MATLAB CODE FOR NON-LINEAR DYNAMIC RESPONSE HISTORY ANALYSIS OF MULTI-DEGREE OF FREEDOM SYSTEMS

George Papazafeiropoulos

Description

Matlab code for the application of the non-linear dynamic response history analysis (Non Linear DRHA) of multi-degree of freedom (MDOF) structures is presented. Direct integration of the equations of motion is used for the calculation of the non-linear dynamic response of a MDOF system subject to dynamic loading. The dynamic response history analysis procedure with direct integration proceeds incrementally, by solving the MDOF system equations for each time step.

For the direct integration of equations of motion, the function NLDRHA_MDOF_1D.m is used. See the following examples:

- example_SDOF_Clough_Penzien.m
- example_SDOF_Rajasekaran.m
- example_Shear_Building_2_NPTEL.m
- example_Energy_SDOF_Chopra.m
- example_SDOF_Chopra.m

for more details.

The function NLDRHA_MDOF_1D.m needs the function Bilinear_Kin_1D.m in order to run properly. In the last function the bilinear elastoplastic hysteretic model with elastic viscous damping is implemented for SDOF or MDOF systems. The MDOF structure modeled with this function consists of lumped masses connected with stiffness and damping elements in series, along a single dimension (i.e. one-dimensional MDOF system, hence the function is named as '1D'). Each lumped mass has one degree of freedom. The first degree of freedom is at the top of the structure and the last at its fixed base. However, the last degree of freedom is not included in the input arguments of the function, i.e. not contained in ndof, as it is always fixed. The nonlinear stiffness is virtually of the bilinear type, where an initial stiffness and a post-yield stiffness are defined. The unloading or reloading curve of this model are parallel to the initial loading curve, and a hysteresis loop is created by continuously loading and unloading the structure above its yield limit. This behavior can be viewed as hardening of the kinematic type.

All the above functions can be used for acceleration time histories of a constant time step size. If this is not the case, then the acceleration time history needs to be resampled by using the MATLAB program file function RESAMPLE.m. The user is encouraged to see the example

- example_Resampling_Nonuniform_Time_History.m

in this last case.

The present code is accompanied by 6 examples in which its application is presented. These examples are taken from various standard textbooks or other material. The results of the examples are verified by the results of the application of the present code.
The author is open to any suggestions or recommendations that the users may have.

Keywords: Dynamic loading, Response history, Structural design, Earthquake engineering, shock, modal superposition, direct integration, damping, resampling, bilinear, elastoplastic, kinematic, hysteretic, nonlinear.

REFERENCES

[1] Chopra, A. K. 2019, Dynamics of Structures, Theory and Applications to Earthquake Engineering.

[2] MathWorks, Inc. MATLAB R2017b. Natick, MA: MathWorks, Inc.; 2017.

[3] National Programme on Technology Enhanced Learning (NPTEL).

[4] Clough, R. W., & Penzien, J. 2003, Dynamics of Structures. Berkeley, CA, USA: Computers & Structures.

[5] Rajasekaran, S., 2009. Structural dynamics of earthquake engineering: theory and application using MATHEMATICA and MATLAB. Elsevier.

ABOUT THE AUTHOR

Google Scholar profile	https://scholar.google.com/citations?hl=en&user=JXRPD7UAAAAJ
RG profile	https://www.researchgate.net/profile/George_Papazafeiropoulos2
LinkedIn profile	https://www.linkedin.com/in/george-papazafeiropoulos-8b6b6025/
MathWorks profile	https://www.mathworks.com/matlabcentral/profile/authors/2438703-george-papazafeiropoulos
Facebook profile	https://www.facebook.com/george.papazafeiropoulos.5
e-mail	gpapazafeiropoulos@yahoo.gr

Dependency Report

The Dependency Report shows dependencies among MATLAB files in a folder (Learn More).

Rerun This Report   Run Report on Current Folder

Show child functions Show parent functions (current folder only)

Show subfunctions

Built-in functions and files in toolbox/matlab are not shown

Report for Folder C:\Users\pc\Desktop\NLDRHA

MATLAB File List	Children (called functions)	Parents (calling functions, current dir. only)
Bilinear_Kin_1D		NLDRHA_MDOF_1D
NLDRHA_MDOF_1D	current dir : Bilinear_Kin_1D	example_Energy_SDOF_Chopra example_SDOF_Chopra example_SDOF_Clough_Penzien example_SDOF_Rajasekaran example_Shear_Building_2_NPTEL
example_Energy_SDOF_Chopra	current dir : NLDRHA_MDOF_1D	
example_SDOF_Chopra	current dir : NLDRHA_MDOF_1D	
example_SDOF_Clough_Penzien	current dir : NLDRHA_MDOF_1D	
example_SDOF_Rajasekaran	current dir : NLDRHA_MDOF_1D	
example_Shear_Building_2_NPTEL	current dir : NLDRHA_MDOF_1D	
helpFun		help_Bilinear_Kin_1D help_NLDRHA_MDOF_1D
help_Bilinear_Kin_1D	current dir : helpFun	
help_NLDRHA_MDOF_1D	current dir : helpFun	

Energy dissipation of SDOF system (Chopra,2020)

Contents

Statement of the problem

- This example comes from the 'Dynamics of Structures - Theory and Applications to Earthquake Engineering' by A. Chopra (edition 2020), Chapter 7, Figure 7.9.1.
- Calculate the variation of the various energy quantities with time for two SDOF systems subjected to the El Centro ground motion. The results presented are for a linearly elastic system with natural period Tn=0.5 sec and damping ratio ksi=0.05, and for an elastoplastic system with the same properties in the elastic range and normalized strength fybar=0.25.

Initialization of structural input data

Set the number of degrees of freedom of the structure, which is equal to the number of its storeys.

```
nDOFs=1;
```

Set the lumped mass equal to unity. This is permitted since the various response measures are normalized.

```
m=1;
```

Set the lateral stiffness.

```
Tn=0.5;
k_hi=m*(2*pi/Tn)^2;
```

Set the spatial distribution of the effective earthquake forces. Earthquake forces are applied at all dofs of the structure.

```
inflvec=ones(nDOFs,1);
```

Load earthquake data

Earthquake acceleration time history of the El Centro earthquake (El Centro, 1940, El Centro Terminal Substation Building)

```
D=load('elcentro.dat');
dt=D(2,1)-D(1,1);
xgtt=9.81*D(:,2);
```

```
t=dt*(0:(numel(xgtt)-1));
```

Post-yield stiffness

```
k_lo=0.001*k_hi;
```

Yield displacement for elastic response

```
uy1=1e10;
```

Set the critical damping ratio

```
ksi=0.05;
```

Initial displacement

```
u0=zeros(nDOFs,1);
```

Initial velocity

```
ut0=zeros(nDOFs,1);
```

Dynamic Response History Analysis (DRHA) with direct integration

Perform DRHA analysis for linear structure

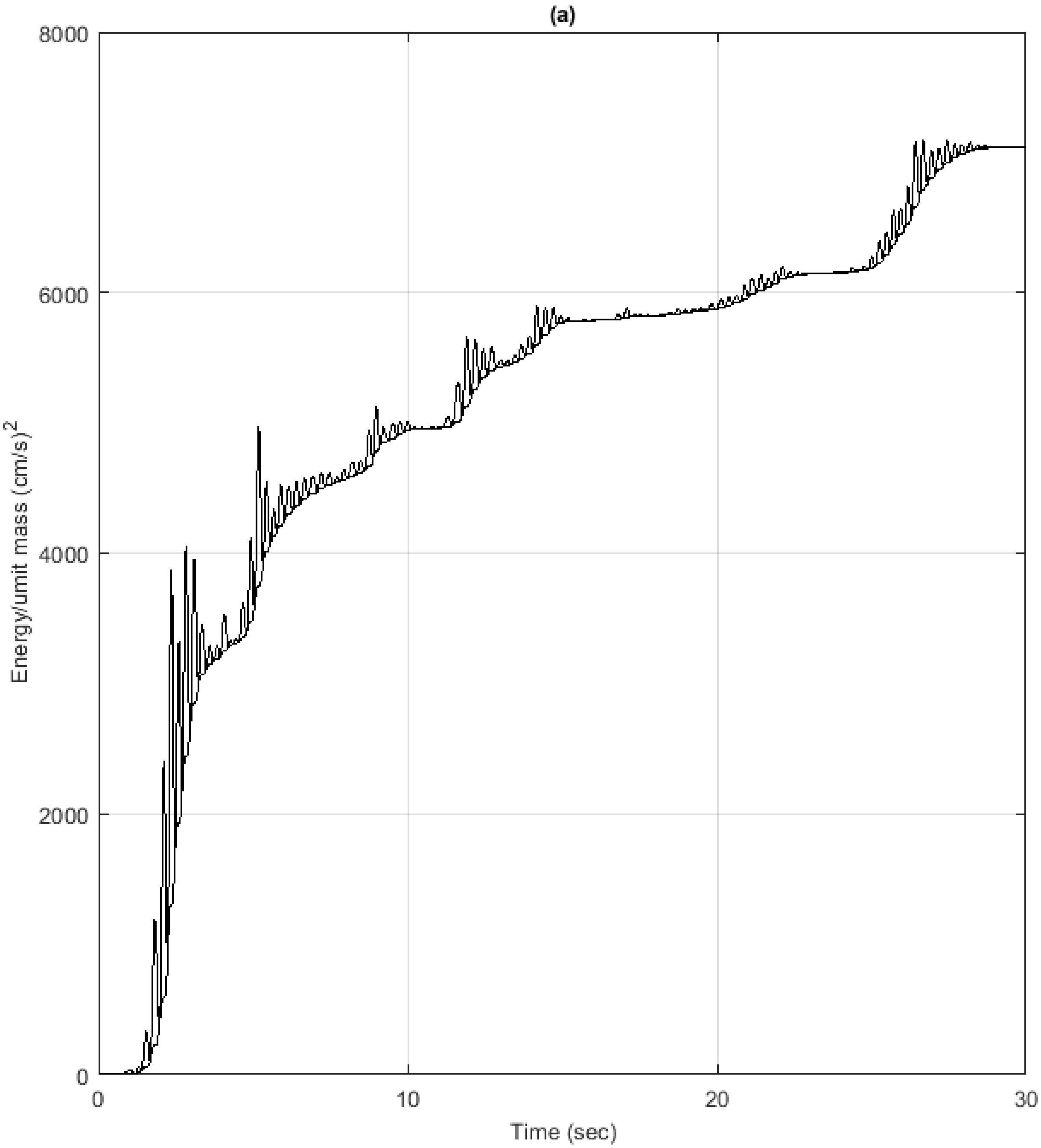
```
[u,ut,utt,Fs,Ey,Es,Ed,jiter] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy1,ksi,u0,ut0);
```

Plot results

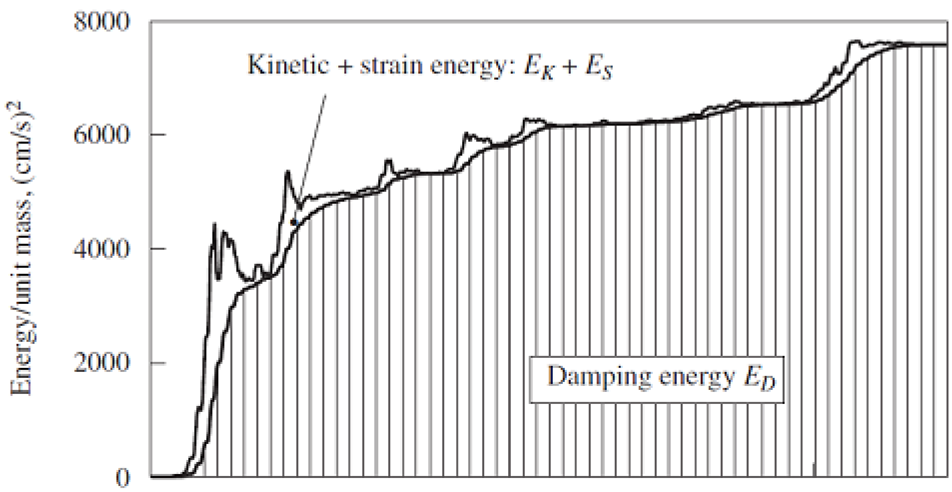
Plot the damping energy and strain energy of the linearly elastic SDOF system.

```
FigHandle=figure('Name','Elastic response','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 700, 750]);
plot(t',cumsum(Ed)*1e4,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t',cumsum(Ed)*1e4+Es*1e4,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold off
xlim([0,30])
ylim([0,8000])
set(gca,'XTick', (0:10:30));
set(gca,'YTick', (0:2000:8000));
xlabel('Time (sec)','FontSize',10);
ylabel('Energy/umit mass (cm/s)^2','FontSize',10);
title('(a)','FontSize',10)
```

```
grid on
```



Verify with the Figure 7.9.1(a) of 'Dynamics of Structures - Theory and Applications to Earthquake Engineering' by A. Chopra.



