ASSIGNMENT
CSM Plasticity

Jordi Parra Porcar

May 2018

Part I
1D plasticity

1 Introduction

In the current study a Matlab algorithm has been implemented for 1D rate-independent/rate-dependent hardening plasticity models, including linear and nonlinear isotropic hardening, and linear kinematic hardening. For the 1D case, the numerical simulation of uniaxial cyclic plastic loading/elastic unloading has been performed for the next cases:

- Rate-independent/rate-dependent perfect plasticity
- Rate-independent/rate-dependent linear isotropic hardening plasticity
- Rate-independent/rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law
- Rate-independent/rate-dependent linear kinematic hardening plasticity
- Rate-independent/rate-dependent nonlinear isotropic and linear kinematic hardening plasticity

For each case the stress-strain curves has been plotted and for rate-dependent plasticity models, also the stress-time curves showing the influence of the viscosity parameter and the loading rate has been studied.

2 Rate-independent/rate-dependent perfect plasticity

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	0	0
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

Perfect plasticity is characterized by an elastic element (E parameter), and an elastic limit σ_e called *yield stress*. In rate-independent perfect plasticity (Figure 1 for $\nu = 0$) we don't have any deformation for stresses smaller than the yield stress (elastic loading) and, as the stress value reaches the yield stress, for any strain increment the stress value keeps constant (perfect plasticity) and no deformation has been experimented. During the elastic unloading the initial state is recovered (σ=0) and than, during compression, as the the compression critical value is reached, perfect plasticity has been experimented again. As the ν value is incremented, the perfect plasticity behaviour decreases (Figure 1).

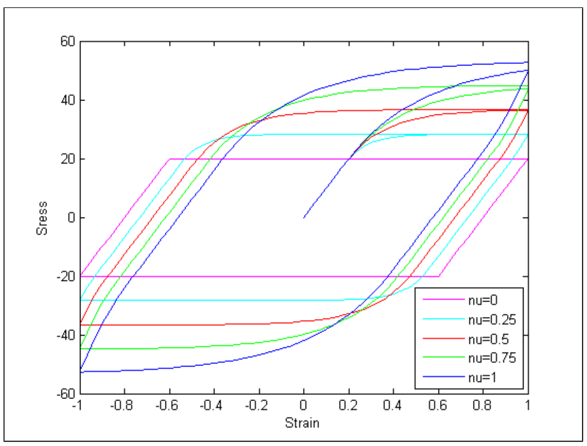

Figure 1: Rate-independent/rate-dependent perfect plasticity

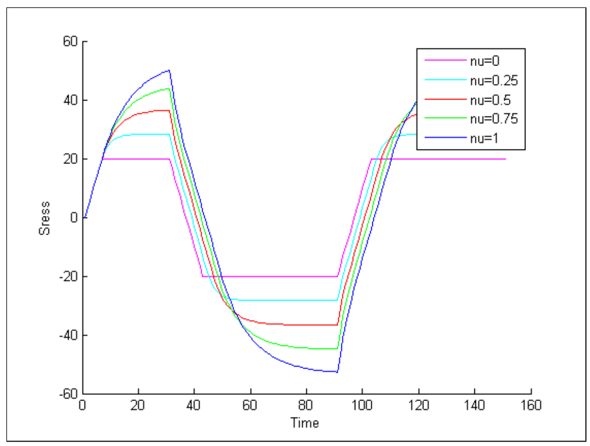

Figure 2: Time-stress rate-dependent perfect plasticity

Figure 3 shows how the inviscid material behaviour is recovered as the time step vale dt increases. The perfect platicity case is recovered for $dt = 1$.

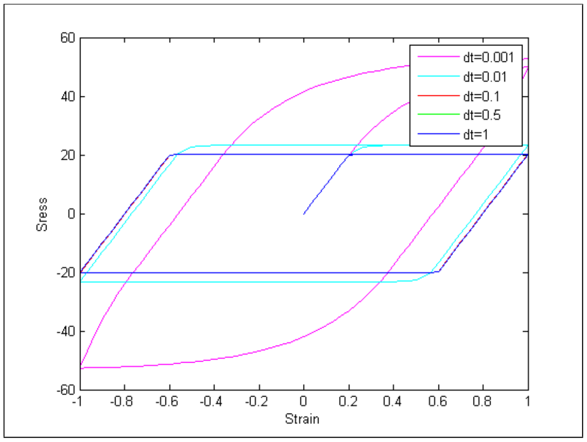

Figure 3: Rate-independent plasticity recovered for different values of dt

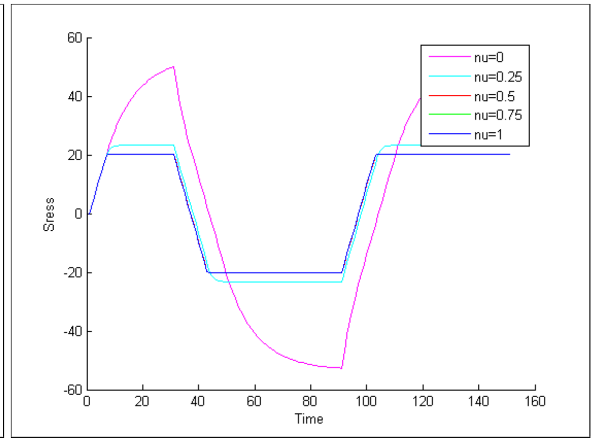

Figure 4: Time-stress rate-independent plasticity recovered

3 Rate-independent/rate-dependent linear isotropic hardening plasticity

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	[0, 10, 30, 40, 50]	30
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 9/4,1]
dt (s)	1E-3	1E-3

In the isotropic hardening plasticity model the material experiments an increasing of the stress limit during plasticity.

Figure 5 shows the model's behaviour with different values of K and how the stress limit and the slopes increase with K. For K = 0 the perfect plasticity model is recovered. It is important to point out how for a values of K bigger than cero, the stress does not close the stress path.

Figure 7 shows the effects of viscosity on the model, where the slope increase with ν.

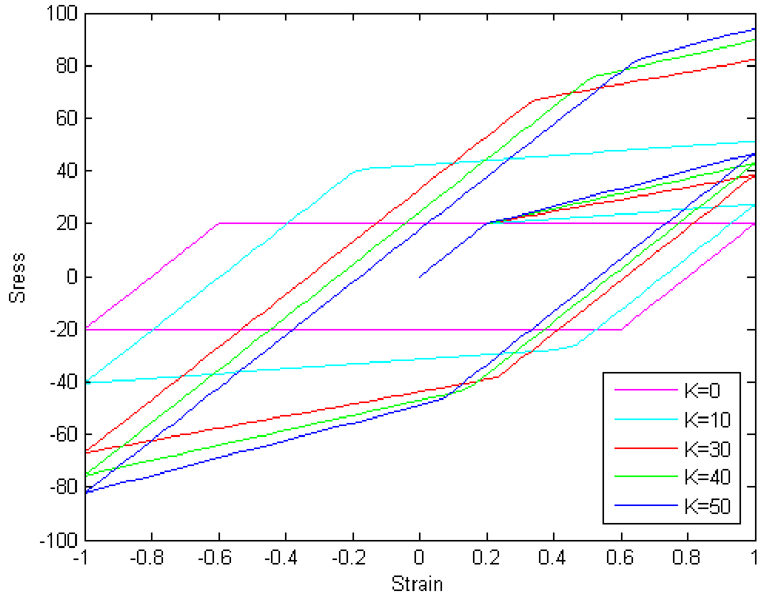

Figure 5: Linear isotropic hardening plasticity different K values (ν=0)

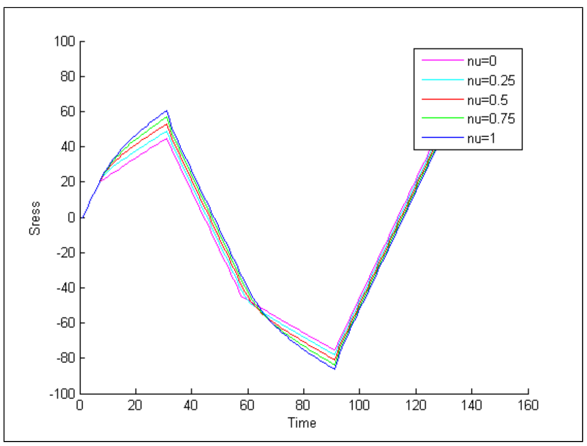

Figure 6: Time-stress linear isotropic hardening plasticity different ν values (K=30)

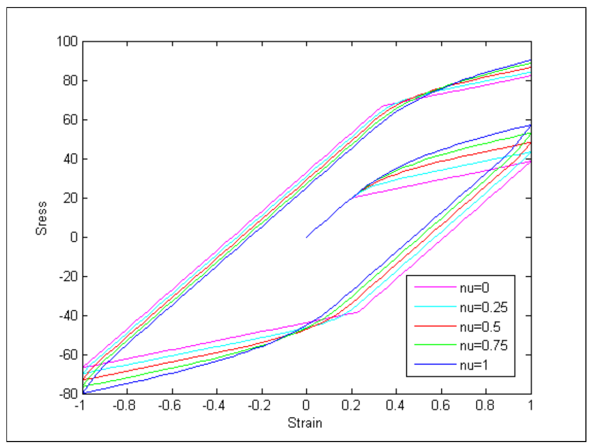

Figure 7: Linear isotropic hardening plasticity different ν values (K=30)

4 Rate-independent/rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	[0, 10, 30, 40, 50]	[0, 10, 30, 40, 50]
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
dt (s)	1E-3	1E-3

Figure 8 and Figure 10 show the behaviour for a non-linear isotropic hardening plasticity case, where an exponential saturation law has been implemented. Due to the exponential saturation law, the slope of the stress is not linear. Even as for the previous case, it is important to point out how the slope increases as K increases.

