Problem 1

11 January 2022 09:59 AM

60/60

1)a EPP system using CDM method

EQ_HW2...


```
%Assignment #2 P1a-Inelastic response of SDOF system using
%Central Difference Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
del_t=0.005;
dt=0.005; % Time step for EPP analysis
% Refning the time axis with dt=0.005
t1=0:0.005:51.180;
% Adding zero padding to the given Earthquake excitation data
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20↙
sec.
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
Tn=0.5; %Natural Period of the system
Z=0.0; %Damping ratio
m=1; %Considering unit mass
Wn=(2*pi)/Tn; %Natural Frequency
k=m*Wn^2; %Linear elastic Stiffness
Wd=Wn*sqrt(1-Z^2); %Damped Natural Frequency
%Defining Parameters required A,B,C,D & A1,B1,C1,D1
A=exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t));
B=exp(-Z*Wn*del_t)*(sin(Wd*del_t)/Wd);
C=(((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((1-2*Z^2)/(Wd*del_t)-(Z/sqrt(1-Z^2)))*sin↙
(Wd*del_t)-(1+((2*Z)/(Wn*del_t)))*cos(Wd*del_t)))/Wn^2;
D=(1-((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((2*Z^2-1)/(Wd*del_t))*sin(Wd*del_t)+((2*Z)/↙
```

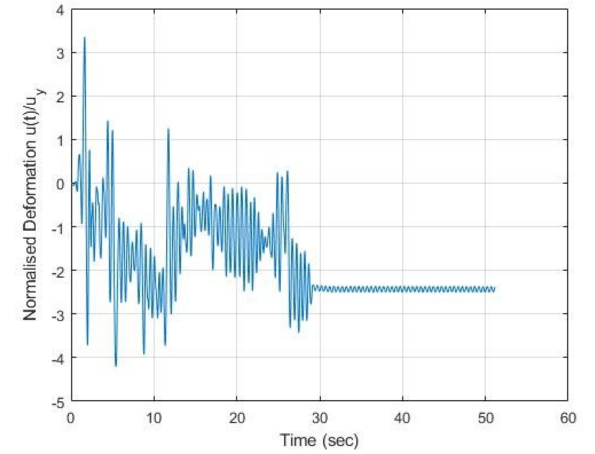


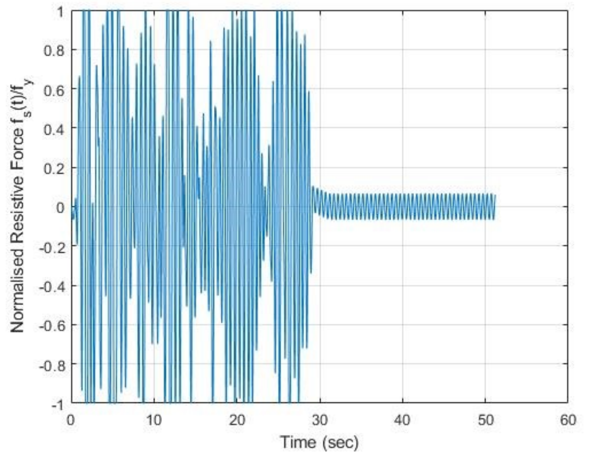
```
(Wn*del_t))*cos(Wd*del_t)))/Wn^2;
A1=-exp(-Z*Wn*del_t)*((Wn/sqrt(1-Z^2))*sin(Wd*del_t));
B1=exp(-Z*Wn*del_t)*(cos(Wd*del_t)-(Z/sqrt(1-Z^2))*sin(Wd*del_t));
C1=((-1/del_t)+exp(-Z*Wn*del_t)*(((Wn/(sqrt(1-Z^2)))+(Z/(del_t*sqrt(1-Z^2))))*sin↙
(Wd*del_t)+(cos(Wd*del_t)/del_t)))/Wn^2;
D1=(1-exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t)))/(Wn^2*del_t);
u=zeros(length(a_g1),1); %Initialising displacement response vector of the SDOF system
v=zeros(length(a_g1),1); %Initialising velocity response vector of the SDOF system
acc=zeros(length(a_g1),1);
```

