

# Computational Solid Mechanics

## Assignment 1 - Continuum Damage Model

Arjun Ajay\*

*Master of Science in Computational Mechanics, UPC, Barcelona*

E-mail: arjunajay100@gmail.com

### Introduction

The report has been divided into two parts. Part I deals with the analysis of the Rate independent model and Part II deals with the analysis of the Rate dependent model. The rate independent non-symmetric tension compression continuum damage model and the rate independent tension only continuum damage model are implemented in MATLAB to analyse the stress state and the evolution of the elastic domain for the Plane strain case. For the rate dependent model, the effects of strain rate, viscous coefficient and time integration parameter ( $\alpha$ ) on the stress path in the stress space and stress-strain plot are analysed.

### Rate Independent Model

#### Tension Only Model

The damage surface in the stress space for a tension only model is shown in the Figure. 1. As seen in the figure, the damage surface is infinite in the compressive domain, implying that the material is infinitely elastic under compression, but it has a finite damage surface under tension.

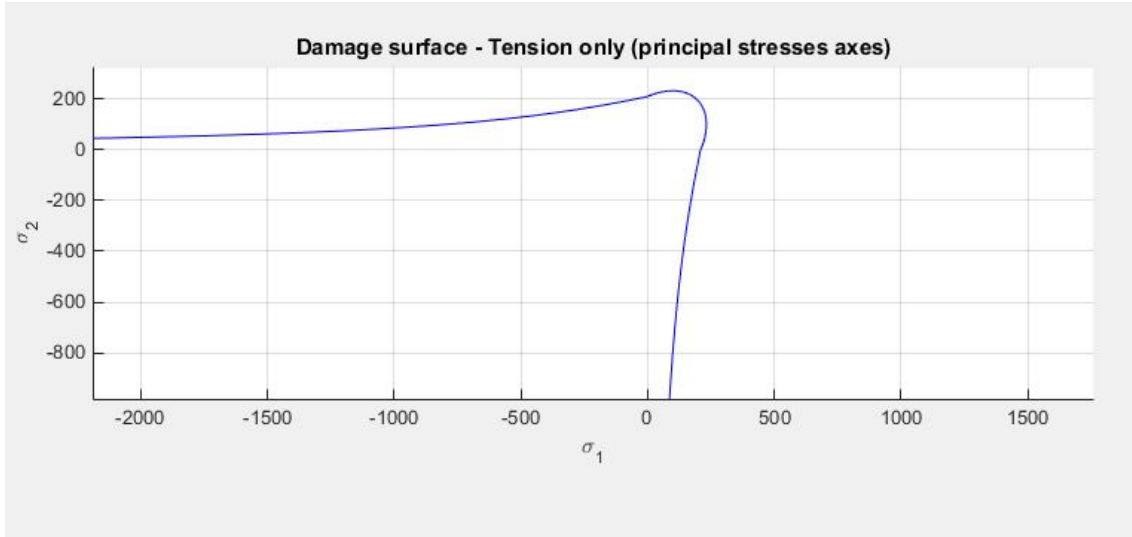


Figure 1: Tension only damage surface

### Linear Hardening/Softening

To understand the effect of Linear Hardening/Softening we consider a test case(I) as shown in Figure. 2 with load states :

$$\begin{aligned} \bar{\sigma}_1^{(0)} &= 0, \bar{\sigma}_2^{(0)} = 0 \\ \bar{\sigma}_1^{(1)} &= 300, \bar{\sigma}_2^{(1)} = 400 \\ \bar{\sigma}_1^{(2)} &= 500, \bar{\sigma}_2^{(2)} = 400 \\ \bar{\sigma}_1^{(3)} &= 500, \bar{\sigma}_2^{(3)} = 0 \end{aligned}$$

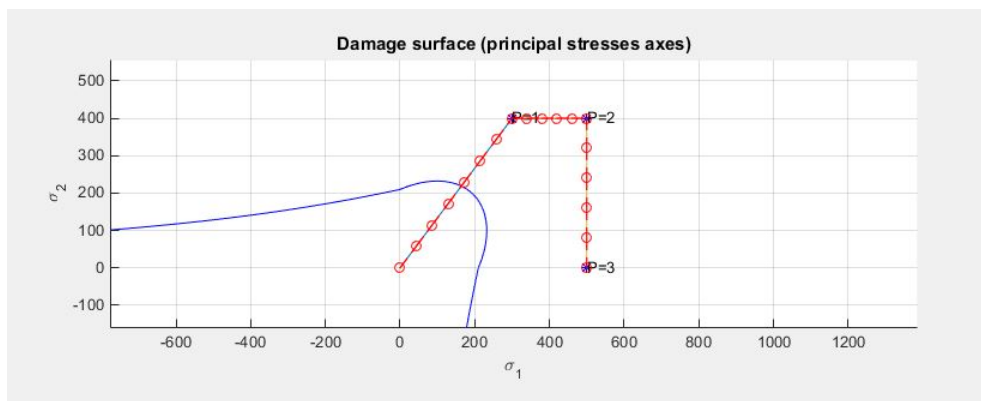


Figure 2: Load state for Tension only model

For linear hardening, the Hardening modulus(H) is taken to be 0.1 and the  $r_0$  value is found to be 1.4. Figure. 3 and Figure. 4 show the path in the stress space and the corresponding plot between hardening variable(q) and strain like variable(r). As shown in Figure.3a and 4a, the damage surface expands as the material hardens while it shrinks for a softening material. Figures 3b and 4b show the linear variation of q with respect to r.

