporated to absorb outgoing waves.
Step 4 – Total-Field/Scattered-Field

Most periodic electromagnetic devices are modeled by using periodic boundaries for the horizontal axis and a PML for the vertical axis. We then implement TF/SF at the vertical center of the grid for testing.

Step 5 – Calculate TRN, REF, and CON

We move the TF/SF interface to a unit cell or two outside of the top PML. We include code to calculate Fourier transforms and to calculate transmittance, reflectance, and conservation of power.
Step 6 – Model a Device to Benchmark

We build a device on the grid that has a known solution. We run the simulation and duplicate the known results to benchmark our new code.

Summary of Code Development Sequence

Step 1 – Basic Update + Dirichlet
Step 2 – Basic Update + Periodic BC
Step 3 – Add PML
Step 4 – TF/SF
Step 5 – Calculate Response
Step 6 – Add a Device and Benchmark
Maxwell’s Equations

We start with Maxwell’s equations in the following form:

\[ \nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} \]
\[ \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \]
\[ \nabla \cdot \vec{B} = 0 \]
\[ \nabla \cdot \vec{D} = \rho_v \]

\[ \vec{D}(t) = \varepsilon_0 \varepsilon \vec{E}(t) \]
\[ \vec{B}(t) = \mu_0 \mu \vec{H}(t) \]
Normalize the Electric Fields

We will now adopt the more conventional approach in FDTD and normalize the electric field according to:

\[ \tilde{E} = \sqrt{\frac{\varepsilon_0}{\mu_0}} \tilde{E} = \frac{1}{\eta_0} \tilde{E} \]

To be consistent, we also need to normalize other parameters related to the electric field.

\[ \tilde{D} = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} \tilde{D} = c_0 \tilde{D} \]
\[ \tilde{P} = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} \tilde{P} = c_0 \tilde{P} \]
\[ \tilde{J} = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} \tilde{J} = c_0 \tilde{J} \]

We won’t be using \( J \) and \( P \).

Normalized Maxwell’s Equations

Using the normalized fields, Maxwell’s equations become

\[ \nabla \times \tilde{E} = -\frac{\mu_r}{c_0} \frac{\partial \tilde{H}}{\partial t} \]
\[ \nabla \times \tilde{H} = \frac{1}{c_0} \frac{\partial \tilde{D}}{\partial t} \]

These equations are independent of \( \varepsilon_r \).

We can derive update equations from these expressions without complicating them with what might need to be done to model \( \varepsilon_r \).

This is a very simple equation that makes it much easier to model more sophisticated dielectrics that may be anisotropic, nonlinear, dispersive, all of the above, or something else altogether.
HDE Algorithm

To make our code more modular, we will modify the FDTD algorithm to what is called the HDE algorithm.

**Update H from E**

\[ \nabla \times \vec{E} = -\frac{\mu_r}{c_0} \frac{\partial \vec{H}}{\partial t} \]

**Update D from H**

\[ \nabla \times \vec{H} = \frac{1}{c_0} \frac{\partial \vec{D}}{\partial t} \]

**Update E from D**

\[ \vec{D} = \varepsilon_r \vec{E} \]

Expand Maxwell’s Equations

\[ \nabla \times \vec{E} = -\frac{\mu_r}{c_0} \frac{\partial \vec{H}}{\partial t} \]  
\[ \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{1}{c_0} \left( \mu_r \frac{\partial H_z}{\partial t} + \mu_\varepsilon \frac{\partial H_y}{\partial t} + \mu_\varepsilon \frac{\partial H_x}{\partial t} \right) \]

\[ \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{1}{c_0} \left( \mu_r \frac{\partial H_x}{\partial t} + \mu_\varepsilon \frac{\partial H_z}{\partial t} + \mu_\varepsilon \frac{\partial H_y}{\partial t} \right) \]

\[ \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial y} = \frac{1}{c_0} \left( \mu_r \frac{\partial H_y}{\partial t} + \mu_\varepsilon \frac{\partial H_z}{\partial t} + \mu_\varepsilon \frac{\partial H_x}{\partial t} \right) \]

