```
function HW4cavity

% compute propagation constants for waveguide cross section modeled
% with triangular cells
%
% August 2, 2018   A. F. Peterson

global pcetond xy;

% read mesh from file 'cylfil.txt'
nnodes = dlmread('cylfil.txt','', [0,0,0,0]);
ncells = dlmread('cylfil.txt','', [0,1,0,1]);
ninner = dlmread('cylfil.txt','', [0,2,0,2]);
nouter = dlmread('cylfil.txt','', [0,3,0,3]);
nunks = nnodes - ninner - nouter;
xy=dlmread('cylfil.txt','', [1,1,nnodes,2]);
nstart=nnodes + 1;
nend=nstart + ncells - 1;
pcetond=dlmread('cylfil.txt','', [nstart,1,nend,3]);

% initialize variables
W=zeros(nunks);
Y=zeros(nunks);

% fill global matrix one cell at a time
for icell=1:ncells

    [eleS,eleT]=elemat(icell);

    %    add contributions from cell 'icell' to global matrix
    for ii=1:3
        ig=pcetond(icell,ii); % 'ig' is the global node for 'ii'
        for jj=1:3
            jg=pcetond(icell,jj);  % 'jg' is the global node for 'jj'
            if(ig <= nunks) % test function at interior node
                if(jg <= nunks) % basis function at interior node
                    W(ig,jg) = W(ig,jg) + eleS(ii,jj);
                    Y(ig,jg) = Y(ig,jg) + eleT(ii,jj);
                end
            end
        end
    end
end

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

E = eig(W,Y); % use [V,E] = eig(W,Y) to get eigenvectors as well

str = 'TM resonant wavenumbers: ';
fprintf(fid,'%s \n',str);
for ii=1:nunks
    reaE=real(sqrt(E(ii)));
```

```
    imaE=imag(sqrt(E(ii)));
    fprintf(fid,'%6d %15.14g %15.14g\n',ii, reaE,imaE);
end

end

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

function [eleS,eleT] = elemat(icell)
%  elemat: construct the element matrix for the contributions of
%          basis and testing functions of the form
%               S = grad Bm dot grad Bn
%                           and
%                   T = Bm times Bn
%          over a triangular cell

global pcetond xy;

eleS(3,3)=0;  eleT(3,3)=0;
n1=pcetond(icell,1);
n2=pcetond(icell,2);
n3=pcetond(icell,3);

x(1)=xy(n1,1);  y(1)=xy(n1,2);
x(2)=xy(n2,1);  y(2)=xy(n2,2);
x(3)=xy(n3,1);  y(3)=xy(n3,2);

b(1)=y(2)-y(3);
b(2)=y(3)-y(1);
b(3)=y(1)-y(2);

c(1)=x(3)-x(2);
c(2)=x(1)-x(3);
c(3)=x(2)-x(1);

Area = abs(b(3)*c(1) - b(1)*c(3))*0.5;

for ii=1:3
    for jj=1:3
        eleS(ii,jj)=(b(ii)*b(jj)+c(ii)*c(jj))/Area/4;
        if(ii == jj)
            eleT(ii,jj)=Area/6;
        else
            eleT(ii,jj)=Area/12;
        end
    end
end
end
```

Published with MATLAB® R2018a