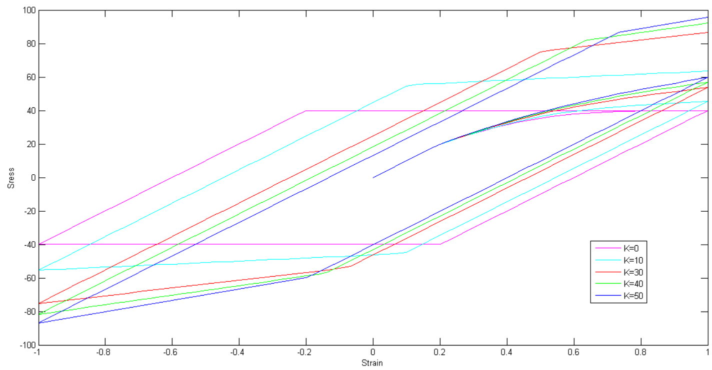

Figure 8: Rate independent nonlinear isotropic hardening plasticity, considering an exponential saturation law different K values

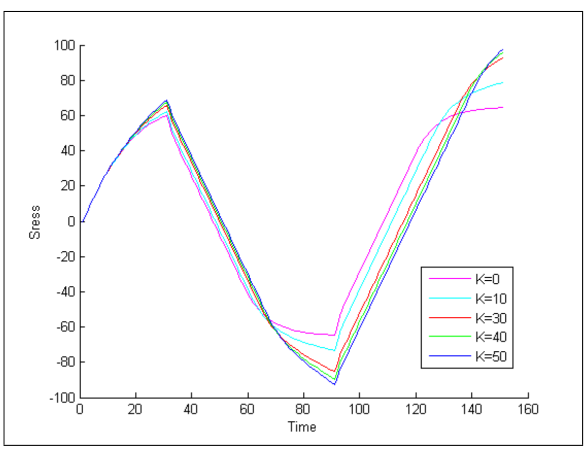

Figure 9: Time-stress Rate dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law for different K values ($\nu = 0.75$) .

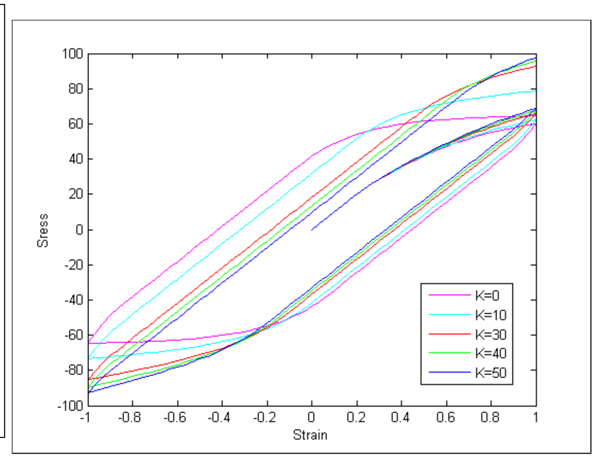

Figure 10: Rate dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law for different K values ($\nu = 0.75$).

Figure 12 shows as the slope increase with ν for a rate dependent model.

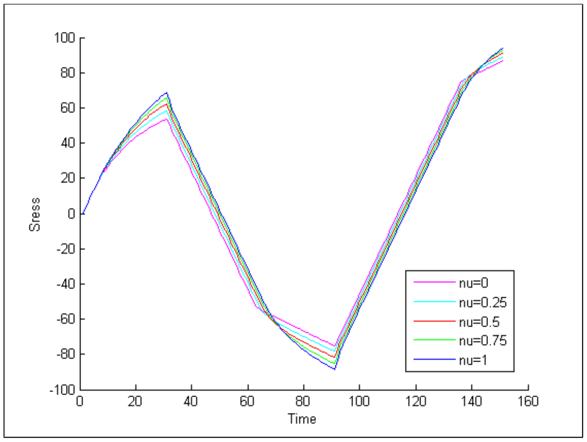

Figure 11: Time-stress Rate dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law for different ν values (K=30).

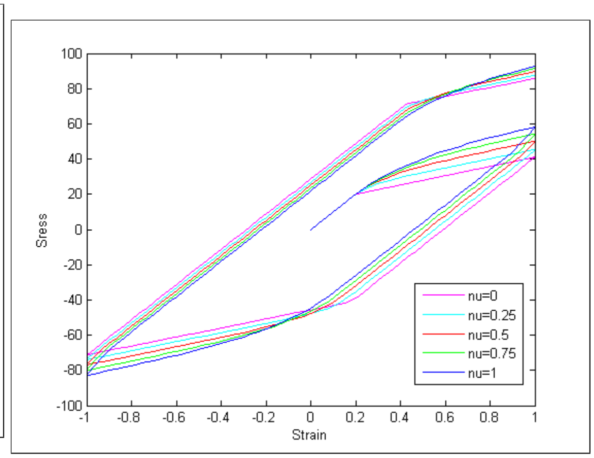

Figure 12: Rate dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law for different ν values (K=30).

5 Rate-independent/rate-dependent linear kinematic hardening plasticity

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	0	0
H	[0, 10, 30, 40, 50]	[0, 10, 30, 40, 50]
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
dt (s)	1E-3	1E-3

Figure 13, Figure 15 and Figure 17 show how the stress closes the stress path, and also how the slope increments as the H increases too.

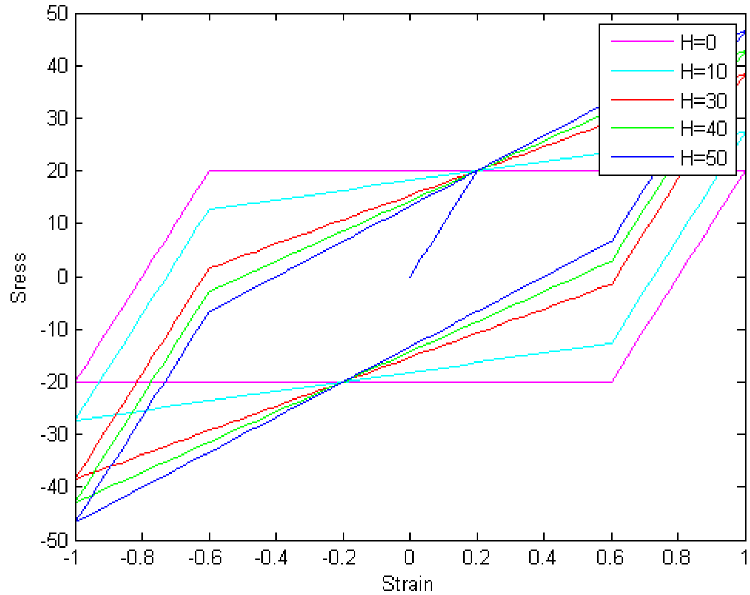

Figure 13: Rate independent linear kinematic hardening plasticity different K values.

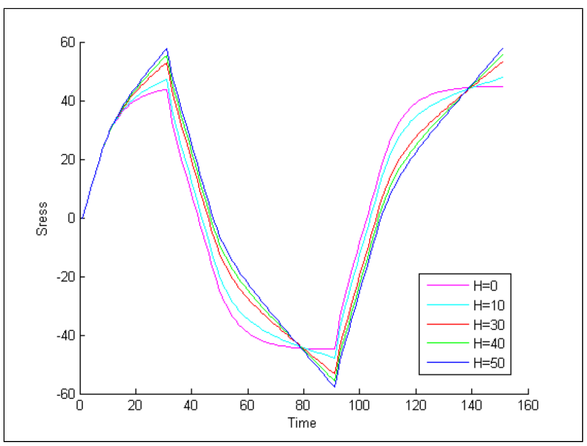

Figure 14: Time-stress linear kinematic hardening plasticity different H values (ν =0.75)

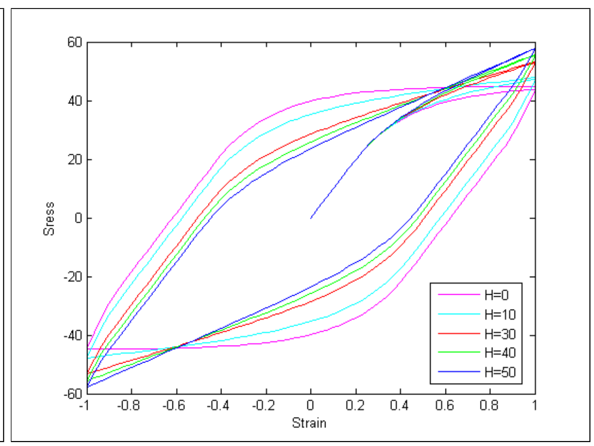

Figure 15: Rate-dependent linear kinematic hardening plasticity different H values (ν =0.75)

6 Rate-independent/rate-dependent nonlinear isotropic and linear kinematic hardening plasticity (K=H=10)

Material properties.