\[ \nabla \times \vec{H} = \frac{1}{c_0} \frac{\partial \vec{D}}{\partial t} \]  
\[ \frac{\partial H_y}{\partial y} - \frac{\partial H_z}{\partial z} = \frac{1}{c_0} \frac{\partial D_z}{\partial t} \]
\[ \frac{\partial H_z}{\partial z} - \frac{\partial H_x}{\partial x} = \frac{1}{c_0} \frac{\partial D_x}{\partial t} \]
\[ \frac{\partial H_x}{\partial x} - \frac{\partial H_y}{\partial y} = \frac{1}{c_0} \frac{\partial D_y}{\partial t} \]

\[ \vec{D} = \varepsilon_r \vec{E} \]  
\[ \vec{D}_x = \varepsilon_r \vec{E}_x + \varepsilon_\mu \vec{E}_y + \varepsilon_\mu \vec{E}_z \]
\[ \vec{D}_y = \varepsilon_r \vec{E}_y + \varepsilon_\mu \vec{E}_x + \varepsilon_\mu \vec{E}_z \]
\[ \vec{D}_z = \varepsilon_r \vec{E}_z + \varepsilon_\mu \vec{E}_x + \varepsilon_\mu \vec{E}_y \]
Assume Only Diagonal Tensors

\[ \nabla \times \vec{E} = -\frac{\mu_r}{c_0} \frac{\partial \vec{H}}{\partial t} \]

\[ \nabla \times \vec{H} = \frac{1}{c_0} \frac{\partial \vec{D}}{\partial t} \]

\[ \vec{D} = \varepsilon_r \vec{E} \]

Governing Equations

These are the primary governing equations from which we will formulate the 2D and 3D FDTD method.

\[ \frac{\partial \vec{E}_x}{\partial y} - \frac{\partial \vec{E}_y}{\partial z} = -\frac{\mu_{xx}}{c_0} \frac{\partial H_x}{\partial t} \]
\[ \frac{\partial \vec{E}_x}{\partial z} - \frac{\partial \vec{E}_z}{\partial x} = -\frac{\mu_{yy}}{c_0} \frac{\partial H_y}{\partial t} \]
\[ \frac{\partial \vec{E}_y}{\partial x} - \frac{\partial \vec{E}_x}{\partial y} = -\frac{\mu_{zz}}{c_0} \frac{\partial H_z}{\partial t} \]
\[ \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial z} = \frac{1}{c_0} \frac{\partial D_x}{\partial t} \]
\[ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \frac{1}{c_0} \frac{\partial D_y}{\partial t} \]
\[ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{1}{c_0} \frac{\partial D_z}{\partial t} \]

\[ \vec{D}_x = \varepsilon_{xx} \vec{E}_x \]
\[ \vec{D}_y = \varepsilon_{yy} \vec{E}_y \]
\[ \vec{D}_z = \varepsilon_{zz} \vec{E}_z \]
Final Analytical Equations

It will be beneficial in the formulation and implementation to calculate the curl terms separately. We start by expressing Maxwell’s equations in the following form.

\[
\begin{align*}
C^E_x &= \frac{\mu_0}{c_0} \frac{\partial H_x}{\partial t} \\
C^E_y &= \frac{\mu_0}{c_0} \frac{\partial H_y}{\partial t} \\
C^E_z &= \frac{\mu_0}{c_0} \frac{\partial H_z}{\partial t} \\
C''_x &= \frac{1}{c_0} \frac{\partial D_x}{\partial t} \\
C''_y &= \frac{1}{c_0} \frac{\partial D_y}{\partial t} \\
C''_z &= \frac{1}{c_0} \frac{\partial D_z}{\partial t}
\end{align*}
\]

\[
\begin{align*}
\vec{D}_x &= \varepsilon_0 \vec{E}_x \\
\vec{D}_y &= \varepsilon_0 \varepsilon_y \vec{E}_y \\
\vec{D}_z &= \varepsilon_0 \varepsilon_z \vec{E}_z
\end{align*}
\]

