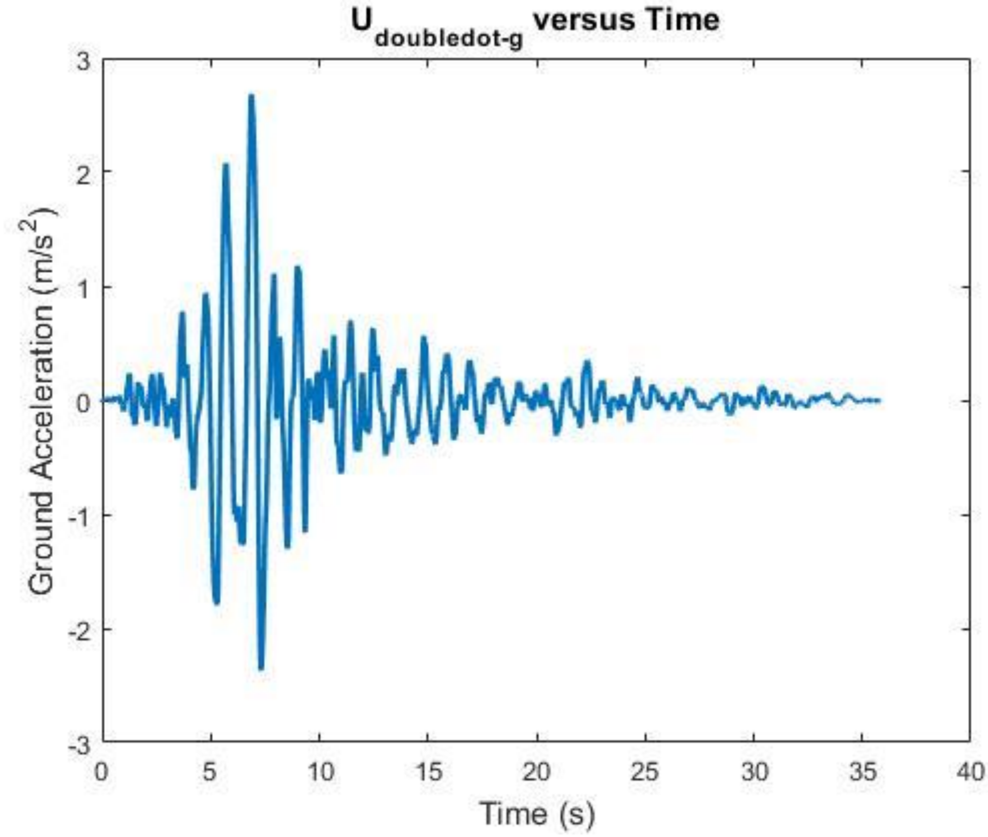
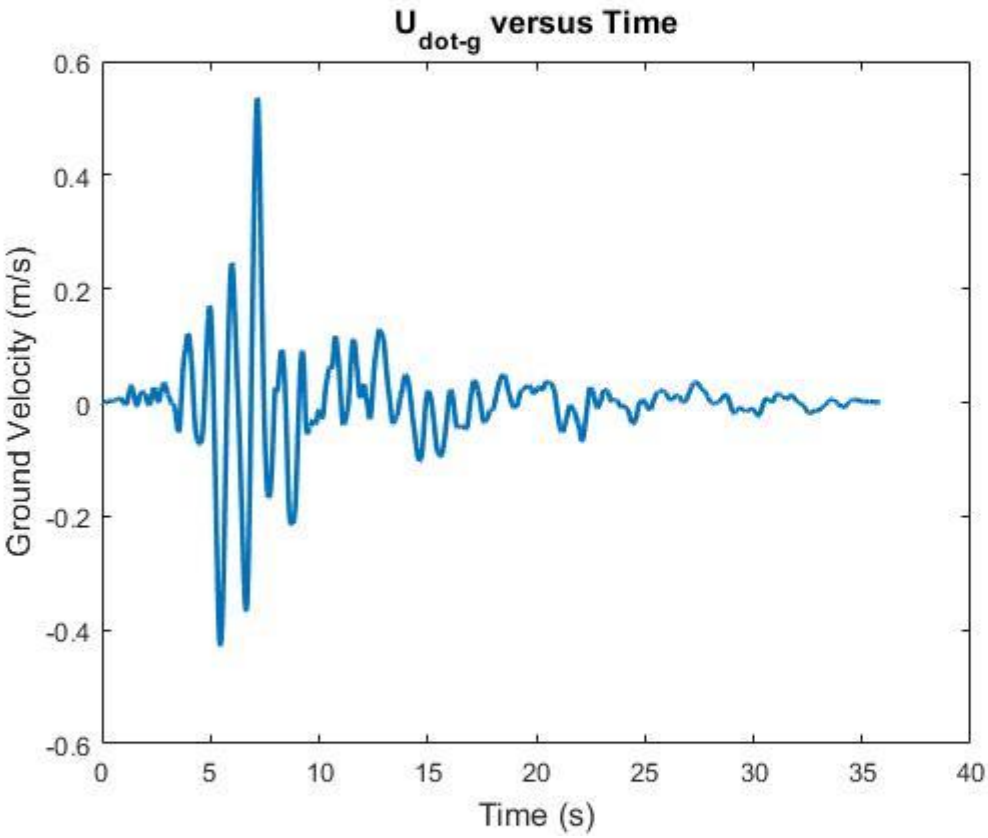
TABLE OF CONTENTS

