```
function FDTDabc

% 2D propagation in a parallel plate waveguide w/ a dielectric slab
%
% short circuit walls at z=0, x=0, x=W;  Mur ABC termination at z=L
%
% one period sin line source excitation at user specified location
%
% dielectric slab at user specified location
%
% September 3, 2018   A. F. Peterson

  L = 2500; %input('Give guide length ');
  A = 2500; %input('Give guide width  ');
  Nz = 300; %input('Number of cells along length ');
  Nx = 300; %input('Number of cells along width ');
  delz = round(L/Nz);
  delx = round(A/Nx);
  Nsamp = round((Nx+1)/2);
  Nzm1=Nz-1;

  %Sz = 0; %input('Cell index of source along length? ');
  %Sx = round(Nx/2); %input('Cell index of source along width?  ');

% initialize variables

  %BigT = 1.0e-6; % transient excitation of 1 microsecond duration
  mu = pi*4.0e-7;
  epsilon = 8.854e-12;
  c = 2.998e8;
  freq = 1.0e6;
  eta=sqrt(mu/epsilon);
  Hy=zeros(Nx,Nz);
  Ex=zeros(Nx,Nz);
  Ez=zeros(Nx,Nz);
  Jx=zeros(Nx,Nz);
  Hymem(Nx)=0;

% Ex(jj,ii) is used to denote Ex(jj,ii-1/2)
% Ez(jj,ii) is used to denote Ez(jj-1/2,ii)
% erx(jj,ii) samples of permittivity at Ex-field locations
% erz(jj,ii) samples of permittivity at Ez-field locations

  epsx=ones(Nx,Nz);
  epsz=ones(Nx,Nz);
  zstart = round(500/delz); %input('starting index in z for slab? ');
  zend = zstart + round(0.4*(c/freq)/delz); %input('  ending index in
 z for slab? ');
  epslb = 5; %input('          epsilon-r for slab? ');

  for ii=zstart:zend
    if(ii == zstart) % values of epsx get average, epsz in slab
```