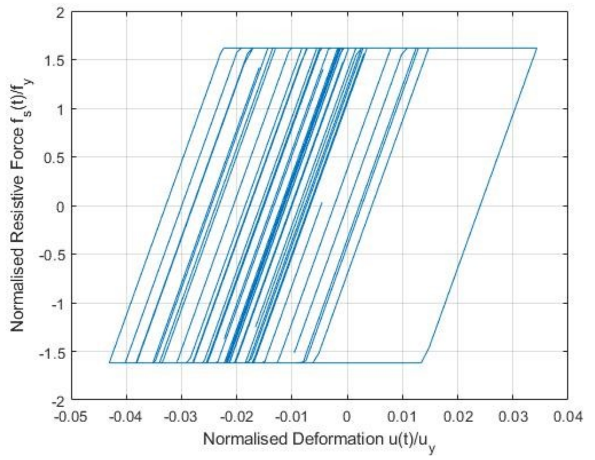
```
for i=1:length(a_g1)-1
    u(i+1)=A*u(i)+B*v(i)-C*a_g1(i)-D*a_g1(i+1);
    v(i+1)=A1*u(i)+B1*v(i)-C1*a_g1(i)-D1*a_g1(i+1);
    acc(i+1)=-a_g1(i+1)-2*Z*Wn*v(i+1)-Wn^2*u(i+1);
end
a_t=a_g1+acc;
umax=abs(max(u));
f_0=k*umax; %Max. Force for system to remain Linear Elastic
% Performing Inelastic Response Analysis
u_epp=zeros(length(a_g1),1);
v_epp=zeros(length(a_g1),1);
a_epp=zeros(length(a_g1),1);
%Initial calculations:
fs=zeros(length(a_g1),1);
fs(1)=k*u_epp(1);
Ry=8; %Yield Strength reduction factor
fy=f_0/Ry; %yield strength of the system
a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs(1))/m; %Initial acceleration
u_0=u_epp(1)-dt*v_epp(1)+0.5*dt^2*a_epp(1);
k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
b=(2*m)/dt^2; %Integration parameter
p_hat=0;du=0;fst=0;
for i=1:length(a_g1)-1
    if i==1
        p_hat=-m*a_g1(1)-a*u_0-fs(1)+b*u_epp(1);
        u_epp(2)=p_hat/k_hat;
        v_epp(1)=(u_epp(2)-u_0)/(2*dt);
        a_epp(1)=(u_epp(2)-2*u_epp(1)+u_0)/dt^2;
    else
        p_hat=-m*a_g1(i)-a*u_epp(i-1)-fs(i)+b*u_epp(i);
        u_epp(i+1)=p_hat/k_hat;
        v_epp(i)=(u_epp(i+1)-u_epp(i-1))/(2*dt);
        a_epp(i)=(u_epp(i+1)-2*u_epp(i)+u_epp(i-1))/dt^2;
    end
    du=u_epp(i+1)-u_epp(i);
    fst=fs(i)+k*du;
    if abs(fst)>fy
        fs(i+1)=sign(fst)*fy;
    else
        fs(i+1)=fst;
    end
end
u_m=max(abs(u_epp)); %Maximum displacement
meu=u_m/(fy/k);
%Finding the residual displacement
```

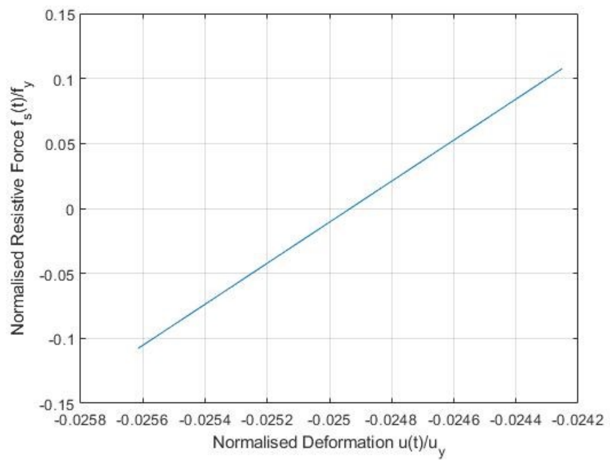


```
if fs(end)>0
    u_r=u_epp(end)-(fs(end)/k);
elseif fs(end)<0
    u_r=u_epp(end)+(fs(end)/k);
end
figure(1)
title('Normalised Time history of deformation response')
plot(t1,u_epp./(fy/k))
xlabel('Time (sec)');
ylabel('Normalised Deformation u(t)/u_{y}');
grid on
figure(2)
title('Normalised Time History of Resistive Force')
plot(t1,fs./fy);
xlabel('Time (sec)');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
figure(3)
title('Normalised Force vs Normalised Deformation Hysteritic response')
plot(u_epp,fs);
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
figure(4)
title('Normalised Force vs Normalised Deformation Hysteritic response for free
vibration phase only');
plot(u_epp(6238:(end),1),fs(6238:(end),1));
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
fprintf('Ductility Demand=%8.3e\n',meu);
fprintf('Normalised Permanent Deformation=%8.3e',u_r);
```