Finite-Difference Approximations
Finite-Difference Equations for $H$

Curl Terms

\[ C_x^E = \frac{\partial E_y}{\partial y} - \frac{\partial E_z}{\partial z} \]
\[ C_y^E = \frac{\partial E_z}{\partial z} - \frac{\partial E_x}{\partial x} \]
\[ C_z^E = \frac{\partial E_x}{\partial x} - \frac{\partial E_y}{\partial y} \]

Maxwell's Equations

\[ C_x^H = \frac{\mu_0}{c_0} \frac{\partial H_x}{\partial t} \]
\[ C_y^H = \frac{\mu_0}{c_0} \frac{\partial H_y}{\partial t} \]
\[ C_z^H = \frac{\mu_0}{c_0} \frac{\partial H_z}{\partial t} \]

Finite-Difference Equations for $D$

Curl Terms

\[ C_x^D = \frac{\partial D_y}{\partial y} - \frac{\partial D_z}{\partial z} \]
\[ C_y^D = \frac{\partial D_z}{\partial z} - \frac{\partial D_x}{\partial x} \]
\[ C_z^D = \frac{\partial D_x}{\partial x} - \frac{\partial D_y}{\partial y} \]

Maxwell's Equations

\[ C_x^D = \frac{1}{c_0} \frac{\partial D_x}{\partial t} \]
\[ C_y^D = \frac{1}{c_0} \frac{\partial D_y}{\partial t} \]
\[ C_z^D = \frac{1}{c_0} \frac{\partial D_z}{\partial t} \]
Finite-Difference Equations for $E$

\[ \tilde{D}_x = \varepsilon_{xx} \tilde{E}_x \]
\[ \tilde{D}_y = \varepsilon_{yy} \tilde{E}_y \]
\[ \tilde{D}_z = \varepsilon_{zz} \tilde{E}_z \]

\[ \tilde{D}_x^{i,j,k} = (\varepsilon_{xx})^{i,j,k} \tilde{E}_x^{i,j,k} \]
\[ \tilde{D}_y^{i,j,k} = (\varepsilon_{yy})^{i,j,k} \tilde{E}_y^{i,j,k} \]
\[ \tilde{D}_z^{i,j,k} = (\varepsilon_{zz})^{i,j,k} \tilde{E}_z^{i,j,k} \]

Reduction to Two Dimensions
Real Electromagnetic Devices

All physical devices are three-dimensional.

3D → 2D (Exact)

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.
3D $\rightarrow$ 2D (Approximate)

Many times it is possible to approximate a 3D device in two dimensions. It is very good practice to at least perform the initial simulations in 2D and only moving to 3D to verify the final design.

Effective indices are best computed by modeling the vertical cross section as a slab waveguide.

A simple average index can also produce good results.

2D Grids are Infinite in the 3rd Dimension

Anything represented on a 2D grid, is actually a device that is of infinite extent along the 3rd dimension.

Assuming the z direction is uniform and of infinite extent, then the field is also uniform and of infinite extent in the z direction.

$$\frac{\partial}{\partial z} = 0$$
Assume Uniform in $z$ Direction

For 2D devices, $\partial/\partial z=0$ and Maxwell’s equations reduce to

$$
\begin{align*}
C_x^e &= \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \\
C_y^e &= \frac{\partial E_y}{\partial x} + \frac{\partial E_x}{\partial y} \\
C_z^e &= \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial x} \\
C_x^h &= \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \\
C_y^h &= \frac{\partial H_y}{\partial x} + \frac{\partial H_x}{\partial y} \\
C_z^h &= \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial x}
\end{align*}
$$

We eliminate the $z$-derivative terms.

$$
\begin{align*}
\dot{D}_i &= \varepsilon_{\mu} E_i \\
\dot{D}_j &= \varepsilon_{\mu} E_j \\
\dot{D}_k &= \varepsilon_{\mu} E_k
\end{align*}
$$
We No Longer Need Grid Index \( k \)

For 2D devices, we only retain grid indices \( i \) and \( j \).