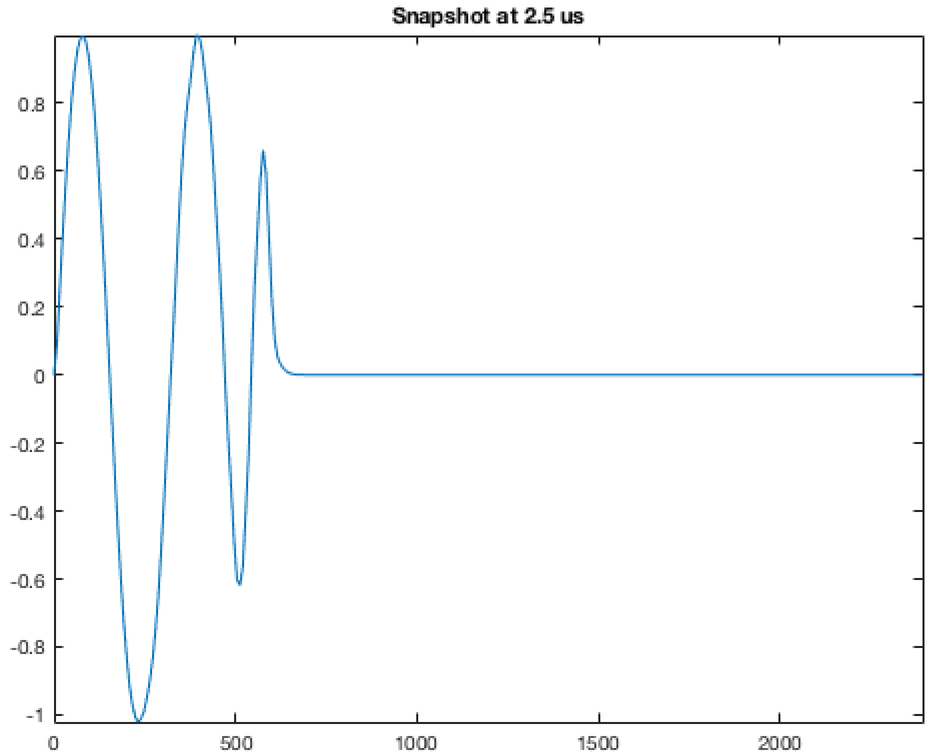
```
      for jj=1:Nx
        epsx(jj,ii)=(epslb+1)/2;
        epsz(jj,ii)=epslb;
      end
    elseif(ii == zend) % values of epsx get average, epsz out of slab
      for jj=1:Nx
        epsx(jj,ii)=(epslb+1)/2;
      end
    else   % values of epsx and epsz are in slab
      for jj=1:Nx
        epsx(jj,ii)=epslb;
        epsz(jj,ii)=epslb;
      end
    end
  end

% determine time step based on background permittivity

  deltmax = 1/c/sqrt((1/delz)^2+(1/delx)^2);
  str = ['CFL time step: ',num2str(deltmax)];  disp(str);
  delt = 1.25e-8; %input('Give time step  ');
  str = ['delta: ',num2str(delt)];  disp(str);

  dtomudz = delt/(mu*delz);
  dtomudx = delt/(mu*delx);
  dtoepdz = delt/(epsilon*delz);
  dtoepdx = delt/(epsilon*delx);
  beta=1/(2*delz/c/delt + 1);
  alpha=beta*(2*delz/c/delt - 1);

% march in time

  fid = fopen('snapshots.txt', 'wt');

   nslab = sqrt(mu/(5*epsilon));
   gamma = (nslab - eta) / (nslab + eta);
   fprintf(fid, 'reflection coefficient: %f \n', gamma);
   fprintf('reflection coefficient: %f \n', gamma);
   tau = 2*nslab / (nslab + eta);
   fprintf(fid, 'transmission coefficient: %f \n\n', tau);
   fprintf('transmission coefficient: %f \n\n', tau);

   for kk = 1:500

%    update source
 ------------------------------------------------------

     t = kk*delt;

     for jj = 1:Nx
        Ex(jj,1)=sin(2*pi*freq*t);
     end
```

```
%    update Hy at time kk+1/2  (Nx by Nz samples)
 ------------------------

     for jj = 1:Nx
         Hymem(jj)=Hy(jj,Nzm1); % store next-to-last values for ABC
     end

     for ii = 1:Nzm1
       for jj = 1:Nx

         if(jj == 1) % zero Ez field at wall of guide jj

            Hy(jj,ii)=Hy(jj,ii)-dtomudz*(Ex(jj,ii+1)-Ex(jj,ii)) ...
                               +dtomudx*(Ez(jj+1,ii)          );

         elseif(jj == Nx)  % zero Ez field at wall of guide jj+1

            Hy(jj,ii)=Hy(jj,ii)-dtomudz*(Ex(jj,ii+1)-Ex(jj,ii)) ...
                               +dtomudx*(           -Ez(jj,ii));

         else % out in the middle of the mesh somewhere

            Hy(jj,ii)=Hy(jj,ii)-dtomudz*(Ex(jj,ii+1)-Ex(jj,ii)) ...
                               +dtomudx*(Ez(jj+1,ii)-Ez(jj,ii));

         end
       end
     end

%    update Hy at absorbing boundary Nz

       for jj = 1:Nx

         if(jj == 1) % zero Ez field at wall of guide jj

            Hy(jj,Nz)=alpha*Hy(jj,Nz)...
                     +beta*(Hy(jj,Nzm1)+Hymem(jj))...
                     +beta*(Ez(jj+1,Nz)          )/eta;

         elseif(jj == Nx)  % zero Ez field at wall of guide jj+1

            Hy(jj,Nz)=alpha*Hy(jj,Nz)...
                     +beta*(Hy(jj,Nzm1)+Hymem(jj))...
                     +beta*(           -Ez(jj,Nz))/eta;

         else % out in the middle of the mesh somewhere

            Hy(jj,Nz)=alpha*Hy(jj,Nz)...
                     +beta*(Hy(jj,Nzm1)+Hymem(jj))...
                     +beta*(Ez(jj+1,Nz)-Ez(jj,Nz))/eta;

         end
       end
```

```
%    update Ex at time kk  (Nx by Nz-1 samples)
 --------------------------

     for ii = 2:Nz  % don't update Ex(x,1), keep at zero for PEC
       for jj = 1:Nx

         Ex(jj,ii)=Ex(jj,ii)-dtoepdz*(Hy(jj,ii)-Hy(jj,ii-1))/
epsx(jj,ii)...
                            -(delt/epsilon)*Jx(jj,ii);

       end
     end

%    update Ez at time kk (Nx-1 by Nz samples)
 ---------------------------

     for ii = 1:Nz
       for jj = 2:Nx

         Ez(jj,ii)=Ez(jj,ii)+dtoepdx*(Hy(jj,ii)-Hy(jj-1,ii))/
epsz(jj,ii);

       end

     end

% store snapshot down centerline in file 'fid'
 ---------------------------

      str = ['result at time = ',num2str(t)];
      fprintf(fid,'%s \n',str);
      for ii=1:Nz
         z = delz * (ii-1);
         fprintf(fid,'%15.14g %15.14g\n',z,Ex(Nsamp,ii));
      end
      fprintf(fid,'%15.14g %15.14g\n',L,0.0);
      fprintf(fid,'\n\n');

      if t > 2.4999e-6 && t < 2.5001e-6
          figure
          z = delz .* (0:Nz-1);
          plot(z, Ex(Nsamp,:))
          axis tight;
          title('Snapshot at 2.5 us');
      end

   end

   fclose(fid);

end
```

CFL time step: 1.8869e-08
delta: 1.25e-08

```
reflection coefficient: -0.381966
transmission coefficient: 0.618034
```



Published with MATLAB® R2018a