The calculated Ductility Demand (μ) = 4.204
Residual Normalized Deformation (Ur/Uy)= 2.453

1)b EPP system using KR-α Method:

```
%Assignment #2 P1b-Inelastic response of SDOF system using
%KR-Alpha Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
t1=0:0.005:51.180; % Refning the time axis with dt=0.005
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20↙
sec.
Tn=0.5; %Natural Period of the system
Z=0.0; %Damping ratio
m=1; %Considering unit mass
Wn=(2*pi)/Tn; %Natural Frequency
k=m*Wn^2; %Linear elastic Stiffness
f_0=k*0.082; %Using the maximum displacement value for this system from the previous↙
LE analysis
% Performing Inelastic Response Analysis
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
u_epp=zeros(length(a_g1),1);
v_epp=zeros(length(a_g1),1);
a_epp=zeros(length(a_g1),1);
dt=0.005; % Time step for EPP analysis
%Initial calculations:
fs=zeros(length(a_g1),1);
fs(1)=k*u_epp(1); %Initial resistive force
Ry=8; %Yield Strength reduction factor
```

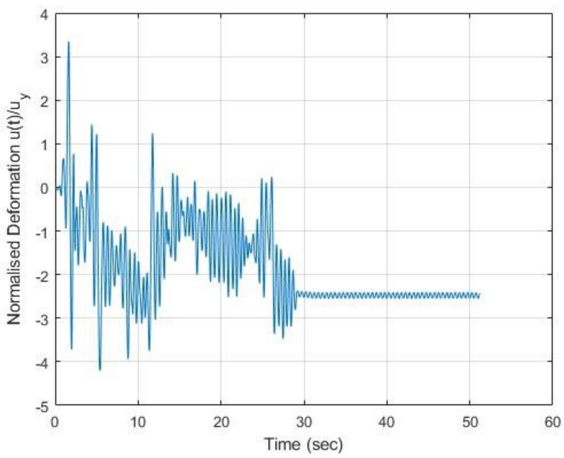


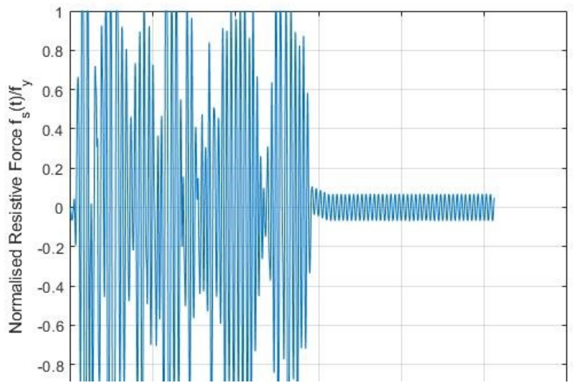
```
fy=f_0/Ry; %yield strength of the system
a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs(1))/m; %Initial acceleration
%Defining the constants to be used
p_inf=1;
am=(2*p_inf-1)/(p_inf+1); af=p_inf/(p_inf+1);
Y=0.5-am+af; B=0.25*(0.5+Y)^2;
a1=m/(m+Y*dt*2*Z*m*Wn+B*dt^2*k);
a2=a1*(0.5+Y);
a3=(am*m+af*Y*dt*2*Z*m*Wn+af*B*dt^2*k)/(m+Y*dt*2*Z*m*Wn+B*dt^2*k);
for i=1:length(a_g1)-1
    v_epp(i+1)=v_epp(i)+dt*a1*a_epp(i);
    u_epp(i+1)=u_epp(i)+dt*v_epp(i)+dt^2*a2*a_epp(i);
    du=u_epp(i+1)-u_epp(i);
    fst=fs(i)+k*du;
    if abs(fst)>fy %state determination
        fs(i+1)=sign(fst)*fy;
    else
        fs(i+1)=fst;
    end
```