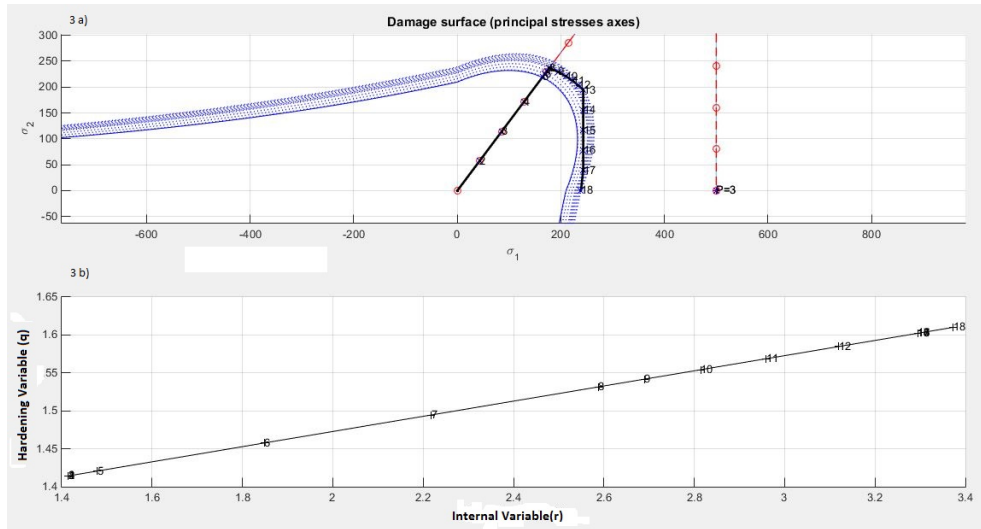


Figure 3: Linear Hardening stress path and corresponding q vs r

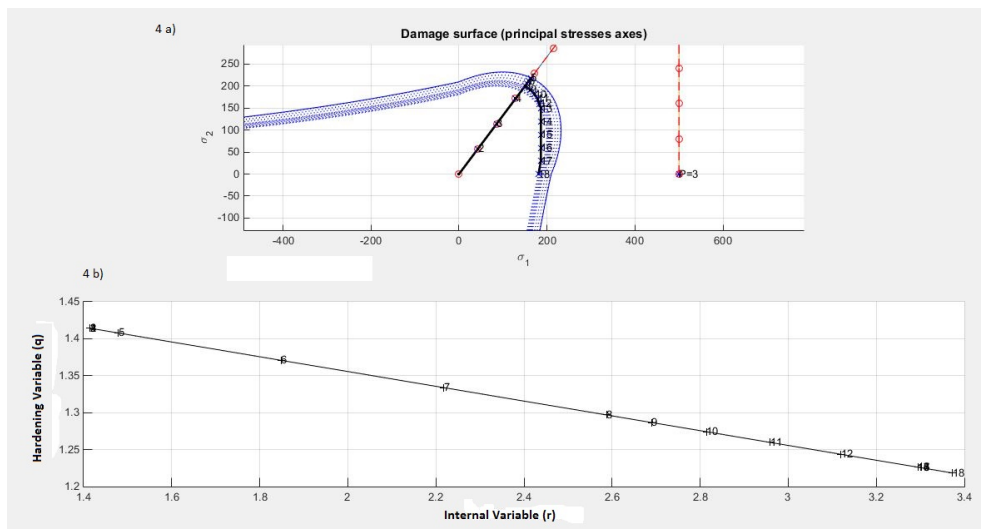


Figure 4: Linear Softening stress path and corresponding q vs r

## Exponential Hardening/Softening

The same test case as for the Linear Hardening is chosen. The minimum value that  $q$  can take is assumed to be  $q_{min} = 1.42 \times 10^{-6}$  and the maximum value is taken to be  $q_{max} = r_0 + 2$ . The value of  $A$  for the exponential model is assumed to be 0.4. The  $q_{max}$  has been exaggerated to clearly see the exponential behavior in the stress strain plot.

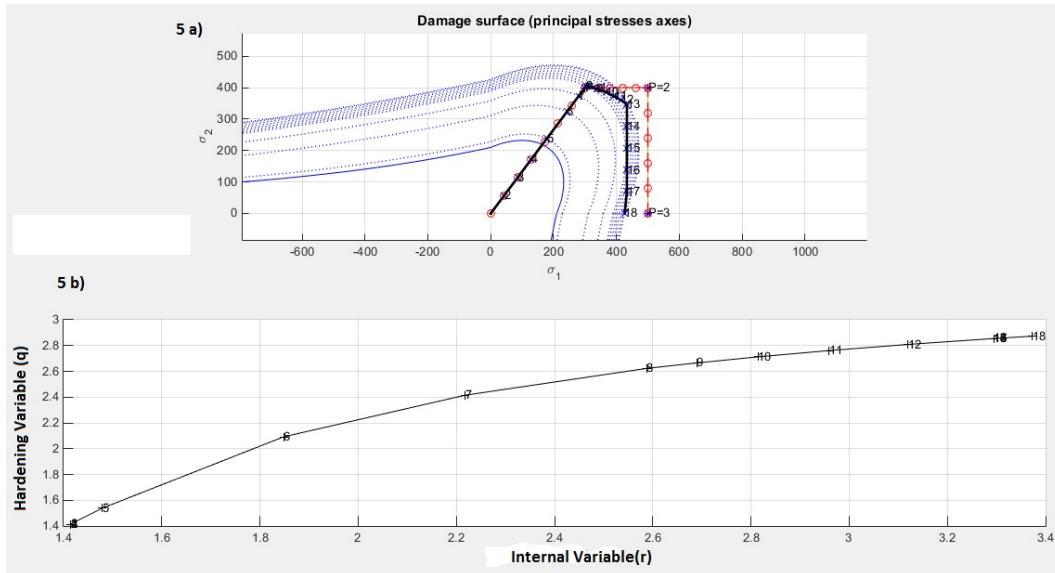


Figure 5: Exponential Hardening stress path and corresponding  $q$  vs  $r$

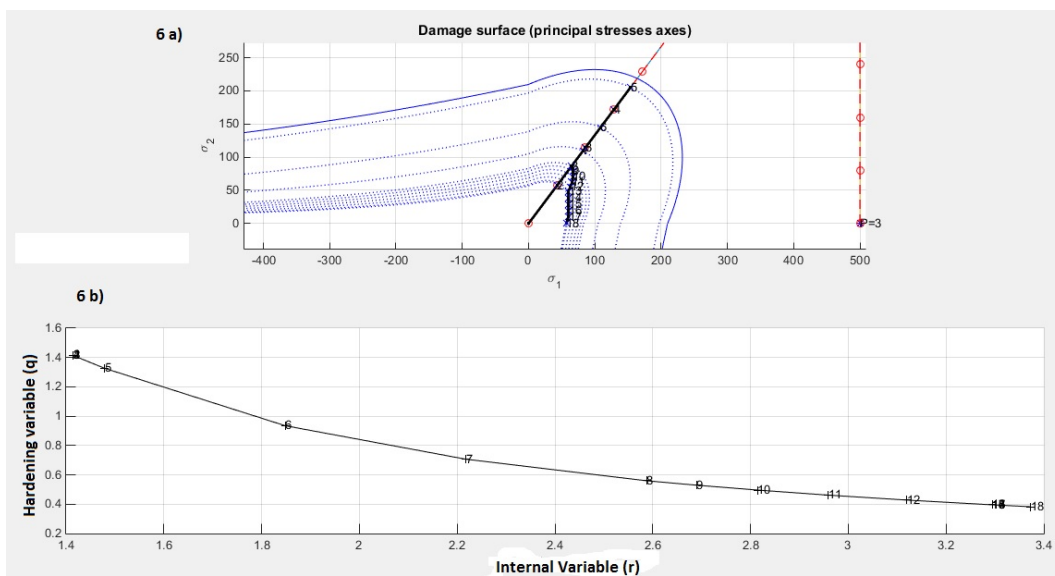


Figure 6: Exponential Softening stress path and corresponding  $q$  vs  $r$

With these constant values defined, Figure. 5 and Figure. 6 show the path in the stress space and the corresponding plot between hardening variable( $q$ ) and strain like variable( $r$ ). As shown in Figure.5a and 6a, the damage surface expands as the material hardens while it shrinks for a softening material. Figures 3b and 4b show the exponential variation of  $q$  with respect to  $r$ .

### Verification of the Model

3 cases of stress increments are chosen to assess the correctness of the implementation. The increments are chosen as:  $\alpha = 300$ ,  $\beta = 200$  and  $\gamma = 400$ . As the effects of linear and exponential hardening/softening has already been discussed, linear hardening with  $H = 0.1$  is chosen for all the 3 cases.

