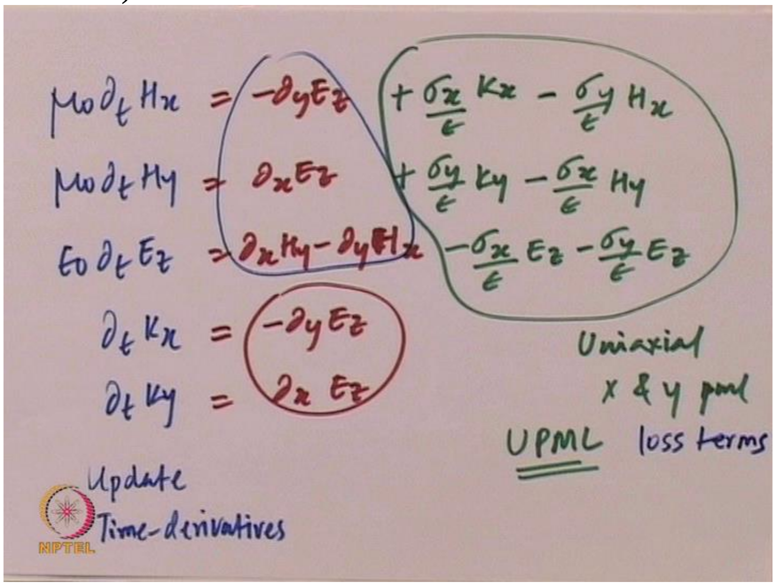
Computational Electromagnetics and Applications
Professor Krish Sankaran
Indian Institute of Technology Bombay
Exercise No 9
Boundary Conditions

Oh boy what we have got into so we have derived an extensive set of equations and these equations are intimidating if you don't understand them conceptually and we have done that for the universal PML that we are going to model in the Matlab environment before I jump into that Matlab code itself I would like to recap the final set of equations that we need to model this particular problem so let's look at this set of equations and get overview of it and we are going to the Matlab model.

(Refer Slide Time: 00: 53)



$$\mu_0 \partial_t H_x = -\partial_y E_z + \frac{\sigma_x}{\epsilon} K_x - \frac{\sigma_y}{\epsilon} H_x$$

$$\mu_0 \partial_t H_y = \partial_x E_z + \frac{\sigma_y}{\epsilon} K_y - \frac{\sigma_x}{\epsilon} H_y$$

$$\epsilon_0 \partial_t E_z = \partial_x H_y - \partial_y H_x - \frac{\sigma_x}{\epsilon} E_z - \frac{\sigma_y}{\epsilon} E_z$$

$$\partial_t K_x = -\partial_y E_z$$

$$\partial_t K_y = \partial_x E_z$$

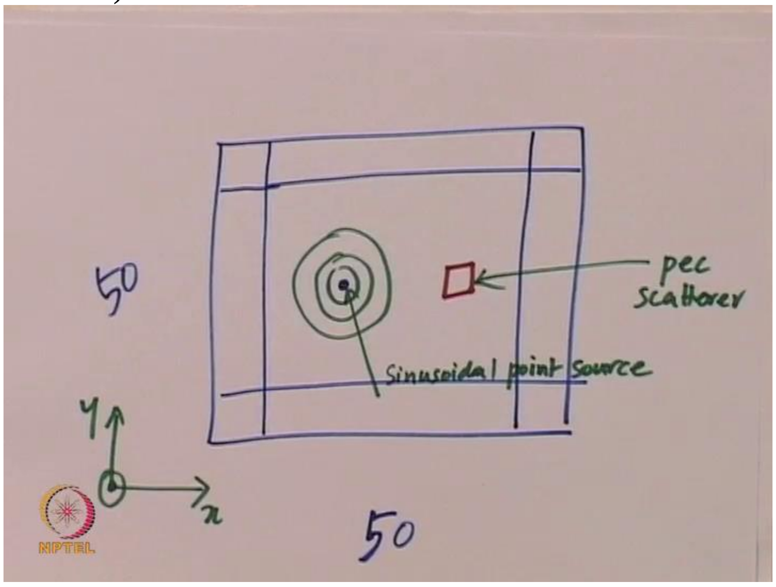
Uniaxial x & y pml

UPML loss terms

Update Time-derivatives

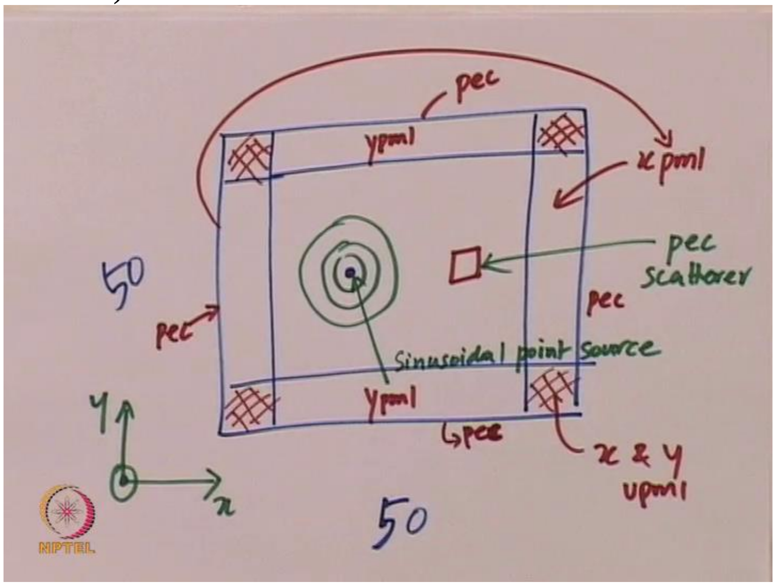
So we have got 1 2 3 4 5 setup update equations and we have got the flux terms that are due to the standard Maxwell equations for 2D problems just math in blue here and we have got to additional equations because we are talking about X and Y oriented uan universal PML and the green terms what we have got here uniaxial X pml last terms so these are the so these are the last terms that we are going to have so now let me describe the problem that we are going to model using a simple diagram.

(Refer Slide Time: 01: 37)



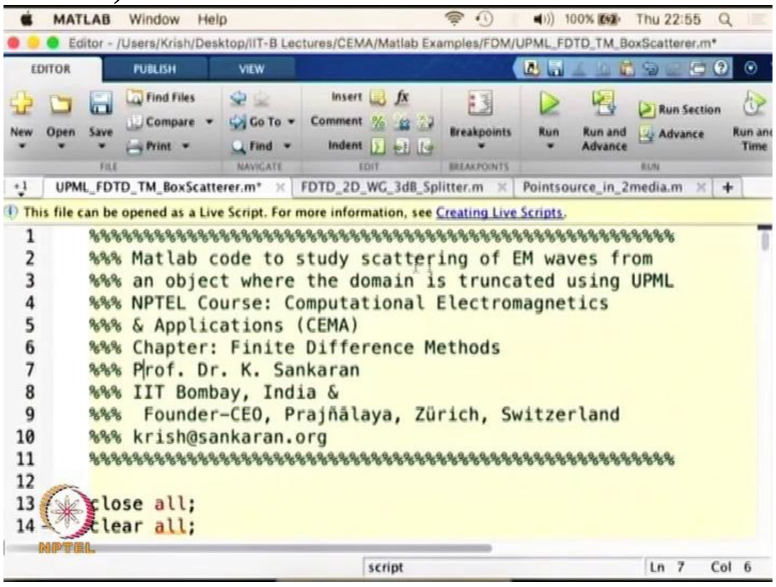
What we have got is a square domain so the domain itself is going to be let’s say 50 by 50 some arbitrary units it could be metres it could be centimetres depending upon whatever interested in let’s say it’s going to be 50 by 50 and I have a point source which is located here. And I have got a scatter which we have modelled it as a small square as well is going to be a perfect electric conductor and now we are going to truncate this particular domain using uni axial PML that are both X and Y oriented so we assume this is the axe and this is the y-axis and the Z is coming out of it out of this paper so what we have got is the Waves that are going to come so these are going to be the sinusoidal point source and this is going to be the Pec scatterer perfect electric conducting scatterer. And how do we run this problem we are going to simulate this problem by truncating this domain using two sets of PML so I am going to draw them like this.

(Refer Slide Time: 03: 33)



So what you have got here is both the x PML this is also X PML and we have got both the y PML so this is y PML and this is y PML and here we have got both the X and Y oriented PML so this is going to be both X and Y u p m l important to know is when we are truncating a problem using p.m. else we have to truncate PML itself it is not just enough to put PML around it the question comes what is going to be that is going to be at the end of the PML so we are going to use perfect electric conductor PC it is going to be PC that is going to sit on all the PML backside so PC is here so the question about PC is the tangential component of the electric field goes to zero there so there is something we have to look into and you will see how VR that in a Matlab code. So this is going to be our problem geometry so let’s go and look into the Matlab code itself.

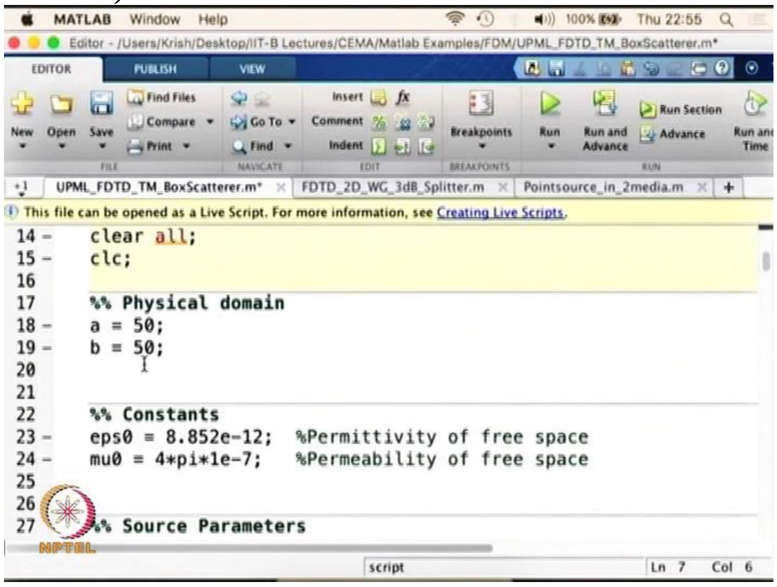
(Refer Slide Time: 05: 05)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 7 Col 6