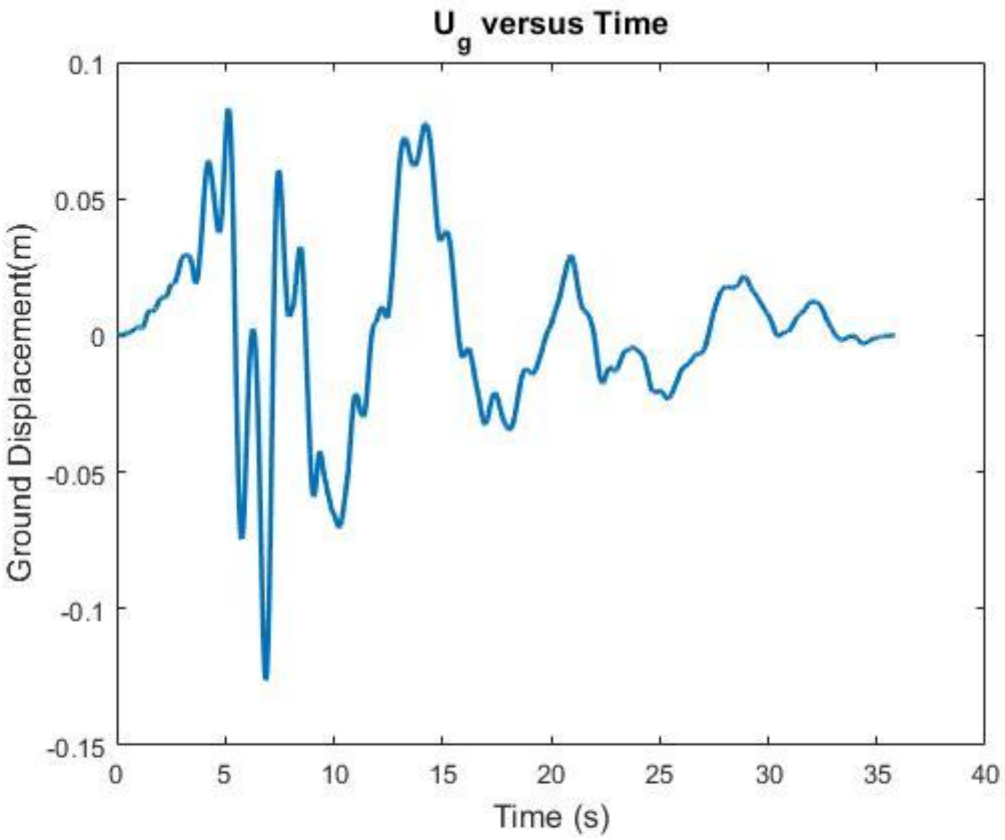
QUESTION-1



$U_{doubledot\text{-}g}$ versus Time
Ground Acceleration (m/s^2)
3
2
1
0
-1
-2
-3
0
5
10
15
20
25
30
35
40
Time (s)



$U_{dot\text{-}g}$ versus Time
Ground Velocity (m/s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
5
10
15
20
25
30
35
40
Time (s)