1. Case one is uniaxial loading and unloading with  $\bar{\sigma}_2 = 0$  and  $\bar{\sigma}_1$  given by :

$$\Delta\bar{\sigma}_1^{(1)} = \alpha$$

$$\Delta\bar{\sigma}_1^{(2)} = -\beta$$

$$\Delta\bar{\sigma}_1^{(3)} = -\gamma$$

The path in the stress space is shown in the Figure. 7 and the corresponding stress ( $\sigma_1$ ) vs strain ( $\epsilon_1$ ) is shown in Figure. 8. As seen in Figure. 7 during the first stress increment(marked by numbers in pink) the material is loaded elastically until it reached the damage surface. Loading beyond that leads to the expansion of the damage surface. During the second stress increment(marked by numbers in red) the material is unloaded elastically and during the third stress increment(marked by numbers in black) the material is again loaded elastically until it reached the new damage surface after which the damage surface further expands.

The material behavior can also be analysed from the Stress1-Strain1 plot( Figure. 8 ). Initially the material is loaded until its elastic limit (numbers 1-8), beyond which it undergoes linear hardening as seen by the change in the slope (8-11). During the

second stress increment, the material is elastically unloaded (11-18) and during the third increment the material is elastically loaded until the new elastic limit (18-22) beyond which it undergoes linear hardening (22-26).

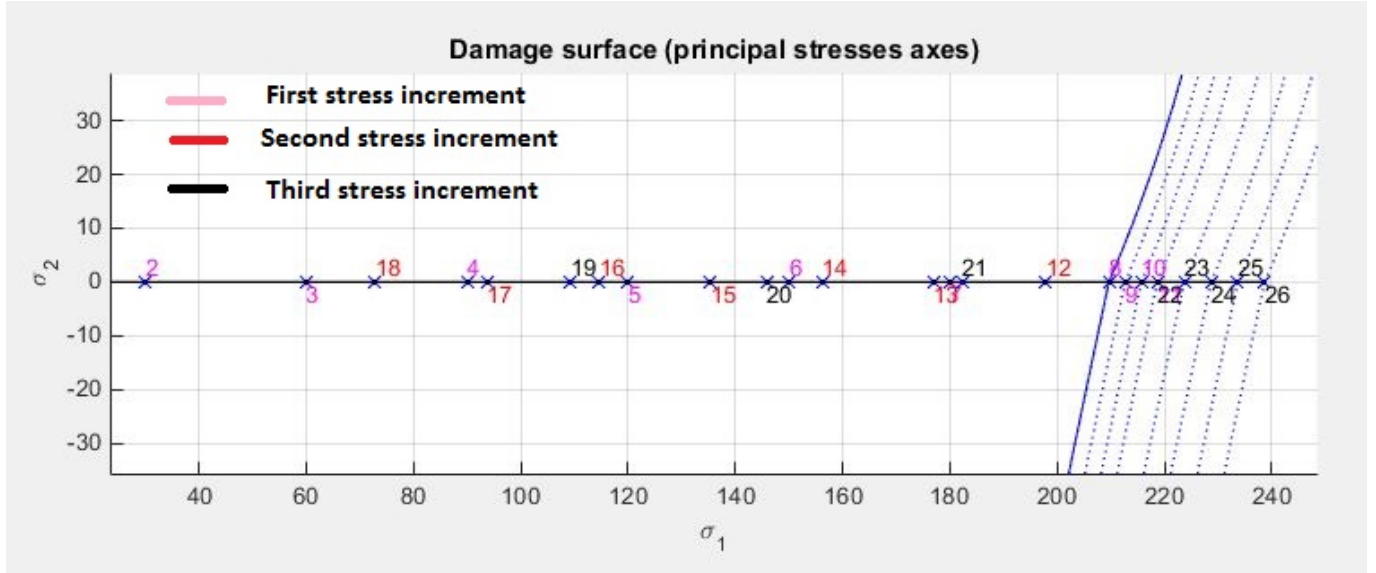


Figure 7: Path in the stress space

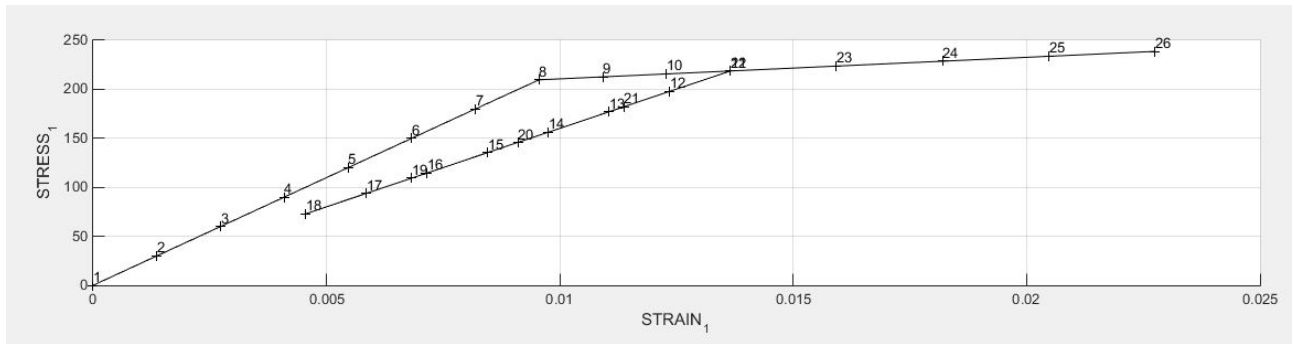


Figure 8: Stress1 strain1 plot

2. Case 2 is a combination of uniaxial loading and biaxial loading and unloading, given by :

$$\begin{aligned} \Delta \bar{\sigma}_1^{(1)} &= \alpha & \Delta \bar{\sigma}_2^{(1)} &= 0 \\ \Delta \bar{\sigma}_1^{(2)} &= -\beta & \Delta \bar{\sigma}_2^{(2)} &= -\beta \\ \Delta \bar{\sigma}_1^{(3)} &= \gamma & \Delta \bar{\sigma}_2^{(3)} &= \gamma \end{aligned}$$

The path in the stress space is shown in the Figure 9 and the corresponding norm stress vs norm strain is shown in Figure 10.