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Matlab code to study scattering of EM waves from
%%% an object where the domain is truncated using UPML
%%% NPTEL Course: Computational Electromagnetics
%%% & Applications (CEMA)
%%% Chapter: Finite Difference Methods
%%% Prof. Dr. K. Sankaran
%%% IIT Bombay, India &
%%%  Founder-CEO, Prajñālaya, Zürich, Switzerland
%%% krish@sankaran.org
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clear all;
```

So this is going to be the Matlab code that we are going to run so the code is to study the scattering of electromagnetic waves from an object where the domain is truncated using u PML uniaxial PML.

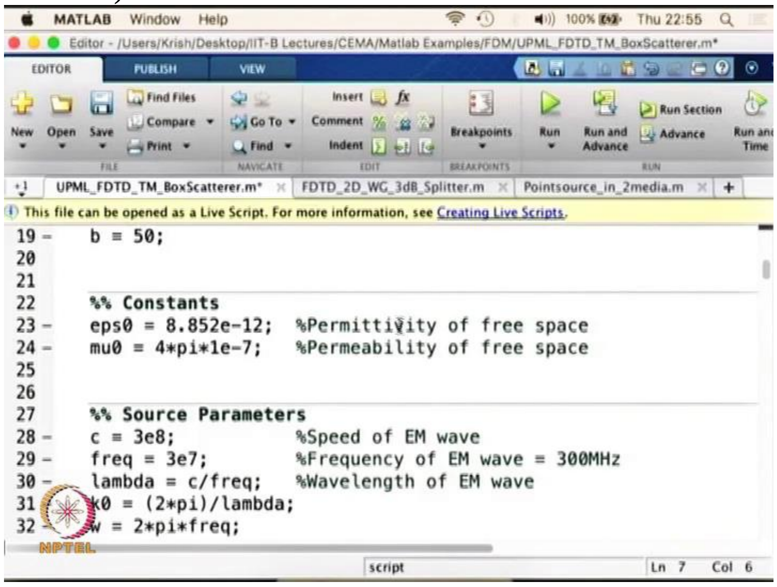
(Refer Slide Time: 05: 20)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 7 Col 6

```
clear all;
clc;

%% Physical domain
a = 50;
b = 50;


%% Constants
eps0 = 8.852e-12;   %Permittivity of free space
mu0 = 4*pi*1e-7;    %Permeability of free space


%% Source Parameters
```

So as I said we are defining the physical domain we are setting its dimensions to be 1550 on the X and Y direction.

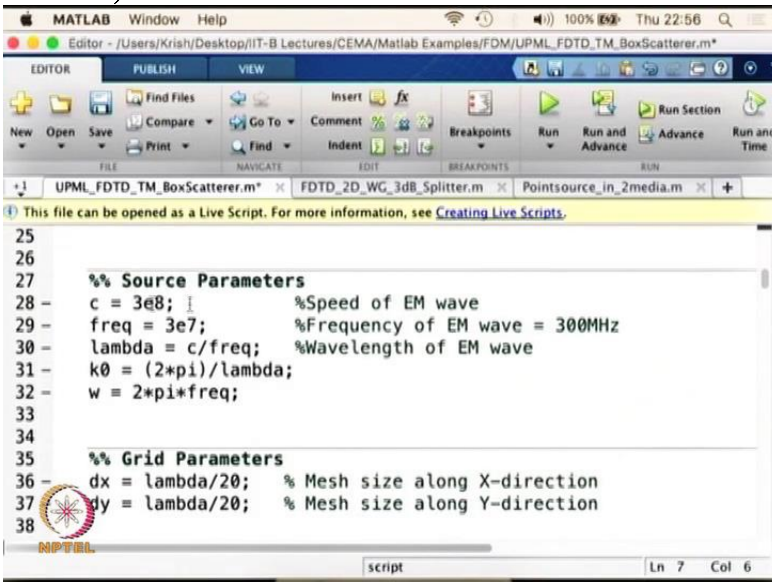
(Refer Slide Time: 05: 28)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 7 Col 6

```
b = 50;


%% Constants
eps0 = 8.852e-12;   %Permittivity of free space
mu0 = 4*pi*1e-7;    %Permeability of free space


%% Source Parameters
c = 3e8;            %Speed of EM wave
freq = 3e7;         %Frequency of EM wave = 300MHz
lambda = c/freq;    %Wavelength of EM wave
k0 = (2*pi)/lambda;
w = 2*pi*freq;
```

We are setting certain physical constants Epsilon 0 and Mu 0 which are the permittivity and permeability of free space.

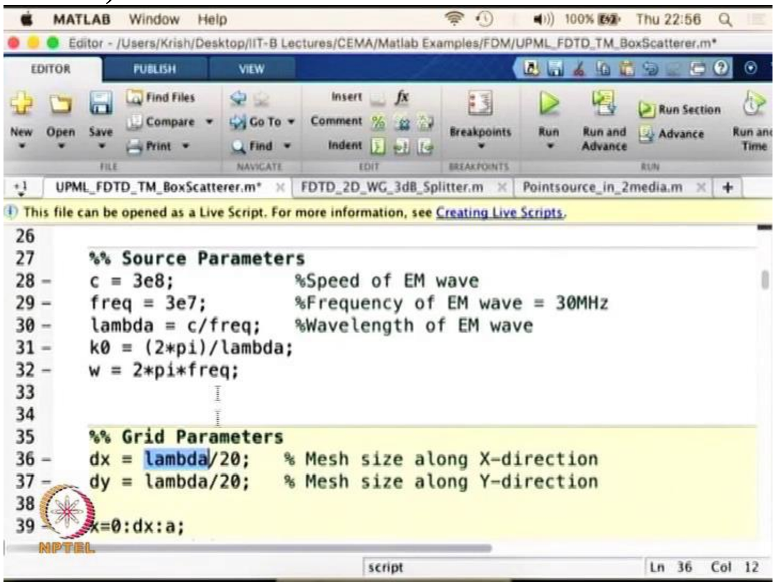
(Refer Slide Time: 05: 35)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
NPTEL

```
%% Source Parameters
c = 3e8;            %Speed of EM wave
freq = 3e7;         %Frequency of EM wave = 300MHz
lambda = c/freq;    %Wavelength of EM wave
k0 = (2*pi)/lambda;
w = 2*pi*freq;


%% Grid Parameters
dx = lambda/20;    % Mesh size along X-direction
dy = lambda/20;    % Mesh size along Y-direction
```

We are setting the source parameters show the velocity of electromagnetic wave propagation the frequency so we are taking the frequency of 30 megahertz it doesn’t matter what your choosing here for this particular simulation we can go also higher frequency but accordingly the parameter for discretisation will change.

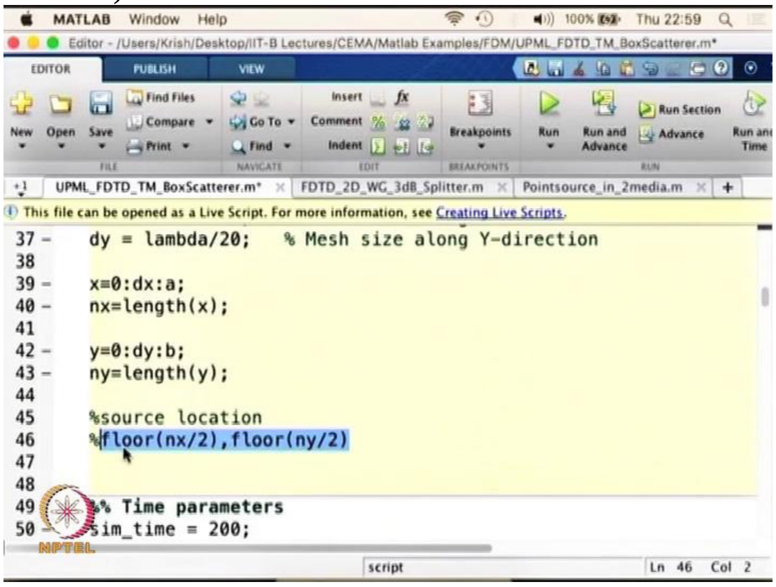
(Refer Slide Time: 05:57)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
NPTEL

```
%% Source Parameters
c = 3e8;            %Speed of EM wave
freq = 3e7;         %Frequency of EM wave = 30MHz
lambda = c/freq;    %Wavelength of EM wave
k0 = (2*pi)/lambda;
w = 2*pi*freq;


%% Grid Parameters
dx = lambda/20;    % Mesh size along X-direction
dy = lambda/20;    % Mesh size along Y-direction

x=0:dx:a;
```

So this is going to be the number of cells that are going to be there for a particular Lambda there should be at least 20 cells that should be sitting with a particular Lambda and here the Lambda value is calculated based on this equation it is C divided by frequency.

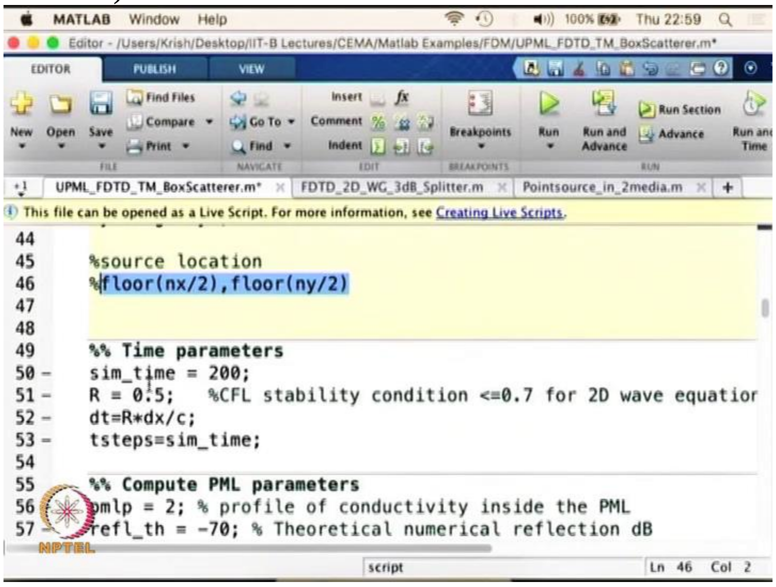
(Refer Slide Time: 06: 15)



```
dy = lambda/20;    % Mesh size along Y-direction

x=0:dx:a;
nx=length(x);

y=0:dy:b;
ny=length(y);

%source location
%floor(nx/2),floor(ny/2)


%% Time parameters
sim_time = 200;
```

So nx is going to be the length of the x step and NY is going to be the length of the by step resource location we have assigned it to be NX by 2 and NY by 2 so it is going to be more or less in the middle

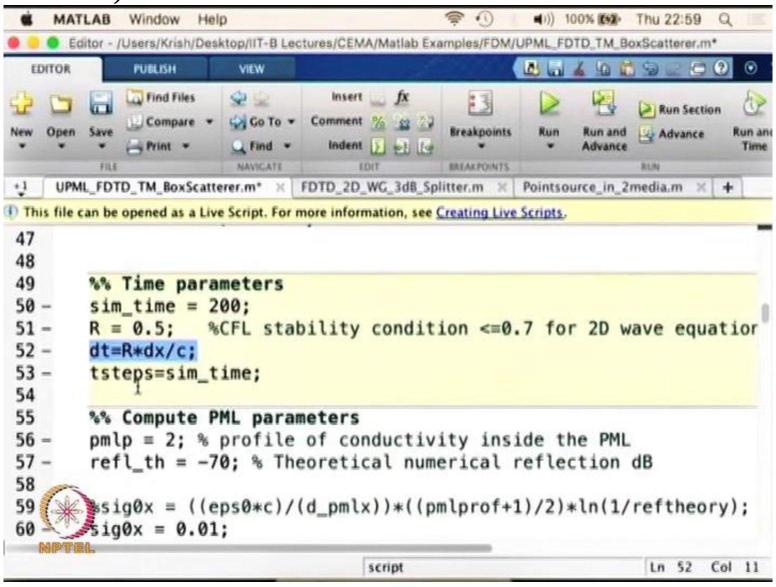
(Refer Slide Time: 06: 35)