\[
C_i^x = \frac{\partial E_i}{\partial y} \quad C_j^y = \frac{\mu}{c_y} \frac{\partial H_j}{\partial t} \quad C_j^z = \frac{\mu}{c_z} \frac{\partial H_j}{\partial t} \quad C_i^z = -\frac{\mu}{c_i} \frac{\partial H_i}{\partial t} \quad C_j^x = \frac{\mu}{c_j} \frac{\partial H_j}{\partial t} \quad C_i^y = \frac{\partial H_i}{\partial y} \quad C_j^z = \frac{\partial H_j}{\partial z} \quad C_i^x = \frac{\partial H_i}{\partial x}
\]

Two Distinct Modes: \( E_z \) and \( H_z \)

Maxwell’s equations have decoupled into two distinct sets of equations.

\[
C_i^x = \frac{\partial E_i}{\partial y} \quad C_j^y = \frac{\mu}{c_y} \frac{\partial H_j}{\partial t} \quad C_j^z = \frac{\mu}{c_z} \frac{\partial H_j}{\partial t} \quad C_i^z = -\frac{\mu}{c_i} \frac{\partial H_i}{\partial t} \quad C_j^x = \frac{\mu}{c_j} \frac{\partial H_j}{\partial t} \quad C_i^y = \frac{\partial H_i}{\partial y} \quad C_j^z = \frac{\partial H_j}{\partial z} \quad C_i^x = \frac{\partial H_i}{\partial x}
\]

\[
\begin{align*}
\partial_i E_i &= \epsilon_{\perp} E_i \\
\partial_j E_j &= \epsilon_{\parallel} E_j \\
\partial_i H_i &= \mu_{\perp} H_i \\
\partial_j H_j &= \mu_{\parallel} H_j
\end{align*}
\]
**Ez Mode**

\[ C_x^e = \frac{\partial \vec{E}_x}{\partial y} \]
\[ C_y^e = -\frac{\mu_0}{c_0} \frac{\partial H_y}{\partial x} \]
\[ C_z^e = \frac{\partial \vec{E}_z}{\partial x} \]
\[ C_x^f = -\frac{\mu_0}{c_0} \frac{\partial H_x}{\partial t} \]
\[ C_y^f = -\frac{\mu_0}{c_0} \frac{\partial H_y}{\partial t} \]
\[ C_z^f = \frac{\partial \vec{H}_z}{\partial t} \]

\[ \vec{D}_x = \varepsilon_0 \vec{E}_x \]
\[ \vec{D}_y = \varepsilon_0 \vec{E}_y \]
\[ \vec{D}_z = \varepsilon_0 \vec{E}_z \]

**Hz Mode**

\[ C_x^h = \frac{\partial \vec{H}_x}{\partial y} \]
\[ C_y^h = \frac{\partial \vec{H}_y}{\partial x} \]
\[ C_z^h = \frac{\partial \vec{H}_z}{\partial t} \]
\[ C_x^f = -\frac{\mu_0}{c_0} \frac{\partial H_x}{\partial t} \]
\[ C_y^f = -\frac{\mu_0}{c_0} \frac{\partial H_y}{\partial t} \]
\[ C_z^f = \frac{\partial \vec{D}_z}{\partial t} \]

\[ \vec{B}_x = \mu_0 \vec{H}_x \]
\[ \vec{B}_y = \mu_0 \vec{H}_y \]
\[ \vec{B}_z = \mu_0 \vec{H}_z \]
Update Equations for $E_z$ Mode

Solving the finite-difference equations for the future time values of the fields associated with the $E_z$ mode leads to:

\[
\begin{align*}
C_z^{E^{ij}} &= -\frac{\mu_0}{c_0} \left[ H_x^{ij} - H_x^{ij-\Delta t} \right] \\
C_y^{E^{ij}} &= -\frac{\mu_0}{c_0} \left[ H_y^{ij} - H_y^{ij-\Delta t} \right] \\
C_{Hz}^{E^{ij}} &= \frac{1}{c_0} \left[ D_z^{ij+\Delta t} - D_z^{ij-\Delta t} \right] \\
D_z^{ij} &= \left( \varepsilon_z^{ij} \right) E_z^{ij} \\
\end{align*}
\]