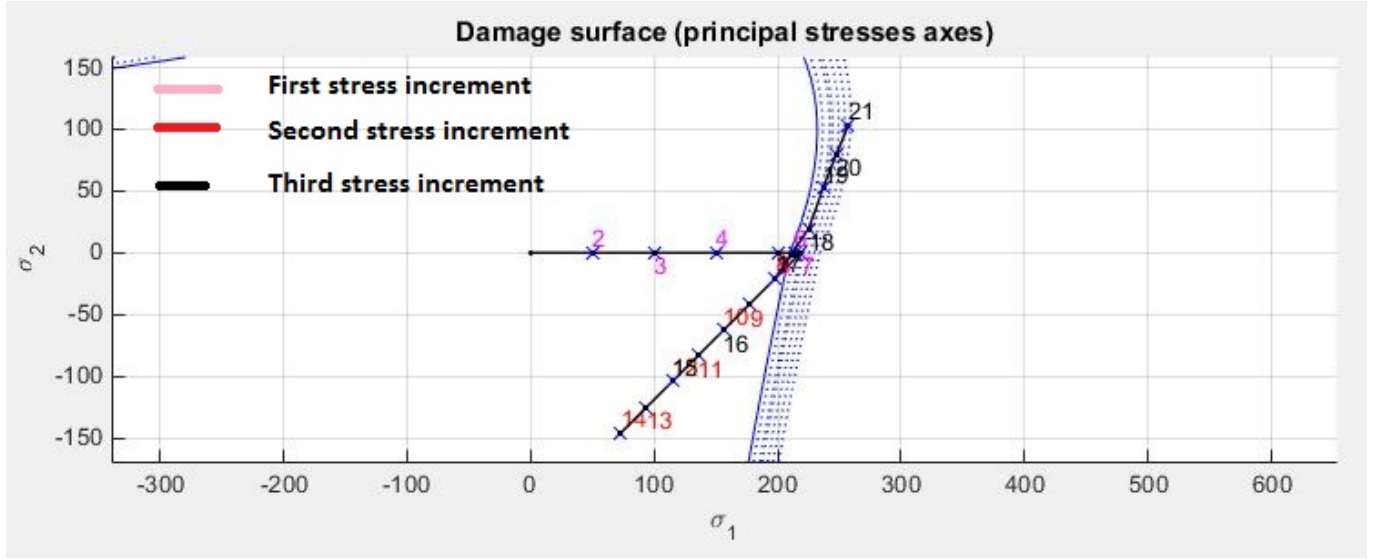


Figure 9: Path in the stress space

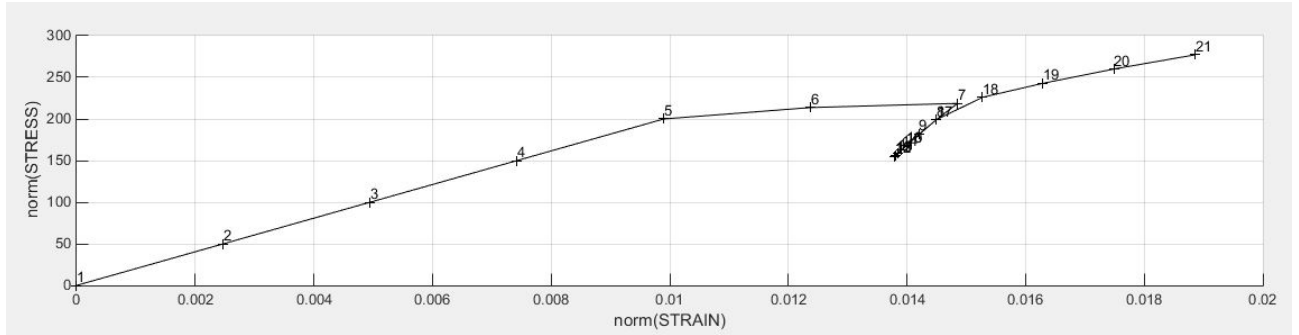


Figure 10: Norm Stress Norm strain plot for case 2

3. Case 3 is biaxial loading and unloading, given by :

$$\begin{aligned} \Delta \bar{\sigma}_1^{(1)} &= \alpha & \Delta \bar{\sigma}_2^{(1)} &= \alpha \\ \Delta \bar{\sigma}_1^{(2)} &= -\beta & \Delta \bar{\sigma}_2^{(2)} &= -\beta \\ \Delta \bar{\sigma}_1^{(3)} &= \gamma & \Delta \bar{\sigma}_2^{(3)} &= \gamma \end{aligned}$$

The path in the stress space is shown in the Figure 11 and the corresponding norm stress vs norm strain is shown in Figure 12.

With these three cases the implementation of the Tensile only model has been verified as the stresses do not exceed the damage surface and are always in the elastic domain.

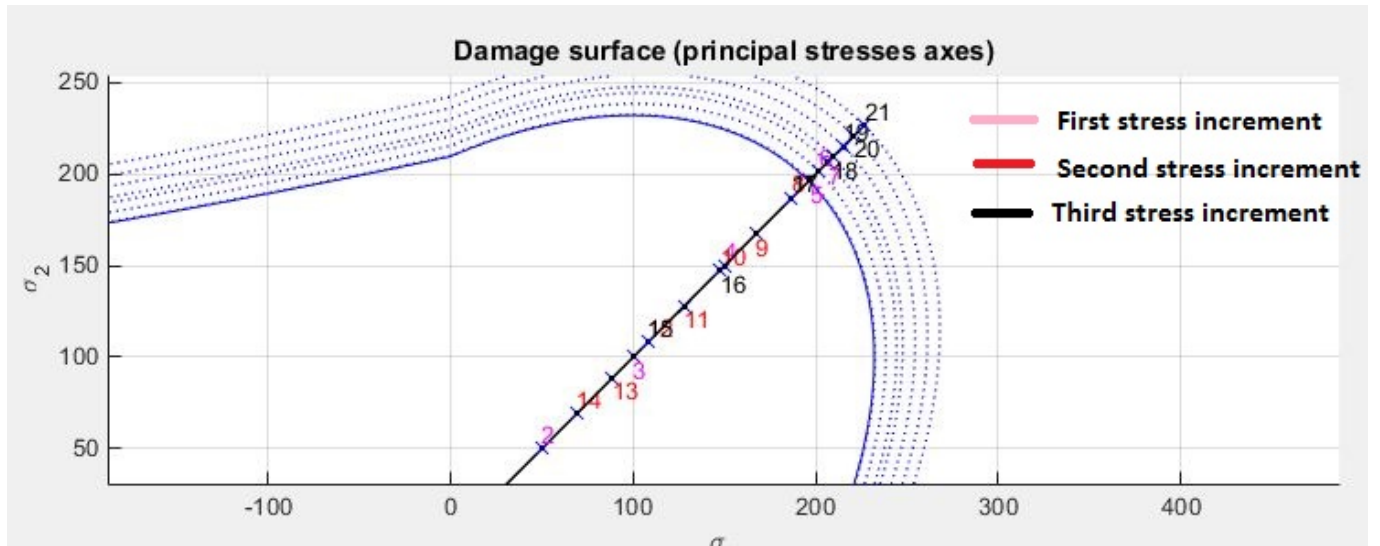


Figure 11: Path in the stress space

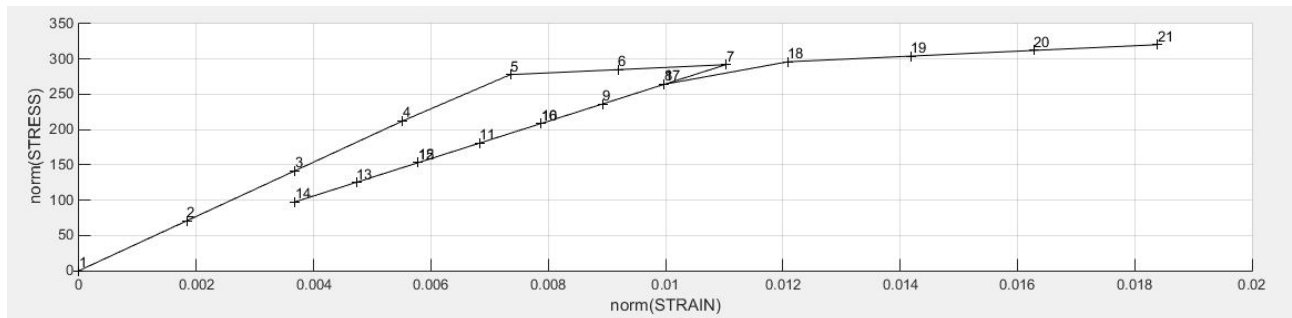


Figure 12: Norm Stress Norm strain plot for case 3

## Non symmetric Model

The damage surface in the stress space for a Non symmetric tension compression model is shown in the figure 13. As seen in the figure, the damage surface has both tension and compression parts, with the material having higher elastic limit in Compression.

