21 November 2016

CE 541A MDOF Power Method

The Power Method for computing (some or all of) the natural frequencies and mode shapes of a MDOF system was given in the notes for the 10/24/2016 lecture (see page 4, which is marked as “(2b) ”).

1. Write a function to compute and return the first k natural frequencies and mode shapes of a MDOF system with n degrees of freedom. Your function should take as inputs, the mass matrix M, the stiffness matrix K, and k (the number of eigenvalues/eigenvectors to compute).

```
function EigenValueAndVector = InverseVectorIterationGeneralised(M,K,k)
format long
% the following program computes the eigen values and eigen vectors
% to any number of degree of freedom system
% setting the number of degrees of freedom
% of the problem
% dof=input('give the total number of degrees of freedom of the system ');
dof=3 ;

% here k is the total number of eigen values we want the
% program to give it as output

% This is the output which contains
% eigen value in the first half section and
% eigen vectors in the second half of the section
EigenValueAndVector_all=zeros(dof,2*dof);

% this is lamda Initial guess
la=0;

for c=1:dof

    % this is the shift in eigen value that has to be done
    shift=la*dof;
    % The shifted value is assigned as the starting value in computation
    la=shift;

    % this matrix is due to the shifting
    Ks=K-shift*M;

    % initial guess for the eigen vector
    xjp1=ones(dof,1);

    for i=1:100
       %we begin iterating for eigen vector from here
       xjp1=inv(Ks)*M*xjp1;
       lamda=(xjp1'*Ks*xjp1)/(xjp1'*M*xjp1);

       % this is the normalised forn of eigen vector
       xjp1=xjp1/(xjp1'*M*xjp1)^0.5;

       % this is the tolerence used to stoop the iteration
       toler=lamda-la;

       % if statement to break if tolerence is reached
```

```
        % this if statement ensures we get convergence
        if(abs(toler)<0.001 )
            break
        end

        % assigning the computed eigen value per iteration to
        % initial guess of eigen value
        la=lamda;
    end

    % shifting the eigen vector bact to original system
    la=lamda+shift;

    % post processing the eigen value for output
    EigenValueAndVector_all(c,c)=la;

    % post processing the eigen vector for out put
    EigenValueAndVector_all(:,3+c)=xjp1;

    % initialising Ks back to K after shifting
    Ks=K;

end
% here k is the total number of eigen values we want the
% program to give it as output
   EigenValueAndVector=zeros(dof,2*k);
   for h=1:k
      EigenValueAndVector(:,h)= EigenValueAndVector_all(:,h);
   end
   for h=1:k
      EigenValueAndVector(:,k+h)= EigenValueAndVector_all(:,dof+h);
   end
   EigenValueAndVector;

   naturalFrequency=EigenValueAndVector(:,1:k);
   for d=1:k
      naturalFrequency(d,d)= sqrt(naturalFrequency(d,d));
   end
   naturalFrequency

end
```

To check the correctness of the above function we will take a sample example problem from the textbook. From page number 432 Example 10.14.

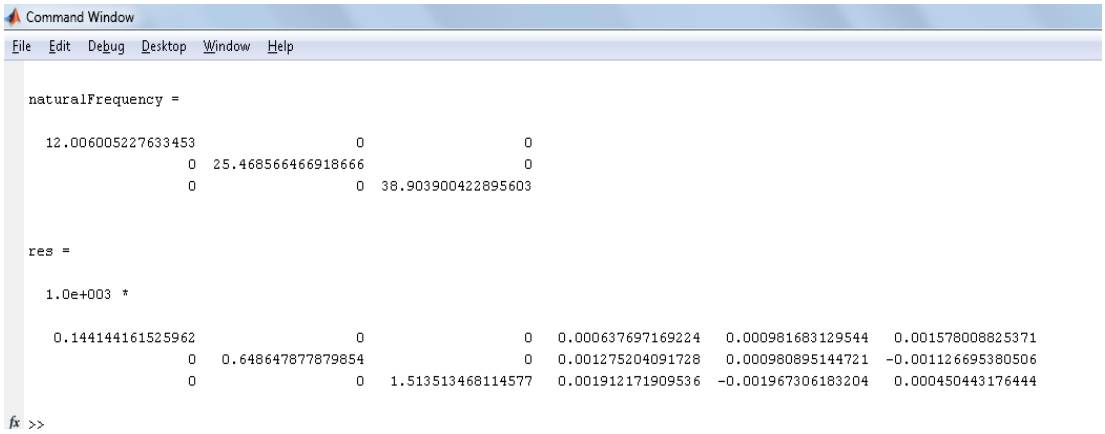
```
clear all
clc
M=0.259*[1 0 0;
    0 1 0;
    0 0 .5];
K=(168/9)*[16 -7 0;
    -7 10 -3;
    0 -3 3];
k=3;

res=InverseVectorIterationGeneralised(M,K,k)
```

21 November 2016

The above function written can compute for any number of degree of freedom system and can compute any number of modes that the user asks the function to compute. To make it work for any degree of freedom system comment out the line highlighted in the program and un comment the line above it. By doing so the program can compute the values for any number of degree of freedom system.

The result obtained is
We find that the result obtained is exactly same as the answer provided in page 436 table 10.15a, 10.15b and 10.15c

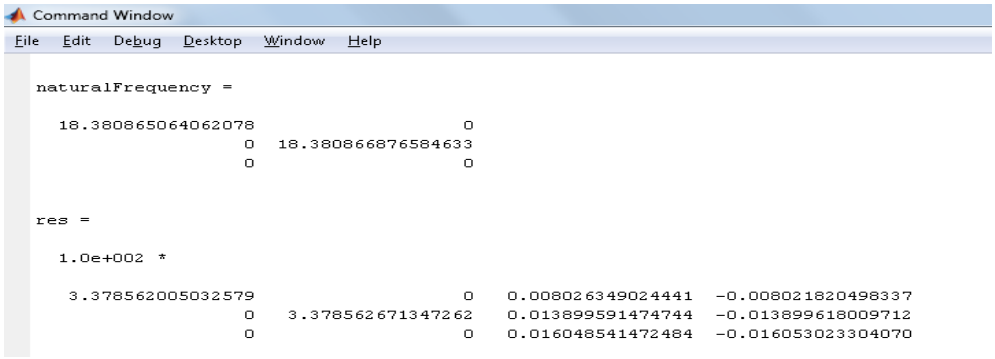
The corresponding values are depicted in the res variable and is depicted in the result below



2. Write a function
or script to that uses the function from #1 to determine the first 2 natural f requencies and mode shapes of the structure in Chopra problem 10.29. How do they compare to the exact answers

```
% problem 10.29
M=0.2588*[1 0 0;
    0 1 0;
    0 0 .5];
K=(326.32)*[2 -1 0;
    -1 2 -1;
    0 -1 1];
k=2;

res=InverseVectorIterationGeneralised(M,K,k)
```

21 November 2016



We find that the eigen value and vector is same as the theoretical results computed hence we can say that our program is correct.

The output produced is such that the first half of the values correspond tho the natural frequencies in radian per second and the second half pertains to the mode shapes.

Fundamental Natural Frequency is 18.38 rd/s
Fundamental mode is [0.802 1.389 1.604]T .]