MATLAB CODE

```
%This code will plot the ground acceleration-time, ground velocity-time,
%and ground displacement-time graphics by utilizing the given data for
%HOMEWORK-I in CE490-INTRODUCTION TO EARTHQUAKE ENGINEERING class.

clear all %clears all workspace.
clc %clears command window.

format long
delta_t=0.005; %defines the given time interval.
u_g_doubledot=load('A02043.txt'); %loads the given ground acceleration data.
u_g_doubledot(end+1)=0; %returns the zero at the end of A02043.txt data meaning that
the earthquake ended.
n=length(u_g_doubledot); %finds the dimension of given ground acceleration data.
T=zeros(n,1); %creates a time array having dimension of nx1 being same as A02043 data.
u_g_dot=zeros(n,1); %creates a ground veleocity array having dimension of nx1 being
same as A02043 data.
u_g=zeros(n,1); %creates a ground displacement array having dimension of nx1 being
same as A02043 data.

for i=1:n-1 %calculates the ground velocity in accordance with equation 3.32 given in
the book.
    u_g_dot(i+1,1)=u_g_dot(i,1)+(u_g_doubledot(i,1)+u_g_doubledot(i+1,1))*delta_t/2;
end

for i=1:n-1 %calculates the ground displacement in accordance with equation 3.34 given
in the book.

u_g(i+1,1)=u_g(i,1)+u_g_dot(i,1)*delta_t+(u_g_doubledot(i+1,1)+u_g_doubledot(i,1))*(d
elta_t^2)/4;
end

for i=1:1:n-1 %increments the time by delta_t.
    T(i+1,1)=T(i,1)+delta_t;
```

```
end

figure('Name','Ground Acceleration'); %Plots the ground acceleration-time graph.
plot(T,u_g_doubledot,'LineWidth',2);
xlabel('Time (s)','FontSize',12);
ylabel('Ground Acceleration (m/s^2)','FontSize',12);
title('U_d_o_u_b_l_e_d_o_t_-_g versus Time','FontSize',12);

figure('Name','Ground Velocity'); %Plots the ground velocity-time graph.
plot(T,u_g_dot,'LineWidth',2);
xlabel('Time (s)','FontSize',12);
ylabel('Ground Velocity (m/s)','FontSize',12);
title('U_d_o_t_-_g versus Time','FontSize',12);

figure('Name','Ground Displacement'); %Plots the ground displacement-time graph.
plot(T,u_g,'LineWidth',2);
xlabel('Time (s)','FontSize',12);
ylabel('Ground Displacement(m)','FontSize',12);
title('U_g versus Time','FontSize',12);
```