We consider the same test case as for the tension only model to analyse the behaviour of the Non symmetric model for linear and exponential hardening/softening. The damage



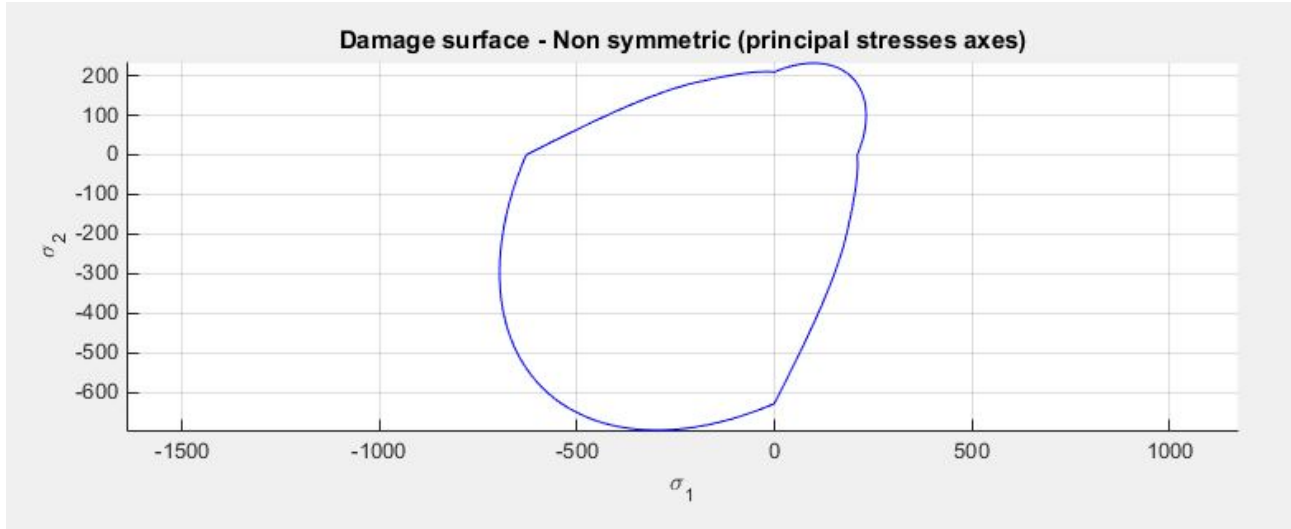


Figure 13: Damage surface for the non-symmetric model

surface undergoes expansion under hardening while it shrinks during softening similar to the case of the Tension only model (see Figures 3, 4, 5 and 6).

### Verification of the Implementation of the Model

The cases one, two and three of the previous section are used to verify the implementation of the model. The same value of  $\alpha$ ,  $\beta$  and  $\gamma$  are used.

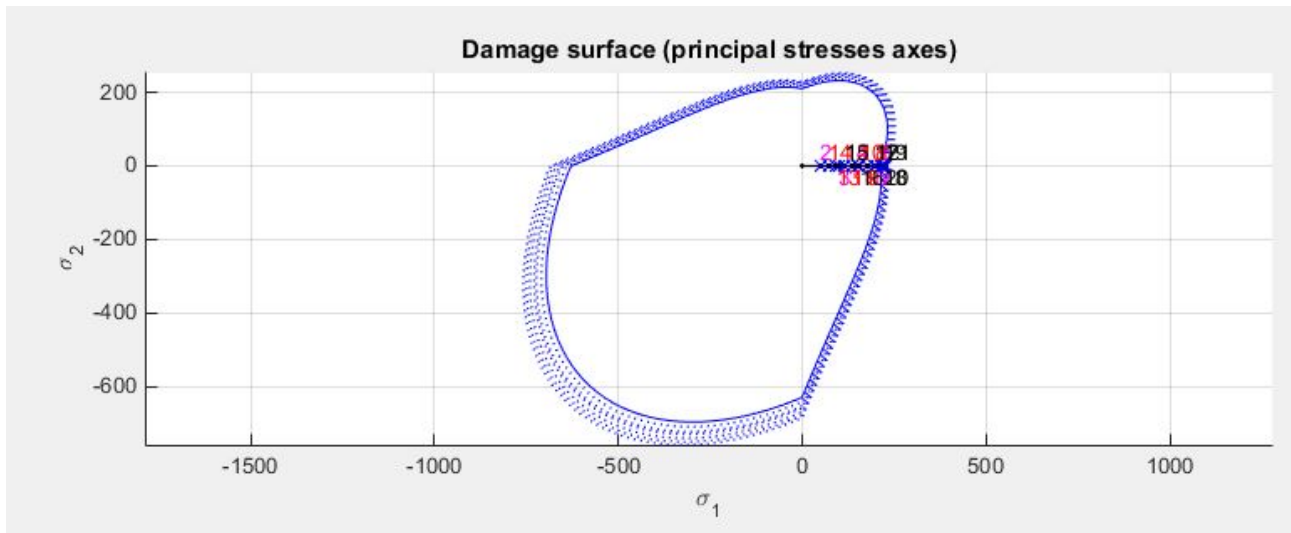


Figure 14: Path in the stress space

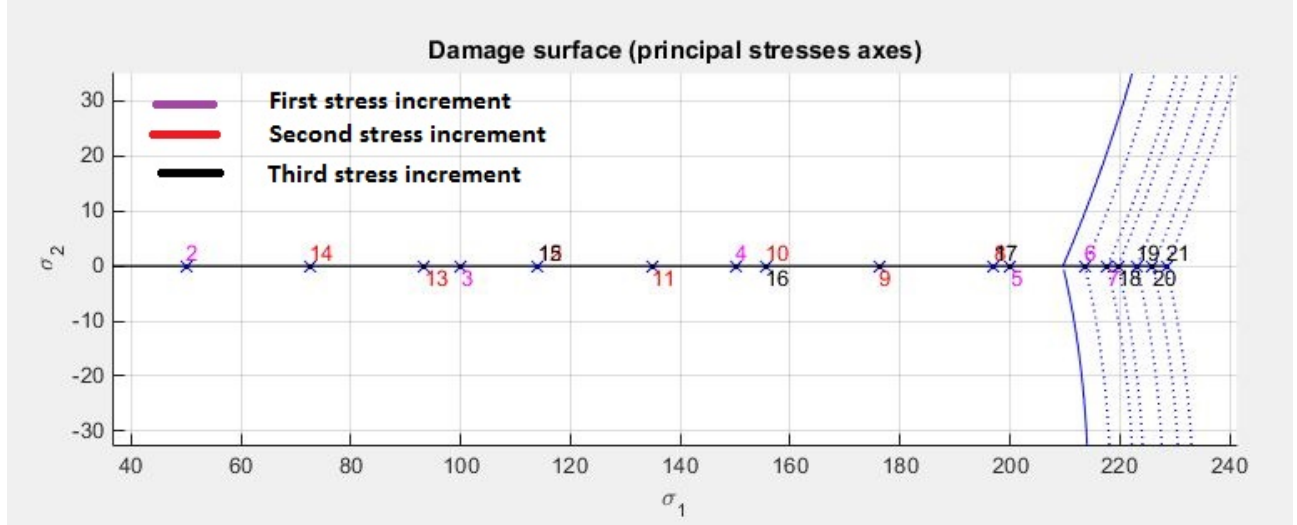


Figure 15: Zoomed in view of the path in the stress space

However, we use exponential hardening here with  $q_{max} = r_0 + (r_0 - q_{min})$  where  $r_0$  is the initial value of the hardening variable ( $q$ ) and  $q_{max}$  and  $q_{min}$  are as defined before and  $A = 0.1$ .

