```
function femtot2

% use finite element formulas to solve the scalar Helmholtz
% equation for EM fields in a dielectric slab and the
% reflection and transmission coefficients
%
% second order Lagrange basis functions

     global cell_to_node x;

% read mesh from file 'inputfil.txt'

 n_nodes = dlmread('inputfil.txt','', [0,0,0,0]);
 n_cells = dlmread('inputfil.txt','', [0,1,0,1]);

 x=dlmread('inputfil.txt','', [1,1,n_nodes,1]);

 nstart= n_nodes + 1;
 nend = nstart + n_cells - 1;
 cell_to_node = dlmread('inputfil.txt','', [nstart,1,nend,3]);

 nstart = nend + 1;
 nend = nend + n_cells;
 epsilon = dlmread('inputfil.txt','', [nstart,1,nend,1]);

% disp(x);
% disp(cell_to_node);
% disp(epsilon);

% initialize variables

  k0 = 2*pi;
  n_unknowns = n_nodes;
  Z = zeros(n_unknowns);
  RHS = zeros(n_unknowns,1);

% fill global matrix one cell at a time using element matrices
  cell = 1;
  for i = 1:2:n_unknowns-1
      [eleS, eleT] = elemat(cell);
      Z(i:i+2,i:i+2) = Z(i:i+2,i:i+2) + eleS -
 k0^2*epsilon(cell)*eleT;
      cell = cell+1;
  end

%  add boundary conditions for RBC terminations
Z(1,1) = Z(1,1) + j*k0;
Z(end,end) = Z(end,end) + j*k0;

% fill excitation vector (right hand side)
 RHS(1) = 1j*2*k0;  % assumes that incident Ey(a)=1
```

```
% solve system of equations

 E = Z\RHS;

% write fields to output file

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

% reflection & tranmission coefficients

 gamma = E(1)-1;
 mag=abs(gamma);
 phs=180*atan2(imag(gamma),real(gamma))/pi;
 str = ['reflection coeff = ',num2str(mag),' ',num2str(phs)];
 disp(str);
 fprintf(fid,'%50s\n\n',str);

 tau = E(n_unknowns);
 mag=abs(tau);
 phs=180*atan2(imag(tau),real(tau))/pi;
 str = ['transmission coeff = ',num2str(mag),' ',num2str(phs)];
 disp(str);
 fprintf(fid,'%44s\n\n',str);

% fields
 for irow=1:n_unknowns
         mag=abs(E(irow));
         phs=180*atan2(imag(E(irow)),real(E(irow)))/pi;
         fprintf(fid,'%6d %15.14g %15.14g\n',irow, mag, phs);

 end

end

%
 ----------------------------------------------------------------------

function [eleS,eleT] = elemat(icell)

%
%  elemat: construct the element matrix for the contributions of
%          basis and testing functions of the form
%                   dT/dx * dB/dx    and    T * B
%          for a 1D cell, second order basis & test functions
%
%          (assumes mid-cell nodes are at exact center of cell)

     global cell_to_node x;

     eleS(3,3)=0;  eleT(3,3)=0;
     n1=cell_to_node(icell,1);
     n2=cell_to_node(icell,2);
%     n3=cell_to_node(icell,3);
```

```
%    compute 3 by 3 element matrix S

     delta = x(n2) - x(n1);

     eleS(1,1)=7;
     eleS(1,2)=-8;
     eleS(1,3)=1;

     eleS(2,1)=-8;
     eleS(2,2)=16;
     eleS(2,3)=-8;

     eleS(3,1)=1;
     eleS(3,2)=-8;
     eleS(3,3)=7;

     eleT(1,1)=4;
     eleT(1,2)=2;
     eleT(1,3)=-1;

     eleT(2,1)=2;
     eleT(2,2)=16;
     eleT(2,3)=2;

     eleT(3,1)=-1;
     eleT(3,2)=2;
     eleT(3,3)=4;

     for ii=1:3
        for jj=1:3
           eleS(ii,jj)=eleS(ii,jj)/(6*delta);
           eleT(ii,jj)=eleT(ii,jj)*delta/15;
        end
     end
end
```

Published with MATLAB® R2018a