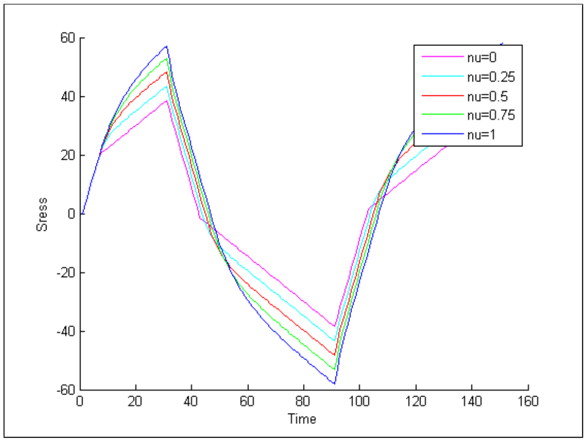

Figure 16: Time-stress linear kinematic hardening plasticity different ν values (H=30)

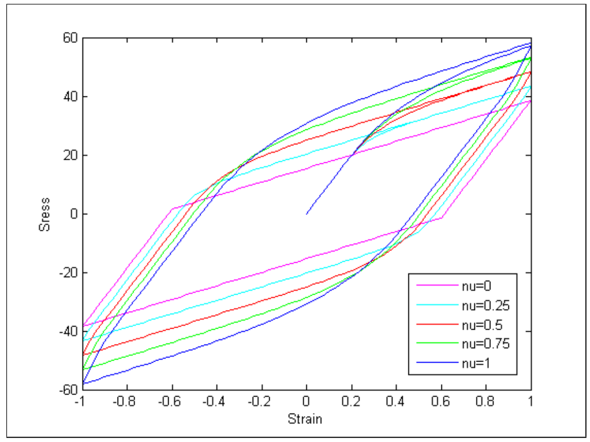

Figure 17: Rate-dependent linear kinematic hardening plasticity different ν values (H=30)

Parameters	Rate-independent	Rate-dependent
E	100	100
K	0	0
H	[10, 30, 50]	[10, 30, 50]
Yield stress	20	20
ν	0	3/4
dt (s)	1E-3	1E-3

In this last case nonlinear isotropic and linear kinematic hardening plasticity are mixed. We can observe that the loading/unloading stress path cycle is not close.

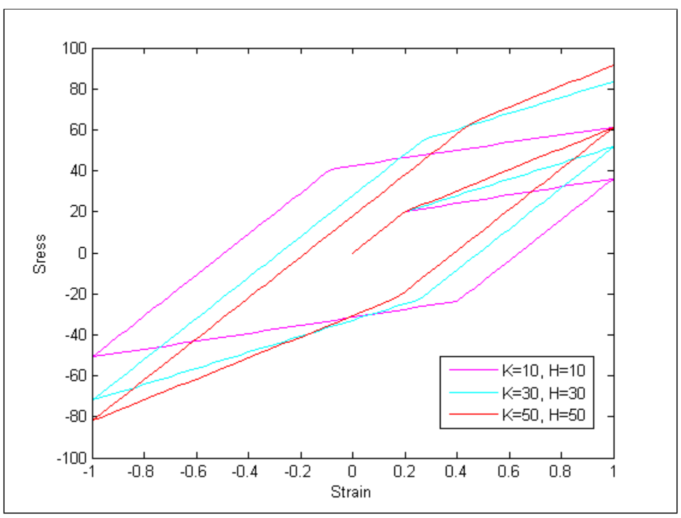

Figure 18: Rate independent nonlinear isotropic and linear kinematic hardening plasticity

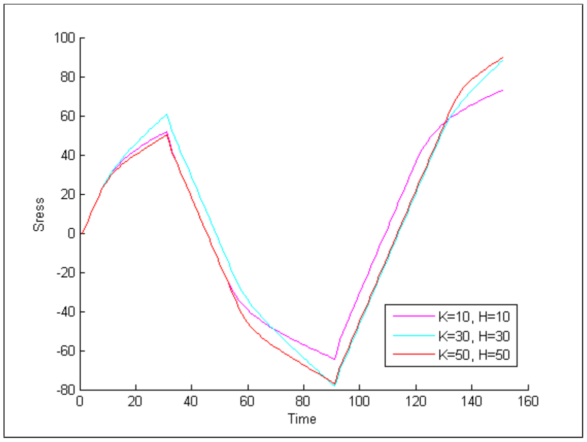

Figure 19: Time-stress nonlinear isotropic and linear kinematic hardening plasticity (ν=0.75).

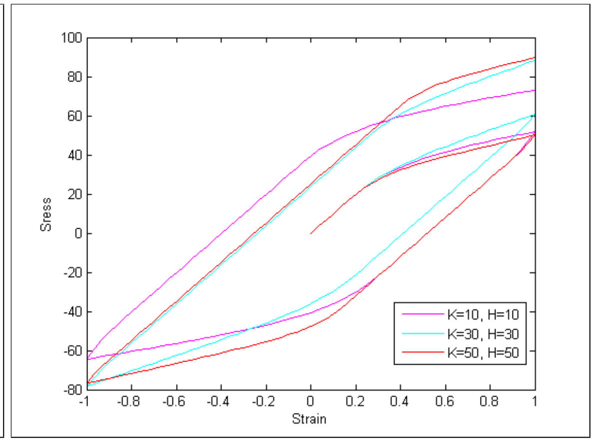

Figure 20: Rate-dependent nonlinear isotropic and linear kinematic hardening plasticity (ν=0.75).

Part II
J2 plasticity

7 Introduction

In the curent study a Matlab algorith has been implemented for J2 rate-independent/rate-dependent hardening plasticity models, including linear and nonlinear isotropic hardening, and linear kinematic hardening. In J2 model we are wornikg in a 3D model and the *Poisson* ratiois introduced. For the J2 case, the numerical simulation of uniaxial cyclic plastic loading/elastic unloading has been performed for the next cases:

- Rate-independent/rate-dependent perfect plasticity
- Rate-independent/rate-dependent linear isotropic hardening plasticity
- Rate-independent/rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law
- Rate-independent/rate-dependent linear kinematic hardening plasticity
- Rate-independent/rate-dependent nonlinear isotropic and linear kinematic hardening plasticity

For each case the stress-strain curves and the dev[stress11]-strain11 curves have been plotted and for rate-dependent plasticity models, also the stress-time curves showing the influence of the viscosity parameter and the loading rate has been studied.

8 Rate-independent/rate-dependent perfect plasticity

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	0	0
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
pois	0.25	0.25
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

In order to observe the perfect plasticity behaviour we must compute the deviatoric stress tensor (Figure 21). The stress-strain plot (Figure 22) does not give us a the perfect plasticity results due to the effects of the *Poisson* ratio, which mobilizes the principal stress components. Figure 24 shows how the perfect plasticity behaviour is recovered for a bigger timesteps.

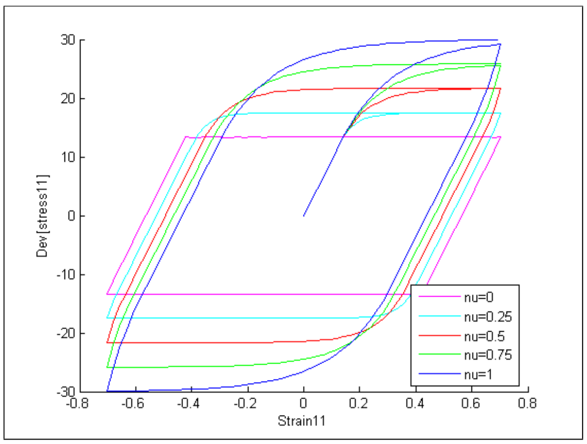

Figure 21: Rate-independent/rate-dependent perfect plasticity Dev[Stress11]-Strain11