1. For case1 (Uniaxial loading and unloading), Figure. 14 and Figure. 15 show the path in the stress space and its zoomed in view. The material behavior is similar to that observed in the tensile only model explained in the Page 5.

In case 2(combination of uniaxial loading and biaxial loading and unloading) and case 3 (biaxial loading and unloading ) we get stress paths similar to that of case 2 and 3 of the tensile only model. This is because, although the material has an elastic limit under compression for the Non symmetric model, the limit is high and the material behaves elastically under compression for the load paths chosen in the cases one, two and three.

## Rate Dependent Model

The rate dependent model is studies for Symmetric tension compression under the plain strain condition. Case(I)(Page-2) is used as the Load path for the analysis of the effect of

various parameters in the Rate dependent model. As Case(I) involves biaxial loading and unloading, norm stress and norm strain, hereafter referred as stress and strain are used in the stress-strain plots.

### Effect of the Viscosity Parameter

Figure. 16 and Figure. 17 show the effect of the viscosity coefficient on the path in the stress space and in the Stress strain plot.

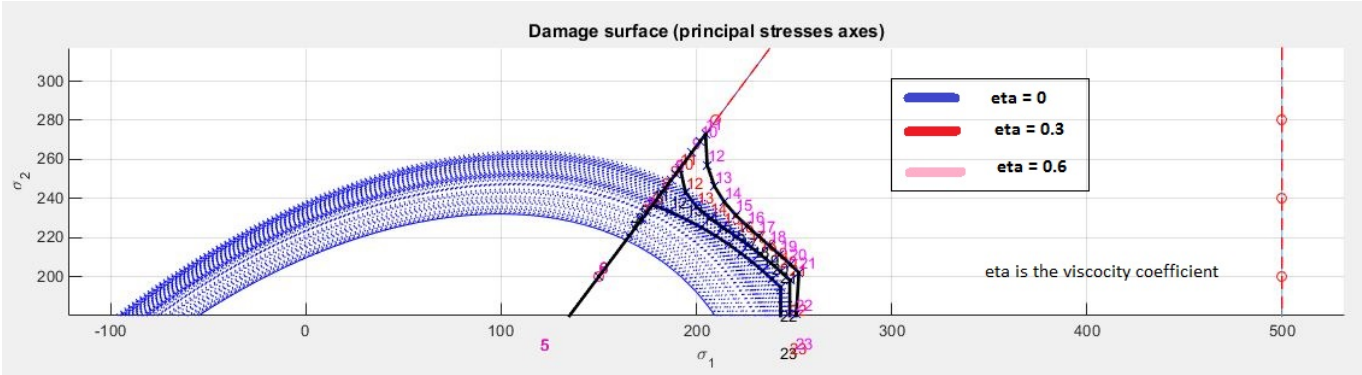


Figure 16: Effect of the viscosity Parameter on the Path in the stress space

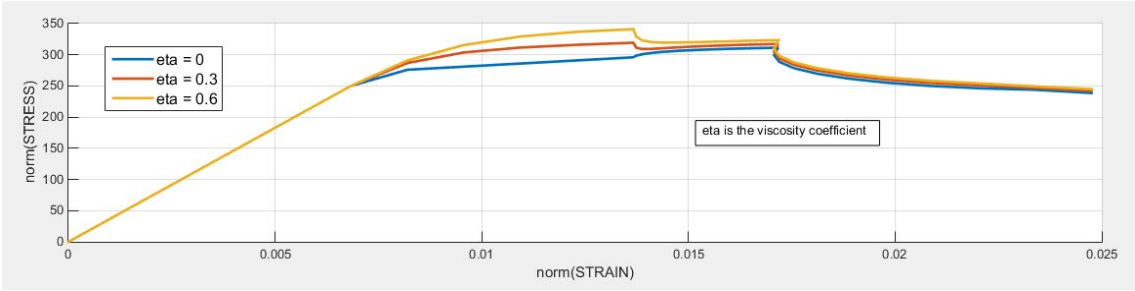


Figure 17: Variation of the stress strain plot with the viscous parameter.

eta = 0 is the non-viscous case and the material behaves like a rate independent model. This is shown in Figure. 16 (marked by blue numbers), where the stresses do not cross the damage surface. It is seen that for higher value of the viscous parameter the stresses in the inelastic domain are further from the damage surface. From Figure. 16, it can be concluded that the viscous parameter does not affect the stress - strain behavior in the elastic domain

but in the inelastic domain the stresses increase with viscosity parameter for the same strain value.

### Effect of the strain rate

In the rate dependent model, the stresses depend not only on the strains but also on the strain rate. Figure. 18 and Figure. 19 show the effect of the strain rate on the path in the stress space and in the Stress strain plot.