QUESTION-2

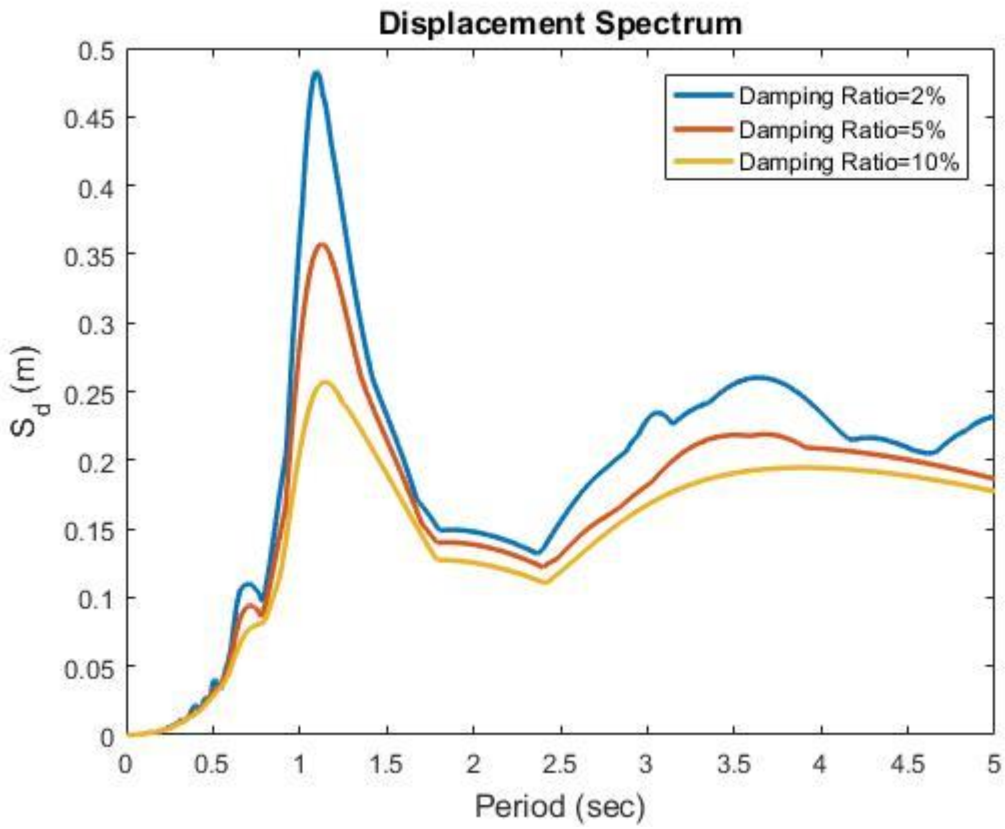
MATLAB CODE

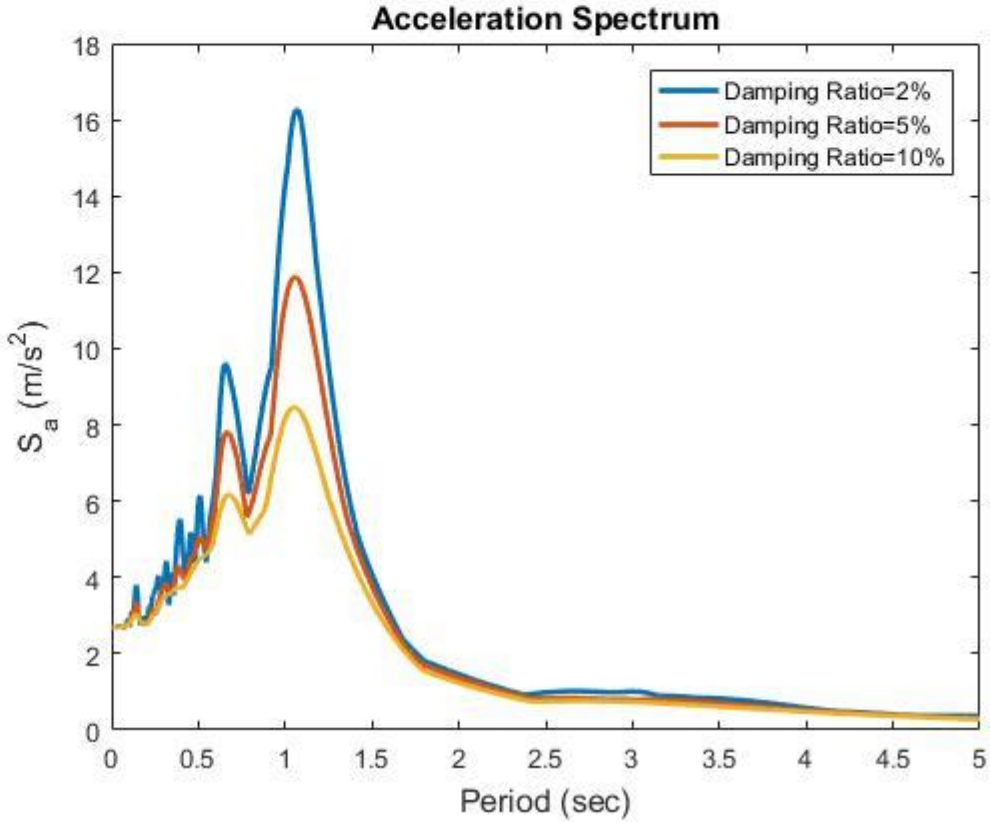
```
%This code will calculate and plot the spectra for 2%, 5% and 10% damping.
%It is important to note that 2% damping code was written at first and then
%it is copied to other m.files so as to change to 5% and 10% damping codes
%and finally they brought back to this file by copying back.

clear all %This clears all workspace.
clc %This clears command window.

delta_t=0.005; %This defines the given time interval.
T_final=5; %This determines the final period.
u_g_doubledot=load('A02043.txt'); %This takes the given ground acceleration data from
.txt file
damping_2=2; %This defines the damping ratio.

u_g_spectrum_2=zeros(length(u_g_doubledot),1); %This creates an array for displacement
spectrum under 2% damping.
u_g_dot_spectrum_2=zeros(length(u_g_doubledot),1); %This creates an array for velocity
spectrum under 2% damping.
u_g_doubledot_spectrum_2=zeros(length(u_g_doubledot),1); %This creates an array for
acceleration spectrum under 2% damping.
u_g_doubledot(end+1)=0; %This returns the zero at the end of ground acceleration.
T(1,1)=0.00+delta_t;

for i=1:round(T_final/delta_t) %This part will perform the calculation
    W_2(i,1)=2*pi/T(i);
    mass_2=1;
    k_2=((W_2(i))^2)*mass_2;
    c_2=2*mass_2*W_2(i)*damping_2/100;
    new_k_2=k_2+2*c_2/delta_t+4*mass_2/(delta_t)^2;

    for j=1:length(u_g_spectrum_2)-1
        u_g_spectrum_2(1,1)=0;
        u_g_dot_spectrum_2(1,1)=0;
        u_g_doubledot_spectrum_2(1,1)=(-mass_2*u_g_doubledot(1,1)-
c_2*u_g_dot_spectrum_2(1,1)-k_2*u_g_spectrum_2(1,1))/mass_2;

        delta_f_2=-mass_2*(u_g_doubledot(j+1)-
u_g_doubledot(j))+(4*mass_2/delta_t+2*c_2)*u_g_dot_spectrum_2(j,1)+2*mass_2*u_g_doubl
edot_spectrum_2(j,1);
```