```

%source location
%floor(nx/2),floor(ny/2)


%% Time parameters
sim_time = 200;
R = 0.5;    %CFL stability condition <=0.7 for 2D wave equatior
dt=R*dx/c;
tsteps=sim_time;

%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB
```

So the simulation time has been set to 200 time steps and we have set the CFL parameter are to be 0.5 of course you can go until 0. 7 for this kind of problems but we have set it to 0.5 so as to make sure that we get stable simulation.

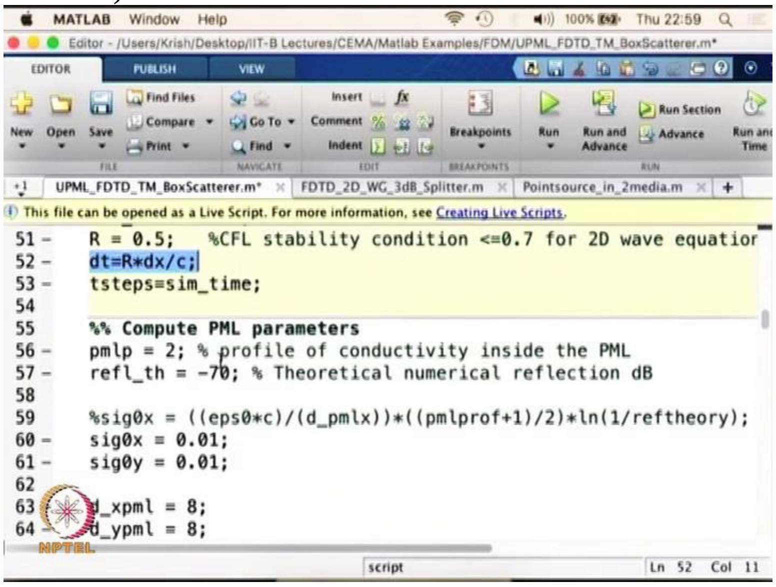
(Refer Slide Time: 06: 52)



```
%% Time parameters
sim_time = 200;
R = 0.5;    %CFL stability condition <=0.7 for 2D wave equatior
dt=R*dx/c;
tsteps=sim_time;

%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB

%sig0x = ((eps0*c)/(d_pmlx))*((pmlprof+1)/2)*ln(1/reftheory);
sig0x = 0.01;
```

And accordingly we get Delta T which is going to be the time discretisation parameter.

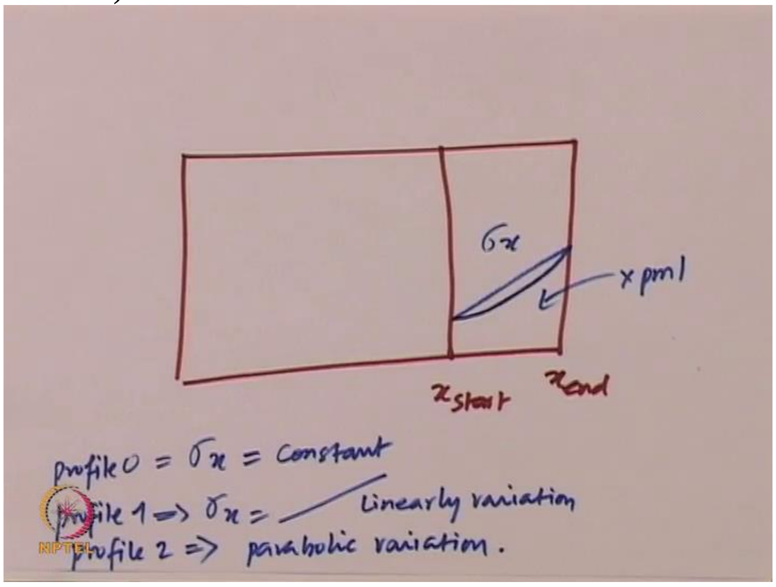
(Refer Slide Time: 07: 00)



```
R = 0.5;    %CFL stability condition <=0.7 for 2D wave equatior
dt=R*dx/c;
tsteps=sim_time;

%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB

%sig0x = ((eps0*c)/(d_pmlx))*((pmlprof+1)/2)*ln(1/reftheory);
sig0x = 0.01;
sig0y = 0.01;

d_xpml = 8;
d_ypml = 8;
```

We have set the value for the profile of the PML to be 2 so I'll explain you what it means in this particular illustration so what you will get is normally a PML that is going to start.

(Refer Slide Time: 07: 15)



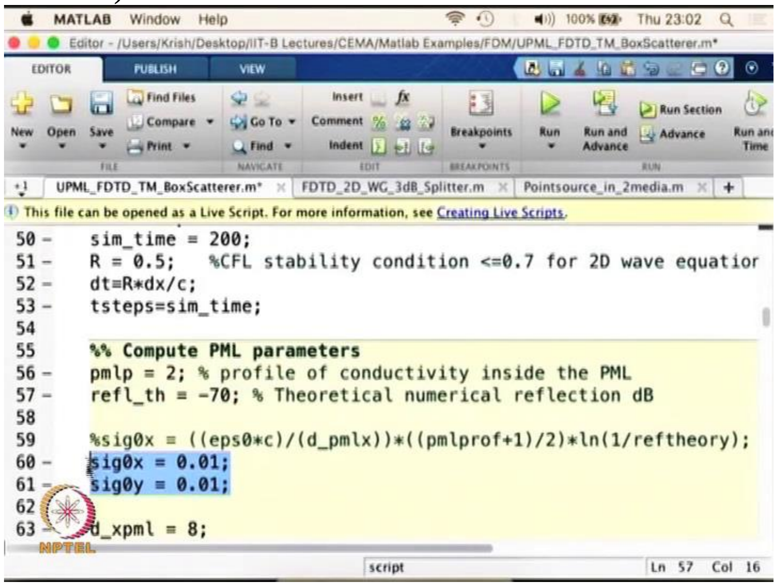
profile 0 = σ_x = constant

profile 1 => σ_x = linearly variation

profile 2 => parabolic variation.

So let's say we're talking about only X oriented PML and this is going to be the starting point of the play PML let's say X start and What happens is the conductivity Sigma x is going to have various values across the entire PML so what you can say is you can either have a constant value so between extract and X and Sigma x is equal to constant is going to be one of the options which we can choose or we can say Sigma X is going to very linearly within the. So this is x PML it is going to vary linearly the value of Sigma X is going to vary linearly the value will be minimum at the starting and maximum at the end or we can go for parabolic value so it is going to have a kind of a curve. So this is going to be for this is Sigma X equal to zero for a constant so the profile will be zero profile is going to be one for the linear variation and profile 2 for the parabolic variation of course you can go higher order but we have chosen a parabolic profile inside the PML.

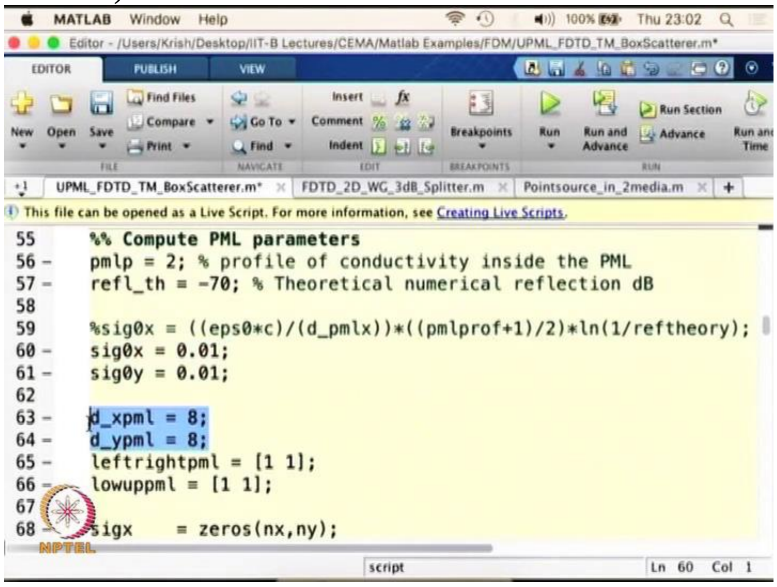
(Refer Slide Time: 09: 04)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 57 Col 16

```
sim_time = 200;
R = 0.5;    %CFL stability condition <=0.7 for 2D wave equatior
dt=R*dx/c;
tsteps=sim_time;

%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB

%sig0x = ((eps0*c)/(d_pmlx))*((pmlprof+1)/2)*ln(1/reftheory);
sig0x = 0.01;
sig0y = 0.01;

d_xpml = 8;
```

You are not going to use this particular term theoretical reflection of course you can compute the value of the maximum value of Sigma inside the X and Y PML using the theoretical reflection I am not going to do that here because it is involving a bit of more Complex illustration in the mathematics for now we will assume the value of maximum Sigma in the X and Y direction going to be 0.01 so we take it as a face value of course there is a wave to derive this using the theoretical reflection that one needs to achieve so let’s not going to it for now we will choose Sigma Sigma X maximum and Sigma why maximum is going to be 0.01.

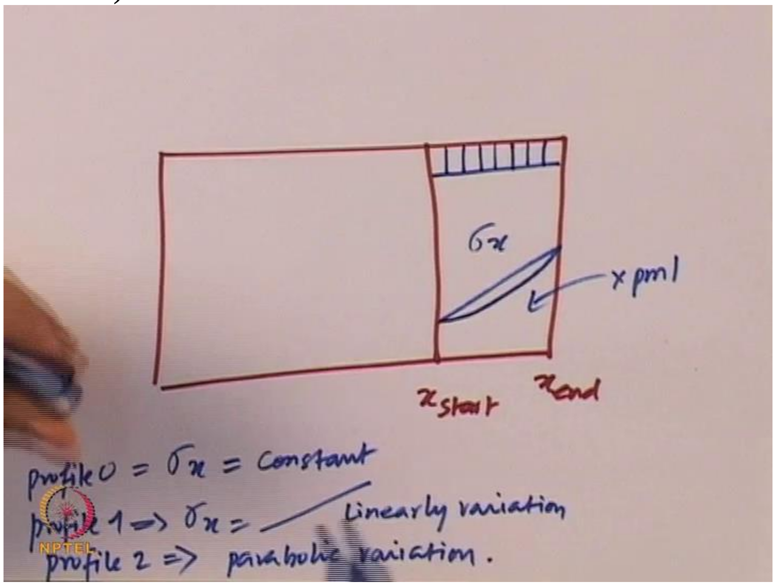
(Refer Slide Time: 09: 54)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m* FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 60 Col 1

