APPENDIX A

Perfectly Matched Layer (PML)

The PML region depicted in Fig. A.1 comprises three PMLs: $R_{up}:\{|x|\le a, c<z\le c+d_1\}$ of dimensions $2a\times d_1$ immediately above the computational domain R, $R_{dn}:\{|x|\le a, -b-d_1\le z<-b\}$ of dimensions $2a\times d_1$ immediately below R, and $R_{rt}:\{a<x\le a+d_2, -b-d_1\le z\le c+d_1\}$ of dimensions $d_2\times(b+c+2d_1)$ immediately to the right of R, where $d_1=N_z\Delta z$ and $d_2=N_x\Delta x$.

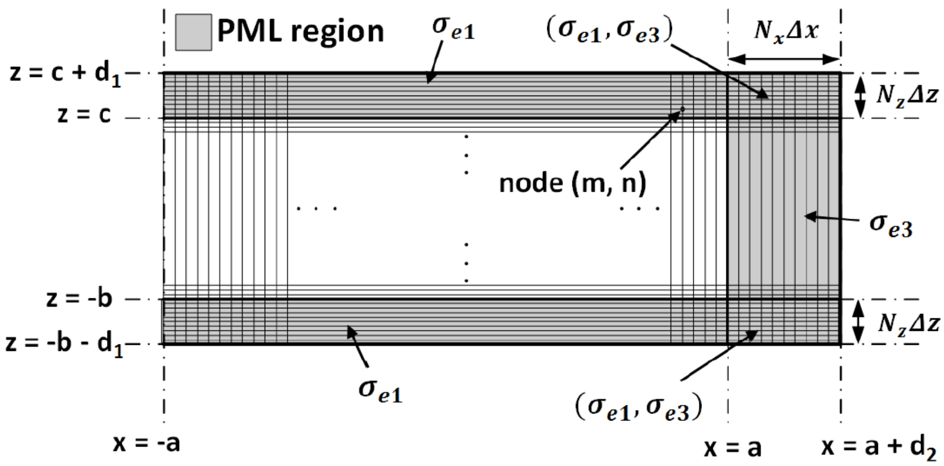

Figure A.1: Schematic of the PML region surrounding R.

The conductivity in R_{up} was prescribed as [Elsherbeni and Demir (2016), Berenger (1994)]

$$\sigma(z)=-\frac{(N_o+1)\ln R_0}{2\eta_0 d_1}\left(\frac{|z-c|}{d_1}\right)^{N_o},\quad |x|\le a,\quad z>c, \tag{A.1}$$

in R_{dn} as

$$\sigma(z) = -\frac{\left(N_o + 1\right)\ln R_0}{2\eta_0\, d_1}\left(\frac{|\, z + b\,|}{d_1}\right)^{N_o}, \quad |\, x\,| \le a, \quad z < -b, \tag{A.2}$$

and in R_{rt} as

$$\sigma(x) = -\frac{\left(N_o + 1\right)\ln R_0}{2\eta_0\, d_2}\left(\frac{|\, x - a\,|}{d_2}\right)^{N_o}, \quad x > a, \quad -b \le z \le c, \tag{A.3}$$

where $R_0 \ll 1$ for satisfactory performance. We chose $R_0 = 10^{-8}$, $N_o = 2$, and $N_x = N_z = 10$ were chosen for all calculations reported in this thesis. The conductivity in the PML region was thus set independent of time but not of space.

FDTD updating equations are replaced in the PML region by

$$E_x^{p+1}(m,n) = C_1(m,n)E_x^p(m,n) + C_2(m,n)\left[H_y^{p+\frac{1}{2}}(m,n) - H_y^{p+\frac{1}{2}}(m,n-1)\right], \tag{A.4}$$

$$E_z^{p+1}(m,n) = C_3(m,n)E_z^p(m,n) + C_4(m,n)\left[H_y^{p+\frac{1}{2}}(m,n) - H_y^{p+\frac{1}{2}}(m-1,n)\right], \tag{A.5}$$

$$H_y^{p+\frac{1}{2}} = H_{yx}^{p+\frac{1}{2}} + H_{yz}^{p+\frac{1}{2}}, \tag{A.6}$$

$$H_{yx}^{p+\frac{1}{2}}(m,n) = C_5(m,n)H_{yx}^{p-\frac{1}{2}}(m,n) + C_6(m,n)\left[E_z^p(m+1,n) - E_z^p(m,n)\right], \tag{A.7}$$

and

$$H_{yz}^{p+\frac{1}{2}}(m,n) = C_7(m,n)H_{yz}^{p-\frac{1}{2}}(m,n) + C_8(m,n)\left[E_x^p(m,n+1) - E_x^p(m,n)\right]. \tag{A.8}$$

Here,

$$C_\ell(m,n) = \frac{2\varepsilon_0 - \sigma_{e\ell}(m,n)\Delta t}{2\varepsilon_0 + \sigma_{e\ell}(m,n)\Delta t}, \quad \ell \in \{1,3\}, \tag{A.9}$$

$$C_2(m,n) = -\frac{2\Delta t}{\left[2\varepsilon_0 + \sigma_{e2}(m,n)\Delta t\right]\Delta z}, \tag{A.10}$$

$$C_4(m,n)=\frac{2\Delta t}{\left[2\varepsilon_0+\sigma_{e4}(m,n)\Delta t\right]\Delta x}, \tag{A.11}$$

$$C_\ell(m,n)=\frac{2\mu_0-\sigma_{m\ell}(m,n)\Delta t}{2\mu_0+\sigma_{m\ell}(m,n)\Delta t}, \quad \ell\in\{5,7\}, \tag{A.12}$$

$$C_6(m,n)=\frac{2\Delta t}{\left[2\mu_0+\sigma_{m6}(m,n)\Delta t\right]\Delta x}, \tag{A.13}$$

and

$$C_8(m,n)=-\frac{2\Delta t}{\left[2\mu_0+\sigma_{m8}(m,n)\Delta t\right]\Delta z}, \tag{A.14}$$

where

$$\left.\begin{aligned}
&\sigma_{e2}(m,n)=\sigma_{e1}(m,n)\\
&\sigma_{e4}(m,n)=\sigma_{e3}(m,n)\\
&\sigma_{m5}(m,n)=\sigma_{m6}(m,n)=\eta_0^2\,\sigma_{e3}(m,n)\\
&\sigma_{m7}(m,n)=\sigma_{m8}(m,n)=\eta_0^2\,\sigma_{e1}(m,n)
\end{aligned}\right\}. \tag{A.15}$$

The conductivity $\sigma_{e1}(m,n)$ is determined in (i) R_{up} using Eq. (A.1) and (ii) R_{dn} using Eq. (A.2). The conductivity $\sigma_{e3}(m,n)$ is determined in R_{rt} using Eq. (A.3). Both $\sigma_{e1}(m,n)$ and $\sigma_{e3}(m,n)$ are used in $R_{up}\cap R_{rt}$ and $R_{dn}\cap R_{rt}$ as appropriate.