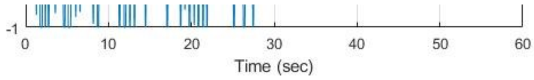
```
    end
    intvel=(1-af)*v_epp(i+1)+af*v_epp(i);
    intf=(1-af)*fs(i+1)+af*fs(i);
    intp=(1-af)*(-m*a_g1(i+1))+af*(-m*a_g1(i));
    a_hat=(intp-2*Z*m*Wn*intvel-intf)/m;
    a_epp(i+1)=(a_hat-a3*a_epp(i))/(1-a3);
end
u_m=max(abs(u_epp)); %Maximum displacement
meu=u_m/(fy/k);
%Finding the residual displacement
if fs(end)>0
    u_r=u_epp(end)-(fs(end)/k);
elseif fs(end)<0
    u_r=u_epp(end)+(fs(end)/k);
end
figure(1)
title('Normalised Time history of deformation response')
plot(t1,u_epp./(fy/k))
xlabel('Time (sec)');
ylabel('Normalised Deformation u(t)/u_{y}');
grid on
figure(2)
title('Normalised Time History of Resistive Force')
plot(t1,fs./fy);
xlabel('Time (sec)');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
figure(3)
title('Normalised Force vs Normalised Deformation Hysteritic response')
plot(u_epp,fs);
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
figure(4)
title('Normalised Force vs Normalised Deformation Hysteritic response for free
vibration phase only');
plot(u_epp(6238:(end),1),fs(6238:(end),1));
```

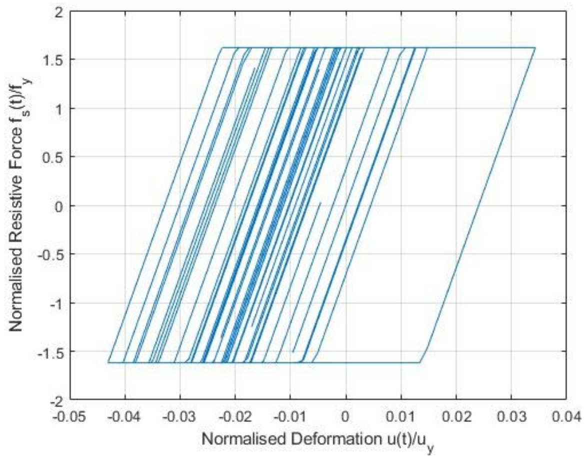


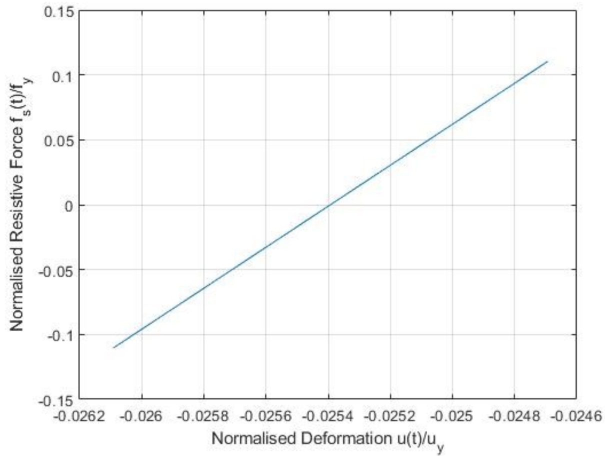
```
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
fprintf('Ductility Demand=%8.3e\n',meu);
fprintf('Normalised Permanent Deformation=%8.3e',u_r);
```











The calculated Ductility Demand (μ) = 4.200
Residual Normalized Deformation (Ur/Uy)= 2.48

1)C EPP system using Newmark-Beta with Newton Raphson Iteration

EQ_HW2...