\[
\begin{align*}
H_x^{ij+\Delta t} &= H_x^{ij} + \left( -\frac{c_0 \Delta t}{\mu_0} \right) C_z^{E^{ij}} \\
H_y^{ij+\Delta t} &= H_y^{ij} + \left( -\frac{c_0 \Delta t}{\mu_0} \right) C_z^{E^{ij}} \\
D_z^{ij+\Delta t} &= D_z^{ij} + \left( c_0 \Delta t \right) C_{Hz}^{E^{ij}} \\
E_z^{ij+\Delta t} &= \left( \frac{1}{\varepsilon_z^{ij}} \right) \left( D_z^{ij} \right)
\end{align*}
\]
Update Equations for $H_z$ Mode

Solving the finite-difference equations for the future time values of the fields associated with the $H_z$ mode leads to:

$$C_z^j v_j = -\frac{\mu_0}{c_0} \left( H_z^{i\pm} - H_z^{i\pm} \right)$$

$$H_z^{i\pm} = H_z^{i\pm} + \left( -\frac{c_0 \Delta t}{\mu_0} \right) C_z^j v_j$$

$$C_x^j v_j = \frac{1}{\epsilon_0} \left( \frac{\partial}{\partial z} \right) \left( \frac{\partial}{\partial z} \right)$$

$$\vec{E}_z = \left( \frac{1}{\epsilon_0} \right) \left( \frac{\partial}{\partial z} \right)$$

Boundary Conditions
Where Are the Boundary Conditions?

All of the spatial derivatives appear only the curl calculations.

\[ C_x^{(i)} = \frac{E_x^{(i+1)} - E_x^{(i)}}{\Delta x} \quad (x-hi) \]
\[ C_x^{(i)} = \frac{E_x^{(i+1)} - E_x^{(i-1)}}{\Delta x} \quad (x-lo) \]
\[ C_y^{(i)} = \frac{E_y^{(i+1)} - E_y^{(i-1)}}{\Delta y} \quad (y-hi) \]
\[ C_y^{(i)} = \frac{H_y^{(i+1)} - H_y^{(i-1)}}{\Delta y} \quad (y-lo) \]

Boundary conditions are handled in the curl computations.

We have “modularized” the boundary conditions by isolating the curl calculations.

DO NOT USE IF STATEMENTS!!!

Your code will not be much shorter and it will run slower if you use if statements.
**Dirichlet Boundary Conditions for CEx**

For CEx, the problem occurs at the y-hi side of the grid.

We fix it explicitly.

\[
C_x^{E_{y,j}} = \begin{cases} 
\frac{\tilde{E}_x^{j+1} - \tilde{E}_x^j}{\Delta y} & 1. j < N_y \\
0 & 2. j = N_y 
\end{cases}
\]

For CEx, the problem occurs at the y-hi side of the grid. We fix it explicitly.

```matlab
% Compute CEx
for nx = 1 : Nx
    for ny = 1 : Ny-1
        CEx(nx,ny) = (Ez(nx,ny+1) - Ez(nx,ny))/dy;
    end
    CEx(nx,Ny) = (0 - Ez(nx, Ny))/dy;
end
```

**Dirichlet Boundary Conditions for CEy**

For CEy, the problem occurs at the x-hi side of the grid.

We fix it explicitly.

\[
C_y^{E_{x,i}} = \begin{cases} 
\frac{\tilde{E}_y^{i+1} - \tilde{E}_y^i}{\Delta x} & 1. i < N_x \\
0 & 2. i = N_x 
\end{cases}
\]

For CEy, the problem occurs at the x-hi side of the grid. We fix it explicitly.

```matlab
% Compute CEy
for ny = 1 : Ny
    for nx = 1 : Nx-1
        CEy(nx,ny) = - (Ez(nx+1,ny) - Ez(nx,ny))/dx;
    end
    CEy(Nx,ny) = - (0 - Ez(Nx, ny))/dx;
end
```
Dirichlet Boundary Conditions for CHz