The field components that need to be updated in R_{up} and R_{dn} are E_x and H_y whereas $E_z\equiv 0$ in those two regions. Similarly, E_z and H_y have to be updated in R_{rt} but $E_x\equiv 0$ in that region.

APPENDIX B

MATALAB Programs

B.1 The following MATLAB programs can be used to obtain the numerical results of Chapters 2 and 3.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%        main_program      %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all
close all
define_parameters;
define_geometry;
define_initial_value;
initialize_pml_boundary;
initialize_permittivity;
run_fdtd_loop;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %              define_parameters              %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c = 3e8;                      % speed of light
mu_0 = 4*pi*1e-7;             % permeability of the free space
eps_0 = 8.854e-12;            % permittivity of the free space
n0 = sqrt (mu_0/eps_0);       % intrinsic impedance of the free space
lambda_c = 600e-9;            % chosen carrier wavelength
wc = 2*pi*c/lambda_c;
wp = 1.352e16;                % plasma angular frequency of silver
tau = 17e-15;                 % relaxation time

eps_A = 1;                                % for air (material A)
eps_B = 1 - (wp^2/(wc*(wc+(1i/tau))));    % Drude model for silver (Material B)
nA = sqrt(eps_A);                         % complex refractive index of material A
nB = sqrt(eps_B);                         % complex refractive index of material B
kc = wc*sqrt(eps_0*mu_0);                 % free-space wavenumber

q = wc*sqrt(eps_0*mu_0)*sqrt(eps_A*eps_B/(eps_A + eps_B)); % SPP wavenumber
```

```
alpha_An = sqrt(wc^2*eps_0*mu_0*eps_A - q^2);
alpha_A = - alpha_An;
alpha_B = sqrt(wc^2*eps_0*mu_0*eps_B - q^2);

del_prop = 1/imag(q);
vp = wc/real(q);
del_A = 1/imag(alpha_A);
del_B = 1/imag(alpha_B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%          define_geometry          %%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

number_of_time_steps =12000;

n_pml_xn = 0;   % number of cells for the thickness of the PML regions in negative x-direction
n_pml_xp = 10;   % number of cells for the thickness of the PML regions in positive x-direction
n_pml_zn = 10;   % number of cells for the thickness of the PML regions in negative z-direction
n_pml_zp = 10;   % number of cells for the thickness of the PML regions in postive z-direction

pml_order = 2;
R_0 = 1e-8;

a = 30000e-9;      % setting x-direction dimension
min_x = -a;
max_x = a;
min_z = -4*del_B; max_z = 2*del_A;      % setting z-direction dimension

dx = 48.34e-9; dz = 7e-9;

domain_min_x = min_x - dx*(n_pml_xn);      % setting domain size
domain_max_x = max_x + dx*n_pml_xp;
domain_min_z = min_z - dz*n_pml_zn;
domain_max_z = max_z + dz*n_pml_zp;

domain_size_x = domain_max_x - domain_min_x;
domain_size_z = domain_max_z - domain_min_z;

nx = round (domain_size_x/dx);          % number of cells in x-direction
nz = round (domain_size_z/dz);          % number of cells in y-direction

pis = n_pml_xn+1;                  % determine the boundaries of the non-PML region
pie = nx-n_pml_xp+1;
pjs = n_pml_zn+1;
pje = nz-n_pml_zp+1;
pjm = round (abs(min_z/dz))+pjs;
```

```
for i = 1:nx
    x_mu_m(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nx+1
    x_mu_m1(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nz
    z_mu_m(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end
for i = 1:nz+1
    z_mu_m1(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end

courant_factor = 0.95;
dt = 1/(c*sqrt((1/dx^2)+(1/dz^2)));
dt = courant_factor*dt;

time      = dt*[0:number_of_time_steps-1].';

Gt = wc.*time.*exp(-wc.*time);                    % pulse function
exp_term = exp(-1i.*wc.*time);
Ht = ((1+1i)/2)*Gt + (exp (-wc*time))/2;          % pulse function

Cexhy = (-dt/(eps_0*dz)).*ones(nx,nz+1);      % coefficients for fields update
Cezhy = (dt/(eps_0*dx)).*ones(nx+1,nz);
Chyez = (dt/(mu_0*dx)).*ones(nx,nz);
Chyex = (-dt/(mu_0*dz)).*ones(nx,nz);

Hy = zeros(nx,nz);
Ex = zeros(nx,nz+1);
Ez = zeros(nx+1,nz);

H_field = figure;
nImages = length(time);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% %                 define_intial_value               %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

z =0; x = -a;

for i = 1:(pje-pjm)
    EA_zi (1,i) = (q/(kc*nA^2))*exp(1i*q*x)*exp(1i*alpha_A*z);
    HA_yi (1,i) = (-1/n0)*exp(1i*q*x)*exp(1i*alpha_A*z);
    z = z+dz;
end
```

```
z = 0 ; x = -a;
for i = 1:((pje-pjm)+1)
    EA_xi(1,i) = (-alpha_A/(kc*nA^2))*exp(1i*q*x)*exp(1i*alpha_A*z);
     z = z+dz;
end

z = min_z; x = -a;

for i = 1:(pjm-pjs)
    EB_xi(1,i) = (1/nB)*(alpha_B/(kc*nB))*exp(1i*q*x)*exp(-1i*alpha_B*z);
    EB_zi (1,i) = (1/nB)*(q/(kc*nB))*exp(1i*q*x)*exp(-1i*alpha_B*z);
    HB_yi (1,i) = (-1/n0)*exp(1i*q*x)*exp(-1i*alpha_B*z);
    z = z+dz;
end

