```
function [lambda detErr lambdaErr  iter] = jrbSturm(K,M,MAXERR,MAXIT)
% (c) Jacob Barton - University of Queensland - 2009
%
% jrbSturm will solve the eigenvalues of the system :
% [K-lambda*M]*[X] = [0] provided all eigenvalues are real.
%
% The function will also solve standard eigen problems. Setting M = []
% will result in the Identity matrix being used.
%
% Syntax: jrbSturm(K,M,MAXERR,MAXIT)
%
% Inputs:
% K      - Stiffness Matrix
% M      - Mass Matrix. (Optional) (Identity Matrix Default)
% MAXERR - Maximum error to solve to. (Optional) (1e-6 Default)
% MAXIT  - Maximum iterations to perform. (Optional) (100 Default)
%
% Outputs:
% lambda       - Eigenvalues in increaseing order
% detErr       - The abs error of the determinates for lambda
% lambdaErr    - The abs error of each respective eigen value
% iter   - The number of iterations taken to solve each eigenvalue
%
% Dependencies:
% This function uses all standard matlab functions, except also requires
% the function jrbSturmCount which should be included in the same
% dirrectory, on the search path, or as a nested function as below.
%
% Reference:
% Matrix Analysis and Structural Dynamics. Applications of earthquake
% engineering. Franklin Y. Cheng. Pages 95-97.

if nargin < 2; M = []; end
if nargin < 3; MAXERR = 1e-6;end
if nargin < 4; MAXIT = 100;end

n = size(K,1);
if isempty(M);M=eye(n);end

Bounds = [0 2];%Starting values, now refine guess to suitable values
while jrbSturmCount(K,M,Bounds(2)) > 0
    Bounds(2)=2*Bounds(2);%do upper bound
end
while jrbSturmCount(K,M,Bounds(1)) < n%do lower bound
    if Bounds(1) == 0;
        Bounds(1) = -2;
    else
        Bounds(1) = Bounds(1)*2;
    end
end

L = Bounds(1);
U = Bounds(2);

lambda = zeros(n,1);
iter = zeros(n,1);
```

```
lambdaErr = zeros(n,1);
detErr = zeros(n,1);
for q = 1:n
    i = 0;
    while abs(U-L) > MAXERR && i < MAXIT
        p = (L+U)/2;
        num_greater = jrbSturmCount(K,M,p);
        if num_greater == n - q + 1
            L = p;
        else
            U = p;
        end
        i = i + 1;
    end
    lambda(q) = (U+L)/2;
    iter(q) = i;
    lambdaErr(q) = abs(U-L)/2;
    detErr(q) = abs(det(K-lambda(q)*M));
    U = Bounds(2);%reset U for next iteration (L will be ok)
end
end

function num_larger = jrbSturmCount(K,M,lamnda)
% (c) Jacob Barton - University of Queensland - 2009
%
% jrbSturmCount will count how many eigen values in the matrix A
% or larger then the value specified for lambda.
%
% Syntax: jrbSturmCount(K,M,lambda)
%
% Inputs:
% K      - Stiffness Matrix
% M      - Mass Matrix
% lambda - Guess for eigenvalue
%
% Outputs:
% num_larger - Number of eigen values greater than lambda

A = K-lamnda*M;

%calculate leading principle minors
sz = size(A,1);
P = zeros(sz+1,1);
P(1) = 1;%det of empty matrix is 1
for i = 1:sz
    P(i+1) = det(A(1:i,1:i));
end
%find the number of times signs agree
num_larger = sum(sign(P(1:end-1))== sign(P(2:end)));
end
```