For CHz, the problem occurs at both the x-lo side of the grid and y-lo.
We fix both of these explicitly.

\[
C_{ij}^{y} = \begin{cases} 
\frac{H_{i+1,j}^{y} - H_{i,j}^{y}}{\Delta x} - \frac{H_{i,j+1}^{y} - H_{i,j}^{y}}{\Delta y} & \text{1. for } i > 1 \text{ and } j > 1 \\
\frac{H_{i+1,j}^{y} - 0}{\Delta x} - \frac{H_{i,j+1}^{y} - H_{i,j}^{y}}{\Delta y} & \text{2. for } i = 1 \text{ and } j > 1 \\
\frac{H_{i+1,j}^{y} - H_{i,j}^{y}}{\Delta x} - \frac{H_{i,j+1}^{y} - 0}{\Delta y} & \text{3. for } i > 1 \text{ and } j = 1 \\
\frac{H_{i+1,j}^{y} - 0}{\Delta x} - \frac{H_{i,j+1}^{y} - 0}{\Delta y} & \text{4. for } i = 1 \text{ and } j = 1 
\end{cases}
\]

MATLAB Code for CHz (1 of 3)

First, we just blindly implement the curl calculation...

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

As expected, this will produce an error when trying to access Hy(0,ny) or Hx(nx,0)
MATLAB Code for CHz (2 of 3)

Second, we handle the problem at $nx = 1$ explicitly by copying the code inside the $nx$ loop, pasting it above, and handling the problem.

```matlab
% Compute CHz
for ny = 1 : Ny
    CHz(1,ny) = (Hy(1,ny) - 0)/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

We still will have an error at $Hx(1,0)$ and $Hx(nx,0)$
```

MATLAB Code for CHz (3 of 3)

Third, we handle the problem at $ny = 1$ explicitly by copying the code inside the $ny$ loop, pasting it above, and handling the problem.

```matlab
% Compute CHz
CHz(1,1) = (Hy(1,1) - 0)/dx ... 
    - (Hx(1,1) - 0)/dy;
for nx = 2 : Nx
    CHz(nx,1) = (Hy(nx,1) - Hy(nx-1,1))/dx ... 
    - (Hx(nx,1) - 0)/dy;
end
for ny = 2 : Ny
    CHz(1,ny) = (Hy(1,ny) - 0)/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
```
Revised FDTD Algorithm for 2D Simulations

**FDTD Algorithm for \( E_z \) Mode**

\[
C'[t] = \begin{cases} 
\frac{E_y^t - E_y^{t+1}}{\Delta y} & j < N_y \\
0 & j = N_y 
\end{cases} \quad C'[t] = \begin{cases} 
\frac{E_y^t - E_y^{t+1}}{\Delta y} & j < N_y \\
0 & j = N_y 
\end{cases} \\
H_{z+}^t - H_{z+}^{t+1} = \left( \frac{\varepsilon_{\mu}}{\mu_{\varepsilon}} \right) C'[t] \\
H_{z+}^t - H_{z+}^{t+1} = \left( \frac{\varepsilon_{\mu}}{\mu_{\varepsilon}} \right) C'[t] \\
H_{z-}^t = 0 & \forall \Delta y \\
H_{z-}^t = 0 & \forall \Delta y \\
H_{z+}^t = 0 & \forall \Delta y \\
H_{z-}^t = 0 & \forall \Delta y \\
\hat{p}[t] = \hat{p}[t] + (\varepsilon_{\mu}) C'[t] \quad \forall \Delta y \\
\hat{p}[t] = \hat{p}[t] + (\varepsilon_{\mu}) C'[t] \quad \forall \Delta y \\
\hat{E}_z^t = \frac{1}{\varepsilon_{\mu}} \hat{p}[t] \\
\hat{E}_z^t = \frac{1}{\varepsilon_{\mu}} \hat{p}[t]
\]

**Simple soft source**
Animation of Basic 2D FDTD Algorithm

STEP 2 of 1000

Dirichlet Boundary Condition

Dirichlet Boundary Condition

Dirichlet Boundary Condition

Dirichlet Boundary Condition