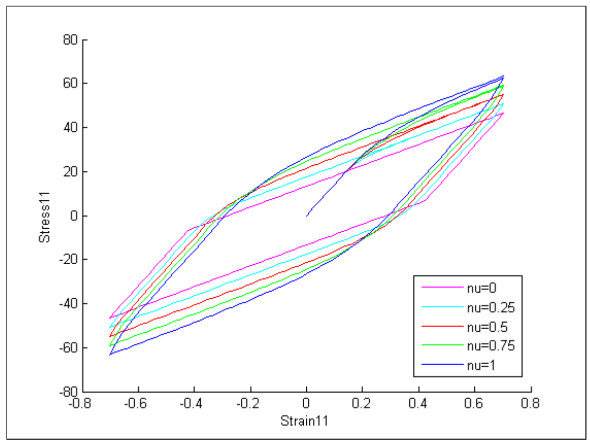

Figure 22: Rate-independent/rate-dependent perfect plasticity Stress11-Strain11

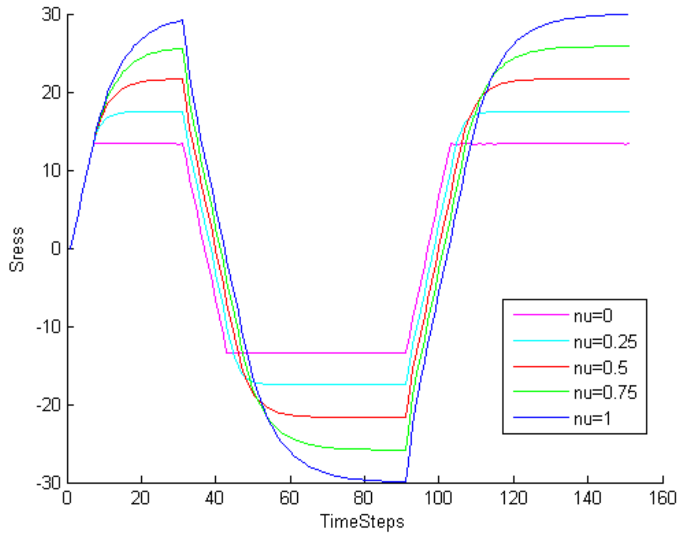

Figure 23: Time-Dev[Stress11] rate-dependent perfect plasticity

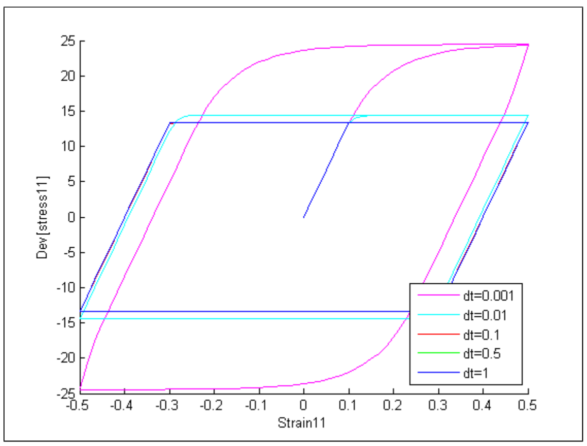

Figure 24: Rate-dependent perfect plasticity different dt ($\nu = 1$) Dev[Stress11]-Strain11

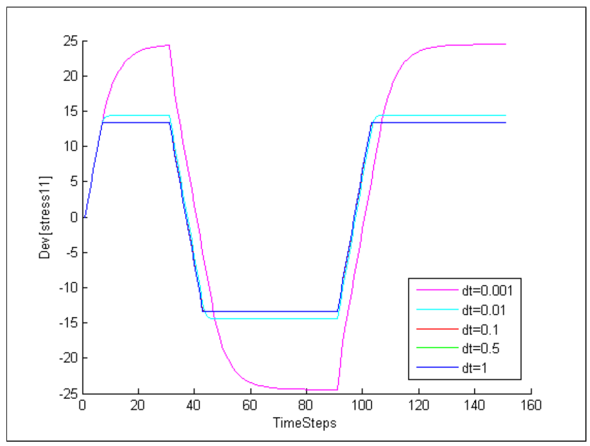

Figure 25: Rate-dependent perfect plasticity ($\nu = 1$) Dev[Stress11]-Steptimes

9 Rate-independent/rate-dependent linear isotropic hardening plasticity

Material properties.

Parameters	Rate-independent	Rate-dependent
E	100	100
K	[0, 10, 30, 40, 50]	[0, 10, 30, 40, 50]
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
pois	0.25	0.25
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

Results for the linear isotropic case point out that the loading/unloading path is open at the end of the cycle and how the yield stress increases depending of the of the isotropic hardening parameter. For K = 0 perfect plasticity is recovered.

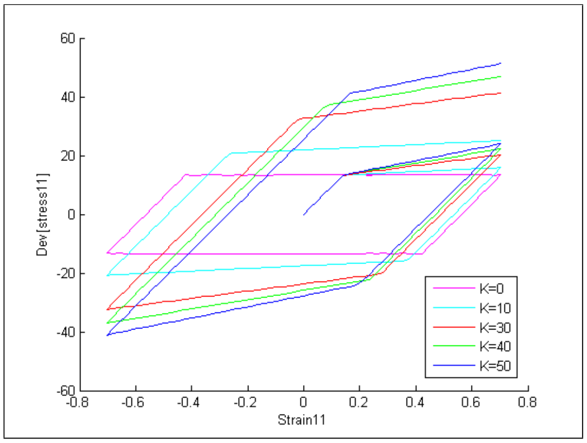

Figure 26: Rate-independent/rate-dependent linear isotropic hardening plasticity Dev[Stress11]-Strain11

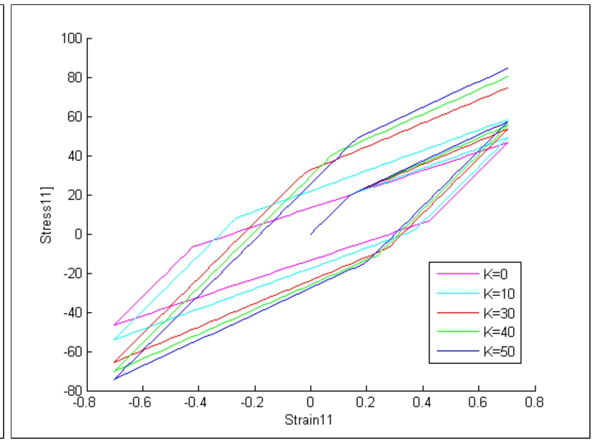

Figure 27: Rate-independent/rate-dependent linear isotropic hardening plasticity Stress11-Strain11

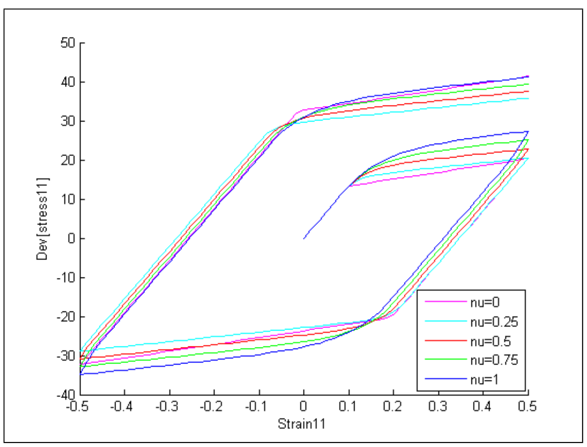

Figure 28: Rate-dependent linear isotropic hardening plasticity Dev[Stress11]-Strain11 (K=30)

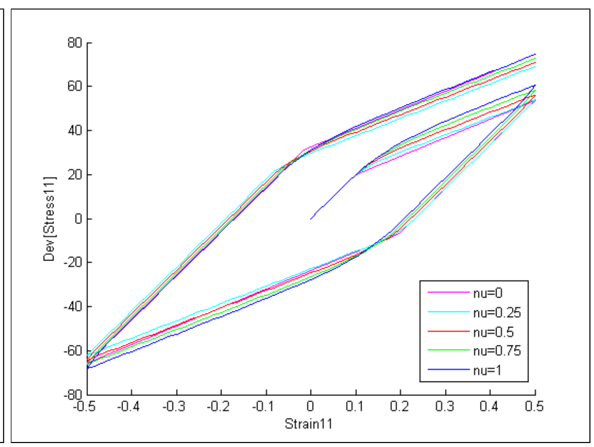

Figure 29: Rate-dependent linear isotropic hardening plasticity Stress11-Strain11 (K=30)

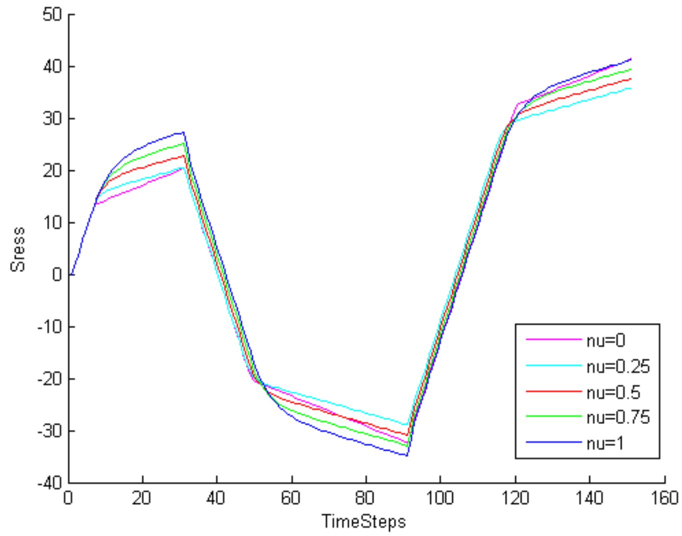

Figure 30: Time-Dev[Stress11] rate-dependent linear isotropic hardening plasticity

10 Rate-independent/rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law

Parameters	Rate-independent	Rate-dependent
E	100	100
K	[0, 10, 30, 40, 50]	[0, 10, 30, 40, 50]
H	0	0
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4 ,1]
pois	0.25	0.25
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

For the nonlinear isotropic hardening plasticity, taking into account the exponential saturation law, the model does not show a lineal plastic hardening behaviour.