```
%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB

%sig0x = ((eps0*c)/(d_pmlx))*((pmlprof+1)/2)*ln(1/reftheory);
sig0x = 0.01;
sig0y = 0.01;

d_xpml = 8;
d_ypml = 8;
leftrightpml = [1 1];
lowuppml = [1 1];
sigx    = zeros(nx,ny);
```

And we are considering the thickness of the PML to be 8 cells.

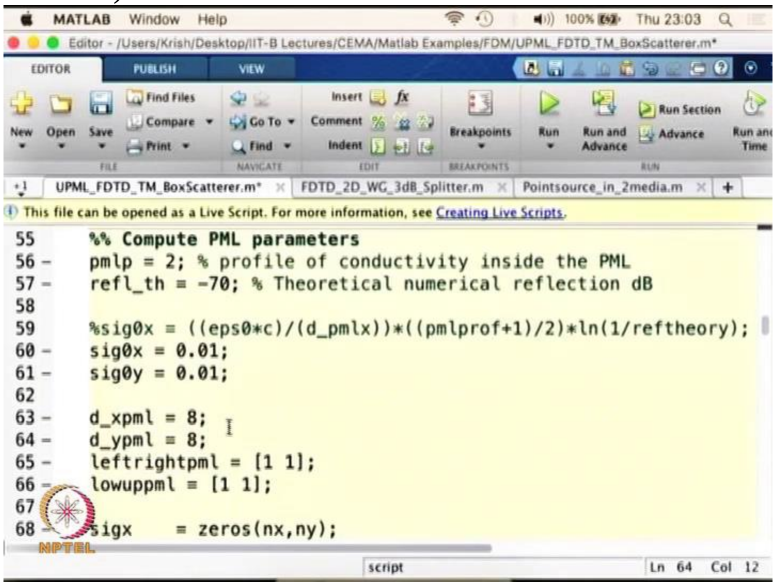
(Refer Slide Time: 10: 07)



σ_x
x pml
x_{start}
x_{end}
profile 0 = σ_x = constant
profile 1 ⇒ σ_x = Linearly variation
profile 2 ⇒ parabolic variation.

So there are going to be 8 cells inside PML so this is 2 this is 4 so there are going to be 8 cells that are going to sit inside the PML of course you can vary this.

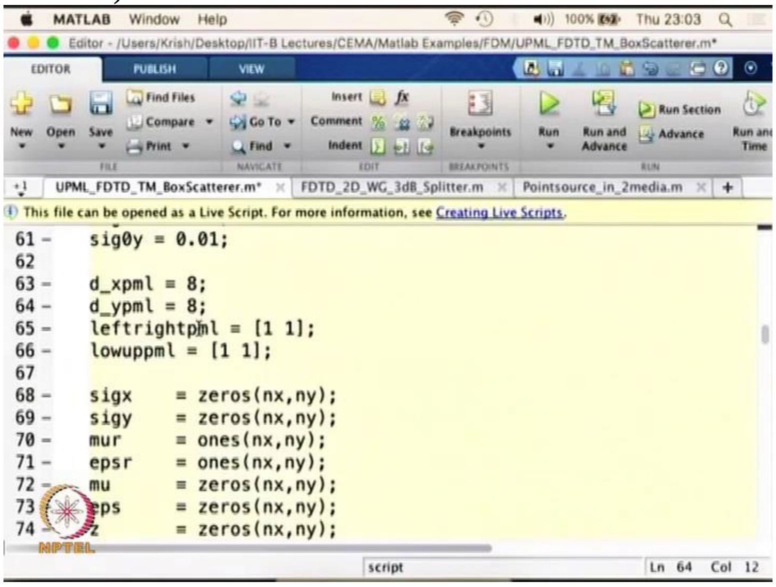
(Refer Slide Time: 10: 15)



```
%% Compute PML parameters
pmlp = 2; % profile of conductivity inside the PML
refl_th = -70; % Theoretical numerical reflection dB

%sig0x = ((eps0*c)/(d_pmlx))*((pmlprof+1)/2)*ln(1/reftheory);
sig0x = 0.01;
sig0y = 0.01;

d_xpml = 8;
d_ypml = 8;
leftrightpml = [1 1];
lowuppml = [1 1];

sigx    = zeros(nx,ny);
```

I wouldn’t do it too much because the idea of putting the PML is to bring the boundary closer and closer to the actual domain of interest if we increase it is going to increase also the computational cost so keep it between 10 to 15 ideally.

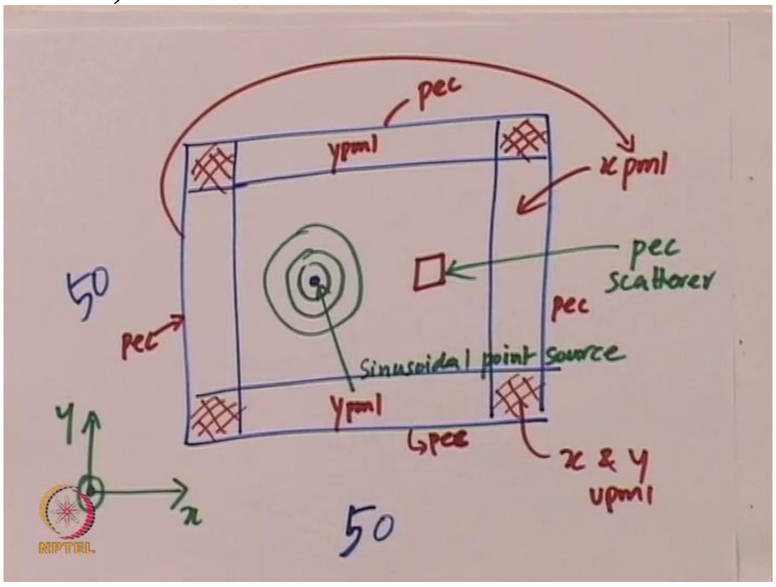
(Refer Slide Time: 10: 36)



```
sig0y = 0.01;

d_xpml = 8;
d_ypml = 8;
leftrightpml = [1 1];
lowuppml = [1 1];

sigx    = zeros(nx,ny);
sigy    = zeros(nx,ny);
mur     = ones(nx,ny);
epsr    = ones(nx,ny);
mu      = zeros(nx,ny);
eps     = zeros(nx,ny);
z       = zeros(nx,ny);
```

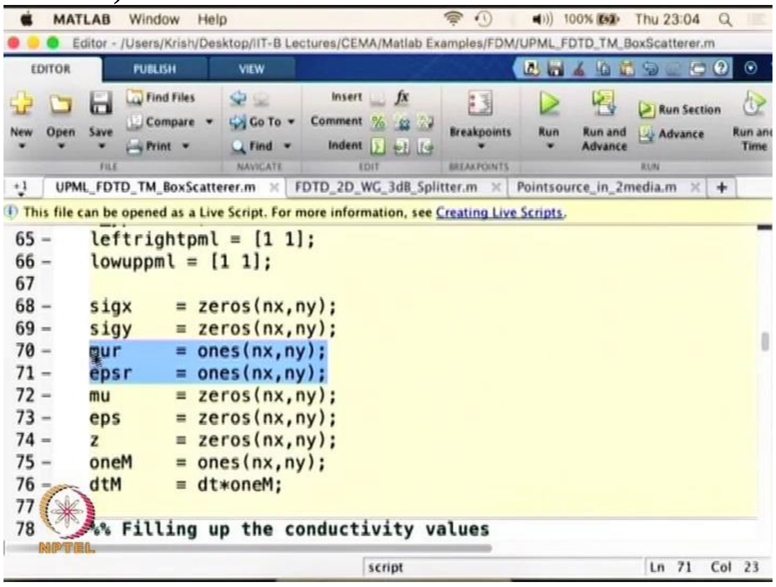
So we said there is going to be both the left and right X PMLS.

(Refer Slide Time: 10: 40)



So that means in our problem what we have here there is going to be both the left and right XML and there is also going to be low and up PML. So there is going to be a low y PML and a up y PML there is going to be a left X PML and right x PML.

(Refer Slide Time: 11: 00)

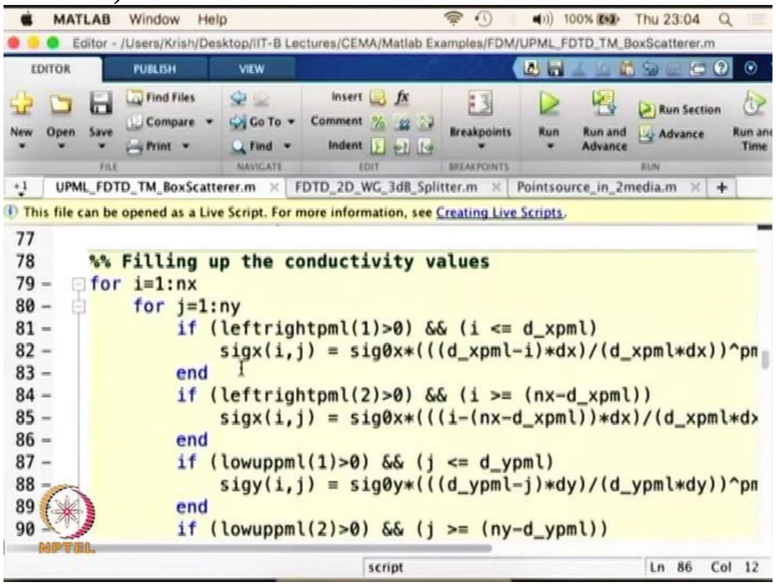

Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
leftrightpml = [1 1];
lowuppml = [1 1];

sigx    = zeros(nx,ny);
sigy    = zeros(nx,ny);
mur     = ones(nx,ny);
epsr    = ones(nx,ny);
mu      = zeros(nx,ny);
eps     = zeros(nx,ny);
z       = zeros(nx,ny);
oneM    = ones(nx,ny);
dtM     = dt*oneM;

%% Filling up the conductivity values
```

So that is a parameter what we are describing here if I put 1111 it means all the four sides are covered by PML. So initially I declare Sigma X and Sigma Y to be zero I declare you are an excellent are relative permeability relative permittivity to be 1 and I declare the value for new and excellent to be zero and the impedance value for space that we are interested in is also declared as zero and we are going to do inside the equations there are some values that we declare just for us to use them inside the equation but it is not important for now.