Exi (1:(pjm-pjs),1) = EB_xi;
Exi ((pjm-pjs)+1:(pje-pjs)+1,1) = EA_xi;
Ezi (1:(pjm-pjs),1) = EB_zi;
Ezi ((pjm-pjs)+1:(pje-pjs),1) = EA_zi;
Hyi (1:(pjm-pjs),1) = HB_yi;
Hyi ((pjm-pjs)+1:(pje-pjs),1) = HA_yi;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                    initialize_pml_boundary                    %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hyx_xn = zeros(n_pml_xn,nz);
Hyz_xn = zeros(n_pml_xn,nz-n_pml_zn-n_pml_zp);
Hyx_xp = zeros(n_pml_xp,nz);
Hyz_xp = zeros(n_pml_xp,nz-n_pml_zn-n_pml_zp);
Hyx_zn = zeros(nx-n_pml_xn-n_pml_xp, n_pml_zn);
Hyz_zn = zeros(nx,n_pml_zn);
Hyx_zp = zeros(nx-n_pml_xn-n_pml_xp, n_pml_zp);
Hyz_zp = zeros(nx,n_pml_zp);

is_pml_xp  = true;
is_pml_zn = true;
is_pml_zp = true;

if is_pml_xp
    sigma_pex_xp = zeros(n_pml_xp,nz);
    sigma_pmx_xp = zeros(n_pml_xp,nz);
    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dx*n_pml_xp);
    rho_e = ([1:n_pml_xp] - 0.75)/n_pml_xp;
    rho_m = ([1:n_pml_xp] - 0.25)/n_pml_xp;
    for ind = 1:n_pml_xp
          sigma_pex_xp(ind,:) = sigma_max * rho_e(ind)^pml_order;
```

```
        sigma_pmx_xp(ind,:) = ...
                (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coefficients updating Ey (Ez)
    Ceye_xp = (2*eps_0 - dt*sigma_pex_xp)./(2*eps_0+dt*sigma_pex_xp);
    Ceyhz_xp= (2*dt/dx)./(2*eps_0 + dt*sigma_pex_xp);

    % Coefficients updating Hyx
    Chzxh_xp  = (2*mu_0 - dt*sigma_pmx_xp)./(2*mu_0+dt*sigma_pmx_xp);
    Chzxey_xp = (2*dt/dx)./(2*mu_0 + dt*sigma_pmx_xp);

    % Coefficients updating Hyz
    Chzyh_xp  =  1;
    Chzyex_xp =  -dt/(dz*mu_0);
end

if is_pml_zn
    sigma_pey_yn = zeros(nx,n_pml_zn);
        sigma_pmy_yn = zeros(nx,n_pml_zn);
        sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dz*n_pml_zn);
        rho_e = ([n_pml_zn:-1:1] - 0.75)/n_pml_zn;
        rho_m = ([n_pml_zn:-1:1] - 0.25)/n_pml_zn;
        for ind = 1:n_pml_zn
            sigma_pey_yn(:,ind) = sigma_max * rho_e(ind)^pml_order;
            sigma_pmy_yn(:,ind) = ...
                (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
        end

    % Coefficients updating Ex
    Cexe_yn  = (2*eps_0 - dt*sigma_pey_yn)./(2*eps_0+dt*sigma_pey_yn);
    Cexhz_yn = -(2*dt/dz)./(2*eps_0 + dt*sigma_pey_yn);

    % Coefficients updating Hzx (Hyx)
    Chzxh_yn  = 1;
    Chzxey_yn = dt/(dx*mu_0);

    % Coefficients updating Hzy (Hyz)
    Chzyh_yn  = (2*mu_0 - dt*sigma_pmy_yn)./(2*mu_0+dt*sigma_pmy_yn);
    Chzyex_yn = -(2*dt/dz)./(2*mu_0 + dt*sigma_pmy_yn);
end

if is_pml_zp
    sigma_pey_yp = zeros(nx,n_pml_zp);
    sigma_pmy_yp = zeros(nx,n_pml_zp);
    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dz*n_pml_zp);
    rho_e = ([1:n_pml_zp] - 0.75)/n_pml_zp;
    rho_m = ([1:n_pml_zp] - 0.25)/n_pml_zp;
    for ind = 1:n_pml_zp
       sigma_pey_yp(:,ind) = sigma_max * rho_e(ind)^pml_order;
```

```
        sigma_pmy_yp(:,ind) = ...
            (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coefficients updating Ex
    Cexe_yp  = (2*eps_0 - dt*sigma_pey_yp)./(2*eps_0+dt*sigma_pey_yp);
    Cexhz_yp = -(2*dt/dz)./(2*eps_0 + dt*sigma_pey_yp);

    % Coefficients updating Hzx
    Chzxh_yp  = 1;
    Chzxey_yp = dt/(dx*mu_0);

    % Coefficients updating Hzy
    Chzyh_yp  = (2*mu_0 - dt*sigma_pmy_yp)./(2*mu_0+dt*sigma_pmy_yp);
    Chzyex_yp = -(2*dt/dz)./(2*mu_0 + dt*sigma_pmy_yp);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          initialize_permittivity          %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_d = round ((lambda_c/dx)/2);
R_left    = round((a/dx)+(pis-1)-n_d);
R_right   = round(((a)/dx)+(pis-1)+n_d);
R_top = round (abs(min_z/dz)+n_pml_zn);
X_lt = wp.^2*tau.*(1-exp(-time./tau));

for i = 1:number_of_time_steps

    chai(i).X_l = zeros (nx,nz+1);
    chai(i).X_l(pis:R_left,1:R_top) = X_lt(i);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%          update_initial_field          %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hyit(:,time_step) = real (Ht(time_step).*Hyi.*exp_term(time_step));
Hy (pis,pjs:pje-1) = Hyit(:,time_step);
Exit(:,time_step) = Gt(time_step).*real (Exi.*exp_term(time_step));
Ex (pis,pjs:pje) = Exit(:,time_step);
Ezit(:,time_step) = Gt(time_step).*real (Ezi.*exp_term(time_step));
Ez (pis,pjs:pje-1) = Ezit(:,time_step);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%          update_pml_magnetic_field      %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nxp1 = nx+1; nzp1 = nz+1;

if is_pml_xp
   Hyx_xp = Chzxh_xp .* Hyx_xp ...
       + Chzxey_xp.*(Ez(pie+1:nxp1,:)-Ez(pie:nx,:));
   Hyz_xp = Chzyh_xp .* Hyz_xp ...
       + Chzyex_xp.*(Ex(pie:nx,pjs+1:pje)-Ex(pie:nx,pjs:pje-1));
end

if is_pml_zn
   Hyx_zn = Chzxh_yn .* Hyx_zn ...
       + Chzxey_yn.*(Ez(pis+1:pie,1:pjs-1)-Ez(pis:pie-1,1:pjs-1));
   Hyz_zn = Chzyh_yn .* Hyz_zn ...
       + Chzyex_yn.*(Ex(:,2:pjs)-Ex(:,1:pjs-1));
end
if is_pml_zp
   Hyx_zp = Chzxh_yp .* Hyx_zp ...
       + Chzxey_yp.*(Ez(pis+1:pie,pje:nz)-Ez(pis:pie-1,pje:nz));
   Hyz_zp = Chzyh_yp .* Hyz_zp ...
       + Chzyex_yp.*(Ex(:,pje+1:nzp1)-Ex(:,pje:nz));
end