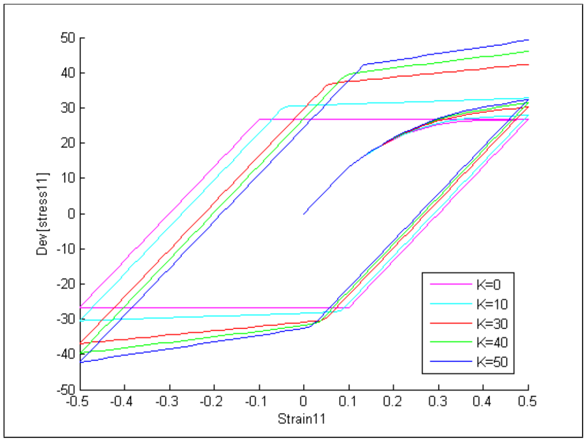

Figure 31: Rate-independent nonlinear isotropic hardening plasticity, considering an exponential saturation law Dev[Stress11]-Strain11

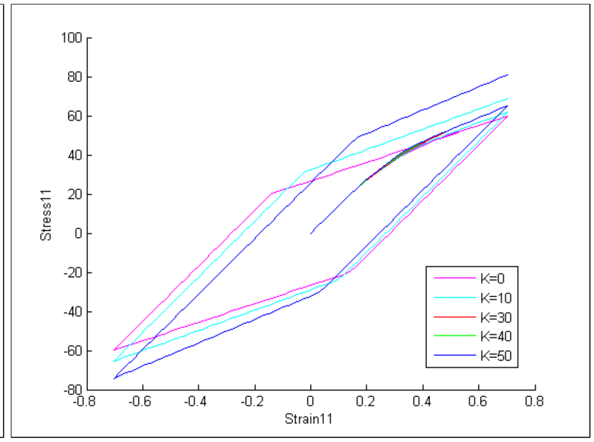

Figure 32: Rate-independent nonlinear isotropic hardening plasticity, considering an exponential saturation law Stress11-Strain11

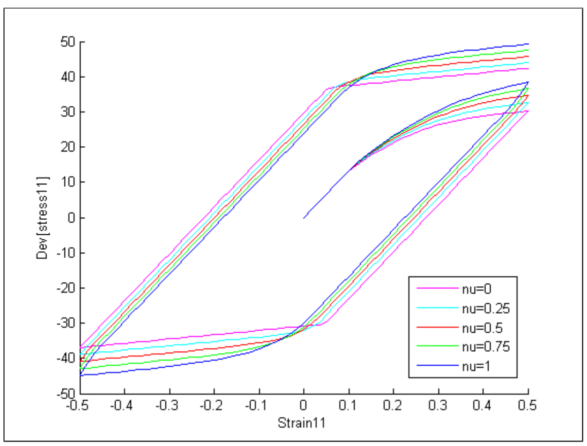

Figure 33: Rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law (K=30) Dev[Stress11]-Strain11

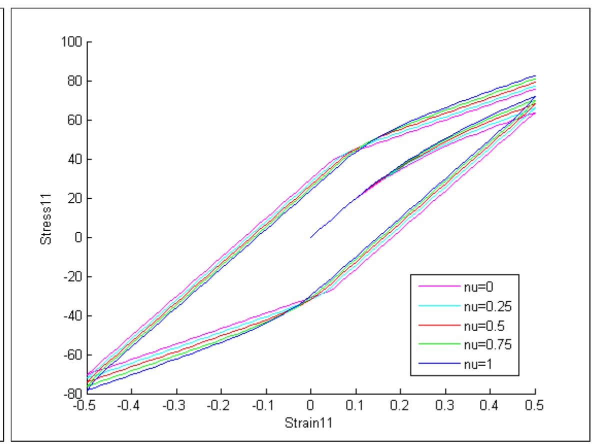

Figure 34: Rate-dependent nonlinear isotropic hardening plasticity, considering an exponential saturation law (K=30) Stress11-Strain11

11 Rate-independent/rate-dependent linear kinematic hardening plasticity

Parameters	Rate-independent	Rate-dependent
E	100	100
K	0	0
H	[0, 10, 30, 40, 50]	[0, 10, 30, 40, 50]
Yield stress	20	20
ν	0	[1/4, 1/2, 3/4,1]
pois	0.25	0.25
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

In linear kinematic hardening we can observe that the loading/unloading path it is closed and as we incremented the H value, the slope increases.

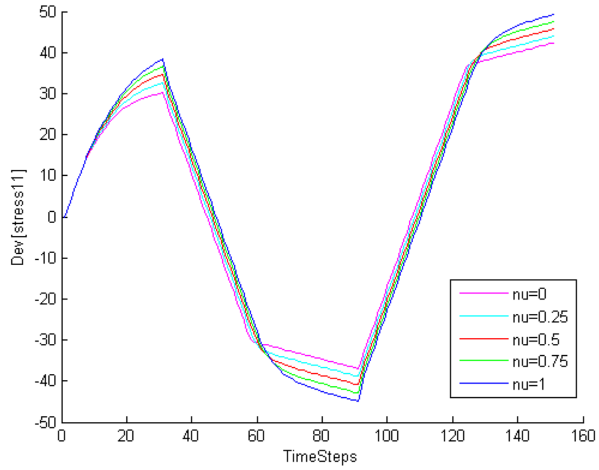

Figure 35: Time-Dev[Stress11] rate-dependent linear isotropic hardening plasticity (K=30).

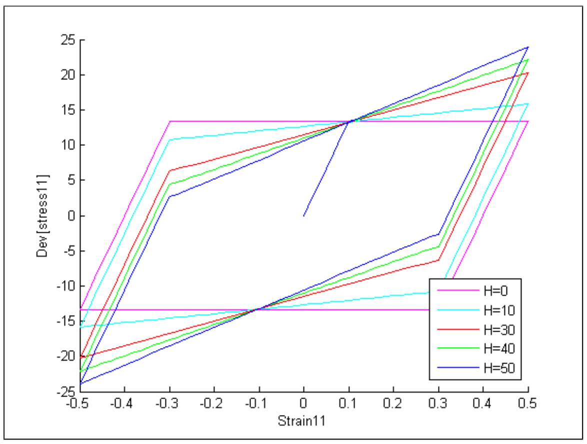

Figure 36: Rate-independent linear kinematic hardening plasticity Dev[Stress11]-Strain11

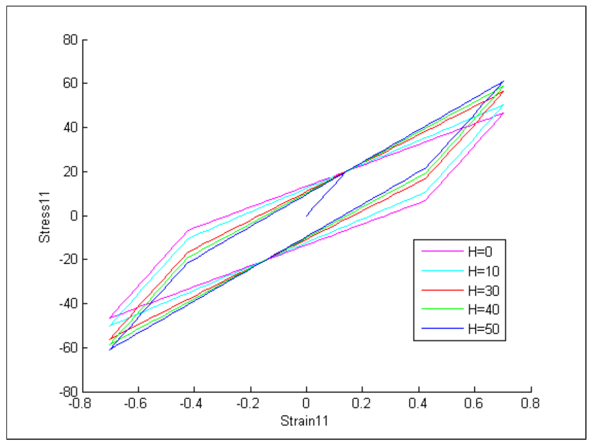

Figure 37: Rate-independent linear kinematic hardening plasticity Stress11-Strain11

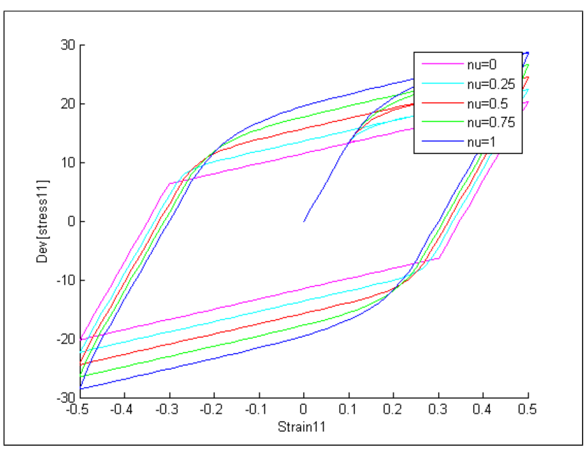

Figure 38: Rate-dependent linear kinematic hardening plasticity (H=30) Dev[Stress11]-Strain11