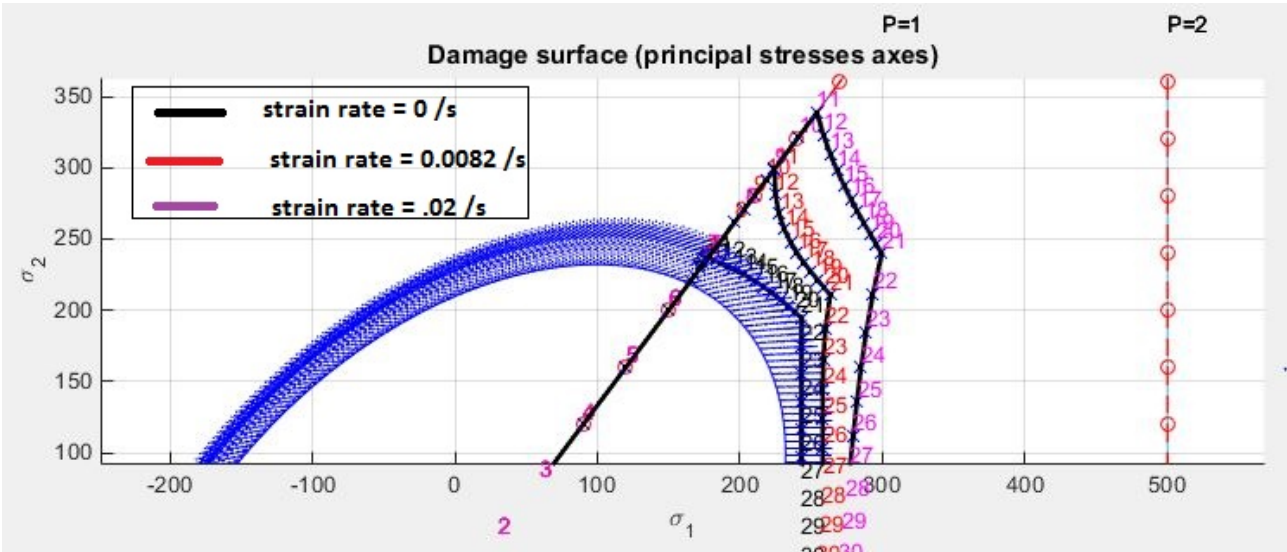


Figure 18: Effect of the Strain rate on the Path in the stress space

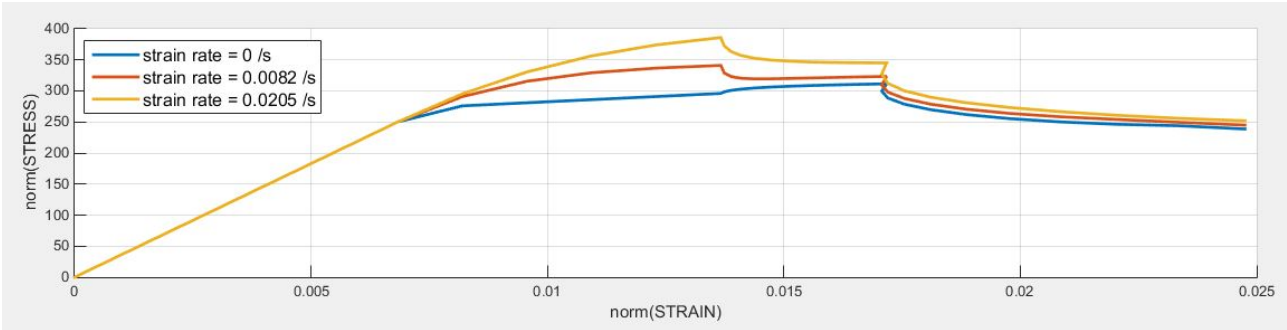


Figure 19: Variation of the stress strain plot with the Strain rate.

The strain rates were varied by changing the total time for the given strain history and

were calculated as the rate of the norm strain.

It is seen that as the strain rate increases, the stresses in the inelastic domain are further away from the damage surface. From Figure. 19, it can be concluded that the strain rate does not affect the stress - strain behavior in the elastic domain but in the inelastic domain the stresses increase with strain rate for the same strain value.

### Effect of $\alpha$ (the time integration parameter)

Figure. 20 shows the effect of the  $\alpha$  parameter on Stress strain plot.

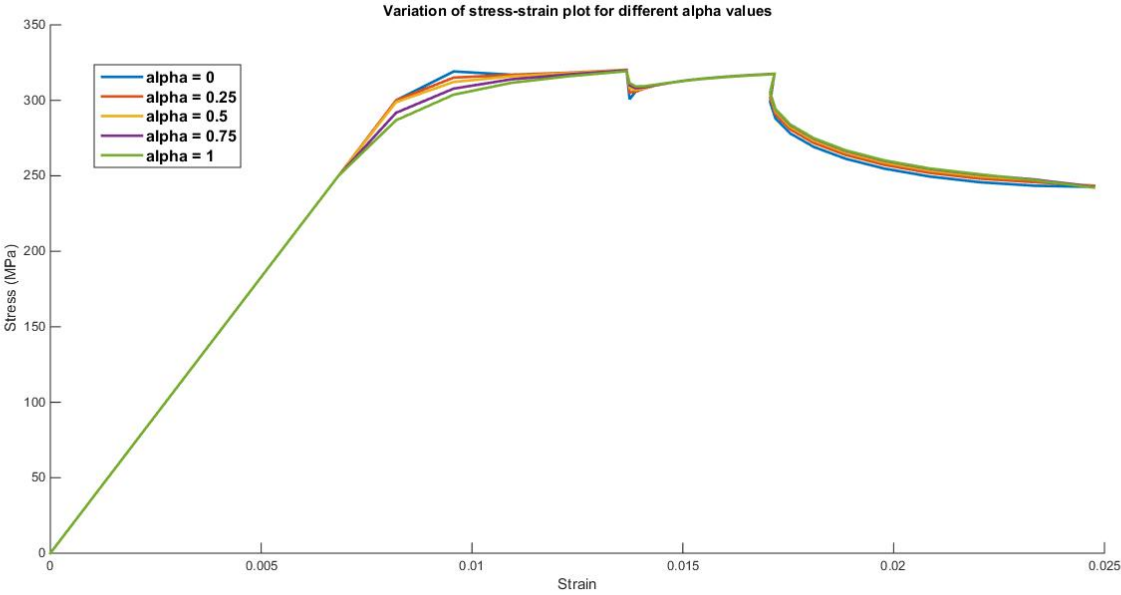


Figure 20: Variation of the stress strain plot with the  $\alpha$  parameter

It is seen that in the elastic domain, there is no effect of  $\alpha$  on the stress-strain plot. In the inelastic domain, as  $\alpha$  increases the value of the stresses are smoothed and there is lesser variation. For  $\alpha$  less than 0.5 for higher time steps  $\Delta t$ , instabilities are induced and the stress strain plots may be erroneous as shown in the Figure. 21.

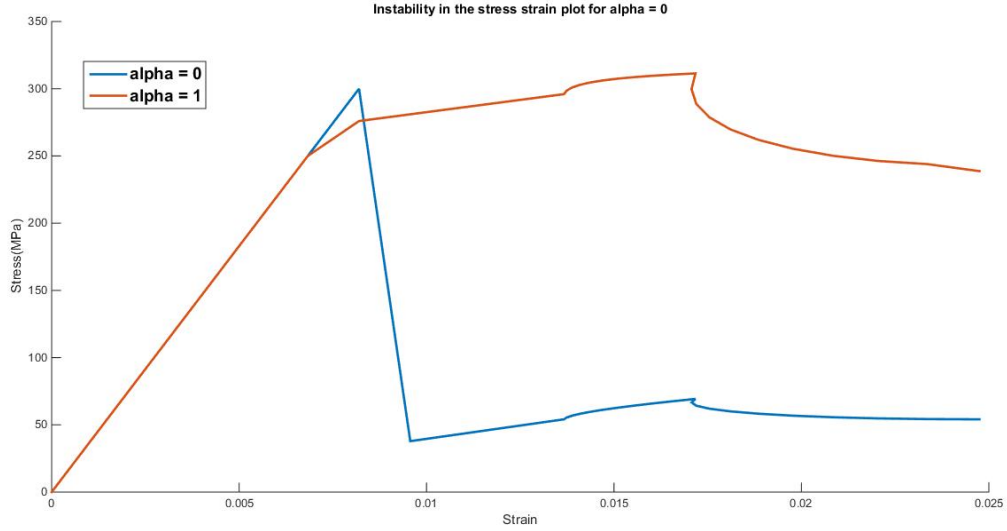


Figure 21: Instability of the stress - strain plot for  $\alpha = 0$

## Effect of the $\alpha$ values on the $C_{11}$ component of the Tangent and Algorithmic Constitutive operator

Figure. 22 and Figure. 23 show the effect of  $\alpha$  on the  $C_{11}$  component of the Tangent and Algorithmic Constitutive operator.

It is seen that as the value of  $\alpha$  increases the elastic domain denoted by a constant  $C_{11}$  (equal to corresponding component of the constitutive matrix) on both the plots decreases. Also, it is noted that fluctuations in the  $C_{11}$  component of the algorithmic constitutive operator are more for higher value of  $\alpha$ .