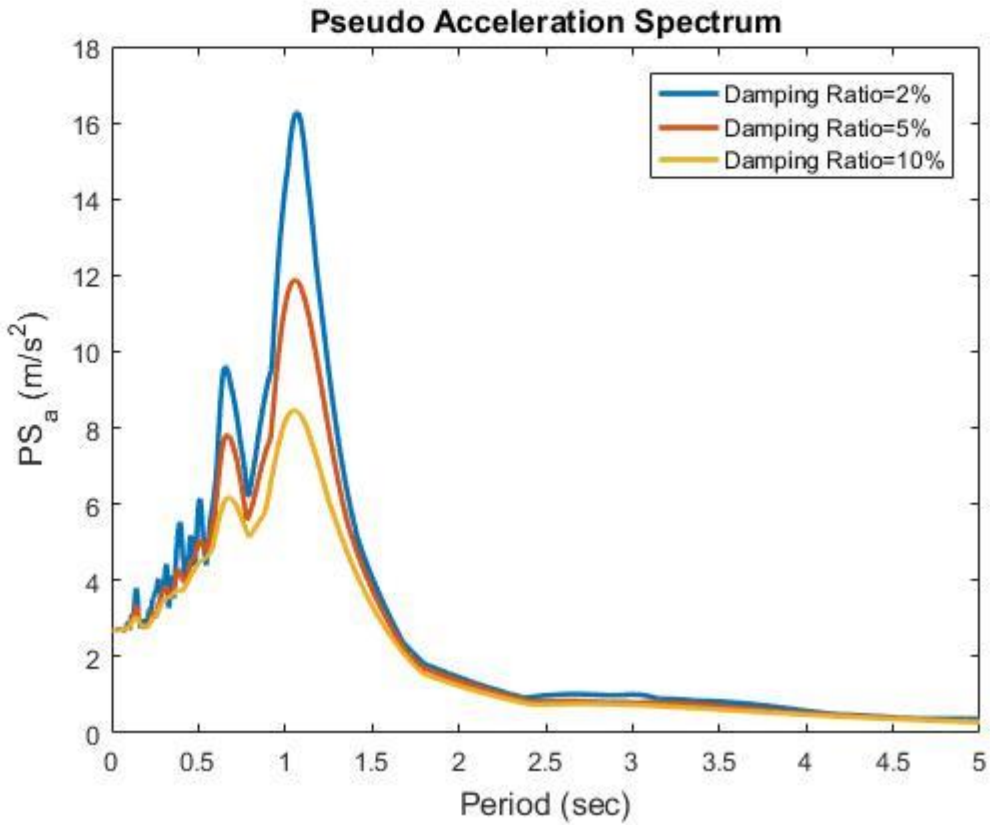
```
        delta_u_spectrum_2=delta_f_2/new_k_2;
        delta_u_dot_spectrum_2=2*delta_u_spectrum_2/delta_t-
2*u_g_dot_spectrum_2(j,1);
        delta_u_doubledot_spectrum_2=(4*delta_u_spectrum_2/(delta_t)^2)-
(4*u_g_dot_spectrum_2(j,1)/delta_t)-(2*u_g_doubledot_spectrum_2(j,1));

        u_g_spectrum_2(j+1,1)=u_g_spectrum_2(j,1)+delta_u_spectrum_2;
        u_g_dot_spectrum_2(j+1,1)=u_g_dot_spectrum_2(j,1)+delta_u_dot_spectrum_2;

u_g_doubledot_spectrum_2(j+1,1)=u_g_doubledot_spectrum_2(j,1)+delta_u_doubledot_spect
rum_2;
    end
    spectral_displacement_2(i,1)=max(abs((u_g_spectrum_2(:,1))));
    spectral_acceleration_2(i,1)=spectral_displacement_2(i)*(W_2(i))^2;
    pseudo_acceleration_2(i,1)=spectral_displacement_2(i)*(W_2(i))^2;
    T(i+1,1)=T(i)+delta_t;

end
u_g_doubledot(end)=[];
T(end)=[];

                   % This part will perform the calculation for 5% damping.

damping_5=5; %This defines the damping ratio.
u_g_spectrum_5=zeros(length(u_g_doubledot),1); %This creates an array for displacement
spectrum under 2% damping.
u_g_dot_spectrum_5=zeros(length(u_g_doubledot),1); %This creates an array for velocity
spectrum under 2% damping.
u_g_doubledot_spectrum_5=zeros(length(u_g_doubledot),1);  %This creates an array for
acceleration spectrum under 2% damping.
u_g_doubledot(end+1)=0; %This returns the zero at the end of ground acceleration.
T(1,1)=0.00+delta_t; %This defines the time.

for i=1:round(T_final/delta_t) %This part will perform the calculation
    W_5(i,1)=2*pi/T(i);
    mass_2=1;
    k_2=((W_5(i))^2)*mass_2;
    c_2=2*mass_2*W_5(i)*damping_5/100;
    new_k_2=k_2+2*c_2/delta_t+4*mass_2/(delta_t)^2;

    for j=1:length(u_g_spectrum_5)-1
        u_g_spectrum_5(1,1)=0;
        u_g_dot_spectrum_5(1,1)=0;
        u_g_doubledot_spectrum_5(1,1)=(-mass_2*u_g_doubledot(1,1)-
c_2*u_g_dot_spectrum_5(1,1)-k_2*u_g_spectrum_5(1,1))/mass_2;

        delta_f_2=-mass_2*(u_g_doubledot(j+1)-
u_g_doubledot(j))+(4*mass_2/delta_t+2*c_2)*u_g_dot_spectrum_5(j,1)+2*mass_2*u_g_doubl
edot_spectrum_5(j,1);
        delta_u_spectrum_2=delta_f_2/new_k_2;
        delta_u_dot_spectrum_2=2*delta_u_spectrum_2/delta_t-
2*u_g_dot_spectrum_5(j,1);
        delta_u_doubledot_spectrum_2=(4*delta_u_spectrum_2/(delta_t)^2)-
(4*u_g_dot_spectrum_5(j,1)/delta_t)-(2*u_g_doubledot_spectrum_5(j,1));

        u_g_spectrum_5(j+1,1)=u_g_spectrum_5(j,1)+delta_u_spectrum_2;
        u_g_dot_spectrum_5(j+1,1)=u_g_dot_spectrum_5(j,1)+delta_u_dot_spectrum_2;

u_g_doubledot_spectrum_5(j+1,1)=u_g_doubledot_spectrum_5(j,1)+delta_u_doubledot_spect
rum_2;
    end