(a)

Dynamic Response History Analysis (DRHA) with direct integration

normalized yield strength

```
fybar=0.25;
```

Yield displacement for nonlinear response

```
uy2=fybar*max(abs(u));
```

Perform DRHA analysis for nonlinear structure

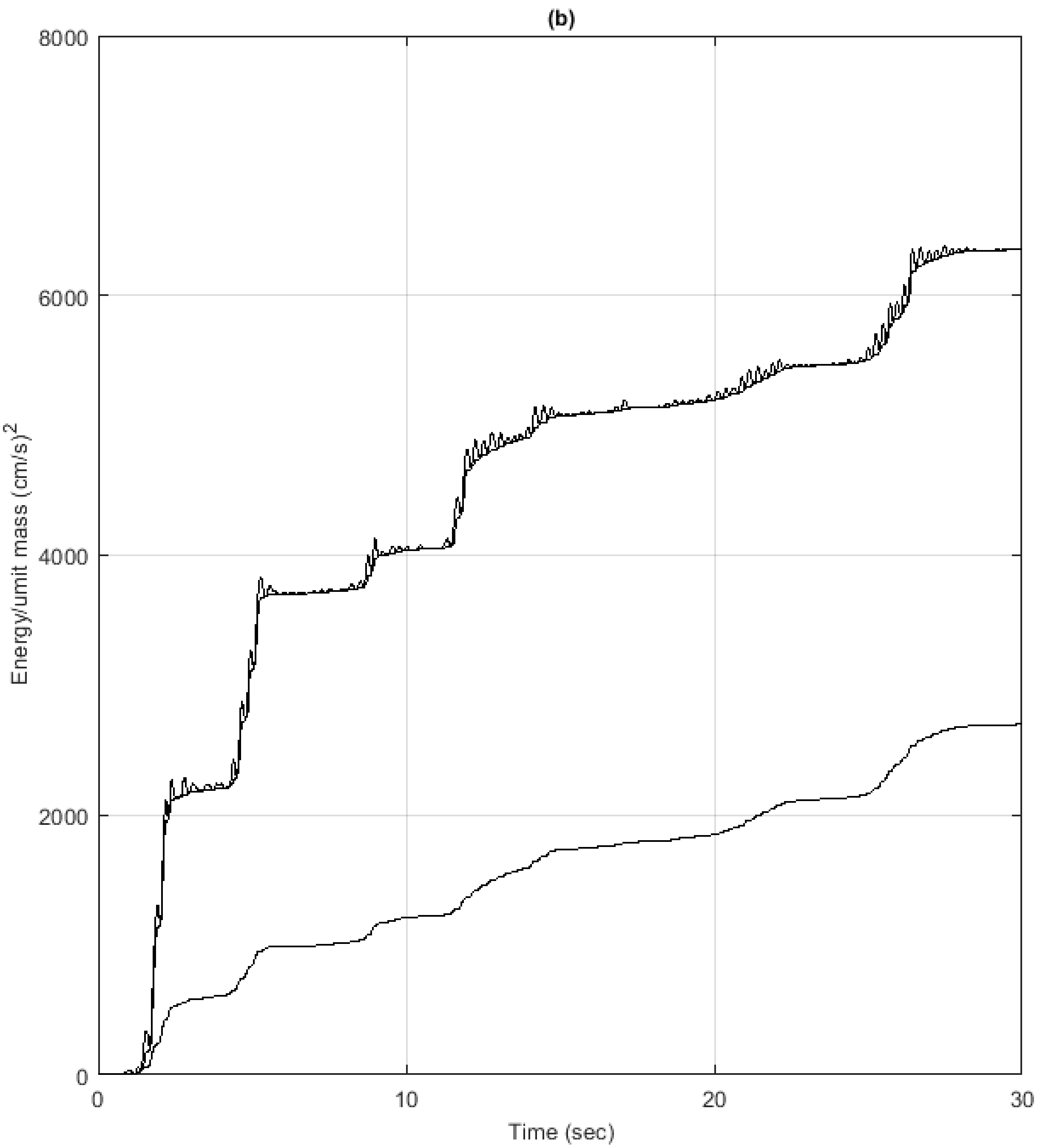
```
[u,ut,utt,Fs,Ey,Es,Ed,jiter] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy2,ksi,u0,ut0);
```

Plot results

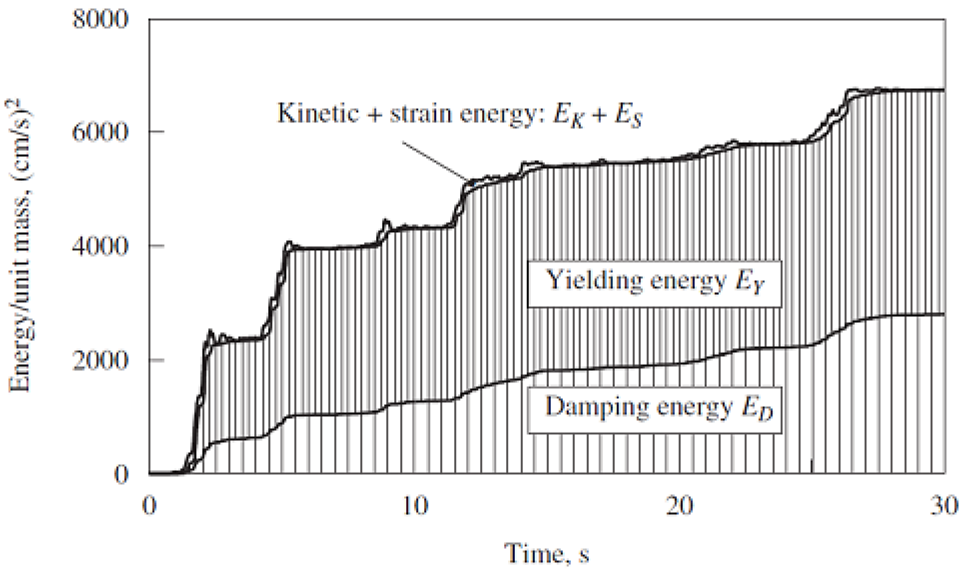
Plot the damping energy and strain energy of the linearly elastic SDOF system.

```
FigHandle=figure('Name','Elastoplastic response','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 700, 750]);
plot(t',cumsum(Ed)*1e4,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t',cumsum(Ed)*1e4+cumsum(Ey)*1e4-Es*1e4,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
plot(t',cumsum(Ed)*1e4+cumsum(Ey)*1e4,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold off
xlim([0,30])
ylim([0,8000])
set(gca,'XTick', (0:10:30));
set(gca,'YTick', (0:2000:8000));
xlabel('Time (sec)','FontSize',10);
ylabel('Energy/umit mass (cm/s)^2','FontSize',10);
title('(b)','FontSize',10)
```

```
grid on
```



Verify with the Figure 7.9.1(b) of 'Dynamics of Structures - Theory and Applications to Earthquake Engineering' by A. Chopra.



(b)

Copyright

- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

Published with MATLAB® R2017b

Resample acceleration time history with variable time step size

Contents

Statement of the problem

- The linear dynamic response history analysis (DRHA) algorithms that are included in this package work only for acceleration time histories defined in terms of time step with constant size. This may not be the case in acceleration time histories with nonuniform time step (i.e. variable time step size). In this case, the time history needs to be resampled, so that an equivalent time history is defined with constant time step size, suitable for use in the various functions of this package. Here an example is provided for converting an acceleration time history with nonuniform time step into an equivalent acceleration time history with uniform (constant size) time step.

Load earthquake data

Initial acceleration time history

```
D=load('elcentro_truncated.dat');
dt=D(2,1)-D(1,1);
t=D(:,1);
xgtt=D(:,2);
```

Define resampling parameters

Set the desired time step of the new acceleration time history that is produced after resampling.

```
dt_new=0.02;
```

Upsampling factor

```
p=1;
```

Downsampling factor

```
q=1;
```

Resampling procedure

Sample rate

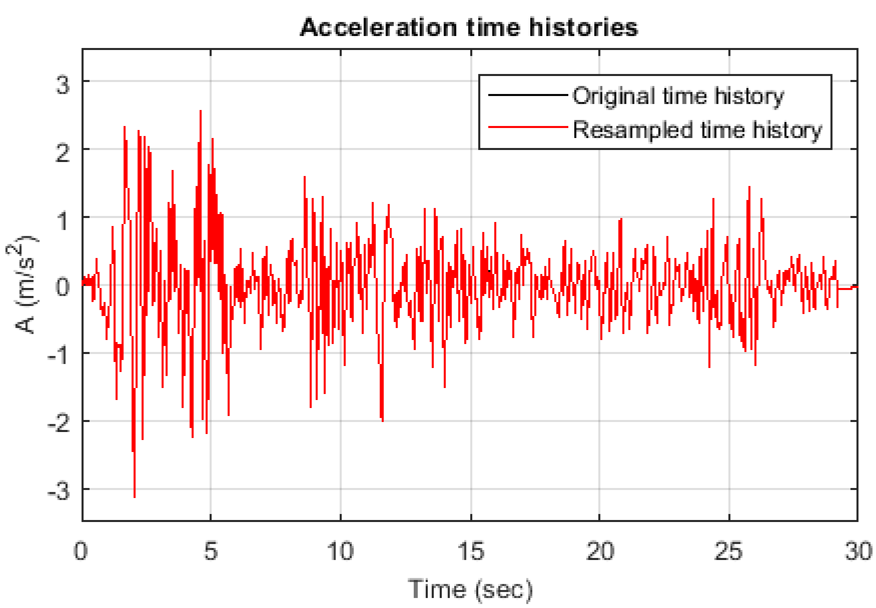
```
fs = 1/dt_new;
```

Resample the original acceleration time history into a new one with constant time step size equal to dt_new

```
[xgtt_new,t_new] = resample(xgtt,t,fs,p,q);
```

Plot original and new acceleration time history

```
FigHandle=figure('Name','Acceleration time histories','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 500, 300]);
plot(t,xgtt,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t_new,xgtt_new,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[1 0 0],'markeredgecolor','r')
grid on
xlim([0,30])
ylim([-3.5,3.5])
xlabel('Time (sec)','FontSize',10);
ylabel('A (m/s^2)','FontSize',10);
title(['Acceleration time histories'],...
    'FontSize',10)
legend('Original time history','Resampled time history')
```



Check uniformity of the time step of the original acceleration time history

```
if any(diff(diff(t))>1e-14)
    disp('The original acceleration time history is nonuniform')
else
    disp('The original acceleration time history is uniform')
end
```

```
The original acceleration time history is nonuniform
```

Check uniformity of the time step of the resampled acceleration time history

```
if any(diff(diff(t_new))>1e-14)
    disp('The resampled acceleration time history is nonuniform')
else
    disp('The resampled acceleration time history is uniform')
end
```

```
The resampled acceleration time history is uniform
```

Copyright

- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

Published with MATLAB® R2017b

SDOF system, dynamic analysis with direct integration (Chopra,2020)

Contents

Statement of the problem

- This example comes from the 'Dynamics of Structures - Theory and Applications to Earthquake Engineering' by A. Chopra (edition 2020), Chapter 7, Figure 7.4.2.
- Consider a linearly elastic system with weight w, natural vibration period Tn = 0.5 sec, and no damping. The excitation selected is the El Centro ground motion. Based on the time variation of the elastic resisting force fS, the peak value of this force fo is given by fo/mg=1.37.
- Consider the response of an elastoplastic system having the same mass and initial stiffness as the linearly elastic system, with normalized strength fybar =0.125. The yield strength of this system is fy=0.125*fo, where fo=1.37*m*g (Fig. 7.4.1); therefore, fy=0.125*(1.37*m*g)=0.171*m*g. Calculate the following for the elastoplastic system: (a) deformation; (b) resisting force and acceleration; (c) time intervals of yielding; (d) force–deformation relation.