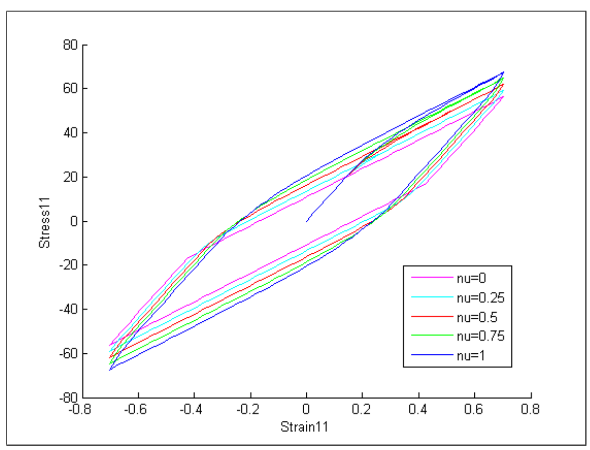

Figure 39: Rate-dependent linear kinematic hardening plasticity (H=30) Stress11-Strain11

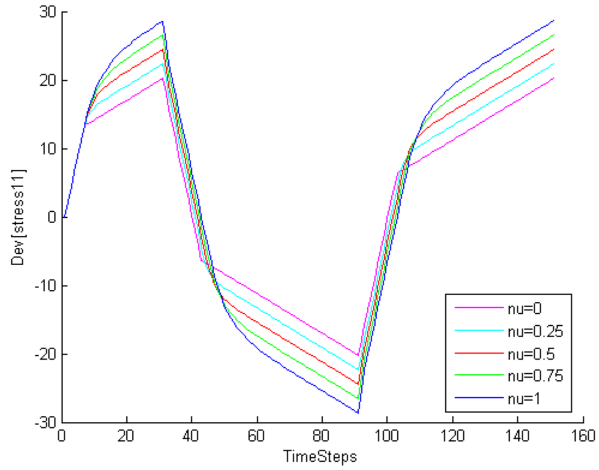

Figure 40: Time-Dev[Stress11] Rate-dependent linear kinematic hardening plasticity (H=30).

12 Rate-independent/rate-dependent nonlinear isotropic and linear kinematic hardening plasticity

Parameters	Rate-independent	Rate-dependent
E	100	100
K	[10, 30, 50]	[10, 30, 50]
H	[10, 30, 50]	[10, 30, 50]
Yield stress	20	20
ν	0	[1/4, 1/2,1]
pois	0.25	0.25
dt (s)	1E-3	1E-3, 1E-2, 1E-1,0.5,1

In this last case nonlinear isotropic and linear kinematic hardening plasticity are mixed. As in the 1D case we can observe that the loading/unloading path is not close.

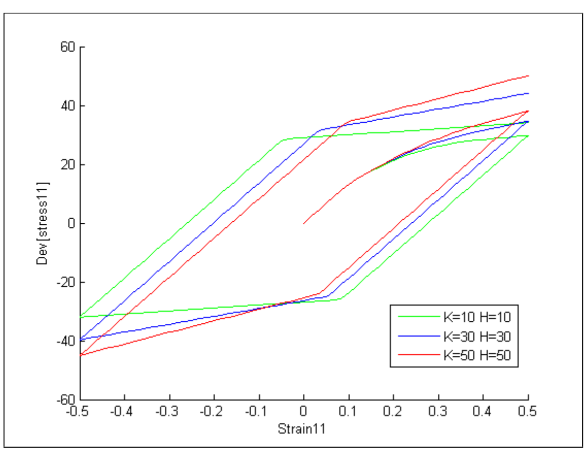

Figure 41: Rate-independent nonlinear isotropic and linear kinematic hardening plasticity Dev[Stress11]-Strain11

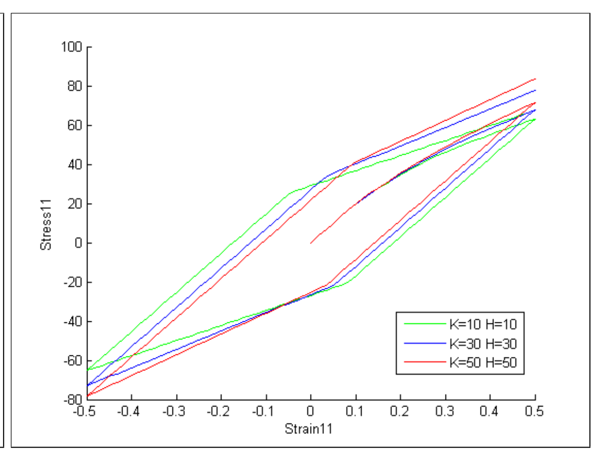

Figure 42: Rate-independent nonlinear isotropic and linear kinematic hardening plasticity Stress11-Strain11

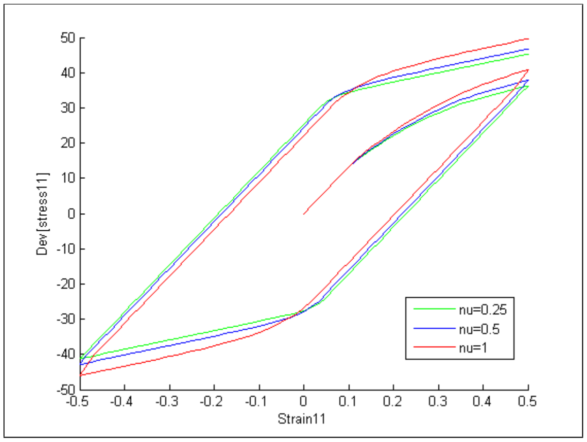

Figure 43: Rate-dependent nonlinear isotropic and linear kinematic hardening plasticity Dev[Stress11]-Strain11

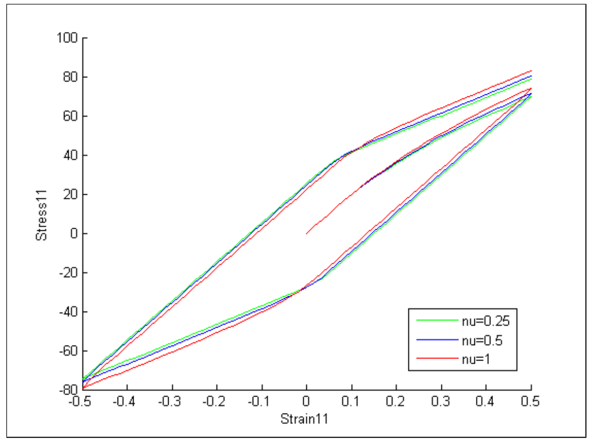

Figure 44: Rate-dependent nonlinear isotropic and linear kinematic hardening plasticity Stress11-Strain11

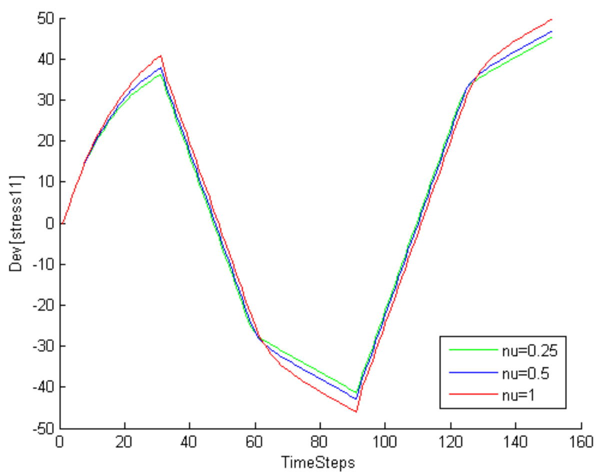

Figure 45: Time-Dev[Stress11] Rate-dependent nonlinear isotropic and linear kinematic hardening plasticity Stress11-Strain11.