    spectral_displacement_5(i,1)=max(abs((u_g_spectrum_5(:,1))));
    spectral_acceleration_5(i,1)=spectral_displacement_5(i)*(W_5(i))^2;
```

```
    pseudo_acceleration_5(i,1)=spectral_displacement_5(i)*(W_5(i))^2;
    T(i+1,1)=T(i)+delta_t;

end

                    % This part will perform the calculation for 5% damping.

damping_10=10; %This defines the damping ratio.
u_g_spectrum_10=zeros(length(u_g_doubledot),1);   %This   creates   an   array   for
displacement spectrum under 2% damping.
u_g_dot_spectrum_10=zeros(length(u_g_doubledot),1);  %This  creates  an  array  for
velocity spectrum under 2% damping.
u_g_doubledot_spectrum_10=zeros(length(u_g_doubledot),1); %This creates an array for
acceleration spectrum under 2% damping.
u_g_doubledot(end+1)=0; %This returns the zero at the end of ground acceleration.

for i=1:round(T_final/delta_t) %This part will perform the calculation
    W_10(i,1)=2*pi/T(i);
    mass_2=1;
    k_2=((W_10(i))^2)*mass_2;
    c_2=2*mass_2*W_10(i)*damping_10/100;
    new_k_2=k_2+2*c_2/delta_t+4*mass_2/(delta_t)^2;

    for j=1:length(u_g_spectrum_10)-1
        u_g_spectrum_10(1,1)=0;
        u_g_dot_spectrum_10(1,1)=0;
        u_g_doubledot_spectrum_5(1,1)=(-mass_2*u_g_doubledot(1,1)-
c_2*u_g_dot_spectrum_10(1,1)-k_2*u_g_spectrum_10(1,1))/mass_2;

        delta_f_2=-mass_2*(u_g_doubledot(j+1)-
u_g_doubledot(j))+(4*mass_2/delta_t+2*c_2)*u_g_dot_spectrum_10(j,1)+2*mass_2*u_g_doub
ledot_spectrum_5(j,1);
        delta_u_spectrum_2=delta_f_2/new_k_2;
        delta_u_dot_spectrum_2=2*delta_u_spectrum_2/delta_t-
2*u_g_dot_spectrum_10(j,1);
        delta_u_doubledot_spectrum_2=(4*delta_u_spectrum_2/(delta_t)^2)-
(4*u_g_dot_spectrum_10(j,1)/delta_t)-(2*u_g_doubledot_spectrum_5(j,1));

        u_g_spectrum_10(j+1,1)=u_g_spectrum_10(j,1)+delta_u_spectrum_2;
        u_g_dot_spectrum_10(j+1,1)=u_g_dot_spectrum_10(j,1)+delta_u_dot_spectrum_2;

u_g_doubledot_spectrum_5(j+1,1)=u_g_doubledot_spectrum_5(j,1)+delta_u_doubledot_spect
rum_2;
    end
    spectral_displacement_10(i,1)=max(abs((u_g_spectrum_10(:,1))));
    spectral_acceleration_10(i,1)=spectral_displacement_10(i)*(W_10(i))^2;
    pseudo_acceleration_10(i,1)=spectral_displacement_10(i)*(W_10(i))^2;
    T(i+1,1)=T(i)+delta_t;

end
u_g_doubledot(end)=[];
T(end)=[];

%plots the spectral displacement.

figure('Name','Spectral Displacement');
plot(T,spectral_displacement_2,'LineWidth',2);
hold on
plot(T,spectral_displacement_5,'LineWidth',2);
hold on
plot(T,spectral_displacement_10,'LineWidth',2);
legend('Damping Ratio=2%','Damping Ratio=5%','Damping Ratio=10%')
xlabel('Period (sec)','FontSize',12);
```