Initialization of structural input data

Set the number of degrees of freedom of the structure, which is equal to the number of its storeys.

```
nDOFs=1;
```

Set the lumped mass equal to unity. This is permitted since the various response measures are normalized.

```
m=1;
```

Set the lateral stiffness.

```
Tn=0.5;
k_hi=m*(2*pi/Tn)^2;
```

Set the spatial distribution of the effective earthquake forces. Earthquake forces are applied at all dofs of the structure.

```
inflvec=ones(nDOFs,1);
```

Load earthquake data

Earthquake acceleration time history of the El Centro earthquake (El Centro, 1940, El Centro Terminal Substation Building)

```
D=load('elcentro.dat');
dt=D(2,1)-D(1,1);
xgtt=9.81*D(:,2);
```

```
t=dt*(0:(numel(xgtt)-1));
```

Post-yield stiffness

```
k_lo=0.001*k_hi;
```

Yield displacement

```
g=9.81;
fy=0.125*(1.37*m*g);
uy=fy/k_hi;
```

Set the critical damping ratio

```
ksi=0.0;
```

Initial displacement

```
u0=zeros(nDOFs,1);
```

Initial velocity

```
ut0=zeros(nDOFs,1);
```

Dynamic Response History Analysis (DRHA) with direct integration

Perform DRHA analysis for nonlinear structure

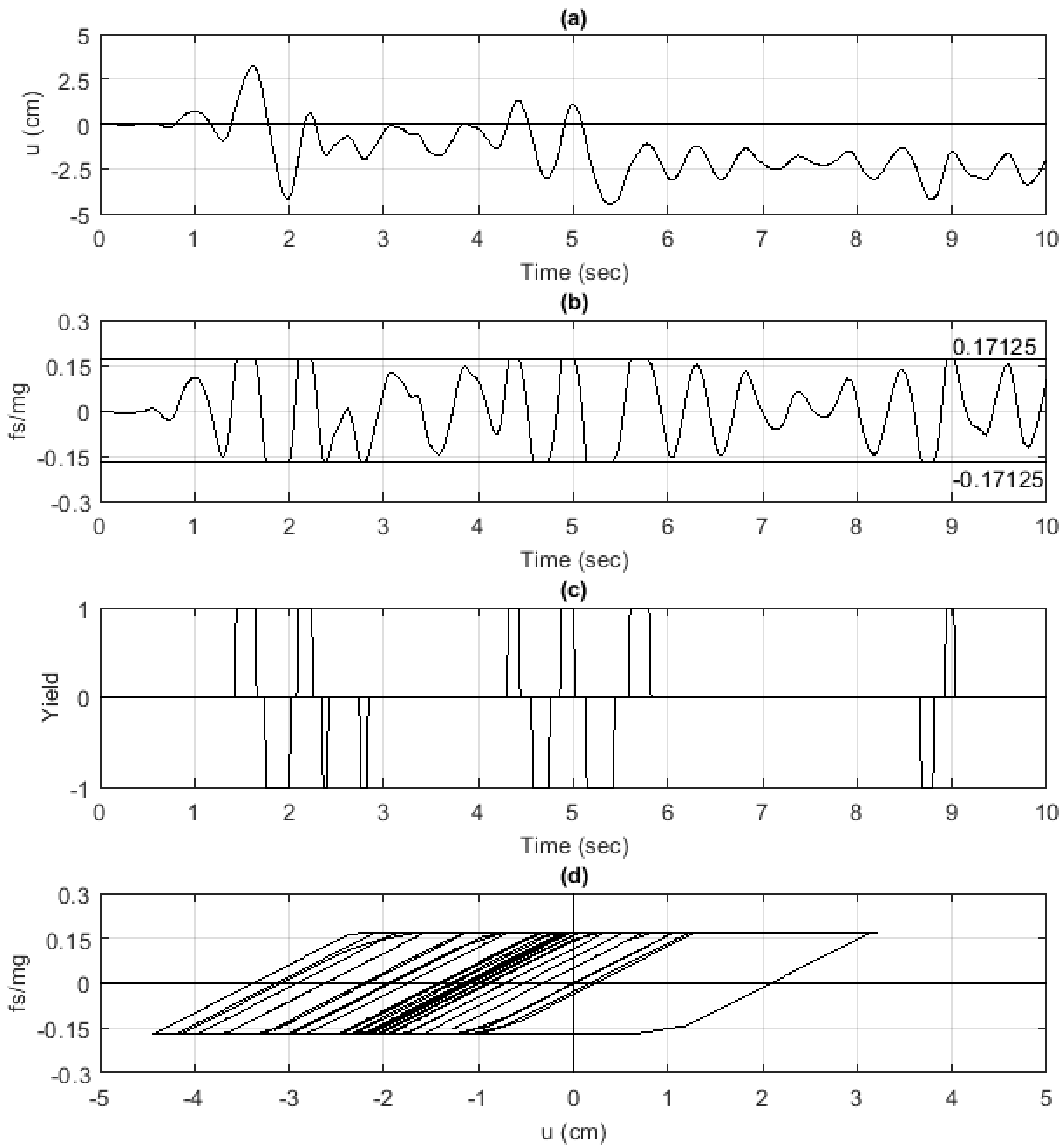
```
[u,ut,utt,Fs,Ey,Es,Ed,jiter] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy,ksi,u0,ut0);
```

Plot results

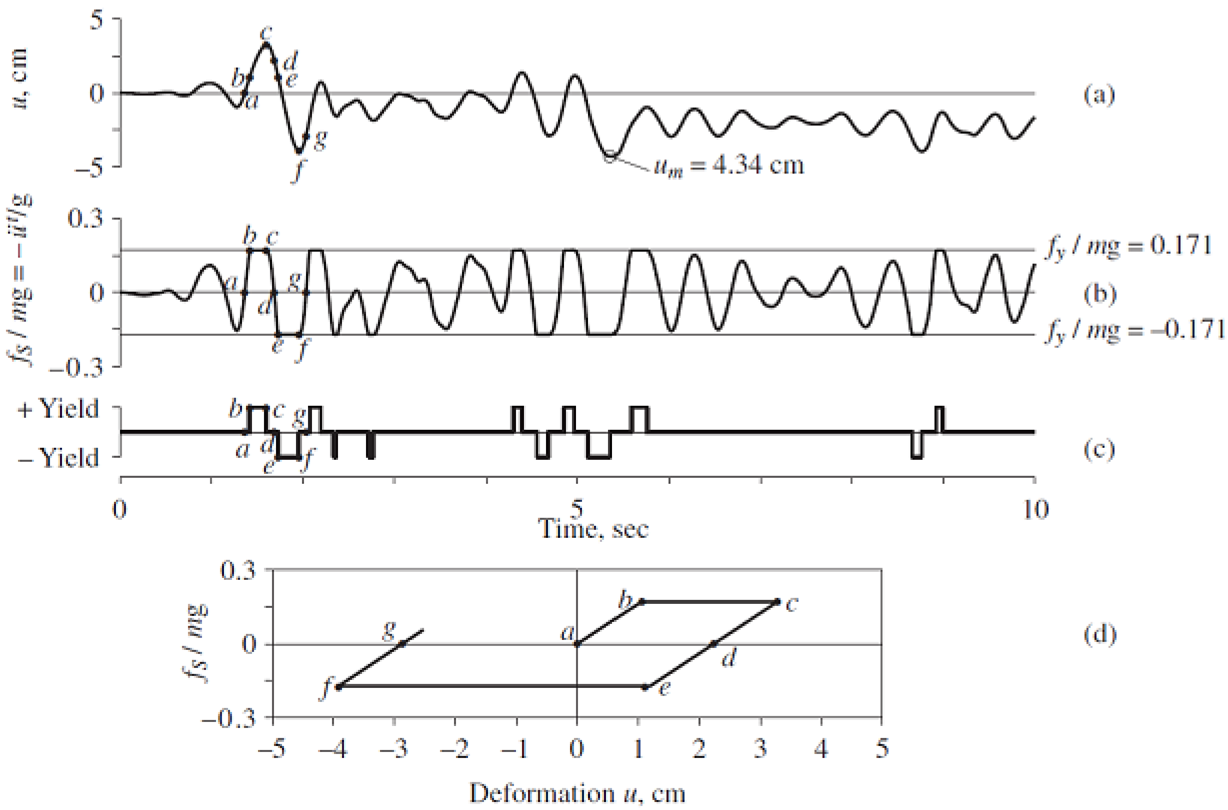
Plot the following: (a) deformation; (b) resisting force and acceleration; (c) time intervals of yielding; (d) force–deformation relation.

```
FigHandle=figure('Name','Elastoplastic response','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 700, 750]);
%1st subplot
subplot(4,1,1)
plot(t,100*u(1,:),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t,zeros(size(t)),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold off
xlim([0,10])
ylim([-5,5])
set(gca,'XTick', (0:10));
```

```
set(gca,'YTick', (-5:2.5:5));
xlabel('Time (sec)','FontSize',10);
ylabel('u (cm)','FontSize',10);
title('(a)','FontSize',10)
grid on
%2nd subplot
subplot(4,1,2)
plot(t,Fs/(m*g),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t,fy/(m*g)*ones(size(t)),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
plot(t,-fy/(m*g)*ones(size(t)),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
text(9,0.22,num2str(fy/(m*g)))
text(9,-0.22,num2str(-fy/(m*g)))
hold off
xlim([0,10])
ylim([-0.3,0.3])
set(gca,'XTick', (0:10));
set(gca,'YTick', [-0.3,-0.15,0,0.15,0.3]);
xlabel('Time (sec)','FontSize',10);
ylabel('fs/mg','FontSize',10);
title('(b)','FontSize',10)
grid on
%3rd subplot
subplot(4,1,3)
plot(t,(abs(Fs-fy)<0.1)-(abs(Fs+fy)<0.1),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t,zeros(size(t)),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold off
xlim([0,10])
ylim([-1,1])
set(gca,'XTick', (0:10));
set(gca,'YTick', [-1,0,1]);
xlabel('Time (sec)','FontSize',10);
ylabel('Yield','FontSize',10);
title('(c)','FontSize',10)
grid on
%4th subplot
subplot(4,1,4)
plot(100*u(1,:),Fs/(m*g),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot([-5,5],zeros(size([-5,5])),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
plot(zeros(size([-0.3,0.3])),[-0.3,0.3],'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold off
xlim([-5,5])
ylim([-0.3,0.3])
set(gca,'XTick', (-5:5));
set(gca,'YTick', (-0.3:0.15:0.3));
xlabel('u (cm)','FontSize',10);
ylabel('fs/mg','FontSize',10);
title('(d)','FontSize',10)
grid on
```



Verify each subplot [(a)-(d)] with the Figure 7.4.2 of 'Dynamics of Structures - Theory and Applications to Earthquake Engineering' by A. Chopra.



- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

Elastoplastic SDOF frame, dynamic analysis with direct integration (Clough & Penzien, 2003)

Contents

Statement of the problem

- This example comes from the 'Dynamics of Structures' book by Clough & Penzien (edition 2003), Example E7-2.
- The response of the elastoplastic SDOF frame shown in Fig. E73 to the loading history indicated is required.
- Input parameters: k=5kips/in, m=0.1kips*sec^2/in, c=0.2kips*sec/in, uy=1.2in, dt=0.1

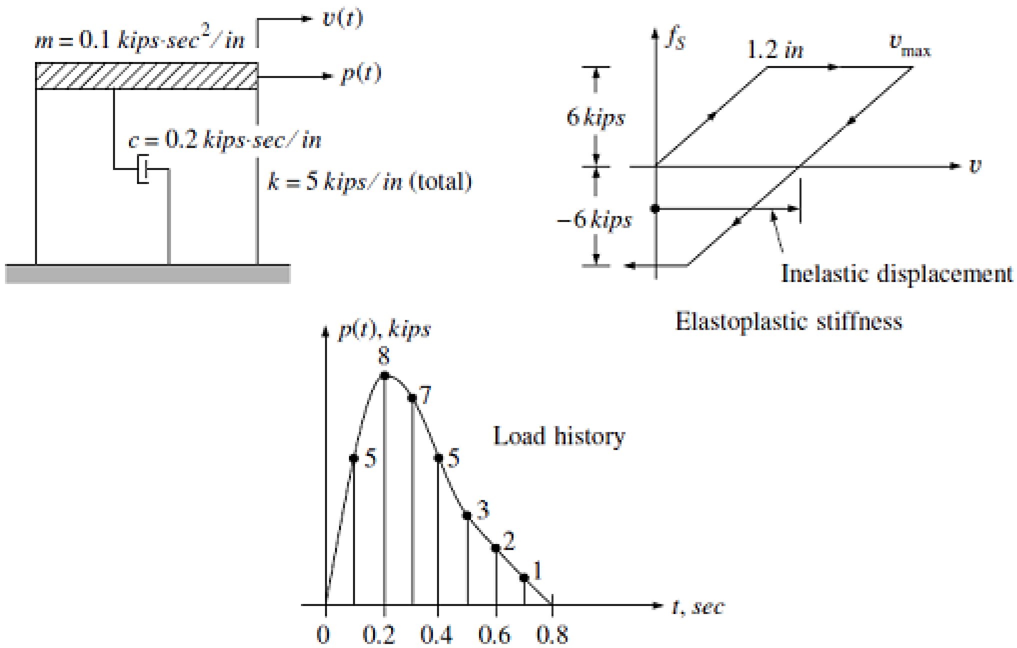


FIGURE E7-3
Elastoplastic frame and dynamic loading.

Initialization of structural input data

Set the number of degrees of freedom of the structure, which is equal to the number of its storeys.

```
nDOFs=1;
```

Set the lateral stiffness.

```
k_hi=5;
```

Set the lumped mass.

```
m=0.1;
```

Set the spatial distribution of the effective earthquake forces. Earthquake forces are applied at all dofs of the structure.

```
inflvec=ones(nDOFs,1);
```

Load earthquake data

Acceleration time history imposed at the base of the structure

```
dt=0.1;
xgtt=-[0;4;7;8;7;5;3;2;1;0.5;0;0;0]/m;
t=0:dt:1.2;
```

Post-yield stiffness

```
k_lo=0.001*k_hi;
```

Yield displacement (two values, a practically infinity value for the linear structure and a finite value for the nonlinear structure)

```
uy1=1e10;
uy2=1.2;
```

Set the critical damping ratio

```
ccrit=2*sqrt(k_hi*m);
c=0.2;
ksi=c/ccrit;
```

Initial displacement

```
u0=zeros(nDOFs,1);
```

Initial velocity

```
ut0=zeros(nDOFs,1);
```

Dynamic Response History Analysis (DRHA) with direct integration

Perform DRHA analysis for the non-yielding linear structure (uy1->inf)

```
[u1,ut1,utt1,Fs1,Ey1,Es1,Ed1,jiter1] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy1,ksi,u0,ut0);
```

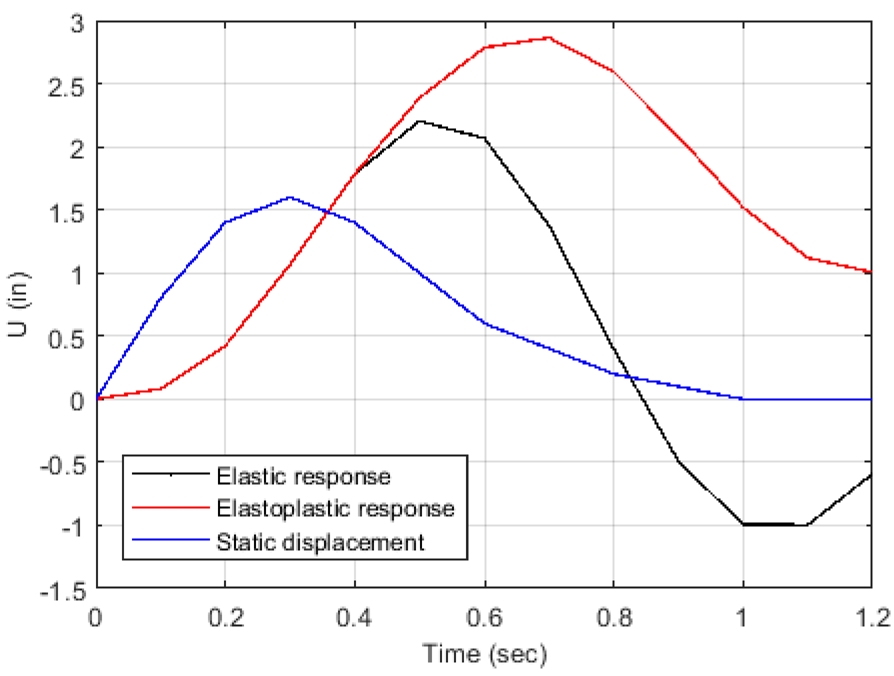
Perform DRHA analysis for the yielding nonlinear structure

```
[u2,ut2,utt2,Fs2,Ey2,Es2,Ed2,jiter2] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy2,ksi,u0,ut0);
```

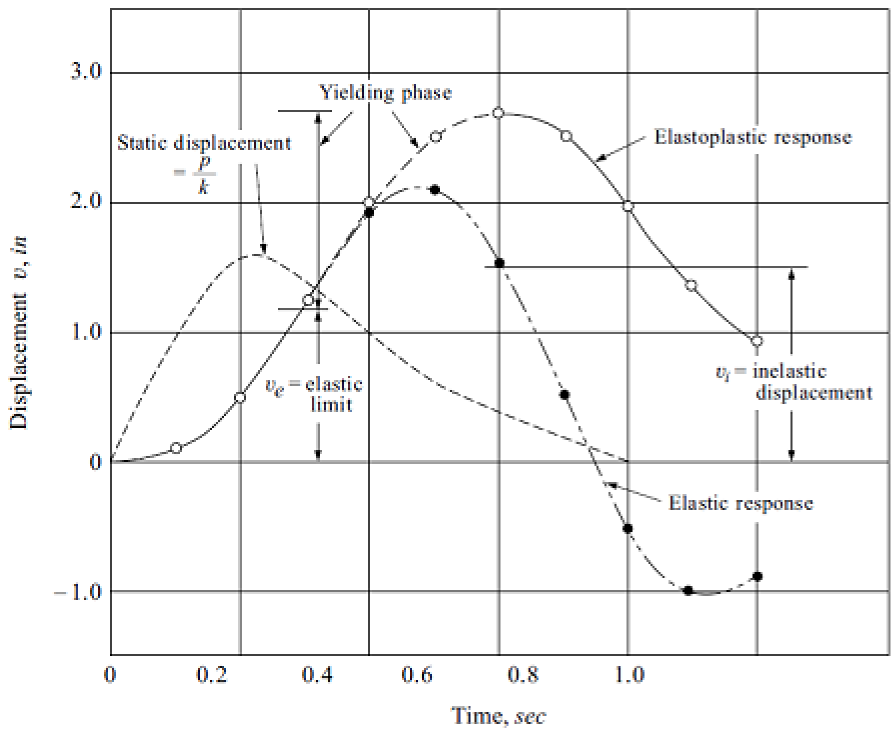
Plot results

Plot the displacement response.

```
FigHandle=figure('Name','Displacement','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 500, 350]);
plot(t,u1,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
hold on
plot(t,u2,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[1 0 0],'markeredgecolor',[1 0 0])
plot(t,-xgtt'*m/k_hi,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 1],'markeredgecolor',[0 0 1])
hold off
grid on
xlim([0,1.2])
ylim([-1.5,3])
xlabel('Time (sec)','FontSize',10);
ylabel('U (in)','FontSize',10);
legend({'Elastic response','Elastoplastic response','Static displacement'},...
    'Location','Southwest')
```



Verify with Figure E7-4 of 'Dynamics of Structures', by Clough & Penzien



Copyright

- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

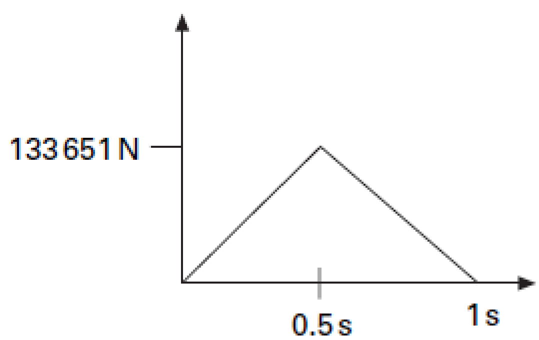
Published with MATLAB® R2017b

Single-storey shear frame, dynamic analysis with direct integration (Rajasekaran, 2009)

Contents

Statement of the problem

- This example comes from the 'Structural dynamics of earthquake engineering Theory and application using MATHEMATICA and MATLAB' book by S. Rajasekaran (edition 2009), Example 7.7.
- A shear frame structure shown in Fig. 7.4a is subjected to time varying force shown in Fig. 7.21. Evaluate the elastic and elasto-plastic response of the structure.
- Input parameters: k=1897251N/m, m=43848kg, c=7767.7Ns/m, Rm=66825.6N, dt=0.001
- NOTE: For the damping coefficient, the value of the Matlab code program 7.10 is valid, c=7767.7Ns/m, and not the one given in the problem statement c=34605.4Ns/m.