(Refer Slide Time: 11: 46)

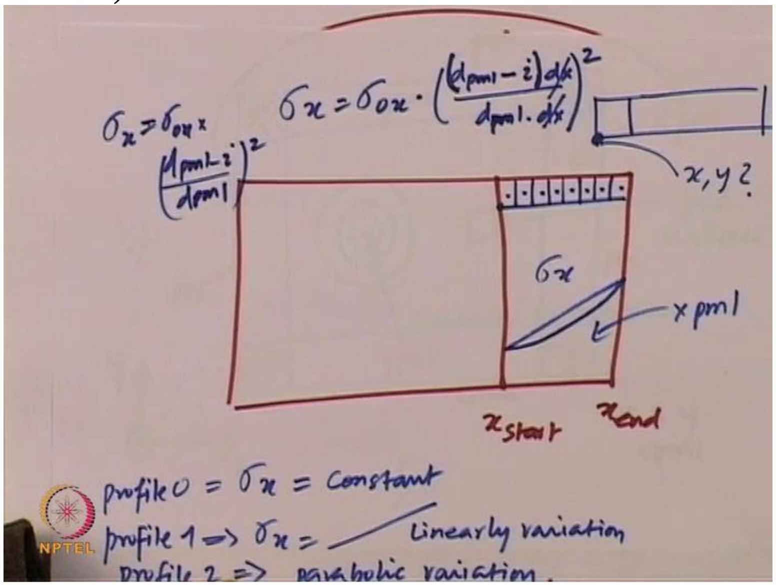

Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```

%% Filling up the conductivity values
for i=1:nx
    for j=1:ny
        if (leftrightpml(1)>0) && (i <= d_xpml)
            sigx(i,j) = sig0x*(((d_xpml-i)*dx)/(d_xpml*dx))^pm
        end
        if (leftrightpml(2)>0) && (i >= (nx-d_xpml))
            sigx(i,j) = sig0x*(((i-(nx-d_xpml))*dx)/(d_xpml*d
        end
        if (lowuppml(1)>0) && (j <= d_ypml)
            sigy(i,j) = sig0y*(((d_ypml-j)*dy)/(d_ypml*dy))^pm
        end
        if (lowuppml(2)>0) && (j >= (ny-d_ypml))
```

So I find out what is going to be the conductivity values so the value that I have to assign into the PML.

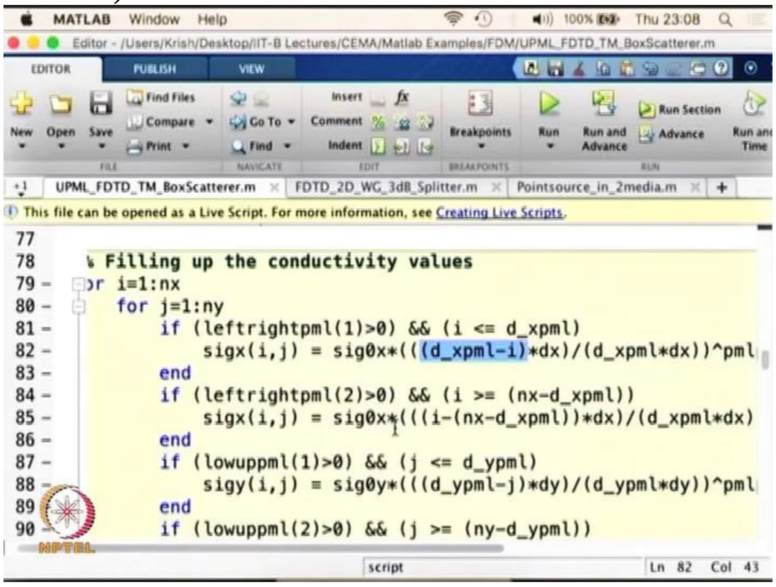
(Refer Slide Time: 11: 52)


$\sigma_x = \sigma_{0x} \times \left(\frac{dpml - i}{dpml}\right)^2$
$\sigma_x = \sigma_{0x} \cdot \left(\frac{(dpml - i)\,dx}{dpml \cdot dx}\right)^2$
x, y ?
σ_x
x pml
x_{start}
x_{end}
profile 0 = σ_x = constant
profile 1 => σ_x = linearly variation
profile 2 => parabolic variation.

So if you see here I said it’s going to vary in a parabolic manner but I have to know what is going to be each of the value at this point this point this point this point this point this point this point. So I am saying the value inside each of the cell is going to depend on the X and Y coordinates of that particular cell so I am going to take the first edge of that particular cell and see what are the X and Y coordinates of this particular cell so if I have a and write it in an elaborated manner so there are going to be different cells initially I take this particular point and ask what is going to be the ex and why value text values important when I am talking about X oriented PML the why value will be important when I am talking about y oriented PML the moment I know X and Y coordinates I can assign the value for the conductivity using this particular equation.

So I am saying the value for PML Sigma x equal to the Sigma not x the maximum value x DP ml minus the co-ordinate I x dx this is going to give me my ex value there should be a bracket here divided by dbml x dx the whole square so I have chosen the second order profile so I am using the square here and you can infect cancel the DX what you will get is Sigma x is equal to Sigma not x x dp ml minus I divided by DP anal whole square so this is what we are saying in this particular equation here in the code.

(Refer Slide Time: 14: 10)



UPML_FDTD_TM_BoxScatterer.m | FDTD_2D_WG_3dB_Splitter.m | Pointsource_in_2media.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

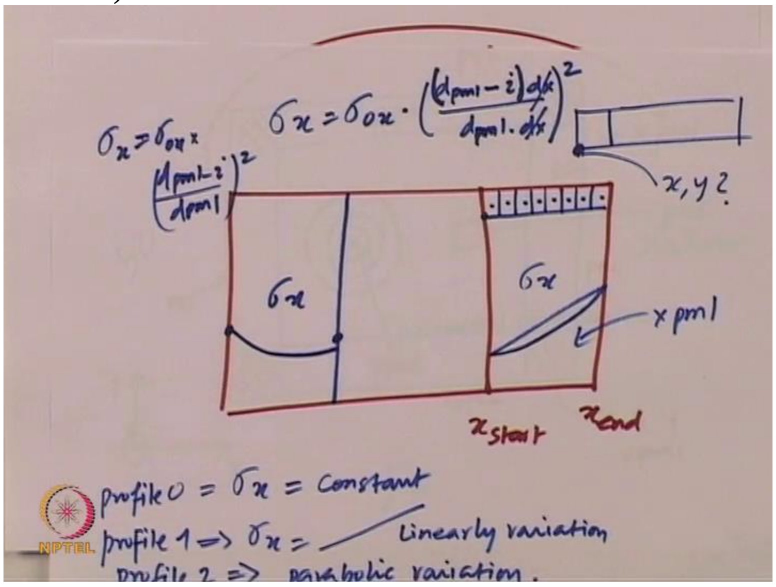
```
% Filling up the conductivity values
for i=1:nx
    for j=1:ny
        if (leftrightpml(1)>0) && (i <= d_xpml)
            sigx(i,j) = sig0x*(((d_xpml-i)*dx)/(d_xpml*dx))^pml
        end
        if (leftrightpml(2)>0) && (i >= (nx-d_xpml))
            sigx(i,j) = sig0x*(((i-(nx-d_xpml))*dx)/(d_xpml*dx)
        end
        if (lowuppml(1)>0) && (j <= d_ypml)
            sigy(i,j) = sig0y*(((d_ypml-j)*dy)/(d_ypml*dy))^pml
        end
        if (lowuppml(2)>0) && (j >= (ny-d_ypml))
```

script | Ln 82 Col 43

C Sigma X not is a maximum value and I am finding out what is the position of the cell from the thickness and subtracting the value so I will know the position of the cell with respect to the thickness the maximum value will be at the distance dxpml which is the length of the PML and a minimum value will be at the point where the PML is starting so when I is the first one so it will have the minimum value when I is equal to dxpml you will have the maximum value.

So now you can do the same things from other domains when you are going to the left oriented xPML what you need to do is you need to swap this as I minus dxpml instead of dxPML minus I because there starting point will be on the first one and then the ending point will be on the left hand side.

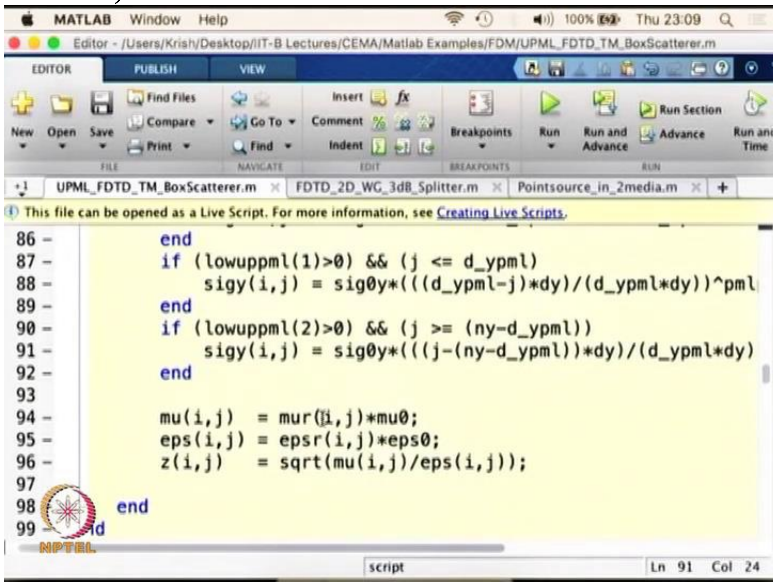
(Refer Slide Time: 15: 14)



$\sigma_x = \sigma_{0x} \times \left(\frac{d_{pml} - i}{d_{pml}}\right)^2$
$\sigma_x = \sigma_{0x} \cdot \left(\frac{(d_{pml} - i)\,dx}{d_{pml} \cdot dx}\right)^2$
x, y ?
σ_x
σ_x
x pml
x_{start}
x_{end}
profile 0 = σ_x = constant
profile 1 => σ_x = Linearly variation
profile 2 => parabolic variation.

What I mean is when we go back here there is going to be also and x oriented PML but on the left hand side so the first point here the rightmost point of this PML will have the lowest value and the leftmost point Will have the maximum value so the value is going to go down where is he at the leftmost point will have the minimum value and the rightmost point will have the maximum value so you have to swap it. And that is why we have swapped this particular time also this you can see while you go in to the code.

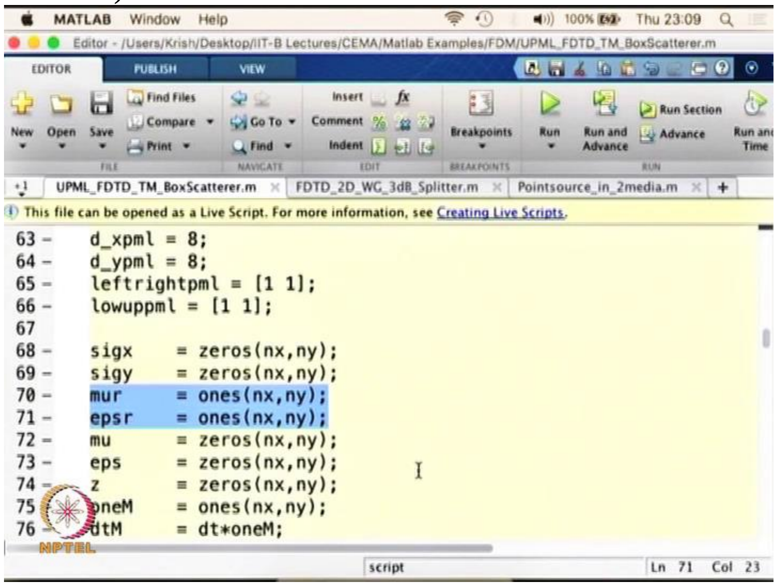
(Refer Slide Time: 15: 44)