29/1/22 6:56 PM EQ_HW2_P1c.m 1 of 3

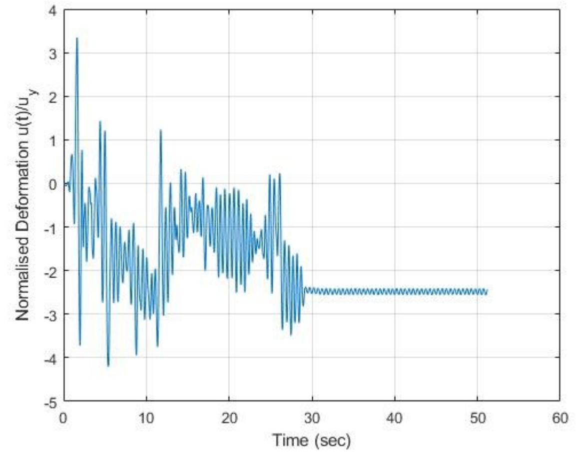
```
%Assignment #2 P1b-Inelastic response of SDOF system using
%KR-Alpha Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
```

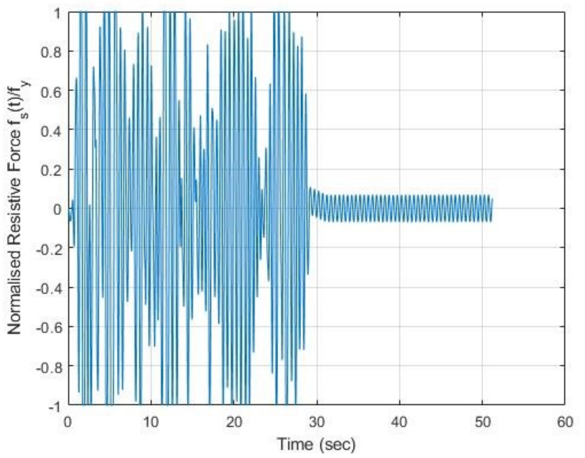
```
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
t1=0:0.005:51.180; % Refning the time axis with dt=0.005
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20↙
sec.
Tn=0.5; %Natural Period of the system
Z=0.0; %Damping ratio
m=1; %Considering unit mass
Wn=(2*pi)/Tn; %Natural Frequency
k=m*Wn^2; %Linear elastic Stiffness
c=2*Z*Wn*m; % Damping coefficient
f_0=k*0.082; %Using the maximum displacement value for this system from the previous↙
LE analysis
% Performing Inelastic Response Analysis
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
u_epp=zeros(length(a_g1),1);
v_epp=zeros(length(a_g1),1);
a_epp=zeros(length(a_g1),1);
dt=0.005; % Time step for EPP analysis
%Initial calculations:
fs=zeros(length(a_g1),1);
fs(1)=k*u_epp(1); %Initial resistive force
```