Hy(1:pis-1,1:pjs-1) = Hyx_xn(:,1:pjs-1) + Hyz_zn(1:pis-1,:);
Hy(1:pis-1,pje:nz)  = Hyx_xn(:,pje:nz) + Hyz_zp(1:pis-1,:);
Hy(pie:nx,1:pjs-1)  = Hyx_xp(:,1:pjs-1) + Hyz_zn(pie:nx,:);
Hy(pie:nx,pje:nz)   = Hyx_xp(:,pje:nz) + Hyz_zp(pie:nx,:);
Hy(1:pis-1,pjs:pje-1) = Hyx_xn(:,pjs:pje-1) + Hyz_xn;
Hy(pie:nx,pjs:pje-1)  = Hyx_xp(:,pjs:pje-1) + Hyz_xp;
Hy(pis:pie-1,1:pjs-1) = Hyx_zn + Hyz_zn(pis:pie-1,:);
Hy(pis:pie-1,pje:nz)  = Hyx_zp + Hyz_zp(pis:pie-1,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%          Cex_function          %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cexn = Cex_function( time_step )

  global chai nz Electric_field nx
  Ex1 = 0;
  i = time_step;
   if  i > 2
       A = zeros (nx,nz+1);
```

```
        for m = 1:i-2
            A  =  A+  chai(m+1).X_l(:,:).*(Electric_field(i-m).Ex(:,:)-  Electric_field(i-m-1).Ex(:,:));
        end
        Cexn(1:nx,1:nz+1) = A + chai(i).X_l(:,:).*Ex1;
    else
        Cexn(1:nx,1:nz+1) =  chai(i).X_l(1:nx,1:nz+1).*Ex1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%          Cez_function              %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Cezn = Cez_function( time_step )

global chai nz Electric_field nx
  Ez1 = 0;
  i = time_step;
    if  i > 2
        A = zeros (nx,nz);
        for m = 1:i-2
            A   =   A+   chai(m+1).X_l(1:nx,1:nz).*(Electric_field(i-m).Ez(1:nx,1:nz)-Electric_field(i-m-1).Ez(1:nx,1:nz));
        end
        Cezn(1:nx,1:nz) = A + chai(i).X_l(1:nx,1:nz).*Ez1;
    else
        Cezn(1:nx,1:nz) =  chai(i).X_l(1:nx,1:nz).*Ez1;
    end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%         update_pml_electric_field        %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if is_pml_xp
    Ez(pie:nx,:) = Ceye_xp .* Ez(pie:nx,:) ...
        + Ceyhz_xp .* (Hy(pie:nx,:)-Hy(pie-1:nx-1,:));
end

if is_pml_zn
    Ex(:,2:pjs)   = Cexe_yn .* Ex(:,2:pjs) ...
        + Cexhz_yn .* (Hy(:,2:pjs)-Hy(:,1:pjs-1));
end
```

```
if is_pml_zp
   Ex(:,pje:nz) = Cexe_yp .* Ex(:,pje:nz) ...
       + Cexhz_yp .* (Hy(:,pje:nz)-Hy(:,pje-1:nz-1));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%                display_field                %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp([num2str(time_step) ' of ' ...
   num2str(number_of_time_steps) ' is completed.']) ;
figure(H_field);
imagesc(x_mu_m1,z_mu_m,Electric_field(time_step).Ez');
   axis xy;
   colorbar;
   line([x_mu_m1(R_left) x_mu_m1(R_left)], [z_mu_m(1) z_mu_m(R_top+1)], 'Color', 'm');
   line([x_mu_m1(1)  x_mu_m1(R_left)],  [z_mu_m(R_top+1)  z_mu_m(R_top+1)], 'Color', 'm');
   line([x_mu_m1(R_right)  x_mu_m1(R_right)],  [z_mu_m(1)  z_mu_m(R_top+1)], 'Color', 'm');
   line([x_mu_m1(R_right)  x_mu_m1(nx)],  [z_mu_m(R_top+1)  z_mu_m(R_top+1)], 'Color', 'm');
   title('Electric field  (z) ');
   drawnow;
   frame(time_step) = getframe(gcf);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %                 run_fdtd_loop                %%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for time_step = 1:number_of_time_steps

   update_initial_field;
   Hy(pis:pie-1,pjs:pje-1) =   Hy(pis:pie-1,pjs:pje-1) ...
       + (Chyex(pis:pie-1,pjs:pje-1).*(Ex(pis:pie-1,pjs+1:pje) - Ex(pis:pie-1,pjs:pje-1))) ...
       + (Chyez(pis:pie-1,pjs:pje-1).*(Ez(pis+1:pie,pjs:pje-1)-Ez(pis:pie-1,pjs:pje-1)));
   update_pml_magnetic_field;
   Magnetic_field (time_step).Hy = Hy;

   Hix(time_step).ex(1:nx,1:nz+1) = Cex_function(time_step);
   Ex (:,pjs+1:pje-1) = Ex (:,pjs+1:pje-1) ...
       + (Cexhy(:,pjs+1:pje-1).*(Hy(:,pjs+1:pje-1)-Hy(:,pjs:pje-2))) ...
       - (dt.*Hix(time_step).ex(:,pjs+1:pje-1));
```

```
    Hiz(time_step).ez(1:nx,1:nz) = Cez_function(time_step);
    Ez (pis+1:pie-1,:) = Ez (pis+1:pie-1,:) ...
        + (Cezhy(pis+1:pie-1,:).*(Hy(pis+1:pie-1,:)-Hy(pis:pie-2,:))) ...
        - (dt.*Hiz(time_step).ez(pis+1:pie-1,:));
    update_pml_electric_field;

    Electric_field(time_step).Ez(1:nx+1,1:nz) = Ez;
    Electric_field(time_step).Ex(1:nx,1:nz+1) = Ex;
    display_fields;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %            Pearson_correlation          %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


close all
clear Power;
clear ds dR x_point1 x_point2
clear Xa padding xcoor PCC

z_point1 = R_top +1;
dR = round ((30*lambda_c)/dx);
dS = round ((40*lambda_c)/dx);

x_point1 = R_left - dR;

for ind  = 1: length (dS)

      x_point2 = R_right + dS(ind);

for i = 1:number_of_time_steps

    Power(ind).xA (i) = -Electric_field(i).Ez(x_point1,z_point1).* Magnetic_field(i).Hy(x_point1,z_point1);
    Power(ind).xB (i) = -Electric_field(i).Ez(x_point2,z_point1).* Magnetic_field(i).Hy(x_point2,z_point1);
end

