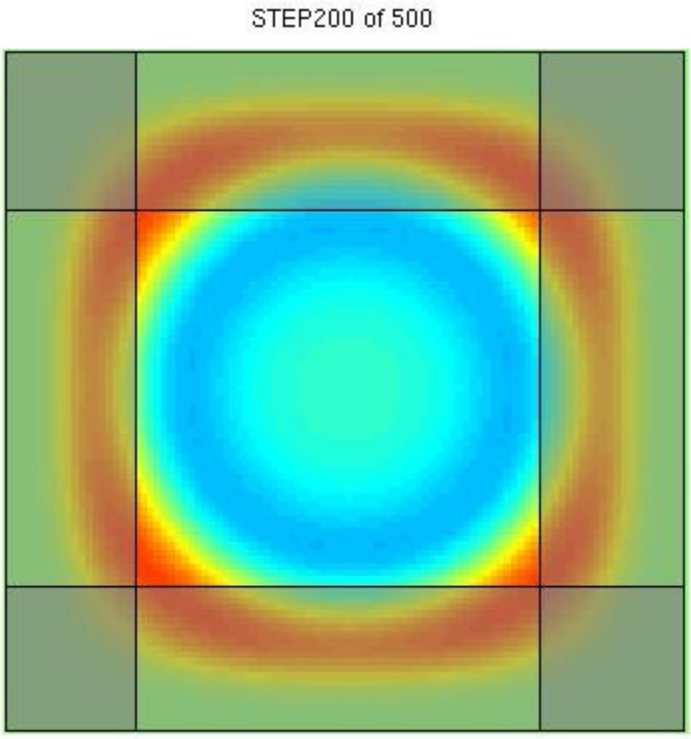
P2 – 2D FDTD with PML



Appendix

P3.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Pre-Program Work
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialize MATLAB
close all; clc;
clear all;

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

%Constants
c0 = 299792458; %m/s
e0 = 8.854187817*10^-12; %F/m
u0 = 1.256637061*10^-6; %H/m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Initialization of Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nx = 100;
Ny = 100;
NPML = [20 21 22 23];
dx = 0.1;
dy = 0.1;
dt = 1.6e-10;
tau = 3.3e-9;
STEPS = 500;


% Compute Grid Axis
xa = [0:Nx-1]*dx;
ya = [0:Ny-1]*dy;

% Compute 2x Grid
Nx2 = 2*Nx;
Ny2 = 2*Ny;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate PML Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Compute sigx
```

```
sigx = zeros(Nx2, Ny2);
for nx=1:2*NPML(1)
  i = 2*NPML(1) - nx + 1;
  sigx(i, :) = (0.5*e0/dt)*(nx/2/NPML(1))^3;
end
for nx=1:2*NPML(2)
  i = Nx2 - 2*NPML(2) + nx;
  sigx(i, :) = (0.5*e0/dt)*(nx/2/NPML(2))^3;
end


% Compute sigy
sigy = zeros(Nx2, Ny2);
for ny=1:2*NPML(3)
  j = 2*NPML(3) - ny + 1;
  sigy(:,j) = (0.5*e0/dt)*(ny/2/NPML(3))^3;
end
for ny=1:2*NPML(4)
  j = Ny2 - 2*NPML(4) + ny;
  sigy(:,j) = (0.5*e0/dt)*(ny/2/NPML(4))^3;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FDTD Initialization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Material Properties
URxx = ones(Nx,Ny);
URyy = ones(Nx,Ny);
ERzz = ones(Nx,Ny);


% Update Coefficients
sigHx = sigx(1:2:Nx2, 2:2:Ny2);
sigHy = sigy(1:2:Nx2, 2:2:Ny2);

mHx0 = (1/dt) + (sigHy/(2*e0));
mHx1 = ((1/dt) - (sigHy/(2*e0)))./mHx0;
mHx2 = -(c0./URxx)./mHx0;
mHx3 = -((c0*dt/e0)*(sigHx./URxx))./mHx0;

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

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

mEz1 = 1./ERzz;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Source
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t0 = 6*tau;
t = [0:STEPS-1]*dt;
g = exp(-((t-t0)/tau).^2);
nx_src = 1+floor(Nx/2);
ny_src = 1+floor(Ny/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FDTD Initialization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Fields
Hx = zeros(Nx,Ny);
Hy = zeros(Nx,Ny);
Dz = zeros(Nx,Ny);
Ez = zeros(Nx,Ny);

%Curl Terms
CEx = zeros(Nx,Ny);
CEy = zeros(Nx,Ny);
CHz = zeros(Nx,Ny);

%Integration Terms
ICEx = zeros(Nx,Ny);
ICEy = zeros(Nx,Ny);
IDz = zeros(Nx,Ny);

figure('Color', 'w');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Execute Simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for T = 1:STEPS

  % Compute Curl of E

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

  for nx=1:Nx
    CEx(nx,Ny) = (Ez(nx,1) - Ez(nx,Ny))/dy;
  end


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

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


  % Update H Integrations
  ICEx = ICEx + CEx;
  ICEy = ICEy + CEy;

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

  %Update Curl of H
  CHz(1,1) = (Hy(1,1) - Hy(Nx,1))/dx - (Hx(1,1) - Hx(1,Ny))/dy;

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

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

  %Update D Integrations
  IDz = IDz + Dz;

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

  %Inject Source
  Dz(nx_src,ny_src) = Dz(nx_src,ny_src) + g(T);

  % Update Ez
  Ez = mEz1.*Dz;


  if mod(T,1) == 0
    draw2d(xa,ya, ERzz, Ez, NPML, 0.03);
    axis equal tight off;
    title(['STEP' num2str(T) ' of ' num2str(STEPS)]);
    drawnow;
  end

  if T==200
    break;
  end
end
```

draw2d.m

```
function [ output_args ] = Draw2D( xa, ya, ERzz, Ez, NPML, ColorAxis)
%DRAW2D Summary of this function goes here
%   Detailed explanation goes here

  [Nx Ny] = size(Ez);

  cla;   hold on;

  imagesc(xa,ya,Ez);
  caxis([-1*ColorAxis, ColorAxis]);

  %Fill in PML

  if NPML(1)
    x = [xa(1) xa(NPML(1)) xa(NPML(1)) xa(1) xa(1)];
    y = [ya(1) ya(1) ya(Ny) ya(Ny) ya(1)];
    c = 0.5 * [1 1 1];
    fill(x,y,c,'FaceAlpha',0.5);
  end
  if NPML(2)
    x = [xa(Nx-NPML(2)) xa(Nx) xa(Nx) xa(Nx-NPML(2)) xa(Nx-NPML(2))];
    y = [ya(1) ya(1) ya(Ny) ya(Ny) ya(1)];
    c = 0.5 * [1 1 1];
    fill(x,y,c,'FaceAlpha',0.5);
  end
  if NPML(3)
    x = [xa(1) xa(Nx) xa(Nx) xa(1) xa(1)];
    y = [ya(1) ya(1) ya(NPML(3)) ya(NPML(3)) ya(1)];
    c = 0.5 * [1 1 1];
    fill(x,y,c,'FaceAlpha',0.5);
  end
  if NPML(4)
    x = [xa(1) xa(Nx) xa(Nx) xa(1) xa(1)];
    y = [ya(Ny-NPML(4)) ya(Ny-NPML(4)) ya(Ny) ya(Ny) ya(Ny-NPML(4))];
    c = 0.5 * [1 1 1];
    fill(x,y,c,'FaceAlpha',0.5);
  end



  hold off;

end
```