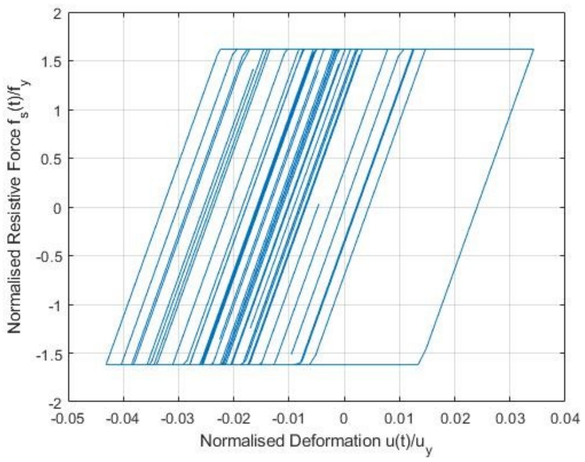


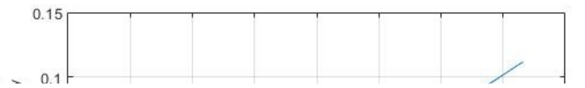
```
Ry=8; %Yield Strength reduction factor
fy=f_0/Ry; %yield strength of the system
a_epp(1)=(-m*a_g1(1)-c*v_epp(1)-fs(1))/m; %Initial acceleration
%Defining the constants to be used
Y=1/2; B=1/4; % Gamma and Beta for Newmark's Constant Avg Acc. method
a1 = (1/(B*dt^2))*m+(Y/(B*dt))*c;
a2 = (1/(B*dt))*m+(Y/B-1)*c;
a3 = (1/(2*B)-1)*m+dt*(Y/(2*B)-1)*c;
kt=zeros(length(a_g1),1);
kt(i)=k; %Initial tangent stiffness equal to the LE stiffness
k_hat=zeros(length(a_g1),1);
p_hat=zeros(length(a_g1),1);
p_resi=zeros(length(a_g1),1);
for i=1:length(a_g1)-1
    u_epp(i+1)=u_epp(i);fs(i+1)=fs(i);kt(i+1)=kt(i);
    p_hat(i+1)=-m*a_g1(i+1)+a1*u_epp(i)+a2*v_epp(i)+a3*a_epp(i);
    p_resi(i+1)=p_hat(i+1)-fs(i+1)-a1*u_epp(i+1);
    while abs(p_resi(i+1))>0.001
        k_hat(i+1)=kt(i+1)+a1;
        du=p_resi(i+1)/k_hat(i+1);
        u_epp(i+1)=u_epp(i+1)+du;
        fst=fs(i+1)+kt(i+1)*du;
        if abs(fst)>fy %state determination
            fs(i+1)=sign(fst)*fy;
            kt(i+1)=0;
        else
            fs(i+1)=fst;
            kt(i+1)=k;
        end
        p_resi(i+1)=p_hat(i+1)-fs(i+1)-a1*u_epp(i+1);
    end
    v_epp(i+1)=(Y/(B*dt))*(u_epp(i+1)-u_epp(i))+(1-(Y/B))*v_epp(i)+dt*(1-(Y/(2*B)))↙
*a_epp(i);
    a_epp(i+1)=(1/(B*dt^2))*(u_epp(i+1)-u_epp(i))-(1/(B*dt))*v_epp(i)-(1/(2*B)-1)↙
*a_epp(i);
end
u_m=max(abs(u_epp)); %Maximum displacement
meu=u_m/(fy/k);
%Finding the residual displacement
if fs(end)>0
    u_r=abs(u_epp(end)-(fs(end)/k))/(fy/k);
elseif fs(end)<0
    u_r=abs(u_epp(end)+(fs(end)/k))/(fy/k);
end
figure(1)
title('Normalised Time history of deformation response')
plot(t1,u_epp./(fy/k))
xlabel('Time (sec)');
ylabel('Normalised Deformation u(t)/u_{y}');
grid on
figure(2)
title('Normalised Time History of Resistive Force')
plot(t1,fs./fy);
xlabel('Time (sec)');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
```

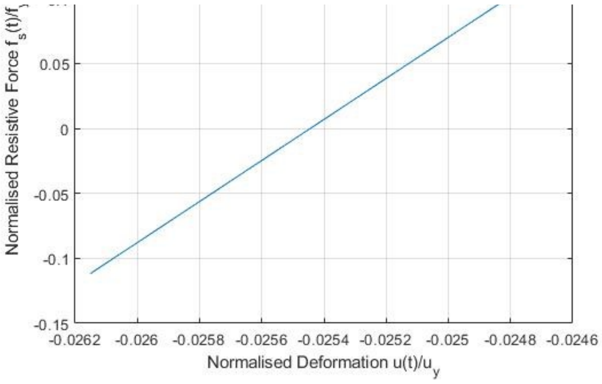
```
grid on
figure(3)
title('Normalised Force vs Normalised Deformation Hysteritic response')
plot(u_epp,fs);
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
figure(4)
title('Normalised Force vs Normalised Deformation Hysteritic response for free
vibration phase only');
plot(u_epp(6238:(end),1),fs(6238:(end),1));
xlabel('Normalised Deformation u(t)/u_{y}');
ylabel('Normalised Resistive Force f_{s}(t)/f_{y}');
grid on
fprintf('Ductility Demand=%8.3e\n',meu);
fprintf('Normalised Permanent Deformation=%8.3e',u_r);
```











The calculated Ductility Demand (μ) = 4.207

Residual Normalized Deformation (Ur/Uy)= 2.482

Comments:

We observe that the 3 methods give very similar results for the EPP system. The CDM and the KR-α method are both explicit whereas the Newton-Raphson driven Newmark-Beta method is implicit.