Initialization of structural input data

Set the number of degrees of freedom of the structure, which is equal to the number of its storeys.

```
nDOFs=1;
```

Set the lateral stiffness.

```
k_hi=1897251;
```

Set the lumped mass.

```
m=43848;
```

Set the spatial distribution of the effective earthquake forces. Earthquake forces are applied at all dofs of the structure.

```
inflvec=ones(nDOFs,1);
```

Load earthquake data

Acceleration time history imposed at the base of the structure

```
dt=0.001;
xgtt=-[133651/m/0.5*(0:dt:0.5),133651/m/0.5*(1-((0.5+dt):dt:1)),...
    zeros(size((1+dt):dt:3))]';
t=0:dt:3;
```

Post-yield stiffness

```
k_lo=0.001*k_hi;
```

Yield displacement (two values, a practically infinity value for the linear structure and a finite value for the nonlinear structure)

```
uy1=1e10;
Rm=66825.6;
uy2=Rm/k_hi;
```

Set the critical damping ratio

```
ccrit=2*sqrt(k_hi*m);
c=7767.7;
ksi=c/ccrit;
```

Initial displacement

```
u0=zeros(nDOFs,1);
```

Initial velocity

```
ut0=zeros(nDOFs,1);
```

Dynamic Response History Analysis (DRHA) with direct integration

Perform DRHA analysis for the non-yielding linear structure (uy1->inf)

```
[u1,ut1,utt1,Fs1,Ey1,Es1,Ed1,jiter1] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy1,ksi,u0,ut0);
```

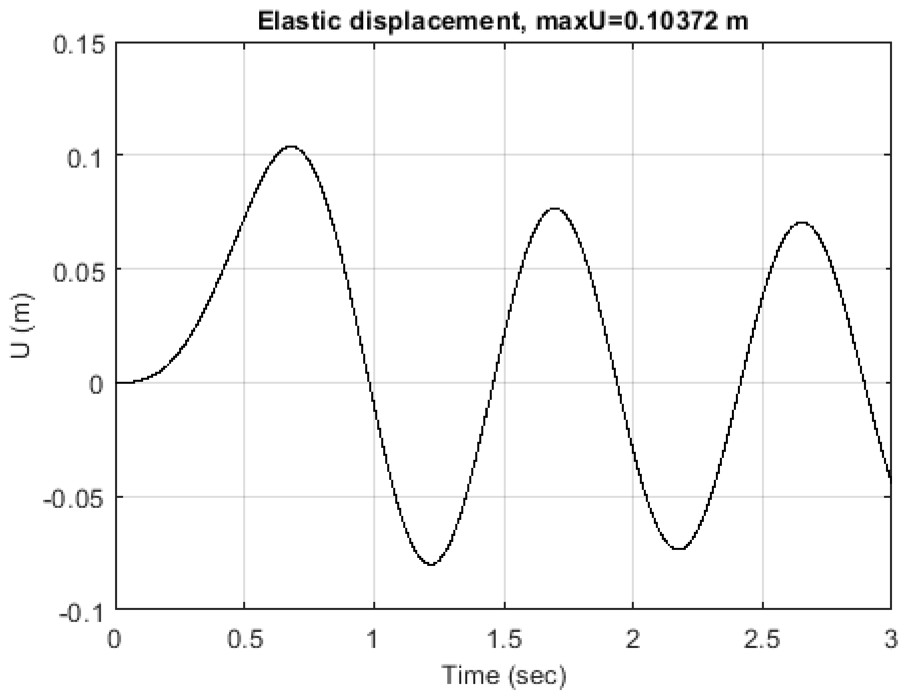
Perform DRHA analysis for the yielding nonlinear structure (uy2=Rm/k_hi)

```
[u2,ut2,utt2,Fs2,Ey2,Es2,Ed2,jiter2] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy2,ksi,u0,ut0);
```

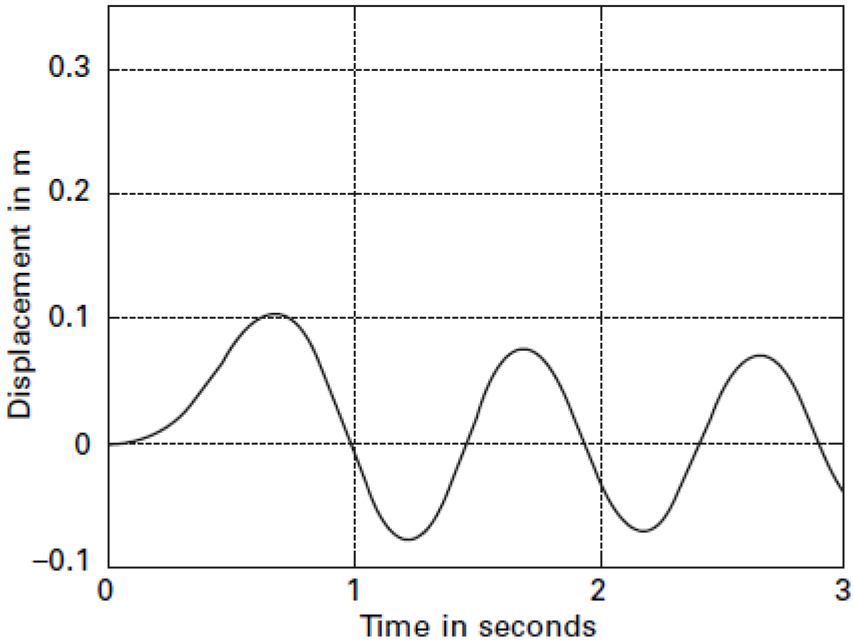
Plot results

Plot the elastic displacement response.

```
FigHandle=figure('Name','Elastic displacement','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 500, 350]);
plot(t,u1,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,3])
ylim([-0.1,0.15])
xlabel('Time (sec)','FontSize',10);
ylabel('U (m)','FontSize',10);
title(['Elastic displacement, maxU=',num2str(max(abs(u1))),' m'],...
    'FontSize',10)
```



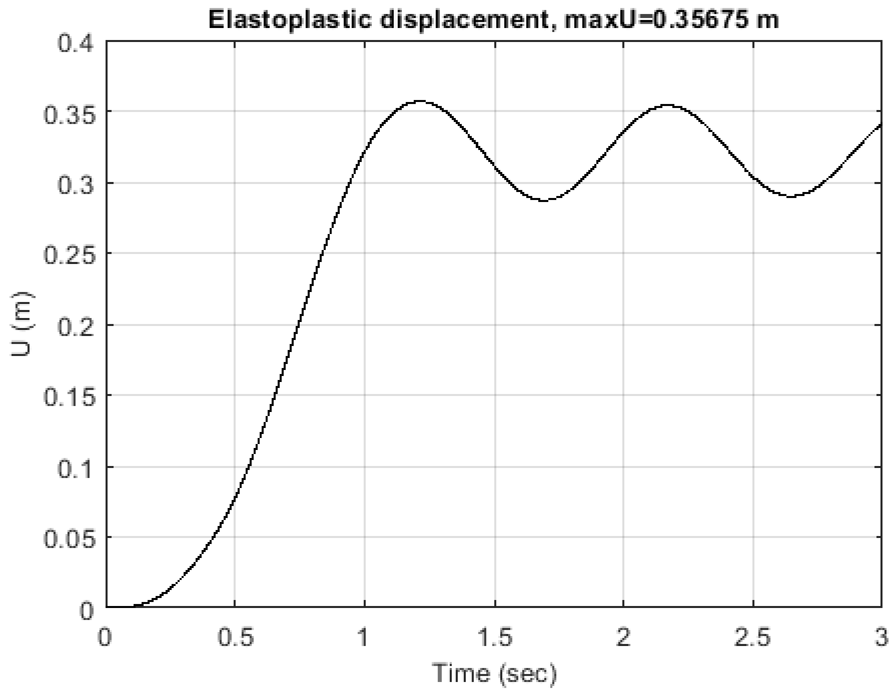
Verify with Figure 7.22(a) of 'Structural dynamics of earthquake engineering Theory and application using MATHEMATICA and MATLAB', by S. Rajasekaran



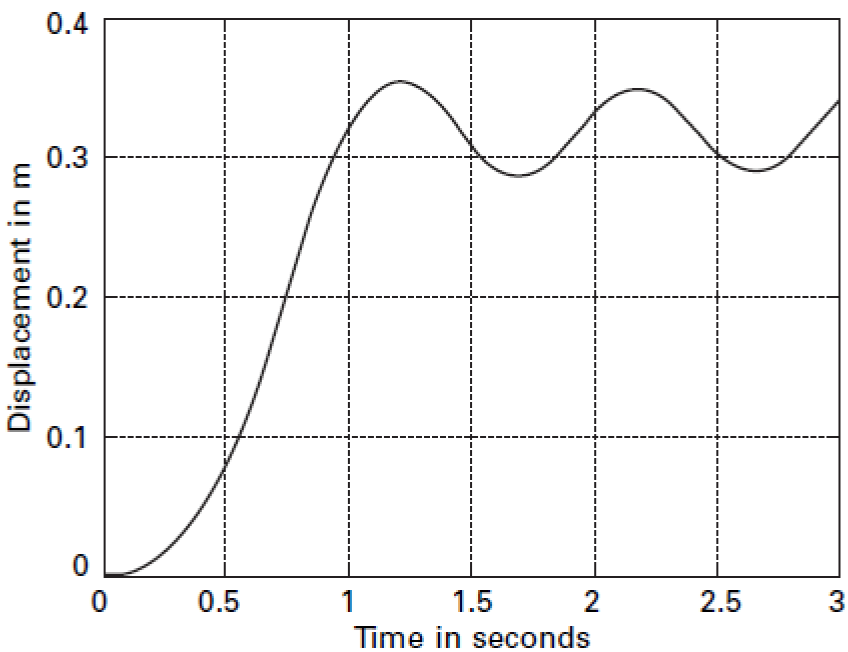
(a)

Plot the elastic displacement response.

```
FigHandle=figure('Name','Elastoplastic displacement','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 500, 350]);
plot(t,u2,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,3])
ylim([0,0.4])
xlabel('Time (sec)','FontSize',10);
ylabel('U (m)','FontSize',10);
title(['Elastoplastic displacement, maxU=',num2str(max(abs(u2))),' m'],...
    'FontSize',10)
```



Verify with Figure 7.22(b) of 'Structural dynamics of earthquake engineering Theory and application using MATHEMATICA and MATLAB', by S. Rajasekaran



(b)

Copyright

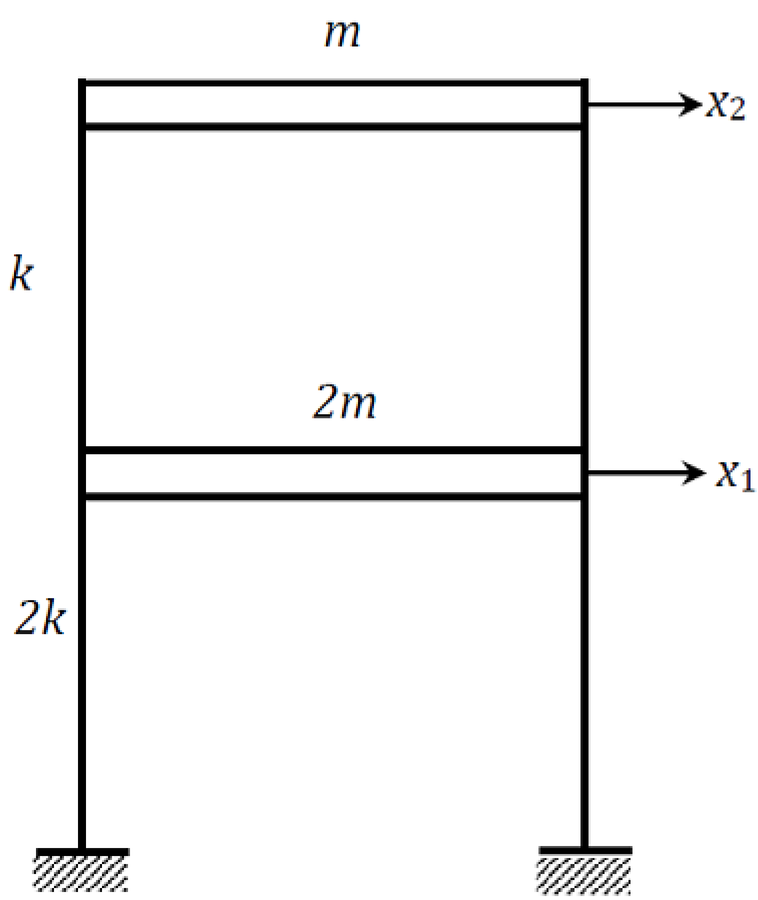
- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

Two-storey shear frame, dynamic analysis with direct integration (NPTEL)

Contents

Statement of the problem

- This example comes from the Introduction to Earthquake Engineering - Web course of NPTEL (National Programme on Technology Enhanced Learning), Chapter 7, Non-linear Seismic Response of Structures, Example 7.3.
- A two-story building is modeled as 2-DOF system and rigid floors as shown in the following figure. Determine the floor displacement responses due to El-Centro, 1940 earthquake ground motion. Take the inter-story stiffness, k =197.392 × 10^3 N/m, the floor mass, m = 2500 kg. The columns of the building are having elasto-plastic behavior with yield displacement of 0.05m.