[Xa(ind).x,Xa(ind).y,Xa(ind).d] = alignsignals(Power(ind).xA,Power(ind).xB);
padding(ind).x = zeros(1,Xa(ind).d) ;  % a zero row vector
Xa(ind).z = [Xa(ind).y , padding(ind).x] ;
xcoor(ind) = (26*lambda_c + ind*lambda_c)/lambda_c;
PCC(ind) = corr2(Xa(ind).x,Xa(ind).z);
```

```
end

plot(xcoor,PCC); figure;
plot(time*10^15,Power(1).xA);
figure;
plot(time*10^15,Power(1).xB);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %            time_to_frequency_domain            %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X] = time_to_frequency_domain(x, dt, frequency_array, time_shift)

number_of_time_steps  = size(x,2);
number_of_frequencies = size(frequency_array,2);
X = zeros(1, number_of_frequencies);
w = 2 * pi * frequency_array;
for n = 1:number_of_time_steps
   t = n * dt + time_shift;
   X = X + x(n) * exp(-1i*w*t);
end
X = X * dt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

B.1.1 To obtain the results of launched signal in Chapter 2, use the following sub-program for "**initialize_permittivity**"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%             initialize_permittivity             %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_d = round ((lambda_c/dx)/2);
R_left    = round((a/dx)+(pis-1)-n_d);
R_right   = round(((a)/dx)+(pis-1)+n_d);
R_top = round (abs(min_z/dz)+n_pml_zn);
X_lt = wp.^2*tau.*(1-exp(-time./tau));
for i = 1:number_of_time_steps
   chai(i).X_l = zeros (nx,nz+1);
   chai(i).X_l(pis:R_left,1:R_top) = X_lt(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

B.1.2 To obtained the results of transmission across a gap in Chapter 2, use the following sub-program for "**initialize_permittivity**"

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          initialize_permittivity          %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_d = round ((lambda_c/dx)/2);
R_left    = round((a/dx)+(pis-1)-n_d);
R_right   = round(((a)/dx)+(pis-1)+n_d);
R_top = round (abs(min_z/dz)+n_pml_zn);
X_lt = wp.^2*tau.*(1-exp(-time./tau));
for i = 1:number_of_time_steps
    chai(i).X_l = zeros (nx,nz+1);
    chai(i).X_l(pis:R_left,1:R_top) = X_lt(i);
    chai(i).X_l((R_right:nx),1:R_top) = X_lt(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

B.1.3 To obtain the results of Chapter 3, use the following sub-program.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%          define_geometry          %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha = 135;
number_of_time_steps =3600;

n_pml_xn = 0;
n_pml_xp = 10;
n_pml_zn = 10;
n_pml_zp = 10;

pml_order = 2;
R_0 = 1e-8;

a = 5000e-9;
b = 5000e-9;
min_x = -a;
max_x = a;
min_z = -b; max_z = 2*del_A;
dx = 10e-9; dz = 10e-9;

domain_min_x = min_x - dx*(n_pml_xn);
domain_max_x = max_x + dx*n_pml_xp;
domain_min_z = min_z - dz*n_pml_zn;
domain_max_z = max_z + dz*n_pml_zp;
```

```
domain_size_x = domain_max_x - domain_min_x;
domain_size_z = domain_max_z - domain_min_z;

nx = round (domain_size_x/dx);
nz = round (domain_size_z/dz);

pis = n_pml_xn+1;
pie = nx-n_pml_xp+1;
pjs = n_pml_zn+1;
pje = nz-n_pml_zp+1;
pjm = round (abs(min_z/dz))+pjs;

for i = 1:nx
   x_mu_m(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nx+1
   x_mu_m1(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nz
   z_mu_m(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end
for i = 1:nz+1
   z_mu_m1(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end

courant_factor = 0.95;
dt = 1/(c*sqrt((1/dx^2)+(1/dz^2)));
dt = courant_factor*dt;

time      = dt*[0:number_of_time_steps-1].';

Gt = wc.*time.*exp(-wc.*time);
exp_term = exp(-1i.*wc.*time);
Ht = ((1+1i)/2)*Gt + (exp (-wc*time))/2;

Cexhy = (-dt/(eps_0*dz)).*ones(nx,nz+1);
Cezhy = (dt/(eps_0*dx)).*ones(nx+1,nz);
Chyez = (dt/(mu_0*dx)).*ones(nx,nz);
Chyex = (-dt/(mu_0*dz)).*ones(nx,nz);

Hy = zeros(nx,nz);
Ex = zeros(nx,nz+1);
Ez = zeros(nx+1,nz);

E_field = figure;
nImages = length(time);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%           initialize_permittiviy          %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if alpha >= 90

    X_lt = wp.^2*tau.*(1-exp(-time./tau));
     theeta = alpha - 90;
      x2 = (b)*tand (theeta);
     nx2 = round (x2/dx);
     R_left    = round((a/dx)+(pis-1));
     R_slide = R_left + nx2;
    R_top = round ((b/dz)+n_pml_zn);
    sy = [R_left; R_left; R_slide];
    sx = [0; R_top; 0];
    SR = poly2mask(sx,sy,nx,nz+1);
   for i = 1:number_of_time_steps

    SR1(i).X_l = SR*X_lt(i);
    SR2(i).X_l = zeros (nx,nz+1);
    SR2(i).X_l(pis:R_left,1:R_top) = X_lt(i);
    chai(i).X_l = SR1(i).X_l + SR2(i).X_l;
end

else

    X_lt = wp.^2*tau.*(1-exp(-time./tau));
    theeta = 90 - alpha;
    x2 = (b)*tand (theeta);
    nx2 = round (x2/dx);
   R_left    = round((a/dx)+(pis-1));
  R_slide = R_left - nx2;
  R_top = round ((b/dz)+n_pml_zn);
   sy = [R_left; R_left; R_slide];
   sx = [0; R_top; 0];
   SR = poly2mask(sx,sy,nx,nz+1);
  for i = 1:number_of_time_steps

    SR1(i).X_l = SR*X_lt(i);
    SR2(i).X_l = zeros (nx,nz+1);
    SR2(i).X_l(pis:R_left,1:R_top) = X_lt(i);
    chai(i).X_l = - SR1(i).X_l + SR2(i).X_l;
end

end

if 0 < R_slide && R_slide < nx
        R_right  = R_slide;
        R_bottom = 1;
```