```
        end
        if (lowuppml(1)>0) && (j <= d_ypml)
            sigy(i,j) = sig0y*(((d_ypml-j)*dy)/(d_ypml*dy))^pml
        end
        if (lowuppml(2)>0) && (j >= (ny-d_ypml))
            sigy(i,j) = sig0y*(((j-(ny-d_ypml))*dy)/(d_ypml*dy)
        end

        mu(i,j)  = mur(i,j)*mu0;
        eps(i,j) = epsr(i,j)*eps0;
        z(i,j)   = sqrt(mu(i,j)/eps(i,j));

    end
end
```

And what you are doing is we are also assigning the value for newer Epsilon r for the particular thing remember we have declared this one as one so we are doing it for free space where the Epsilon r and Mu r taken to be ones because we have declared them as once in this particular step.

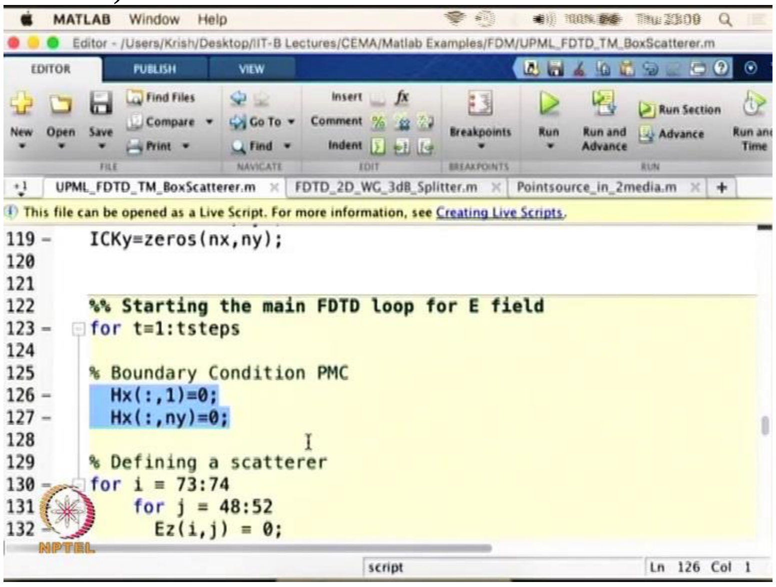
(Refer Slide Time: 16: 08)



```
d_xpml = 8;
d_ypml = 8;
leftrightpml = [1 1];
lowuppml = [1 1];

sigx    = zeros(nx,ny);
sigy    = zeros(nx,ny);
mur     = ones(nx,ny);
epsr    = ones(nx,ny);
mu      = zeros(nx,ny);
eps     = zeros(nx,ny);
z       = zeros(nx,ny);
oneM    = ones(nx,ny);
dtM     = dt*oneM;
```

So she here these are the values that we have declared as once

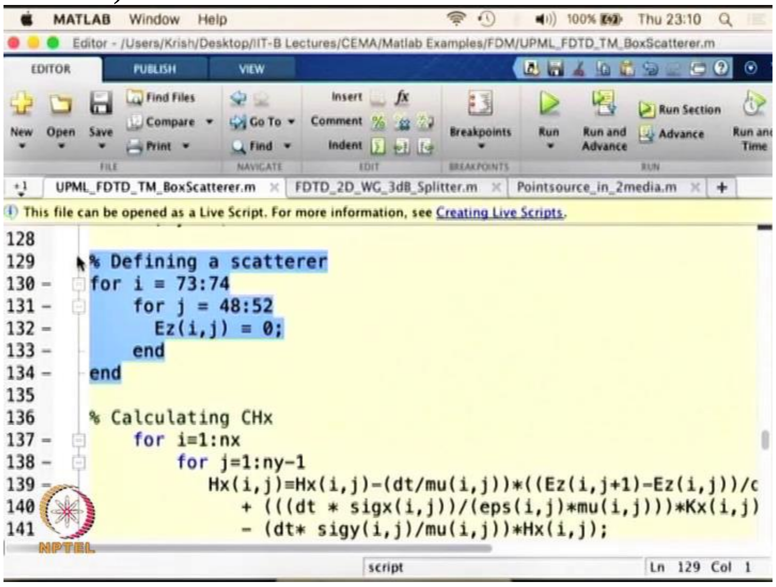
(Refer Slide Time: 16: 14)


MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 126 Col 1

```
ICKy=zeros(nx,ny);


%% Starting the main FDTD loop for E field
for t=1:tsteps

% Boundary Condition PMC
  Hx(:,1)=0;
  Hx(:,ny)=0;

% Defining a scatterer
for i = 73:74
    for j = 48:52
      Ez(i,j) = 0;
```

And we are going to initialise the fields we are setting the boundary conditions I said all the boundaries of the PML are going to be truncated using the perfect magnetic conductor or perfect electric conductor.

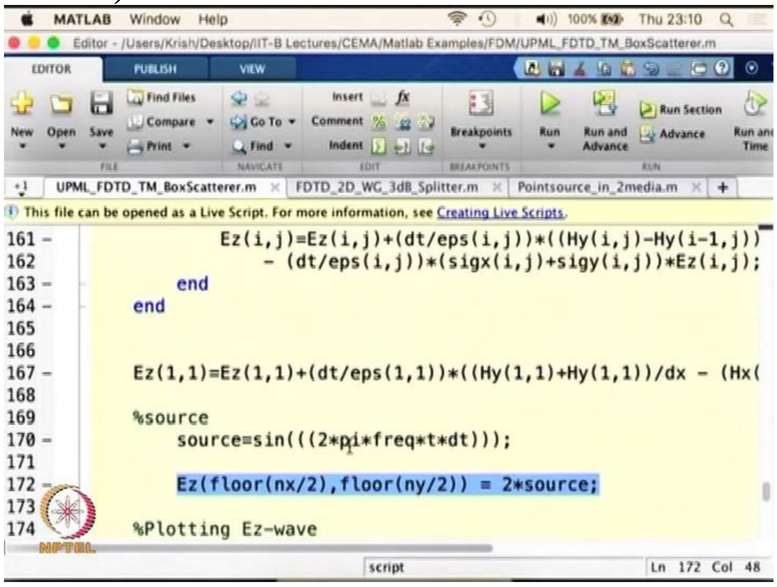
(Refer Slide Time: 16: 32)


MATLAB Window Help
Thu 23:10
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m
EDITOR PUBLISH VIEW
UPML_FDTD_TM_BoxScatterer.m FDTD_2D_WG_3dB_Splitter.m Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 129 Col 1

```

% Defining a scatterer
for i = 73:74
    for j = 48:52
      Ez(i,j) = 0;
    end
end

% Calculating CHx
    for i=1:nx
        for j=1:ny-1
           Hx(i,j)=Hx(i,j)-(dt/mu(i,j))*((Ez(i,j+1)-Ez(i,j))/c
               + (((dt * sigx(i,j))/(eps(i,j)*mu(i,j)))*Kx(i,j)
               - (dt* sigy(i,j)/mu(i,j))*Hx(i,j);
```

And what we are saying here is we are defining a scatter and the position of the scatterer is going to be defined by this particular equation

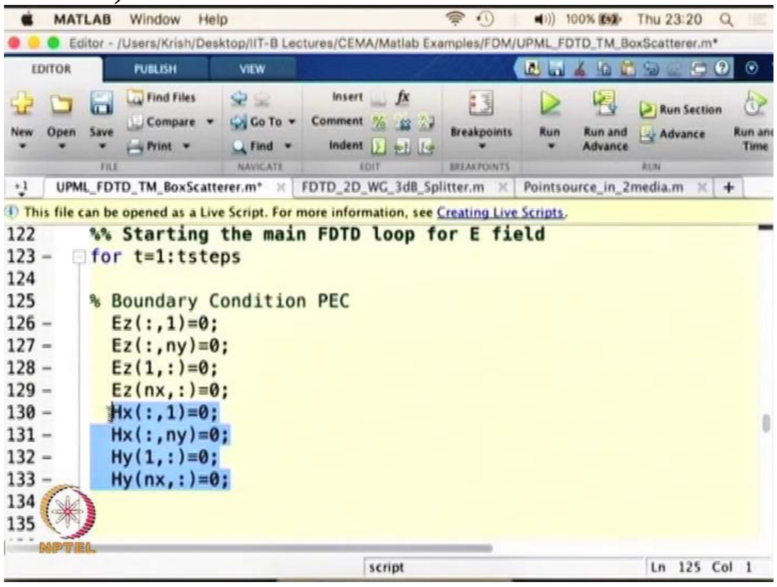
(Refer Slide Time: 16: 43)


Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m
UPML_FDTD_TM_BoxScatterer.m
FDTD_2D_WG_3dB_Splitter.m
Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
                Ez(i,j)=Ez(i,j)+(dt/eps(i,j))*((Hy(i,j)-Hy(i-1,j))
                   - (dt/eps(i,j))*(sigx(i,j)+sigy(i,j))*Ez(i,j);
            end
        end


        Ez(1,1)=Ez(1,1)+(dt/eps(1,1))*((Hy(1,1)+Hy(1,1))/dx - (Hx(

        %source
            source=sin(((2*pi*freq*t*dt)));

            Ez(floor(nx/2),floor(ny/2)) = 2*source;

        %Plotting Ez-wave
```

So the scatterer is the one which we are having here and we are saying that the source value is going to be at this point and we are assigning it to the EZ component here and they are going to be exactly in the middle.