Initialization of structural input data

Set the number of degrees of freedom of the structure, which is equal to the number of its storeys.

```
nDOFs=2;
```

Set the lateral stiffness of each storey in N/m.

```
k_hi=197.392e3*[1;2];
```

Set the lumped mass at each floor in kg.

```
m=2500*[1;2];
```

Set the spatial distribution of the effective earthquake forces. Earthquake forces are applied at all dofs of the structure.

```
inflvec=ones(nDOFs,1);
```

Load earthquake data

Earthquake acceleration time history of the El Centro earthquake (El Centro, 1940, El Centro Terminal Substation Building)

```
D=load('elcentro.dat');
dt=D(2,1)-D(1,1);
xgtt=9.81*D(:,2);
t=dt*(0:(numel(xgtt)-1));
```

Post-yield stiffness

```
k_lo=0.01*k_hi;
```

Yield displacement

```
uy=[0.05;0.05];
```

Set the critical damping ratio

```
ksi=0.02;
```

Initial displacement

```
u0=zeros(nDOFs,1);
```

Initial velocity

```
ut0=zeros(nDOFs,1);
```

Time integration algorithm

```
AlgID='U0-V0-Opt';
```

Minimum absolute value of the eigenvalues of the amplification matrix

```
rinf=1;
```

Maximum tolerance of convergence of the Full Newton Raphson method for numerical computation of acceleration

```
maxtol=0.01;
```

Maximum number of iterations per increment

```
jmax=200;
```

Infinitesimal acceleration for the calculation of the derivative required for the convergence of the Newton-Raphson iteration

```
dak=eps;
```

Dynamic Response History Analysis (DRHA) with direct integration

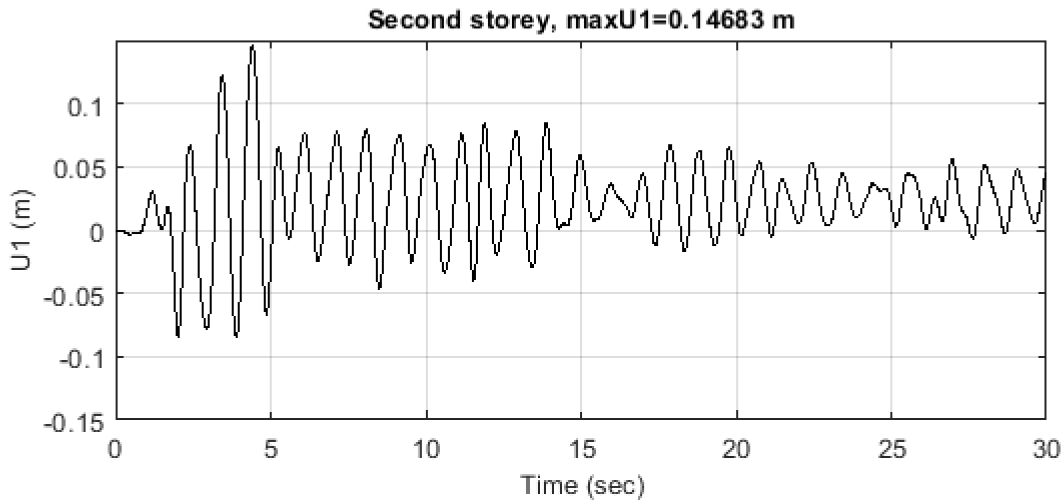
Perform DRHA analysis for nonlinear structure

```
[u,ut,utt,Fs,Ey,Es,Ed,jiter] = NLDRHA_MDOF_1D(dt,xgtt,m,...
    inflvec,k_hi,k_lo,uy,ksi,u0,ut0,AlgID,rinf,maxtol,jmax,dak);
```

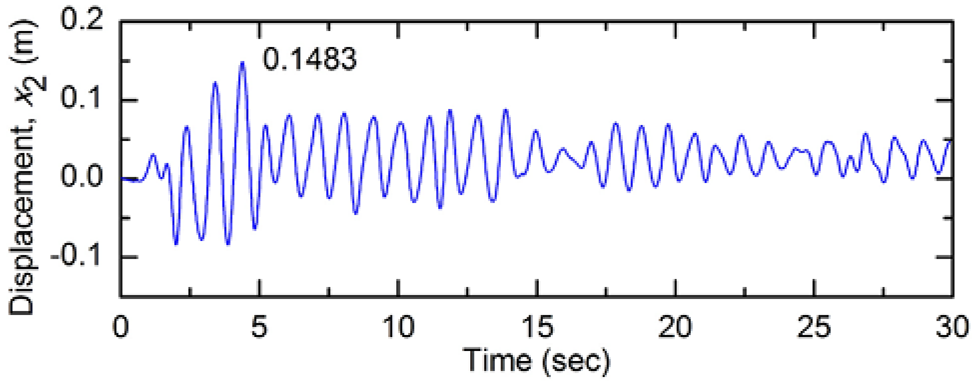
Plot results

Plot the displacement time history of the second storey (1st storey according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Roof displacement','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(t,u(1,:),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,30])
ylim([-0.15,0.15])
xlabel('Time (sec)','FontSize',10);
ylabel('U1 (m)','FontSize',10);
title(['Second storey, maxU1=',num2str(max(abs(u(1,:)))),' m'],...
    'FontSize',10)
```

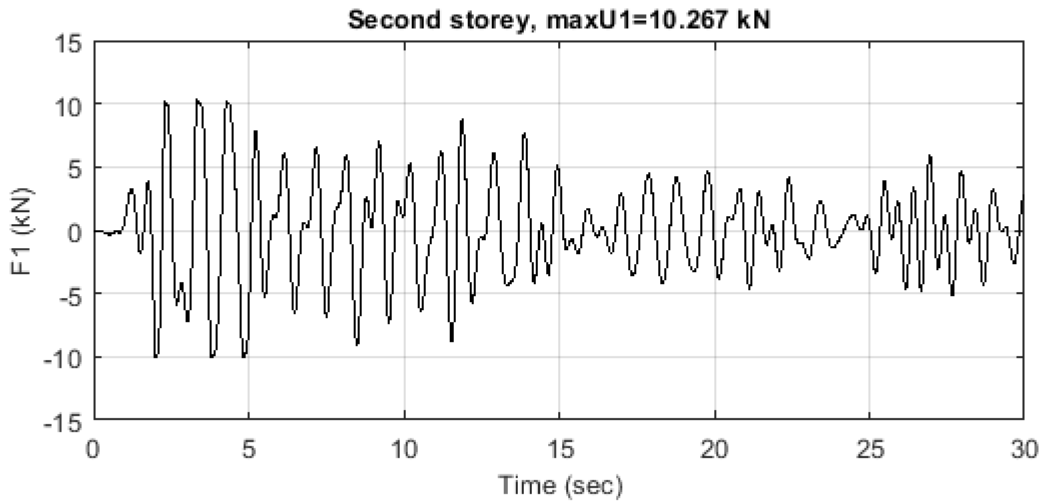


Verify with the top subplot of Figure 7.10 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL

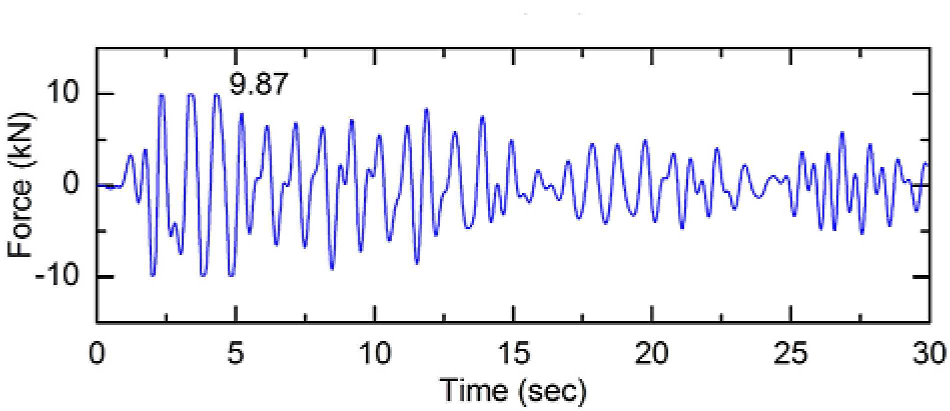


Plot the force time history of the second storey (1st storey according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Roof force','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(t,Fs(1,:)/1000,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,30])
ylim([-15,15])
xlabel('Time (sec)','FontSize',10);
ylabel('F1 (kN)','FontSize',10);
title(['Second storey, maxU1=',num2str(max(abs(Fs(1,:)/1000))),' kN'],...
    'FontSize',10)
```

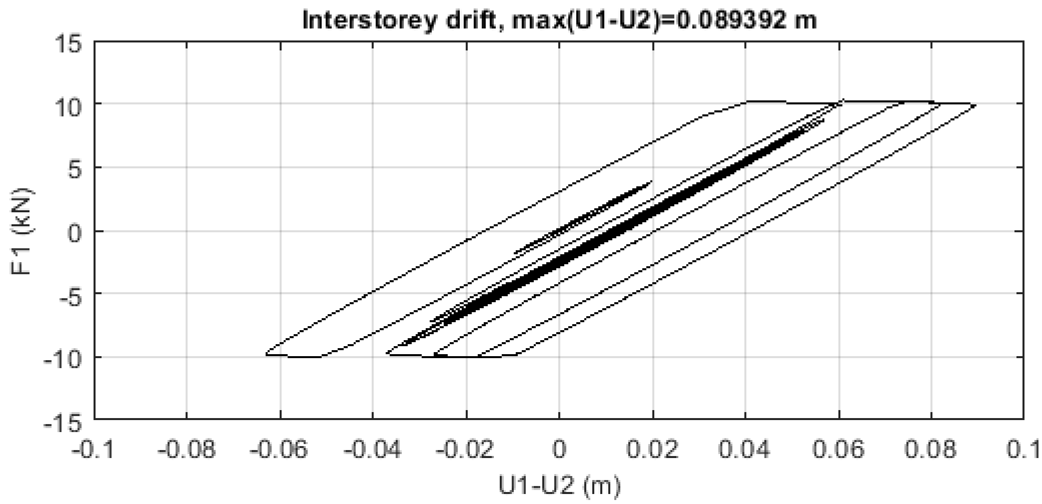


Verify with the middle subplot of Figure 7.10 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL

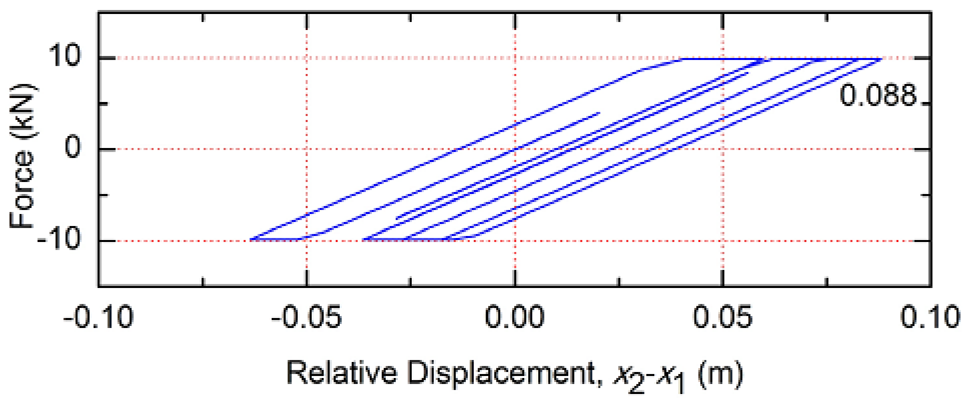


Plot the force versus relative displacement (interstorey drift) between the second and first storeys (1st and 2nd storeys respectively according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Roof force','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(u(1,:)-u(2,:),Fs(1,:)/1000,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([-0.1,0.1])
ylim([-15,15])
xlabel('U1-U2 (m)','FontSize',10);
ylabel('F1 (kN)','FontSize',10);
title(['Interstorey drift, max(U1-U2)=',num2str(max(u(1,:)-u(2,:))),' m'],...
    'FontSize',10)
```

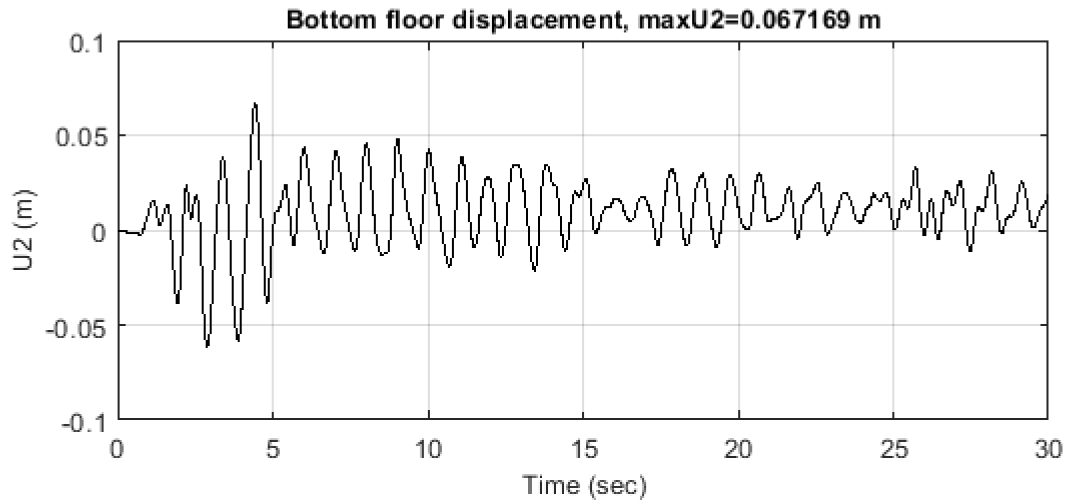


Verify with the bottom subplot of Figure 7.10 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL

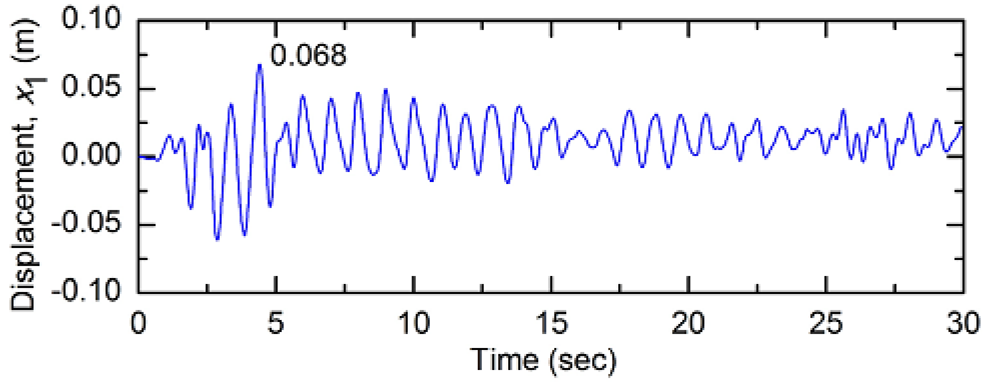


Plot the displacement time history of the first storey (2nd storey according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Bottom floor displacement','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(t,u(2,:),'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,30])
ylim([-0.1,0.1])
xlabel('Time (sec)','FontSize',10);
ylabel('U2 (m)','FontSize',10);
title(['Bottom floor displacement, maxU2=',num2str(max(u(2,:))),' m'],...
    'FontSize',10)
```

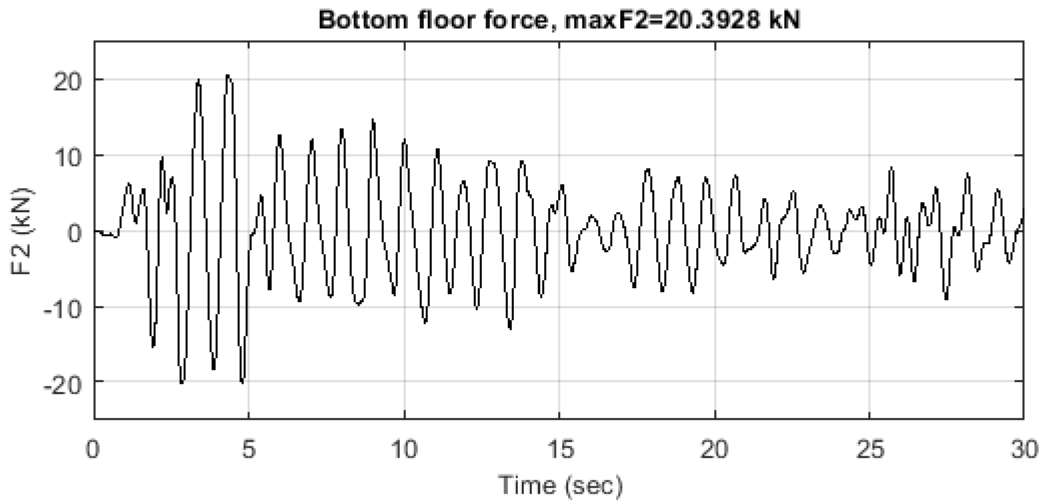


Verify with the top subplot of Figure 7.11 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL

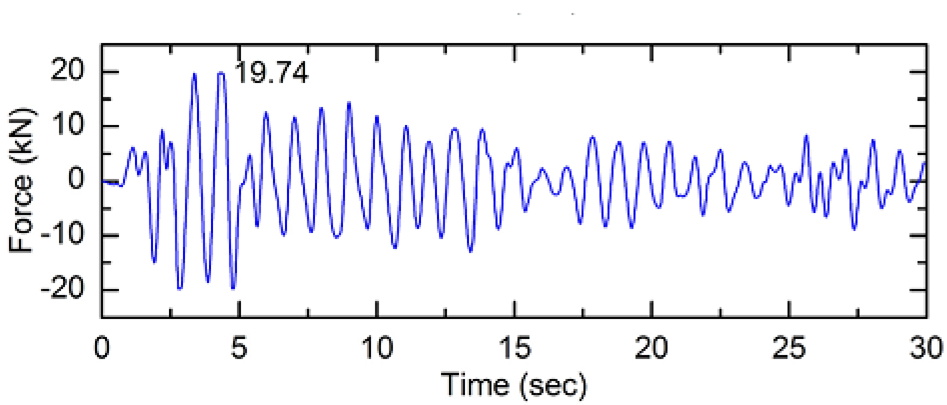


Plot the force time history of the first storey (2nd storey according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Bottom floor force','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(t,(Fs(1,:)+Fs(2,:))/1000,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([0,30])
ylim([-25,25])
xlabel('Time (sec)','FontSize',10);
ylabel('F2 (kN)','FontSize',10);
title(['Bottom floor force, maxF2=',num2str(max((Fs(1,:)+Fs(2,:))/1000)),' kN'],...
    'FontSize',10)
```

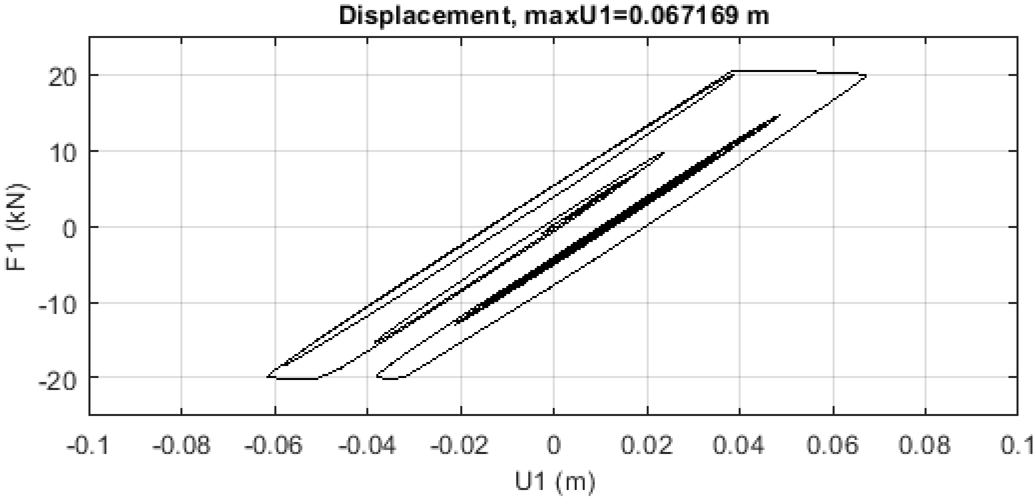


Verify with the middle subplot of Figure 7.11 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL

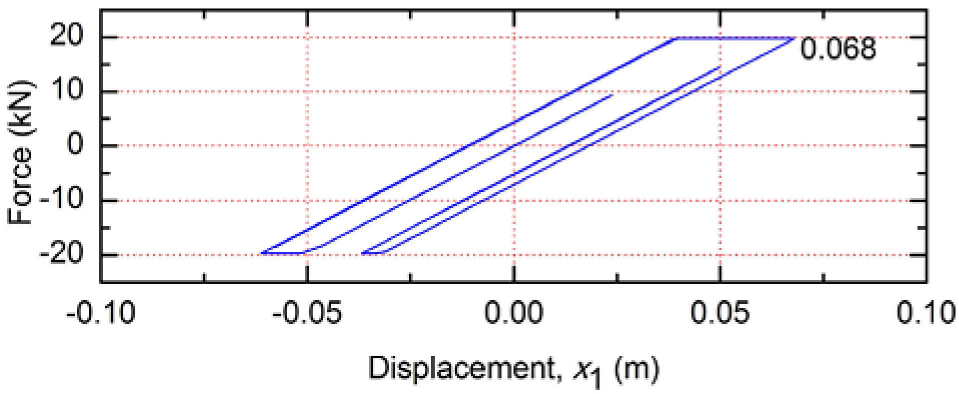


Plot the bottom floor force versus its displacement (i.e. 2nd storey according to the numbering convention of the NLDRHA_MDOF_1D function).

```
FigHandle=figure('Name','Bottom floor force','NumberTitle','off');
set(FigHandle,'Position',[50, 50, 600, 250]);
plot(u(2,:),(Fs(1,:)+Fs(2,:))/1000,'LineWidth',1.,'Marker','.',...
    'MarkerSize',1,'Color',[0 0 0],'markeredgecolor','k')
grid on
xlim([-0.1,0.1])
ylim([-25,25])
xlabel('U1 (m)','FontSize',10);
ylabel('F1 (kN)','FontSize',10);
title(['Displacement, maxU1=',num2str(max(u(2,:))),' m'],...
    'FontSize',10)
```



Verify with the bottom subplot of Figure 7.11 of Chapter 7, Non-linear Seismic Response of Structures, Example 7.3 of NPTEL



Copyright

- Major, Infrastructure Engineer, Hellenic Air Force
- Civil Engineer, M.Sc., Ph.D. candidate, NTUA
- Email: gpapazafeiropoulos@yahoo.gr

Published with MATLAB® R2017b

Bilinear_Kin_1D

Documentation of the Bilinear_Kin_1D function.