```
    else if R_slide <= 0
          R_right = 1;
          R_bottom1 = (a + (n_pml_xp*dx))/ tand (theeta);
          R_bottom = R_top +1 - round (R_bottom1/dz);
       else
       R_right = nx;
       R_bottom1 = (a + (n_pml_xp*dx))/ tand (theeta);
       R_bottom = R_top +1 - round (R_bottom1/dz);
       end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%              display_field              %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(E_field);

   imagesc(x_mu_m1,z_mu_m,Electric_field(time_step).Ez');
   axis xy;
   colorbar;
   line([x_mu_m1(1)    x_mu_m1(R_left)],[z_mu_m(R_top+1)    z_mu_m(R_top+1)],
'Color', 'm');
   line([x_mu_m1(R_left)          x_mu_m1(R_right)],          [z_mu_m(R_top+1)
z_mu_m(R_bottom)], 'Color','m');
   title('Electric field  (z) ');
   drawnow;
   frame(time_step) = getframe(gcf);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%          Pearson_correlation          %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear Power;
clear ds dR x_point1 x_point2
clear Xa padding xcoor PCC
close all

dR = round ((8*lambda_c)/dx);
dS = 1.*lambda_c;
dx1 = dS.*sind (theeta);
dz1 = dS.*cosd (theeta);
```

```
ndx1 = round (dx1./dx);
ndz1 = round (dz1./dz);

z_point1 = R_top +1;
x_point1 = R_left - dR;

for ind  = 1: length (dS)
  z_point2 = R_top - ndz1(ind)
  x_point2 = R_left + ndx1(ind)
for i = 1:number_of_time_steps
    Power(ind).xA (i) = -Electric_field(i).Ez(x_point1,z_point1).* Magnetic_field(i).Hy(x_point1,z_point1);
    Power(ind).B1 (i) = -Electric_field(i).Ez(x_point2,z_point2).* Magnetic_field(i).Hy(x_point2,z_point2);
    Power(ind).B2 (i) = -Electric_field(i).Ex(x_point2,z_point2).* Magnetic_field(i).Hy(x_point2,z_point2);
    Power(ind).xB (i) = Power (ind).B1(i)*sind (theeta) + Power (ind).B2(i)*cosd (theeta);

end

[Xa(ind).x,Xa(ind).y,Xa(ind).d] = alignsignals(Power(ind).xA,Power(ind).xB);
padding(ind).x = zeros(1,Xa(ind).d) ;  % a zero row vector
Xa(ind).z = [Xa(ind).y , padding(ind).x] ;
xcoor(ind) = ind - 1;

 PCC(ind) = corr2(Xa(ind).x,Xa(ind).z)
end
plot(time*10^15,Power(1).xA);
figure
plot(time*10^15,Power(1).xA./Norm_value);
figure
plot(time*10^15,Power(1).xA./Sum_Power);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

B.2 The following MATLAB program can be used to obtain the numerical results of Chapter 4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%          main_program          %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

define_parameters;
define_geometry;
define_initial_value;
```

```
initialize_pml_boundary;
Initialize_permittivity;
run_fdtd_loop;
save elec.mat Electric_field -v7.3;
save magn.mat Magnetic_field -v7.3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%          define_parameters          %%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c = 3e8;
mu_0 = 4*pi*1e-7;
eps_0 = 8.854e-12;
n0 = sqrt (mu_0/eps_0);
lambda_c = 1200e-9;
wc = 2*pi*c./lambda_c;

%%%%%%%%%%%%%%%%%%%% defining material A (Silicon) %%%%%%%%%%

eps_inf = 1;
del_eps1 = 8.93; del_eps2 = 1.855;
w1 = 3.42*(2*pi*3e14); %% in rad/sec
w2 = 2.72*(2*pi*3e14);
Gamma1 = 0.425*(2*pi*3e14);        Gamma2 = 0.123*(2*pi*3e14);
gamma1 = 0.087*(2*pi*3e14);        gamma2 = 2.678*(2*pi*3e14);

eps_Si = eps_inf + (del_eps1.*(w1^2 - (1i*gamma1.*wc))./(w1^2 - (1i*2.*wc*Gamma1)
- wc.^2)) ...
    + (del_eps2.*(w2^2 - (1i*gamma2.*wc))./(w2^2 - (1i*2.*wc*Gamma2) - wc.^2));

alpha1 = sqrt (w1^2-Gamma1^2);
alpha2 = sqrt (w2^2-Gamma2^2);
beta1 = (w1^2-(gamma1.*Gamma1))./alpha1;
beta2 = (w2^2-(gamma2.*Gamma2))./alpha2;

%%%%%%%%%  Defining material B (Bulk Silver) %%%%%%%%%%%%

wp = 1.352e16;
tau = 17e-15;
eps_Ag = 1 - (wp^2./(wc.*(wc+(1i/tau))));

eps_A = eps_Si;
eps_B = eps_Ag;
nA = sqrt(eps_A);
nB = sqrt(eps_B);
kc = wc*sqrt(eps_0*mu_0);
q = wc.*sqrt(eps_0.*mu_0).*sqrt(eps_A.*eps_B./(eps_A + eps_B));
```

```
alpha_An = sqrt(wc.^2.*eps_0.*mu_0.*eps_A - q.^2);
alpha_A = - alpha_An;
alpha_B = sqrt(wc.^2.*eps_0.*mu_0.*eps_B - q.^2);

del_prop = 1./imag(q);
vp = wc./real(q);
del_A = 1./imag(alpha_A);
del_B = 1./imag(alpha_B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%            define_geometry            %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

number_of_time_steps = 12000;

n_pml_xn = 0;
n_pml_xp = 10;
n_pml_zn = 0;
n_pml_zp = 0;

pml_order = 2;
R_0 = 1e-8;

a = 5000e-9;   % should be less than del_prop/2
min_x = -a;
max_x = 2*a;
min_z = -10*del_B; max_z = 8*del_A;

dx = 25.000e-09;
dz = 5.7605e-09;

domain_min_x = min_x - dx*(n_pml_xn);
domain_max_x = max_x + dx*n_pml_xp;
domain_min_z = min_z - dz*n_pml_zn;
domain_max_z = max_z + dz*n_pml_zp;

domain_size_x = domain_max_x - domain_min_x;
domain_size_z = domain_max_z - domain_min_z;

nx = round (domain_size_x/dx);
nz = round (domain_size_z/dz);
pis = n_pml_xn+1;
pie = nx-n_pml_xp+1;
pjs = n_pml_zn+1;
pje = nz-n_pml_zp+1;
pjm = round (abs(min_z/dz))+pjs;
```

```
for i = 1:nx
    x_mu_m(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nx+1
    x_mu_m1(1,i) = ((domain_min_x-dx) + (i*dx)).*10^6;
end
for i = 1:nz
    z_mu_m(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end
for i = 1:nz+1
    z_mu_m1(1,i) = ((domain_min_z-dz) + (i*dz)).*10^6;
end

courant_factor = 0.90;
dt = 1/(c*sqrt((1/dx^2)+(1/dz^2)));
dt = courant_factor*dt;

time       = dt*[0:number_of_time_steps-1].';