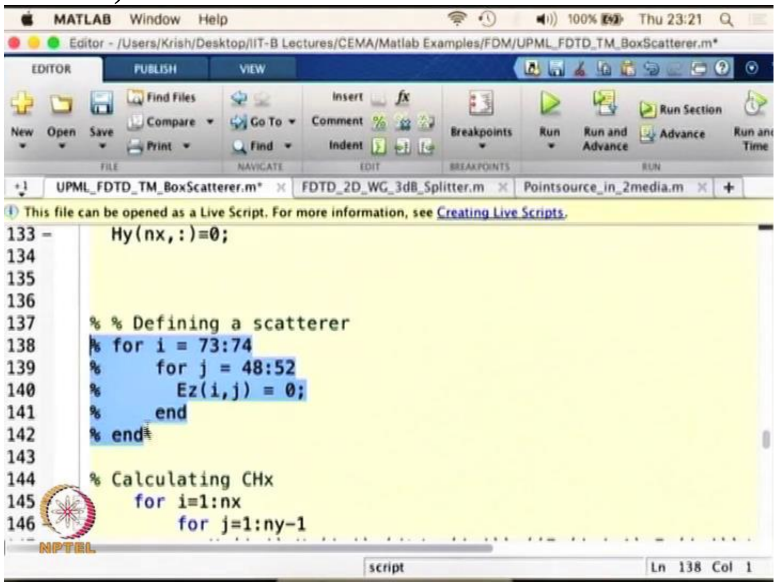
(Refer Slide Time: 17: 00)


Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
UPML_FDTD_TM_BoxScatterer.m*
FDTD_2D_WG_3dB_Splitter.m
Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
%% Starting the main FDTD loop for E field
for t=1:tsteps

% Boundary Condition PEC
  Ez(:,1)=0;
  Ez(:,ny)=0;
  Ez(1,:)=0;
  Ez(nx,:)=0;
  Hx(:,1)=0;
  Hx(:,ny)=0;
  Hy(1,:)=0;
  Hy(nx,:)=0;
```

So now we are going to set up the boundary conditions for truncating the PML itself so we said we will go for the PEC condition we are putting the tangential component of the electric field and the normal components according to the position of the magnetic field it will be zero.

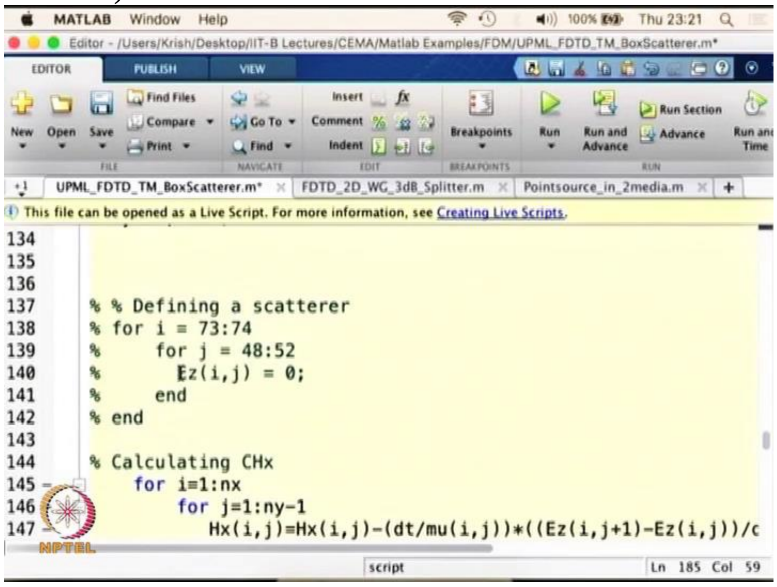
(Refer Slide Time: 17: 18)



```
    Hy(nx,:)=0;



% % Defining a scatterer
% for i = 73:74
%     for j = 48:52
%       Ez(i,j) = 0;
%     end
% end

% Calculating CHx
  for i=1:nx
      for j=1:ny-1
```

And we are defining the scatter not initially Although we can look where the scatterer is going to be initially when we are running the program we are not going to define the scatter so these are commented now.

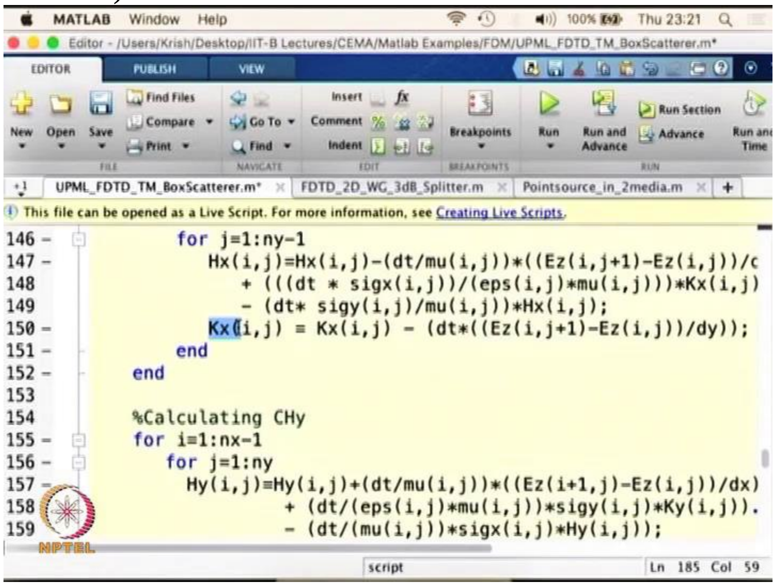
(Refer Slide Time: 17: 34)



```



% % Defining a scatterer
% for i = 73:74
%     for j = 48:52
%       Ez(i,j) = 0;
%     end
% end

% Calculating CHx
  for i=1:nx
      for j=1:ny-1
         Hx(i,j)=Hx(i,j)-(dt/mu(i,j))*((Ez(i,j+1)-Ez(i,j))/c
```

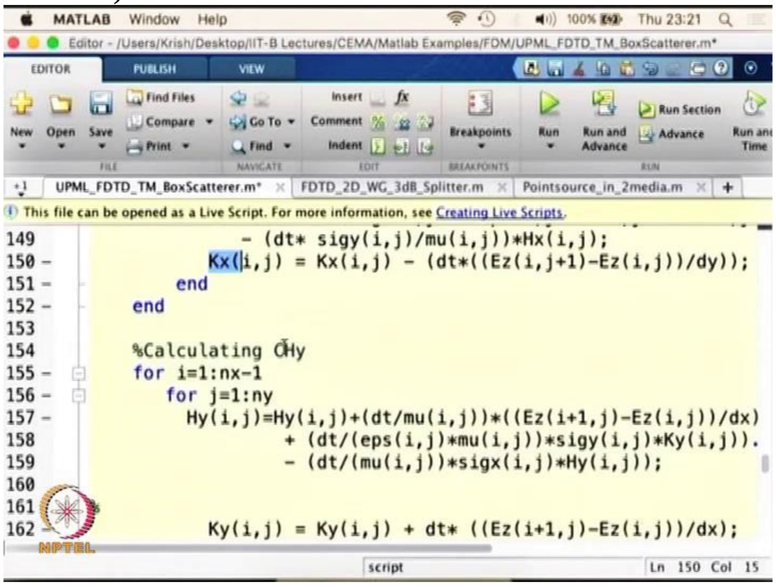
But we have shown where the scatter is going to be using a rectangle that is Martin the magenta colour so it is going to be the position so when we are going to simulate it for the first time you will see that there is a scatter position but the scatter is not defined the moment we define the scatter we have to put the tangential component of electric field to be zero then it will act as a perfect but for now we have commented it.

(Refer Slide Time: 18: 00)


```
        for j=1:ny-1
            Hx(i,j)=Hx(i,j)-(dt/mu(i,j))*((Ez(i,j+1)-Ez(i,j))/c
                + (((dt * sigx(i,j))/(eps(i,j)*mu(i,j)))*Kx(i,j)
                - (dt* sigy(i,j)/mu(i,j))*Hx(i,j);
            Kx(i,j) = Kx(i,j) - (dt*((Ez(i,j+1)-Ez(i,j))/dy));
        end
    end

    %Calculating CHy
    for i=1:nx-1
        for j=1:ny
          Hy(i,j)=Hy(i,j)+(dt/mu(i,j))*((Ez(i+1,j)-Ez(i,j))/dx)
                  + (dt/(eps(i,j)*mu(i,j))*sigy(i,j)*Ky(i,j)).
                  - (dt/(mu(i,j))*sigx(i,j)*Hy(i,j));
```

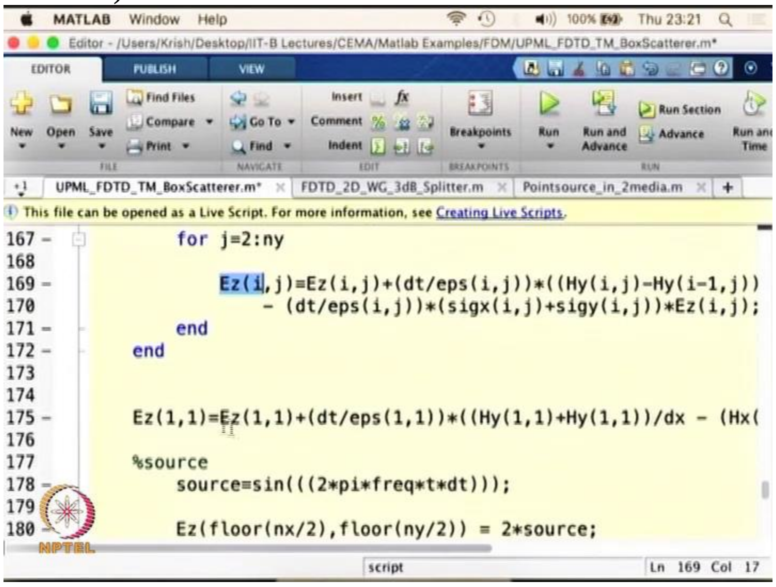
And the calculation for HX we will have kx component

(Refer Slide Time: 18: 06)


```
                - (dt* sigy(i,j)/mu(i,j))*Hx(i,j);
            Kx(i,j) = Kx(i,j) - (dt*((Ez(i,j+1)-Ez(i,j))/dy));
        end
    end

    %Calculating CHy
    for i=1:nx-1
        for j=1:ny
          Hy(i,j)=Hy(i,j)+(dt/mu(i,j))*((Ez(i+1,j)-Ez(i,j))/dx)
                  + (dt/(eps(i,j)*mu(i,j))*sigy(i,j)*Ky(i,j)).
                  - (dt/(mu(i,j))*sigx(i,j)*Hy(i,j));