13 Matlab code 1D

13.1 main

```
clear all
clc
close all

pp = 1; %non lineal isotropic hardening if pp == 1
nn = 1;  %Exponential saturation law + linear hardening
E = 100; % elastic
K= 30; % isotropic
H= 0; % kinematic
yeld_stress = 20;
stress_inf = 40;
delta_ci = 0.3;
nu = 0.75;
dt = 0.001;%time discretization
istep = 30;

v = [E,K,H];
C = diag(v);

%Rate-independent/rate-dependent perfect plasticity
STRAIN_LOAD = [0 1 0 -1 0 1];
%strain_hsitory
[strain, total_strain_n1] = strain_history( istep, STRAIN_LOAD );
%computing strain
strain_p_n = [0 0 0];  %plastic strain
time = [1:length(strain)];
[sigma_n1,strain_p_n, strain_e] = prova(pps,stress_inf,nu,pp,nn,E,K,H,dt, total_strain_n1,...
yeld_stress, C, strain_p_n,delta_ci);
%plot
color= ['m','c','r','g','b'];
%strain = strain';
for i = 1:5
     figure(1)
     plot(strain, sigma_n1(2:end,1))
     hold on
     legend ('nu=0','nu=0.25','nu=0.5','nu=0.75','nu=1')
     xlabel('Strain')
     ylabel('Sress')

     figure(2)
     hold on
     plot(time, sigma_n1(2:end,1),'r')
     legend('K=10, H=10');
     legend ('nu=0','nu=0.25','nu=0.5','nu=0.75','nu=1')
     xlabel('Time')
     ylabel('Sress')
end
```

13.2 functions

```
function [sigma_n1,strain_p_n, strain_e] = prova(pps,stress_inf,nu,pp,nn,E,K,H,dt, total_strain_n1,...
yeld_stress, C, strain_p_n, delta_ci)
% Compute the trial state at time n+1

strain_p_n = zeros(length(total_strain_n1(:,1)),3);
strain_e = zeros(length(total_strain_n1(:,1)),3);
sigma_n1 = zeros(length(total_strain_n1(:,1)),3);

if nu == 0
   dt = 1;
end

for i = 1:(length(total_strain_n1(:,1)))
  strain_p_trial_n1 = strain_p_n(i,:);
  strain_e_trial_n1 = total_strain_n1(i,:) - strain_p_trial_n1;
  sigma_trial_n1 = (total_strain_n1(i,:) - strain_p_trial_n1)*C;

  if nn == 1 %Exponential saturation law + linear hardening
       sigma_trial_n1(2) = -Exp_sat_law( stress_inf, yeld_stress, K, delta_ci, strain_p_trial_n1);
  end
  %yield function
      f_sigma_trial_n1 = abs(sigma_trial_n1(1) - sigma_trial_n1(3))- yeld_stress + sigma_trial_n1(2);

  %pure plastic case
   if f_sigma_trial_n1 <= 0 %elastic step
      strain_p_n(i+1,:) = strain_p_trial_n1;
      strain_e(i+1,:) = strain_e_trial_n1 ;
      sigma_n1(i+1,:) = sigma_trial_n1;
   else
          if pp == 1 % Newton-Raphson iterative solution algorithm (Nonlinear isotropic hardening )
             gama_new = NRM(yeld_stress,stress_inf,f_sigma_trial_n1,K,E,H,dt,nu, delta_ci,...
             strain_p_trial_n1 );
          else
              gama_new = 1/dt*(E+K+H+nu/dt)^-1 * f_sigma_trial_n1;
          end
   sigma_n1(i+1,1) = sigma_trial_n1(1)- dt*gama_new*E*sign(sigma_trial_n1(1) - sigma_trial_n1(3));
   if pp == 1;
            sigma_n1(i+1,2) = -Exp_sat_law( stress_inf, yeld_stress, K, delta_ci,...
            strain_p_trial_n1+gama_new*dt);
   else
            sigma_n1(i+1,2) = sigma_trial_n1(2) - dt*gama_new * K;
   end
   sigma_n1(i+1,3) = sigma_trial_n1(3) + dt*gama_new * H*sign(sigma_trial_n1(1) - sigma_trial_n1(3));

   strain_p_n(i+1,1) = strain_p_trial_n1(1) + dt*gama_new*sign(sigma_trial_n1(1)- sigma_trial_n1(3));
   strain_p_n(i+1,2) = strain_p_trial_n1(2) + dt*gama_new;
   strain_p_n(i+1,3) = strain_p_trial_n1(3) - dt*gama_new*sign(sigma_trial_n1(1)- sigma_trial_n1(3));
         end
     end
end

function [ gamma_new] = NRM(yeld_stress,stress_inf,f_sigma_trial_n1,K,E,H,dt,nu,delta_ci,...
strain_p_trial_n1 )
```

```
k = 0;
gama_k_n1 = 0.0001;
tol = 1e-8;
[ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_trial_n1, gama_k_n1,K,E,H,dt,nu,delta_ci,...
strain_p_trial_n1 );
while  tol < abs(gn1)
   %solve the linearized equation
   gama_k_n1 = gama_k_n1 - gn1/Dgn1;
   [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_trial_n1, gama_k_n1,K,E,H,dt,nu,delta_ci,...
   strain_p_trial_n1 );
   k = k+1;
end
   gamma_new = gama_k_n1

end

function [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_trial_n1, gama_k_n1,K,E,H,dt,nu,...
delta_ci, strain_p_trial_n1);

%Nonlinear residual scalar equation on the plastic multiplayer
aa = Exp_sat_law( stress_inf, yeld_stress, K, delta_ci, strain_p_trial_n1 );
bb = Exp_sat_law( stress_inf, yeld_stress, K, delta_ci, strain_p_trial_n1 + gama_k_n1*dt );
gn1 = f_sigma_trial_n1 - gama_k_n1*dt*(E+H+nu/dt)-(bb - aa);

%delta_ci = delta_ci * strain_p_trial_n1(2);

ddPI = (stress_inf - yeld_stress)*(dt*delta_ci*exp(-1*(delta_ci*(strain_p_trial_n1(2)+gama_k_n1*dt)))
Dgn1 = -(E+H+nu/dt)*dt - ddPI*dt;
end

function [ dPI ] = Exp_sat_law( stress_inf, yeld_stress, K, delta_ci, strain_p_trial_n1);
dPI = (stress_inf - yeld_stress)*(1-exp(-delta_ci*strain_p_trial_n1(2)))+ K*strain_p_trial_n1(2);
end

function [strain, total_strain_n1] = strain_history( istep, STRAIN_LOAD )

strain = zeros(1,sum(istep)+1);

tramo_b=[];
for i=1:length(STRAIN_LOAD)-1;
     e = linspace(STRAIN_LOAD(i),STRAIN_LOAD(i+1),istep+1);
     e = e(2:end-1);
     tramo_a = [STRAIN_LOAD(i) e];
     tramo_b = [tramo_b tramo_a] ;
end
tramo_b = [tramo_b STRAIN_LOAD(end)] ;
strain = tramo_b;
total = length(strain)
total_strain_n1 = zeros(total,3);
for j = 1:total
    total_strain_n1(j,:) = [strain(1,j) 0 0];
end
end
```