Gt = wc.*time.*exp(-wc.*time);
exp_term = exp(-1i.*wc.*time);
Ht = ((1+1i)/2)*Gt + (exp (-wc*time))/2;

Cexhy = (-dt/(eps_0*dz)).*ones(nx,nz+1);
Cezhy = (dt/(eps_0*dx)).*ones(nx+1,nz);
Chyez = (dt/(mu_0*dx)).*ones(nx,nz);
Chyex = (-dt/(mu_0*dz)).*ones(nx,nz);

Hy = zeros(nx,nz);
Ex = zeros(nx,nz+1);
Ez = zeros(nx+1,nz);

E_field = figure;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%              define_initial_value          %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

z =0; x = -a;

for i = 1:(nz-pjm)
    EA_zi (1,i) = (q/(kc*nA^2))*exp(1i*q*x)*exp(1i*alpha_A*z);
    HA_yi (1,i) = (-1/n0)*exp(1i*q*x)*exp(1i*alpha_A*z);

    z = z+dz;
end
z = 0 ; x = -a;
```

```
for i = 1:(nz-pjm+1)
    EA_xi(1,i) = (-alpha_A/(kc*nA^2))*exp(1i*q*x)*exp(1i*alpha_A*z);
     z = z+dz;
end

z = min_z; x = -a;

for i = 1:pjm
    EB_xi(1,i) = (1/nB)*(alpha_B/(kc*nB))*exp(1i*q*x)*exp(-1i*alpha_B*z);
    EB_zi (1,i) = (1/nB)*(q/(kc*nB))*exp(1i*q*x)*exp(-1i*alpha_B*z);
    HB_yi (1,i) = (-1/n0)*exp(1i*q*x)*exp(-1i*alpha_B*z);
    z = z+dz;
end

Exi (1:pjm,1) = EB_xi;
Exi (pjm+1:nz+1,1) = EA_xi;
Ezi (1:pjm,1) = EB_zi;
Ezi ((pjm)+1:nz,1) = EA_zi;
Hyi (1:pjm,1) = HB_yi;
Hyi (pjm+1:nz,1) = HA_yi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                initialize_pml_boundary                %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hyx_xn = zeros(n_pml_xn,nz);
Hyz_xn = zeros(n_pml_xn,nz-n_pml_zn-n_pml_zp);
Hyx_xp = zeros(n_pml_xp,nz);
Hyz_xp = zeros(n_pml_xp,nz-n_pml_zn-n_pml_zp);
Hyx_zn = zeros(nx-n_pml_xn-n_pml_xp, n_pml_zn);
Hyz_zn = zeros(nx,n_pml_zn);
Hyx_zp = zeros(nx-n_pml_xn-n_pml_xp, n_pml_zp);
Hyz_zp = zeros(nx,n_pml_zp);

is_pml_xp  = true;

if is_pml_xp

    sigma_pex_xp = zeros(n_pml_xp,nz);
    sigma_pmx_xp = zeros(n_pml_xp,nz);

    sigma_max = -(pml_order+1)*eps_0*c*log(R_0)/(2*dx*n_pml_xp);
    rho_e = ([1:n_pml_xp] - 0.75)/n_pml_xp;
    rho_m = ([1:n_pml_xp] - 0.25)/n_pml_xp;
    for ind = 1:n_pml_xp
        sigma_pex_xp(ind,:) = sigma_max * rho_e(ind)^pml_order;
        sigma_pmx_xp(ind,:) = ...
```

```
        (mu_0/eps_0) * sigma_max * rho_m(ind)^pml_order;
    end

    % Coeffiecients updating Ey (Ez)
    Ceye_xp = (2*eps_0 - dt*sigma_pex_xp)./(2*eps_0+dt*sigma_pex_xp);
    Ceyhz_xp= (2*dt/dx)./(2*eps_0 + dt*sigma_pex_xp);

    % Coeffiecients updating Hzx (Hyx)
    Chzxh_xp  = (2*mu_0 - dt*sigma_pmx_xp)./(2*mu_0+dt*sigma_pmx_xp);
    Chzxey_xp = (2*dt/dx)./(2*mu_0 + dt*sigma_pmx_xp);

    % Coeffiecients updating Hzy (Hyz)
    Chzyh_xp  =  1;
    Chzyex_xp =  -dt/(dz*mu_0);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                initialize_permittivity               %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

R_left    = round(a/dx);
R_top = round (abs(min_z/dz));

X_Si        =       (del_eps1.*exp(-Gamma1.*time).*((beta1.*sin(alpha1.*time))       +
(gamma1.*cos(alpha1.*time)))) +...
    (del_eps2.*exp(-Gamma2.*time).*((beta2.*sin(alpha2.*time))                        +
(gamma2.*cos(alpha2.*time))));

X_Ag = wp.^2*tau.*(1-exp(-time./tau));

%%%  when Rc occupied by siliver and Rd occupied by silicon

for i = 1:number_of_time_steps

    chai(i).X_l = zeros (nx,nz+1);
    chai(i).X_l(1:nx,1:R_top) = X_Ag(i);
    chai(i).X_l(1:nx,R_top+1:nz+1) = X_Si(i);
end

%% when Rc occupied by silver and Rd occupied by silver

%for i = 1:number_of_time_steps

%    chai(i).X_l = zeros (nx,nz+1);
%    chai(i).X_l(1:nx,1:R_top) = X_Ag(i);
%    chai(i).X_l(R_left+1:nx,R_top+1:nz+1) = X_Ag(i);
%    chai(i).X_l(1:R_left,R_top+1:nz+1) = X_Si(i);
```

```
%end

%%% when Rc occupied by silicon and Rd occupied by silicon

%for i = 1:number_of_time_steps

    %chai(i).X_l = zeros (nx,nz+1);
    %chai(i).X_l(1:R_left,1:R_top) = X_Ag(i);
    %chai(i).X_l(R_left+1:nx,1:R_top) = X_Si(i);
    %chai(i).X_l(1:nx,R_top+1:nz+1) = X_Si(i);
%end

%%%  when Rc occupied by air and Rd occupied by air

%for i = 1:number_of_time_steps

    %chai(i).X_l = zeros (nx,nz+1);
    %chai(i).X_l(1:R_left,1:R_top) = X_Ag(i);
    %chai(i).X_l(1:R_left,R_top+1:nz+1) = X_Si(i);