            Ky(i,j) = Ky(i,j) + dt* ((Ez(i+1,j)-Ez(i,j))/dx);
```

Calculation for HY will have k y component.

(Refer Slide Time: 18: 11)



Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*

UPML_FDTD_TM_BoxScatterer.m* | FDTD_2D_WG_3dB_Splitter.m | Pointsource_in_2media.m

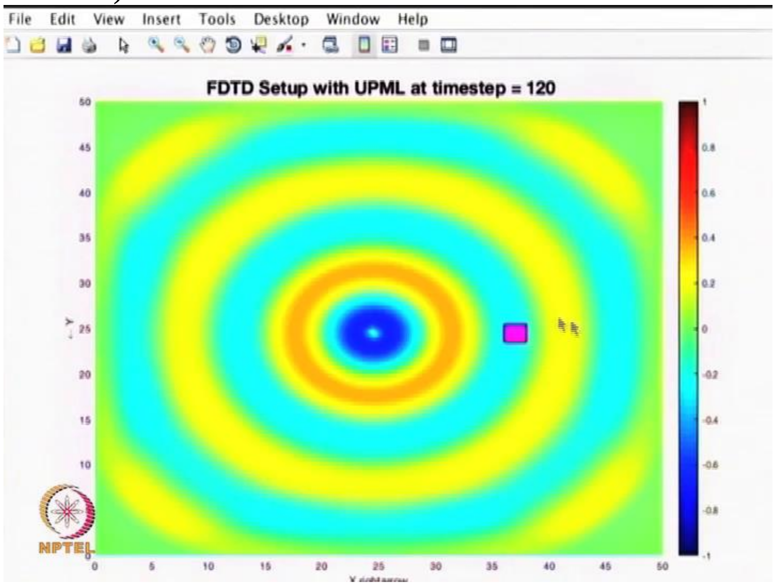
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
            for j=2:ny

                Ez(i,j)=Ez(i,j)+(dt/eps(i,j))*((Hy(i,j)-Hy(i-1,j))
                    - (dt/eps(i,j))*(sigx(i,j)+sigy(i,j))*Ez(i,j);
            end
        end


        Ez(1,1)=Ez(1,1)+(dt/eps(1,1))*((Hy(1,1)+Hy(1,1))/dx - (Hx(

        %source
            source=sin(((2*pi*freq*t*dt)));

            Ez(floor(nx/2),floor(ny/2)) = 2*source;
```

script | Ln 169 Col 17

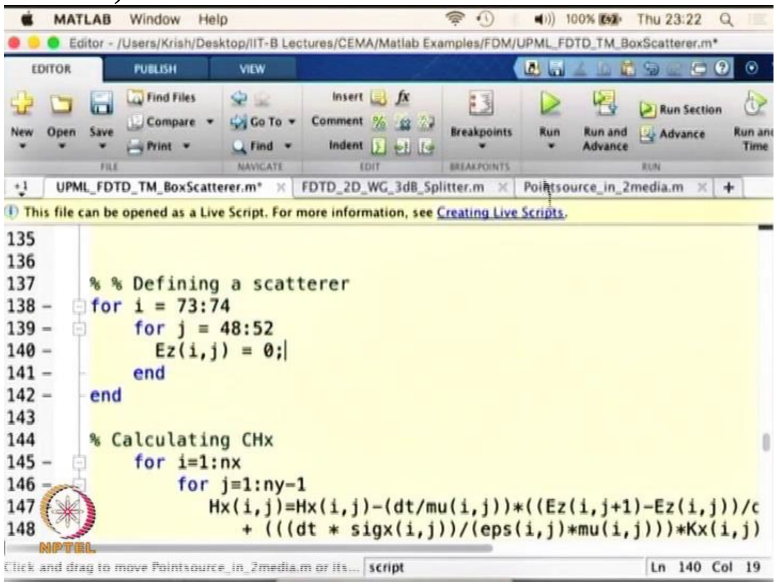
And we have the EZ component and we keep updating it and the source location is exactly in the middle we have taken twice the magnitude of the source just as to increase the contrast in the simulation so as to have a high amplitude so let’s not simulate this particular program and see what we can expect from this particular simulation.

(Refer Slide Time: 18: 32)



So what you are saying here is there is a point source and this is the location of the scatterer as I said right now it is not a scatterer it is just a marker so you see the wave is moving through it without any scattering and the PML starts from this particular point you See that it starts to decay and slowly starts to disappear what is getting absorbed and the field is going unaffected by the scatter as I said it is still not a scatter so the field is getting unaffected by the location because it is still not defined as a scatter.

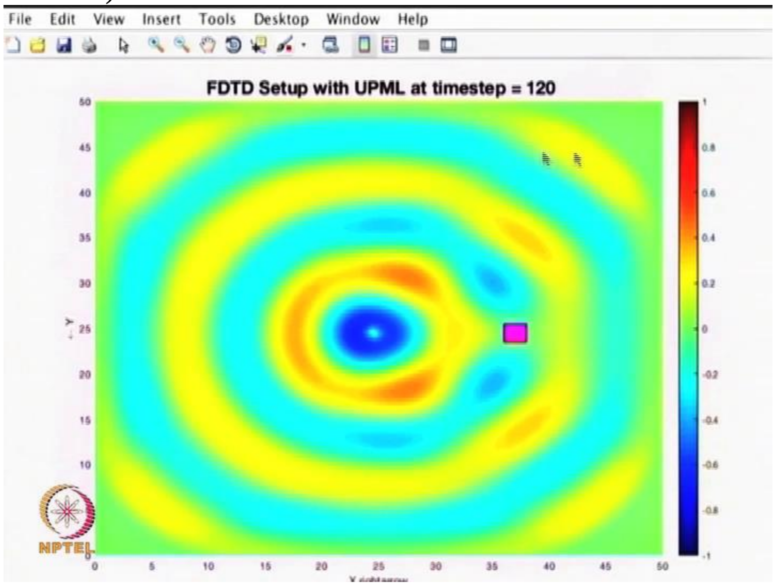
(Refer Slide Time: 19: 10)


Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FDM/UPML_FDTD_TM_BoxScatterer.m*
UPML_FDTD_TM_BoxScatterer.m*
FDTD_2D_WG_3dB_Splitter.m
Pointsource_in_2media.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script
Ln 140 Col 19

```

% % Defining a scatterer
for i = 73:74
    for j = 48:52
      Ez(i,j) = 0;
    end
end

% Calculating CHx
    for i=1:nx
        for j=1:ny-1
           Hx(i,j)=Hx(i,j)-(dt/mu(i,j))*((Ez(i,j+1)-Ez(i,j))/c
              + (((dt * sigx(i,j))/(eps(i,j)*mu(i,j)))*Kx(i,j)
```

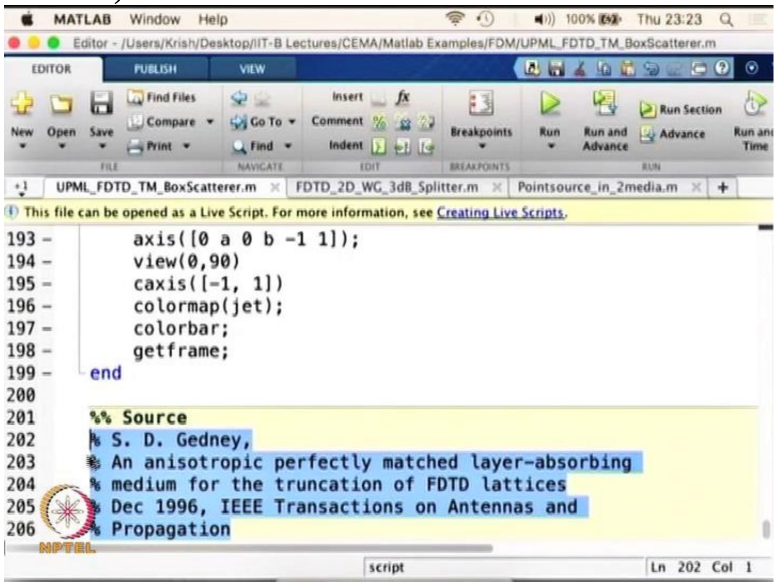
So now we will go back to the code and we will define it as a scatterer by uncommenting this particular. And run the same code and see what we can simulate using this particular situation.

(Refer Slide Time: 19: 30)

3


Now it is a perfect electric conductor so it is going to reflect everything that is impinging on it so you can see it cuts the incoming wave the incoming wave is getting bifurcated using this particular scatterer and you can see that the XML is absorbing whatever is incoming the y PML is absorbing and the complex domain is also absorbing and we have PEC on all other sides.

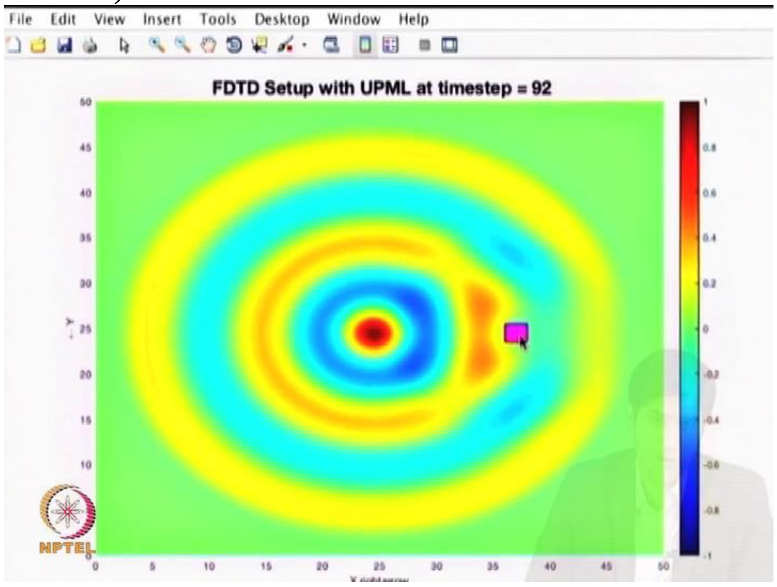
(Refer Slide Time: 19: 59)



```
    axis([0 a 0 b -1 1]);
    view(0,90)
    caxis([-1, 1])
    colormap(jet);
    colorbar;
    getframe;
end

%% Source
% S. D. Gedney,
% An anisotropic perfectly matched layer-absorbing
% medium for the truncation of FDTD lattices
% Dec 1996, IEEE Transactions on Antennas and
% Propagation
```

So we can test such a problem for various applications so what I would recommend is you take this particular code practice it for yourself simulating it for various problems and as I said before the source of this particular code comes from the theoretical discussions that has been presented in a phenomenal paper written by Gedney an anisotropic perfectly matched layer absorbing medium for truncation finite difference time domain lattices it was published in 1996 in the IEEE transactions on Antennas and propagation.

(Refer Slide Time: 20: 37)



So with this what we have done is we have showcased a very important point that one needs to learn if one says he is doing computational electromagnetic sir computational modelling which is nothing but the perfectly matched layer domain truncations we will be using this over and over again in various methods that we are going to apply or various methods that we are going to learn so it's important that you get a good grip on this particular method and that to using finite difference method later on we will do finite element finite volume so on and so forth.

We will come back to it again and again and even in the method of finite volumes we will discuss much in detail about the implementation of such methods but for now I think we have covered the most important aspect that I promised that I will cover I request you to take this code and practice it for yourself see it for yourself how you can model it and also I encourage you to write your own code.

So don’t take this code directly of course you wanted to give you the code but we are also afraid of giving quotes because sometimes people take the code and simulate it is just a black box for them in some cases what we will do it we will genuinely avoid giving codes so as to make you learn and make you program so that’s intention if you are feeling encoding that means you are not learn computational electromagnetics entirely.

So we would like you to learn to quote so in that sense sometimes we ask you genuinely to quote this particular exercise is an excellent example for you to practice coding.

So I encourage you to do that and that being said we have come to the end of this interesting long module so will stop here and you will look at some of the applications of the PML in the following lectures as well if you have any specific questions please post your questions on the forum we will be very happy to answer those questions and help you understand the perfectly match layer technique thank you!