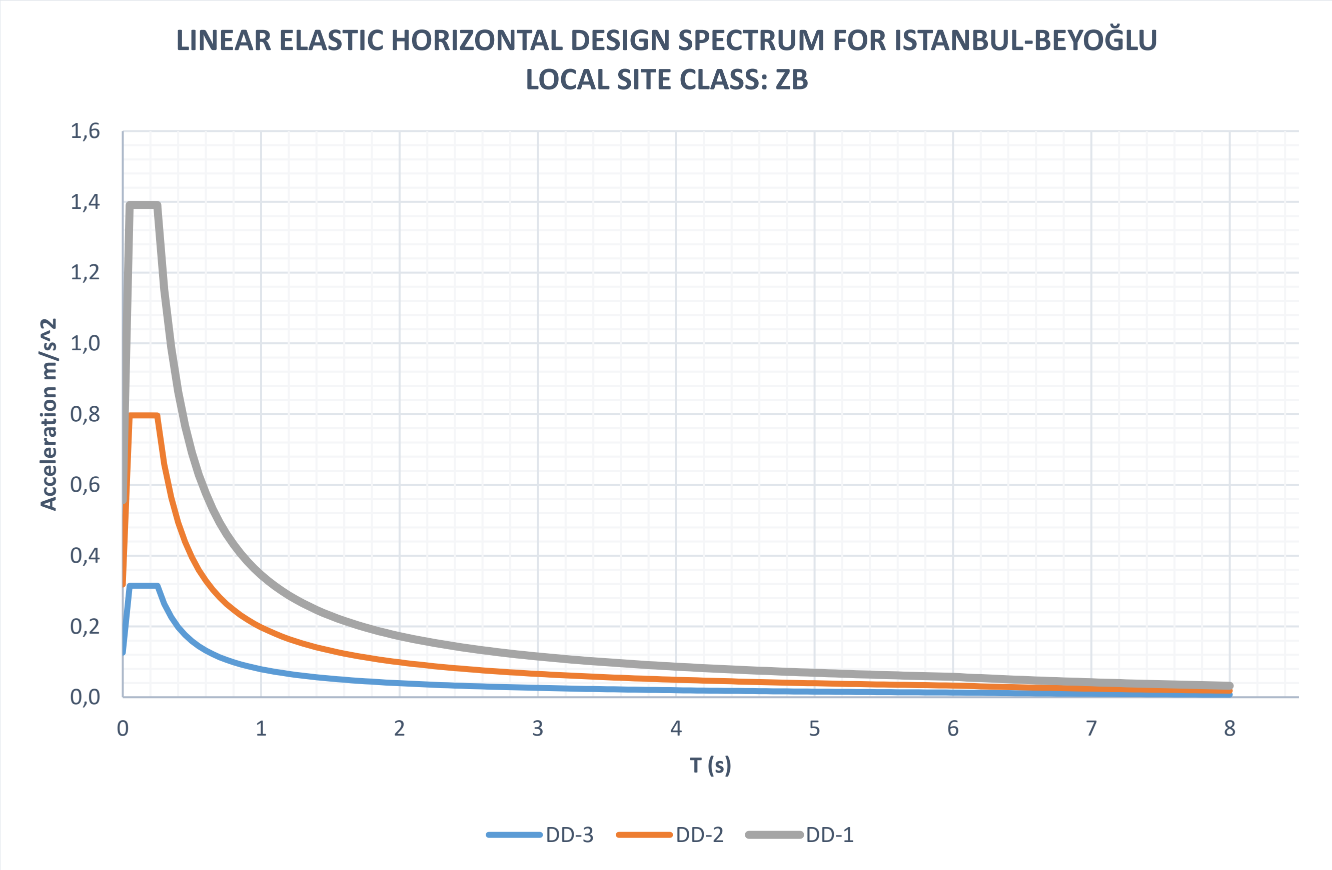
```
ylabel('S_d (m)','FontSize',12);
title('Displacement Spectrum','FontSize',12)

%plots the spectral acceleration.

figure('Name','Spectral Acceleration');
plot(T,spectral_acceleration_2,'LineWidth',2);
hold on
plot(T,spectral_acceleration_5,'LineWidth',2);
hold on
plot(T,spectral_acceleration_10,'LineWidth',2);
legend('Damping Ratio=2%','Damping Ratio=5%','Damping Ratio=10%')
xlabel('Period (sec)','FontSize',12);
ylabel('S_a (m/s^2)','FontSize',12);
title('Acceleration Spectrum','FontSize',12)

 %plots the pseudo acceleration.

figure('Name','Pseudo Acceleration');
plot(T,pseudo_acceleration_2,'LineWidth',2);
hold on
plot(T,pseudo_acceleration_5,'LineWidth',2);
hold on
plot(T,pseudo_acceleration_10,'LineWidth',2);
legend('Damping Ratio=2%','Damping Ratio=5%','Damping Ratio=10%')
xlabel('Period (sec)','FontSize',12);
ylabel('PS_a (m/s^2)','FontSize',12);
title('Pseudo Acceleration Spectrum','FontSize',12)
```