```
helpFun('Bilinear_Kin_1D')
```

```
 Bilinear elastoplastic hysteretic model with elastic viscous damping

 [F,K,C,KSTAT,D] = BILINEAR_KIN_1D(U,UT,K_HI,K_LO,UY,M,KSI,KSTAT,D)

 Description
     Define the internal force vector, tangent stiffness matrix and
     tangent damping matrix of a bilinear elastoplastic hysteretic
     structure with elastic damping as a function of displacement and
     velocity.
     The MDOF structure modeled with this function consists of lumped
     masses connected with stiffness and damping elements in series, along
     a single dimension (i.e. one-dimensional MDOF system, hence the
     function is named as '1D'). Each lumped mass has one degree of
     freedom. The first degree of freedom is at the top of the structure
     and the last at its fixed base. However, the last degree of freedom
     is not included in the input arguments of the function, i.e. not
     contained in ndof, as it is always fixed. The nonlinear stiffness
     is virtually of the bilinear type, where an initial stiffness and a
     post-yield stiffness are defined. The unloading or reloading curve of
     this model are parallel to the initial loading curve, and a
     hysteresis loop is created by continuously loading and unloading the
     structure above its yield limit. This behavior can be viewed as
     hardening of the kinematic type.
     An appropriate reference for this function definition is Hughes,
     Pister & Taylor (1979): "Implicit-explicit finite elements in
     nonlinear transient analysis". This function should be defined in
     accordance with equations (3.1), (3.2) and (3.3) of this paper. This
     representation has as special cases nonlinear elasticity and a class
     of nonlinear “rate-type” viscoelastic materials. Tangent stiffness
     and tangent damping matrices are the "consistent" linearized
     operators associated to f in the sense of [Hughes & Pister,
     "Consistent linearization in mechanics of solids", Computers and
     Structures, 8 (1978) 391-397].

 Input parameters
     U [double(:ndof x 1)] is the absolute displacement (with respect to
         ground). ndof is the number of degrees of freedom of the model.
     UT [double(:ndof x 1)] is the absolute velocity (with respect to
         ground).
     K_HI [double(:ndof x 1)] is the initial stiffness of the system
         before its first yield, i.e. the high stiffness. Give the
         stiffness of each storey from top to bottom.
     K_LO [double(:ndof x 1)] is the post-yield stiffness of the system,
         i.e. the low stiffness. Give the stiffness of each storey from
         top to bottom.
     UY [double(:ndof x 1)] is the yield limit of the stiffness elements
         of the structure. The element is considered to yield, if the
         interstorey drift between degrees of freedom i and i+1 exceeds
         UY(i). Give the yield limit of each storey from top to bottom.
     M ([ndof x ndof]) is the lumped mass matrix of the system. Give
```

```
        the lumped mass of each storey from top to bottom.
    KSI (scalar): ratio of critical viscous damping of the system,
        assumed to be unique for all damping elements of the structure.
    KSTAT [double(:ndof x 1)] is the stiffness vector which takes into
        account any plastic response of the structure. It is used to
        record the status of the structure so that it is known before the
        next application of this function at a next (time) step.
        Initialize by setting KSTAT=K_HI.
    D [double(:ndof x 1)] is the equilibrium displacement vector which
        takes into account any plastic response of the structure. It is
        used to record the status of the structure so that it is known
        before the next application of this function at a next (time)
        step. Initialize by setting D=zeros(ndof,1).

Output parameters
    F [double(:ndof x 1)]: internal force vector of the structure (sum of
        forces due to stiffness and damping) at displacement u and
        velocity UT
    K ([ndof x ndof]): tangent stiffness matrix (nonlinear function
        of displacement u and velocity UT). It is equivalent to the
        derivative d(F)/d(U)
    C ([ndof x ndof]): tangent damping matrix (nonlinear function
        of displacement u and velocity UT). It is equivalent to the
        derivative d(F)/d(U)
    KSTAT [double(:ndof x 1)] is the stiffness vector which takes into
        account any plastic response of the structure. It is used to
        record the status of the structure so that it is known before the
        next application of this function at a next (time) step.
    D [double(:ndof x 1)] is the equilibrium displacement vector which
        takes into account any plastic response of the structure. It is
        used to record the status of the structure so that it is known
        before the next application of this function at a next (time)
        step.

Examples
    % 1) SDOF system
    u=0:0.2:4;
    ut=0.001*ones(1,numel(u));
    u=[u,u(end:-1:1)];
    ut=[ut,-ut];
    u=[u,-u];
    ut=[ut,ut(end:-1:1)];
    u=[u u];
    ut=[ut ut];
    k_hi=1000;
    k_lo=1;
    uy=2;
    M=1;
    ksi=0.05;
    k=k_hi;
    d=0;
    f=zeros(1,numel(u));
    for i=1:numel(u)
        [f(i),K,C,k,d] = Bilinear_Kin_1D(u(i),ut(i),k_hi,k_lo,uy,M,...
            ksi,k,d);
    end
    figure()
    plot(u,f)
    % 2) System with 2 DOFs
    u=0:0.2:4;
    ut=0.001*ones(1,numel(u));
```

```
    u=[u;u/2];
    ut=[ut;ut/2];
    u=[u,u(:,end:-1:1)];
    ut=[ut,-ut];
    u=[u,-u];
    ut=[ut,ut(:,end:-1:1)];
    u=[u u];
    ut=[ut ut];
    k_hi=[1000;2000];
    k_lo=[1;1];
    uy=[1;1];
    M=eye(2);
    ksi=0.05;
    k=k_hi;
    d=zeros(2,1);
    f=zeros(2,size(u,2));
    for i=1:size(u,2)
        [f(:,i),K,C,k,d] = Bilinear_Kin_1D(u(:,i),ut(:,i),k_hi,k_lo,...
            uy,M,ksi,k,d);
    end
    figure()
    plot(u(1,:),f(1,:))
    figure()
    plot(u(2,:),f(2,:))
```

```
 ________________________________________________________________________
 Copyright (c) 2015-2021
     George Papazafeiropoulos
     Major, Infrastructure Engineer, Hellenic Air Force
     Civil Engineer, M.Sc., Ph.D. candidate, NTUA
     Email: gpapazafeiropoulos@yahoo.gr
 ________________________________________________________________________
```

NLDRHA_MDOF_1D

Documentation of the NLDRHA_MDOF_1D function.

```
helpFun('NLDRHA_MDOF_1D')
```

```
Non Linear Dynamic Response History Analysis of a bilinear kinematic MDOF
shear building with elastic damping

[U,UT,UTT,FS,EY,ES,ED,JITER] = NLDRHA_MDOF_ENERGY(DT,XGTT,M,INFLVEC,...
    K_HI,K_LO,UY,KSI,U0,UT0,ALGID,RINF,MAXTOL,JMAX,DAK)

Description
    General nonlinear direct time integration of equation of motion of a
    bilinear elastoplastic hysteretic shear building, or any other MDOF
    system with elastic damping, with lumped masses connected with
    stiffness and damping elements in series (1D).
    The General Single Step Single Solve (GSSSS) family of algorithms
    published by X.Zhou & K.K.Tamma (2004) is employed for direct time
    integration of the equations of motion of the Multiple Degree of
    Freedom (MDOF) dynamic problem. Selection among 9 algorithms, all
    designed according to the above journal article, can be made in this
    routine. These algorithms encompass the scope of Linear Multi-Step
    (LMS) methods and are limited by the Dahlquist barrier theorem
    (Dahlquist,1963).

Input parameters
    DT [double(1 x 1)] is the size of the time step
    XGTT [double(:NumSteps x 1)] is the acceleration time history which
        is imposed at the lumped masses of teh structure, according to
        the influence vector INFLVEC.
    M [double(:ndof x 1)] is the lumped masses of the structure. Define
        the lumped masses from the top to the bottom, excluding the fixed
        dof at the base
    INFLVEC [double(:ndof x 1)] is the influence vector. It determines
        the dofs at which the acceleration prescribed in xgtt will be
        imposed.
    K_HI ([ndof x 1]): is the initial stiffness of the system before
        its first yield, i.e. the high stiffness. Give the stiffness of
        each storey from top to bottom.
    K_LO ([ndof x 1]): is the post-yield stiffness of the system,
        i.e. the low stiffness. Give the stiffness of each storey from
        top to bottom.
    UY ([ndof x 1]): is the yield limit of the stiffness elements of
        the structure. The element is considered to yield, if the
        interstorey drift between degrees of freedom i and i+1 exceeds
        UY(i). Give the yield limit of each storey from top to bottom.
    KSI [double(1 x 1)] is the ratio of critical viscous damping of the
        system, assumed to be unique for all damping elements of the
        structure.
    ALGID [char(1 x :inf)] is the algorithm to be used for the time
        integration. It can be one of the following strings for superior
        optimally designed algorithms:
            'generalized a-method': The generalized a-method (Chung &
            Hulbert, 1993)
            'HHT a-method': The Hilber-Hughes-Taylor method (Hilber,
```

```
          Hughes & Taylor, 1977)
          'WBZ': The Wood–Bossak–Zienkiewicz method (Wood, Bossak &
          Zienkiewicz, 1980)
          'U0-V0-Opt': Optimal numerical dissipation and dispersion
          zero order displacement zero order velocity algorithm
          'U0-V0-CA': Continuous acceleration (zero spurious root at
          the low frequency limit) zero order displacement zero order
          velocity algorithm
          'U0-V0-DA': Discontinuous acceleration (zero spurious root at
          the high frequency limit) zero order displacement zero order
          velocity algorithm
          'U0-V1-Opt': Optimal numerical dissipation and dispersion
          zero order displacement first order velocity algorithm
          'U0-V1-CA': Continuous acceleration (zero spurious root at
          the low frequency limit) zero order displacement first order
          velocity algorithm
          'U0-V1-DA': Discontinuous acceleration (zero spurious root at
          the high frequency limit) zero order displacement first order
          velocity algorithm
          'U1-V0-Opt': Optimal numerical dissipation and dispersion
          first order displacement zero order velocity algorithm
          'U1-V0-CA': Continuous acceleration (zero spurious root at
          the low frequency limit) first order displacement zero order
          velocity algorithm
          'U1-V0-DA': Discontinuous acceleration (zero spurious root at
          the high frequency limit) first order displacement zero order
          velocity algorithm
          'Newmark ACA': Newmark Average Constant Acceleration method
          'Newmark LA': Newmark Linear Acceleration method
          'Newmark BA': Newmark Backward Acceleration method
          'Fox-Goodwin': Fox-Goodwin formula
    U0 [double(:ndof x 1)] is the initial displacement. Give the initial
        displacement of each storey from top to bottom.
    UT0 [double(:ndof x 1)] is the initial velocity. Give the initial
        velocity of each storey from top to bottom.
    RINF [double(1 x 1)] is the minimum absolute value of the eigenvalues
        of the amplification matrix. For the amplification matrix see
        eq.(61) in Zhou & Tamma (2004).
    MAXTOL [double(1 x 1)] is the maximum tolerance of convergence of the
        Full Newton Raphson method for numerical computation of
        acceleration.
    JMAX [double(1 x 1)] is the maximum number of iterations per
        increment. If JMAX=0 then iterations are not performed and the
        MAXTOL parameter is not taken into account.
    DAK [double(:ndof x 1)] is the infinitesimal acceleration for the
        calculation of the derivetive required for the convergence of the
        Newton-Raphson iteration.

Output parameters
    U [double(:ndof x :NumSteps)] is the time-history of displacement
    UT [double(:ndof x :NumSteps)] is the time-history of velocity
    UTT [double(:ndof x :NumSteps)] is the time-history of acceleration
    FS [double(:ndof x :NumSteps)] is the time-history of the internal
        force of the structure analysed.
    EY [double(:ndof x :NumSteps)] is the time history of the sum of the
        energy dissipated by yielding during each time step and the
        recoverable strain energy of the system (incremental).
        cumsum(EY(i,:))-ES(i,:) gives the time history of the
        total energy dissipated by yielding from the start of the dynamic
        analysis.
    ES [double(:ndof x :NumSteps)] is the time-history of the recoverable
```

```
        strain energy of the system (total and not incremental).
    ED [double(:ndof x :NumSteps)] is the time-history of the energy
        dissipated by viscoelastic damping during each time step
        (incremental). cumsum(ED(i,:)) gives the time history of the
        total energy dissipated at dof i from the start of the dynamic
        analysis.
    JITER [double(1 x :NumSteps)] is the iterations that are made per
        increment until convergence
```