```
%--------------------------------------------------------------------------
% Copyright (c) 2016, Anna Liakou
% All rights reserved.
%
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are
% met:
%
% * Redistributions of source code must retain the above copyright
% notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
% notice, this list of conditions and the following disclaimer in
% the documentation and/or other materials provided with the distribution
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
% AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
% IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
% ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
% LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
% CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
% SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
% INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
% CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
% ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
% POSSIBILITY OF SUCH DAMAGE.
% ------------------------------------------------------------------------
% Response of Rayleigh beam for unit impulse load with beta=0.1
% Fourier series - 6 modes
%--------------------------------------------------------------------------
function main()
clear all
close all
%% Geometrical properties::
%------------------------------------------------------
density=2700; % density
radius=sqrt(0.04);  % radius of the beam for circular cross section
E=100000000000; % Young Modulus
I=pi*(radius^4)/4; % Moment of inertia
Area=pi*radius^2; % Area of cross-sectional area
Period=1/sqrt(E*I/(density*Area)); % Time period for unit-length beam
%% Initialize variables::
%------------------------------------------------------------
B=zeros(6,1);C=zeros(6,1);A=zeros(6,1);
omega=zeros(6,1);
%% Define::
% tot = # of time steps
% tot_l= # of spatial positions
tot=1000;tot_l=100;
time=linspace(0,4,tot);
x=linspace(0,1,tot_l);
% Roots of frequency equation for specific slenderness ratio
% beta^2=I/(Area*L^2)=0.1
% For the first 6 modes
%------------------------------------------------------------
A(1)=1.8699;B(1)=1.8380;
A(2)=4.5885;B(2)=4.1705;
A(3)=7.6512;B(3)=6.0766;
```

```
A(4)=10.6856;B(4)=7.3014;
A(5)=13.8062;B(5)=8.098;
A(6)=16.9389;B(6)=8.6117;
alpha=coef_R(A,B);
for i=1:6
  omega(i)=sqrt((A(i)^2-B(i)^2)/(I/Area));
end
%% Apply orthogonality condition::
%--------------------------------------------------------------------------
for i=1:6
eig=@(x) (sin(A(i)*x)-A(i)*sinh(B(i)*x)/B(i)-alpha(i)*(cos(A(i)*x)-
cosh(B(i)*x))).^2;
der=@(x) (I/Area)*(A(i)*(cos(A(i)*x)-
sinh(B(i)*x))+alpha(i)*(A(i)*sin(A(i)*x)+B(i)*sinh(B(i)*x))).^2;
WW(i) = integral(eig,0,1);
Der2(i)=integral(der,0,1);
C(i)=1/sqrt(WW(i)+Der2(i));
end
%% Calculate normalized normal modes and initialize the forced term for every
mode separately:
%----------------------------------------------------------------------------
-------
Sin_f=zeros(6,tot);
for i=1:6
    Sin_f(i,:)=sin(time*omega(i));
    eig_L(i)=C(i)*(sin(A(i))-A(i)*sinh(B(i))/B(i)-alpha(i)*(cos(A(i))-
cosh(B(i))))';
end

%% Calculate final displacement for every mode:
%----------------------------------------------------------------
W1=eig_R(tot,tot_l,eig_L,B,A,alpha,1,x,Sin_f,omega,C);
W2=eig_R(tot,tot_l,eig_L,B,A,alpha,2,x,Sin_f,omega,C);
W3=eig_R(tot,tot_l,eig_L,B,A,alpha,3,x,Sin_f,omega,C);
W4=eig_R(tot,tot_l,eig_L,B,A,alpha,4,x,Sin_f,omega,C);
W5=eig_R(tot,tot_l,eig_L,B,A,alpha,5,x,Sin_f,omega,C);
W6=eig_R(tot,tot_l,eig_L,B,A,alpha,6,x,Sin_f,omega,C);
W=-(W1+W2+W3+W4+W5+W6);
%% Plot result::
%--------------------------------------------------------------------
figure
plot(time(:),W(tot_l,:))
title('Displacement - time at free end')
xlabel('\tau')
ylabel('u')
end

%% Calculate constant term of the general formula for the eigenfunction when
we deal with cantilever beam
%----------------------------------------------------------------------------
----------
function alpha=coef_R(A,B)
alpha=zeros(6,1);
for i=1:6
alpha(i)=((A(i)^2)*sin(A(i))+A(i)*B(i)*sinh(B(i)))/((A(i)^2)*cos(A(i))+(B(i)^
2)*cosh(B(i)));
```

```
end
end
%% Derive the forced vibration for every mode::
%---------------------------------------------------------------------------
----------
function W=eig_R(tot,tot_l,eig_L,B,A,alpha,k,x,Sin_f,omega,C)
for i=1:tot_l
    Wn(i)=eig_L(k)*C(k)*(sin(A(k)*x(i))-(A(k)/B(k))*sinh(B(k)*x(i))-
alpha(k)*(cos(A(k)*x(i))-cosh(B(k)*x(i))));
    for j=1:tot
        W(i,j)=Wn(i)*Sin_f(k,j)/omega(k);
    end
end
end
```