%end

%%%  when Rc occupied by air and Rd occupied by silver

%for i = 1:number_of_time_steps

    %chai(i).X_l = zeros (nx,nz+1);
    %chai(i).X_l(1:nx,1:R_top) = X_Ag(i);
    %chai(i).X_l(1:R_left,R_top+1:nz+1) = X_Si(i);

%end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                    run_fdtd_loop           %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for time_step = 1:number_of_time_steps

    update_initial_field;
    update_magnetic_field;
    update_pml_magnetic_field;
    Magnetic_field (time_step).Hy = Hy;
    Hix(time_step).ex(1:nx,1:nz+1) = Cex_function(time_step);
    Hiz(time_step).ez(1:nx,1:nz) = Cez_function(time_step);
    update_electric_field;
```

```
    update_pml_electric_field;

    Electric_field(time_step).Ez(1:nx+1,1:nz) = Ez;
    Electric_field(time_step).Ex(1:nx,1:nz+1) = Ex;
    display_field;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%            update_initial_field          %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hyit(:,time_step) = real (Ht(time_step).*Hyi.*exp_term(time_step));
Hy (1,:) = Hyit(:,time_step);
Exit(:,time_step) = Gt(time_step).*real (Exi.*exp_term(time_step));
Ex (1,:) = Exit(:,time_step);
Ezit(:,time_step) = Gt(time_step).*real (Ezi.*exp_term(time_step));
Ez (1,:) = Ezit(:,time_step);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%            update_magnetic_field          %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Hy(pis:pie-1,pjs:pje-1) =   Hy(pis:pie-1,pjs:pje-1) ...
        + (Chyex(pis:pie-1,pjs:pje-1).*(Ex(pis:pie-1,pjs+1:pje) - Ex(pis:pie-1,pjs:pje-1))) ...
        + (Chyez(pis:pie-1,pjs:pje-1).*(Ez(pis+1:pie,pjs:pje-1)-Ez(pis:pie-1,pjs:pje-1)));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%            update_pml_magnetic_field          %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nxp1 = nx+1; nzp1 = nz+1;

if is_pml_xp
    Hyx_xp = Chzxh_xp .* Hyx_xp ...
        + Chzxey_xp.*(Ez(pie+1:nxp1,:)-Ez(pie:nx,:));
    Hyz_xp = Chzyh_xp .* Hyz_xp ...
        + Chzyex_xp.*(Ex(pie:nx,pjs+1:pje)-Ex(pie:nx,pjs:pje-1));
end
```

```
Hy(1:pis-1,1:pjs-1) = Hyx_xn(:,1:pjs-1) + Hyz_zn(1:pis-1,:);
Hy(1:pis-1,pje:nz)  = Hyx_xn(:,pje:nz) + Hyz_zp(1:pis-1,:);
Hy(pie:nx,1:pjs-1)  = Hyx_xp(:,1:pjs-1) + Hyz_zn(pie:nx,:);
Hy(pie:nx,pje:nz)   = Hyx_xp(:,pje:nz) + Hyz_zp(pie:nx,:);
Hy(1:pis-1,pjs:pje-1) = Hyx_xn(:,pjs:pje-1) + Hyz_xn;
Hy(pie:nx,pjs:pje-1)  = Hyx_xp(:,pjs:pje-1) + Hyz_xp;
Hy(pis:pie-1,1:pjs-1) = Hyx_zn + Hyz_zn(pis:pie-1,:);
Hy(pis:pie-1,pje:nz)  = Hyx_zp + Hyz_zp(pis:pie-1,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          update_electric_field          %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ex (:,pjs+1:pje-1) = Ex (:,pjs+1:pje-1) ...
        + (Cexhy(:,pjs+1:pje-1).*(Hy(:,pjs+1:pje-1)-Hy(:,pjs:pje-2))) ...
        - (dt.*Hix(time_step).ex(:,pjs+1:pje-1));

Ez (pis+1:pie-1,:) = Ez (pis+1:pie-1,:) ...
        + (Cezhy(pis+1:pie-1,:).*(Hy(pis+1:pie-1,:)-Hy(pis:pie-2,:))) ...
        - (dt.*Hiz(time_step).ez(pis+1:pie-1,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          update_pml_electric_field          %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if is_pml_xp
    Ez(pie:nx,:) = Ceye_xp .* Ez(pie:nx,:) ...
        + Ceyhz_xp .* (Hy(pie:nx,:)-Hy(pie-1:nx-1,:));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          display_field          %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imagesc(x_mu_m1,z_mu_m,Electric_field(time_step).Ez');
    axis xy;
    colorbar;
line([x_mu_m1(1) x_mu_m1(nx)],[z_mu_m(R_top+1) z_mu_m(R_top+1)],'Color','m');
title('Electric field  (z) ');
    drawnow;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%          pearson_and_concordance_correlations       %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear Power;
clear ds dR x_point1 x_point2
clear Xa padding xcoor PCC CCC1 CCC2

dR = round ((3.5*lambda_c)/dx);
dS = round ((3.0*lambda_c)/dx);

z_point1 = R_top +1;
x_point1 = R_left - dR;
time_steps = 12000;

for ind  = 1: length (dS)

  z_point2 = z_point1;
  x_point2 = R_left + dS(ind);

for i = 1:time_steps

    Power(ind).xA (i) = -Electric_field(i).Ez(x_point1,z_point1).*Magnetic_field(i).Hy(x_point1,z_point1);
    Power(ind).xB1 (i) = -Electric_field(i).Ez(x_point2,z_point2).*Magnetic_field(i).Hy(x_point2,z_point2);

end

Power(ind).xB = (max (Power(ind).xA)/max (Power(ind).xB1))*Power(ind).xB1;
[Xa(ind).x,Xa(ind).y,Xa(ind).d] = alignsignals(Power(ind).xA, Power(ind).xB);
padding(ind).x = zeros(1,Xa(ind).d) ;  % a zero row vector
Xa(ind).z = [Xa(ind).y , padding(ind).x] ;
VarX = Xa.x;
VarY = Xa.z;
meanX = mean (VarX);
meanY = mean (VarY);
varianceX = var (VarX); sdX = sqrt (varianceX);
varianceY = var (VarY); sdY = sqrt (varianceY);
fact1 = sdX/sdY;
fact2 = (meanX-meanY)/(sqrt(sdX*sdY));
Cb = 2/(fact1+ (1/fact1)+ fact2^2);

 PCC(ind) = corr2(Xa(ind).x,Xa(ind).z)
 CCC1(ind) = 2*PCC(ind)*sqrt (varianceX*varianceY)./((meanY-meanX)^2 + varianceX + varianceY);
 CCC2(ind) = PCC(ind)*Cb
```

```
end

xcoor = dS*dx/lambda_c;

figure
plot(time*10^15,Power(1).xA);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```