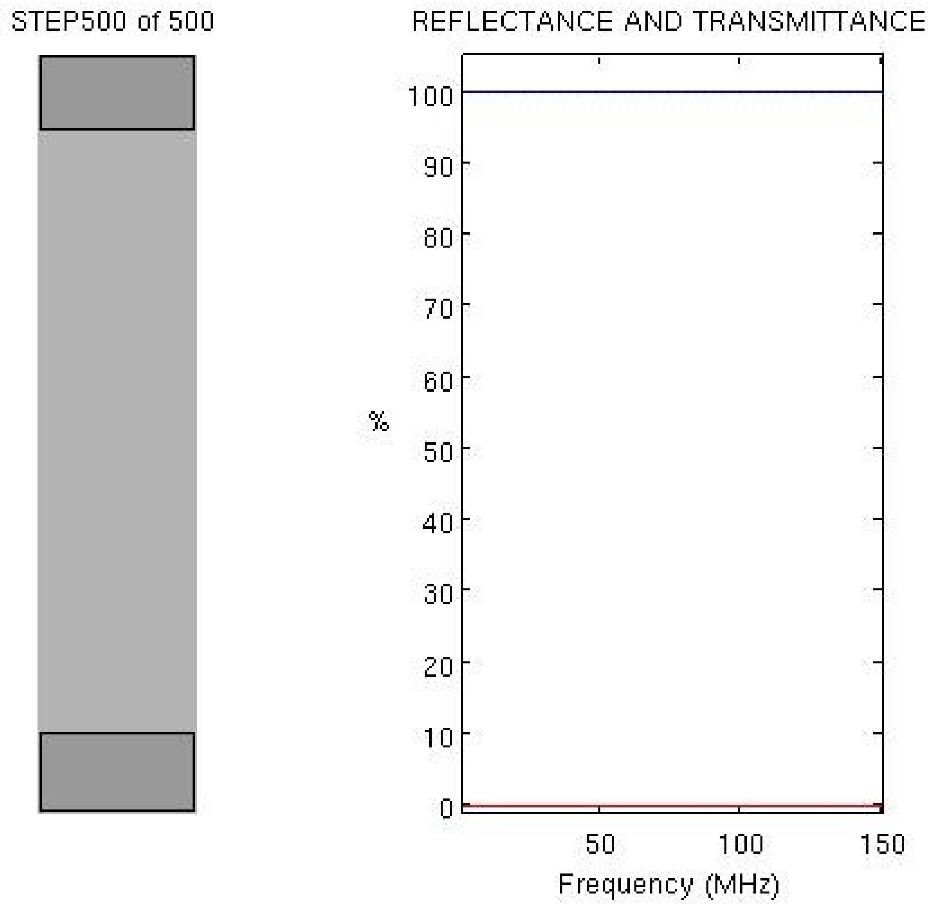
Compute TRN and FREF of Planewave with No Material




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

Nx = 41;
Ny = 200;
NPML = [0 0 20 20];
dx = 0.1;
dy = 0.1;
dt = 1.6e-10;
tau = 3.3e-9;
STEPS = 500;

% FREQ Parameters

NFREQ = 150;
fmax = 150*megahertz;
fmin = 1*megahertz;
FREQ = linspace(fmin, fmax, NFREQ);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Grid Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
% 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;
ta = [0:STEPS-1]*dt;
ny_src = Ny/2;%NPML(3)+2;
A = -sqrt(ERzz(1,ny_src)/URyy(1,ny_src));    % H Amplitude
deltsrc = 0.5*dy/c0 + dt/2; % Delay between E and H


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  REF and TRN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K = exp(-1i*2*pi*dt*FREQ); %Kernels for sweep across grid
EREF = zeros(Nx, NFREQ);  % Steady-State Reflected
ETRN = zeros(Nx, NFREQ);  % Steady-State Transmitted
SRC = zeros(1, NFREQ);    % Source transform

% Position of Recording planes
ny_ref = NPML(3) + 1;
ny_trn = Ny - NPML(4);

% Refractive indices in Recodring planes
nref = sqrt(ERzz(1,ny_ref));
ntrn = sqrt(ERzz(1,ny_trn));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%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);
```

```matlab
%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


  % TF/SF Source
  Ezsrc = exp(-((T*dt-t0)/tau).^2);
  CEx(:,ny_src-1) = CEx(:,ny_src-1) - Ezsrc/dy;


  % 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 ny=2:Ny
    CHz(1,ny) = (Hy(1,ny)-Hy(Nx,ny))/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

% TF/SF Source
Hx_src = A*exp(-((T*dt-t0+deltsrc)/tau).^2);
CHz(:,ny_src) = CHz(:,ny_src) - Hx_src/dy;

%Update D Integrations
IDz = IDz + Dz;

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


% Update Ez
Ez = mEz1.*Dz;


%%%%%%%%%%%%%%%%%% Compute Power %%%%%%%%%%%%%%%%%%%%

for f = 1:NFREQ
  EREF(:,f) = EREF(:,f) + (K(f)^T*Ez(:,ny_ref))*dt;
  ETRN(:,f) = ETRN(:,f) + (K(f)^T*Ez(:,ny_trn))*dt;
  SRC(f) = SRC(f) + (K(f)^T*Ezsrc)*dt;
end;



if mod(T,10) == 0
  subplot(121);
  draw2d(xa,ya, ERzz, Ez, NPML, 0.03);
  axis equal tight off;
  title(['STEP' num2str(T) ' of ' num2str(STEPS)]);
  drawnow;

  REF = zeros(1,NFREQ);
  TRN = zeros(1,NFREQ);

  for f = 1: NFREQ
    %Wave Vector Components
    lam0 = c0/FREQ(f);
    k0 = 2*pi/lam0;
    kzinc = k0*nref;
    m = [-floor(Nx/2):floor(Nx/2)]';
    kx = -2 * pi*m/(Nx*dx);
    kzR = sqrt((k0*nref)^2 - kx.^2);
    kzT = sqrt((k0*ntrn)^2 - kx.^2);

    %REF
    ref = EREF(:,f)/SRC(f);
```

```
      ref = fftshift(fft(ref))/Nx;
      ref = real(kzR/kzinc) .* abs(ref).^2;
      REF(f) = sum(ref);


      %TRN
      trn = ETRN(:,f)/SRC(f);
      trn = fftshift(fft(trn))/Nx;
      trn = real(kzT/kzinc) .* abs(trn).^2;
      TRN(f) = sum(trn);
    end

    CON = REF + TRN;

    subplot(122);
    plot(FREQ/megahertz,100*REF,'-r'); hold on;
    plot(FREQ/megahertz,100*TRN,'-b');
    plot(FREQ/megahertz,100*CON,':k'); hold off;
    axis([FREQ(1)/megahertz FREQ(NFREQ)/megahertz -1 105]);
    xlabel('Frequency (MHz)');
    ylabel('%','Rotation',0,'HorizontalAlignment','right');
    title('REFLECTANCE AND TRANSMITTANCE');
  end

end
```