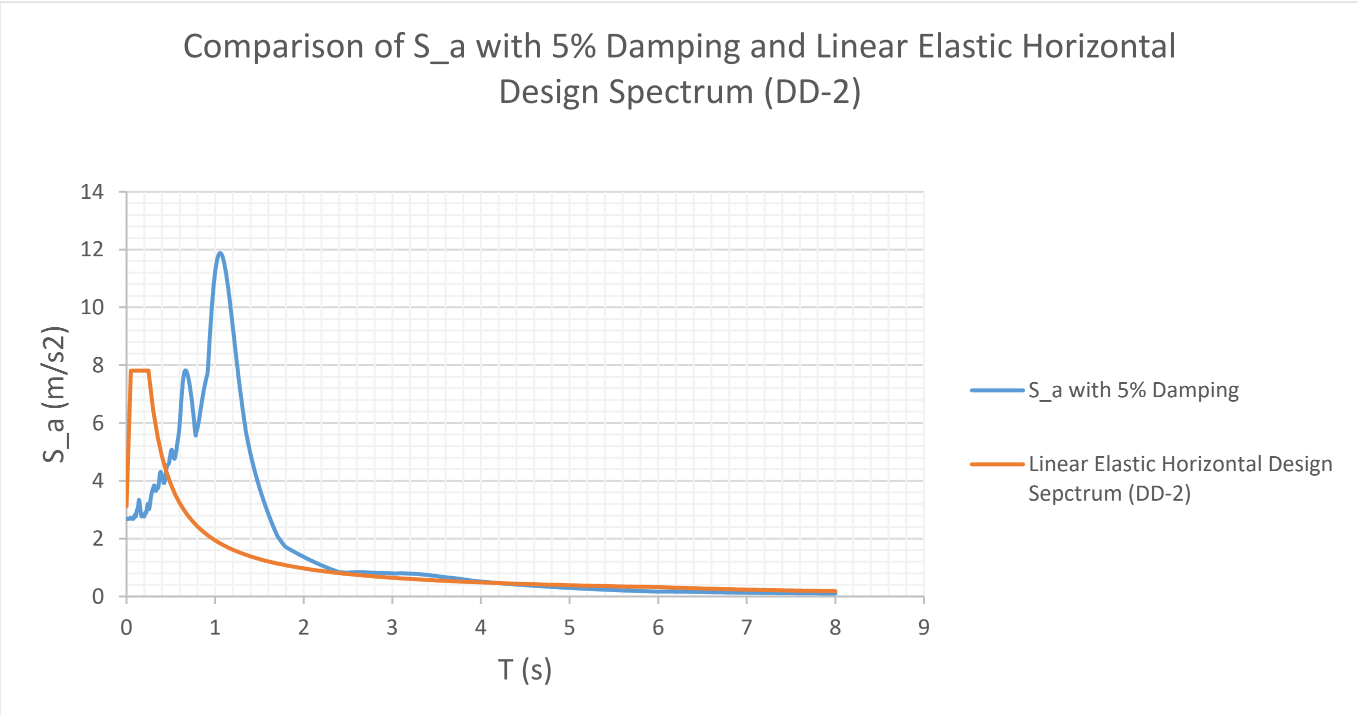

Acceleration Spectrum
Damping Ratio=2%
Damping Ratio=5%
Damping Ratio=10%
S_a (m/s^2)
Period (sec)


Pseudo Acceleration Spectrum
Damping Ratio=2%
Damping Ratio=5%
Damping Ratio=10%
PS_a (m/s^2)
Period (sec)

QUESTION-3



QUESTION-4



QUESTION-5

Damping ratio plays a prominent role in spectral acceleration, displacement and velocity, which is not calculated in this homework. It is because damping is the way of energy dissipation in structures and in question 3, it is obvious to see that the spectra of acceleration and displacement decreases when the damping ratios increase. It is important to note that damping ratio affects the spectral acceleration less in periods larger than 1.5-2 when compared to periods smaller than 1.5-2. This concludes that damping ratio will considerably affect the performance of the structures under earthquakes in smaller periods, yet it is not change the behavior of the structural system in longer periods, thereby not influencing the design parameter.