14 Matlab code J2

14.1 main

```
clear all
clc
close all
pp = 0;
nn = 0; % non lineal isotropic hardening if nn == 1
E = 100; % elastic
K=0; % isotropic
H= 30; % kinematic
yeld_stress = 20;
delta = 30;
pois = 0.5;
nu=1;
stress_inf = 40;
%time discretization
dt = 0.001;
istep = 30;
HH = H*eye(6);
[ce,mu] = tensor(E,pois);
%strain_hsitory
[ total_strain] = strain_history1( ce, istep);

[dev_sigma,sigma_n,strain_p_n, strain_e_n] = prova(mu,stress_inf,nu,pp,nn,E,K,H,HH,dt,delta,...
total_strain, yeld_stress, ce);
time = [1:length(total_strain)]';
color= ['m','c','r','g','b'];
for i = 1:5
     figure(1)
     hold on
     plot(total_strain(:,1),d_palst(:,i),color(i))
     legend ('nu=0','nu=0.25','nu=0.5','nu=0.75','nu=1')
     xlabel('Strain11')
     ylabel('Dev[stress11]')

     figure(2)
     hold on
     plot(total_strain(:,1),pperf_palst(:,i),color(i))
     legend ('nu=0','nu=0.25','nu=0.5','nu=0.75','nu=1')
     xlabel('Strain11')
     ylabel('Stress11')

     figure(3)
     hold on
     plot(time,d_palst(:,i) ,color(i))
     legend ('nu=0','nu=0.25','nu=0.5','nu=0.75','nu=1')
     xlabel('TimeSteps')
     ylabel('Dev[stress11]')

end
```

```
function [dev_sigma,sigma_n,strain_p_n, strain_e_n] = prova(mu,stress_inf,nu,pp,nn,E,K,H,HH,...
dt,delta, total_strain, yeld_stress, ce)
% Compute the trial state at time n+1
total_strain_iso = zeros(length(total_strain(:,1)),1);
total_strain_ki = zeros(length(total_strain(:,1)),6);
strain_p_n = zeros(length(total_strain(:,1)),6);
strain_p_iso = zeros(length(total_strain(:,1)),1);
strain_p_ki = zeros(length(total_strain(:,1)),6);
strain_e_n = zeros(length(total_strain(:,1)),6);
strain_e_iso = zeros(length(total_strain(:,1)),1);
strain_e_ki = zeros(length(total_strain(:,1)),6);
sigma_n = zeros(length(total_strain(:,1)),6);
q_n = zeros(length(total_strain(:,1)),1);
q_bar_n = zeros(length(total_strain(:,1)),6);
dev_sigma = zeros(length(total_strain(:,1)),6);

if nu == 0
   dt = 1;
end

for i = 1:length(total_strain(:,1))-1

strain_p_n_trial = strain_p_n(i,:);
strain_p_iso_trial = strain_p_iso(i,:);
strain_p_ki_trial = strain_p_ki(i,:);

strain_e_n_trial = total_strain(i+1,:) - strain_p_n_trial;
strain_e_iso_trial = total_strain_iso(i+1,:) - strain_p_iso_trial;
strain_e_ki_trial = total_strain_ki(i+1,:) - strain_p_ki_trial;

sigma_n_trial = strain_e_n_trial*ce';
q_n_trial = strain_e_iso_trial*K;
q_bar_n_trial = strain_e_ki_trial*HH;


  %deviatoric part of sigma
   dev_sigma_trial = deviatoric(sigma_n_trial);
   den = dev_sigma_trial - q_bar_n_trial;
   n_trial = den / norm(den);

     if nn == 1 %Exponential saturation law + linear hardening
      q_n_trial = -1*Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial);

   end

   %yield function
      f_sigma_n_trial = norm(den) - sqrt(2/3)*(yeld_stress - q_n_trial);

%pure plastic case
   if f_sigma_n_trial <= 0 %elastic step
      gamma = 0;

      sigma_n(i+1,:) = sigma_n_trial;
      q_n(i+1) = q_n_trial ;
```

```
      q_bar_n(i+1,:) = q_bar_n_trial;
      dev_sigma(i+1,:)  = dev_sigma_trial;%deviatoric(sigma_trial_n1);

      strain_p_n(i+1,:) = strain_p_n_trial;
      strain_p_iso(i+1,:) = strain_p_iso_trial;
      strain_p_ki(i+1,:) = strain_p_ki_trial;

      strain_e_n(i+1,:) = strain_e_n_trial;
      strain_e_iso(i+1,:) = strain_e_iso_trial;
      strain_e_ki(i+1,:) = strain_e_iso_trial;

   else

           if pp == 1 % Newton-Raphson iterative solution algorithm (Nonlinear isotropic hardening )

              gamma = NRM(yeld_stress,stress_inf,f_sigma_n_trial,mu,K,E,H,dt,...
              nu, delta,strain_p_iso_trial );

           else
              gamma = (2*mu+2/3*K+2/3*H+nu/dt)^(-1) *f_sigma_n_trial/dt;
           end

   sigma_n(i+1,:)= sigma_n_trial - dt*gamma*2*mu*n_trial;
if pp == 1
   q_n(i+1,:)= -Exp_sat_law(stress_inf,yeld_stress,K,delta,strain_p_iso_trial+gamma*dt*sqrt(2/3));
else
    q_n(i+1,:) = q_n_trial - dt*gamma*K*sqrt(2/3);
end
   q_bar_n(i+1,:)= q_bar_n_trial + dt* gamma*2/3*H*n_trial;
   dev_sigma(i+1,:)  = deviatoric(sigma_n_trial) - dt*gamma*2*mu*n_trial;

   strain_p_n(i+1,:)= strain_p_n_trial + dt*gamma*n_trial;
   strain_p_iso(i+1,:)= strain_p_iso_trial + dt*gamma*sqrt(2/3);
   strain_p_ki(i+1,:)= strain_p_ki_trial - dt*gamma*n_trial;
   end
end
end

function [ gamma] = NRM(yeld_stress,stress_inf,f_sigma_n_trial,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial )
%UNTITLED4 Summary of this function goes here
%   Detailed explanation goes here
k = 0;
gama_k_n1 = 0;
tol = 1e-6;

[ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial );
while abs(gn1) > tol
   %solve the linearized equation
   gama_k_n1 = gama_k_n1 - gn1/Dgn1;
   [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
   dt,nu,delta,strain_p_iso_trial );
   k = k+1;
end
```

```
   gamma = gama_k_n1;

end

function [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial)

%Nonlinear residual scalar equation on the plastic multiplayer
aa = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial );
bb = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial + gama_k_n1*sqrt(2/3)*dt );
gn1 = f_sigma_n_trial - gama_k_n1*dt*(2*mu+2/3*H+nu/dt)-sqrt(2/3)*(bb - aa);
ddPI = (stress_inf - yeld_stress)*delta*sqrt(2/3)*...
dt*exp(-delta*(strain_p_iso_trial + sqrt(2/3)*gama_k_n1*dt))+K*dt*sqrt(2/3);
Dgn1 = -dt*((2*mu+(2/3)*H+nu/dt) + 2/3*ddPI);
end

function [ dPI ] = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial)

dPI = (stress_inf - yeld_stress)*...
(1-exp(-delta*strain_p_iso_trial))+ K*strain_p_iso_trial;
%q_non_lin = -dPI;
end

function [ dev_sigma ] = deviatoric(sigma_n_trial  )

s = sigma_n_trial ;
trace = s(1)+s(2)+s(3);
dev_sigma= [s(1)-(1/3)*trace,s(2)-(1/3)*trace,s(3)-(1/3)*trace,s(4),s(5),s(6)];
end

function [ce,mu] = tensor( E, pois)
mu = E/(2*(1-pois));
lame = pois*E/(1+pois)*(1-2*pois);

     ce    = zeros(6,6);             % Init.

for i=1:3
     ce(i,i)=2*mu + lame;
end
for i=4:6
    ce(i,i) = mu;
end
        ce(1,2)=lame;
        ce(1,3)=lame;
        ce(2,3)=lame;
        ce(2,1)=lame;
        ce(3,1)=lame;
        ce(3,2)=lame;
end

function [  total_strain  ] = strain_history1( ce, istep )
ce_1 = inv(ce);
step=5*istep+1;
stress_load = [0 100 0 -100 0 100];
```

```
stress = zeros(6,step);
strain = zeros(6,step);
tramo_b=[];
for i=1:length(stress_load)-1
     e = linspace(stress_load(i),stress_load(i+1),istep+1);
     e = e(2:end-1);
     tramo_a = [stress_load(i) e];
     tramo_b = [tramo_b tramo_a] ;
end
tramo_b = [tramo_b stress_load(end)] ;
stress(1,:) = tramo_b;
for i=1:step
    strain(:,i) = ce_1*stress(:,i);
end
for i=1:3
total_strain = strain';
end
```

14.2 functions

```
function [ gamma] = NRM(yeld_stress,stress_inf,f_sigma_n_trial,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial )
%UNTITLED4 Summary of this function goes here
%   Detailed explanation goes here
k = 0;
gama_k_n1 = 0;
tol = 1e-6;

[ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial );
while abs(gn1) > tol
   %solve the linearized equation
   gama_k_n1 = gama_k_n1 - gn1/Dgn1;
   [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
   dt,nu,delta,strain_p_iso_trial );
   k = k+1;
end
   gamma = gama_k_n1;

end

function [ gn1,Dgn1 ] = nrse(yeld_stress,stress_inf,f_sigma_n_trial, gama_k_n1,mu,K,E,H,...
dt,nu,delta,strain_p_iso_trial)

%Nonlinear residual scalar equation on the plastic multiplayer
aa = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial );
bb = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial + gama_k_n1*sqrt(2/3)*dt );
gn1 = f_sigma_n_trial - gama_k_n1*dt*(2*mu+2/3*H+nu/dt)-sqrt(2/3)*(bb - aa);
ddPI = (stress_inf - yeld_stress)*delta*sqrt(2/3)*...
dt*exp(-delta*(strain_p_iso_trial+ sqrt(2/3)*gama_k_n1*dt))+K*dt*sqrt(2/3);
Dgn1 = -dt*((2*mu+(2/3)*H+nu/dt) + 2/3*ddPI);
end

function [ dPI ] = Exp_sat_law( stress_inf, yeld_stress, K, delta, strain_p_iso_trial)
```

```
dPI = (stress_inf - yeld_stress)*(1-exp(-delta*strain_p_iso_trial))+ K*strain_p_iso_trial;
%q_non_lin = -dPI;
end

function [ dev_sigma ] = deviatoric(sigma_n_trial  )

s = sigma_n_trial ;
trace = s(1)+s(2)+s(3);
dev_sigma= [s(1)-(1/3)*trace,s(2)-(1/3)*trace,s(3)-(1/3)*trace,s(4),s(5),s(6)];
end

function [ce,mu] = tensor( E, pois)
mu = E/(2*(1-pois));
lame = pois*E/(1+pois)*(1-2*pois);

     ce    = zeros(6,6);            % Init.

for i=1:3
     ce(i,i)=2*mu + lame;
end
for i=4:6
    ce(i,i) = mu;
end
        ce(1,2)=lame;
        ce(1,3)=lame;
        ce(2,3)=lame;
        ce(2,1)=lame;
        ce(3,1)=lame;
        ce(3,2)=lame;
end

function [  total_strain  ] = strain_history1( ce, istep )
ce_1 = inv(ce);
step=5*istep+1;
stress_load = [0 100 0 -100 0 100];
stress = zeros(6,step);
strain = zeros(6,step);
tramo_b=[];
for i=1:length(stress_load)-1
     e = linspace(stress_load(i),stress_load(i+1),istep+1);
     e = e(2:end-1);
     tramo_a = [stress_load(i) e];
     tramo_b = [tramo_b tramo_a] ;
end
tramo_b = [tramo_b stress_load(end)] ;
stress(1,:) = tramo_b;
for i=1:step
    strain(:,i) = ce_1*stress(:,i);
end
for i=1:3
total_strain = strain';
end
```