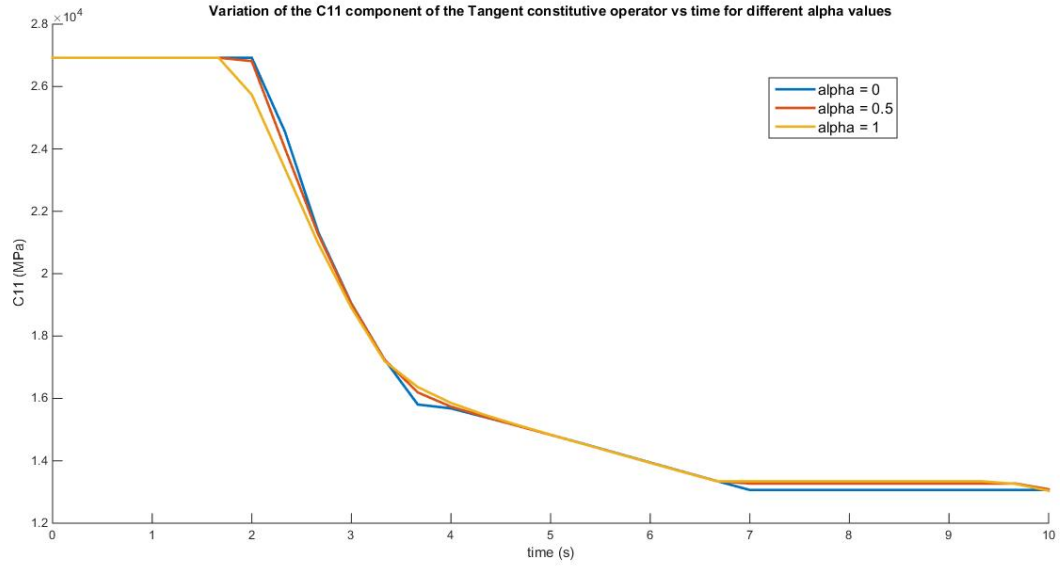


Figure 22: Effect of  $\alpha$  on the C11 component of  $C_{tan}$

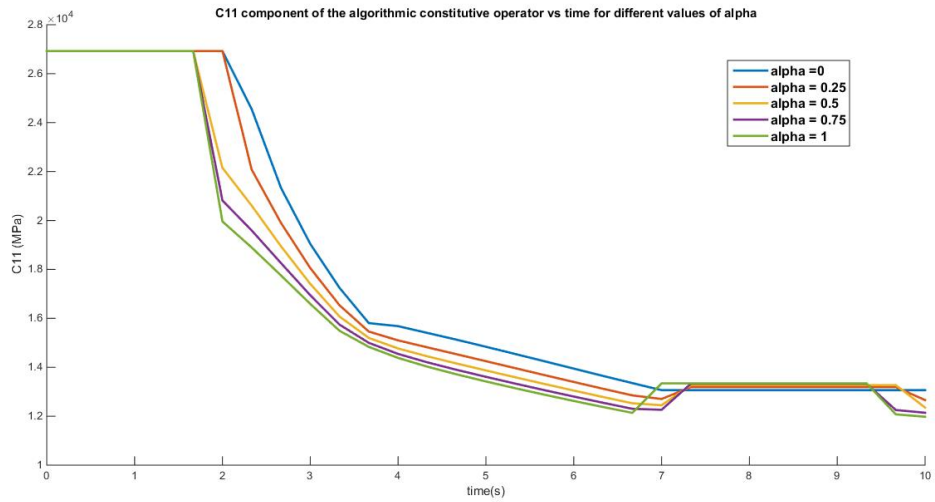


Figure 23: Effect of  $\alpha$  on the C11 component of  $C_{alg}$

## General Remarks and Conclusion

1. It is seen that for 0 viscosity parameter and  $\alpha = 1$  the rate independent model is recovered from the rate dependent model
2. For a very large time length (equivalent to a strain rate close to 0) and  $\alpha = 1$  the rate independent model is recovered from the rate dependent model irrespective of the value of the viscosity parameter.
3. For the viscosity parameter = 0, the strain rate has no effect on the variation of stress with respect to strain.
4. In the Elastic domain, both the rate independent and the rate dependent models behave in the same manner, which was expected at the beginning of the study.



# Appendix

This section contains the modified MATLAB codes used in the report.

## Damage\_main\_visc.m

This function implements the damage surface under loading and returns the stress and other variables to plot. A common function has been implemented for both the inviscid and viscous model.

```
function
[sigma_v,vartoplot,LABELPLOT,TIMEVECTOR,c11_t,c11_ta]=damage_main_visc
(Eprop,ntype,istep, strain,MDtype,n,TimeTotal)
global hplotSURF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CONTINUUM DAMAGE MODEL
% -----
% Given the almanshi strain evolution ("strain(totalstep,mstrain)") and
a set of
% parameters and properties, it returns the evolution of the cauchy
stress and other variables
% that are listed below.
%
% INPUTS <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
% -----
% Eprop(1) = Young's modulus (E)
% Eprop(2) = Poisson's coefficient (nu)
% Eprop(3) = Hardening(+)/Softening(-) modulus (H)
% Eprop(4) = Yield stress (sigma_y)
% Eprop(5) = Type of Hardening/Softening law (hard_type)
%           0 --> LINEAR
%           1 --> Exponential
% Eprop(6) = Rate behavior (viscpr)
%           0 --> Rate-independent (inviscid)
%           1 --> Rate-dependent (viscous)
%
% Eprop(7) = Viscosity coefficient (eta) (dummy if inviscid)
% Eprop(8) = ALPHA coefficient (for time integration), (ALPHA)
%           0<=ALPHA<=1 , ALPHA = 1.0 --> Implicit
%           ALPHA = 0.0 --> Explicit
%           (dummy if inviscid)
%
% ntype = PROBLEM TYPE
%       1 : plane stress
%       2 : plane strain
%       3 : 3D
%
% istep = steps for each load state (istep1,istep2,istep3)
%
% strain(i,j) = j-th component of the linearized strain vector at the
i-th
%              step, i = 1:totalstep+1
%
% MDtype = Damage surface criterion %
%       1 : SYMMETRIC
%       2 : ONLY-TENSION
%       3 : NON-SYMMETRIC
%
%
```



```

% -----
sigma_v = cell(totalstep+1,1) ;
TIMEVECTOR = zeros(totalstep+1,1) ;
delta_t = TimeTotal./istep/length(istep) ;
c11_t = zeros(totalstep+1,1);           %c11 component of the
tangent constitutive operator
c11_ta = zeros(totalstep+1,1);         %c11 of the algorithmic
constitutive operative
% Elastic constitutive tensor
% -----
[ce] = tensor_elastic01 (Eprop, ntype);
% Initz.
% -----
% Strain vector
% -----
eps_n1 = zeros(mstrain,1);
% Historic variables
% hvar_n(1:4) --> empty
% hvar_n(5) = q --> Hardening variable
% hvar_n(6) = r --> Internal variable
hvar_n = zeros(mhist,1) ;

% INITIALIZING (i = 1) !!!!
% *****i*
i = 1 ;
r0 = sigma_u/sqrt(E);
hvar_n(5) = r0; % r_n
hvar_n(6) = r0; % q_n
eps_n1 = strain(i,:);
sigma_n1 = ce*eps_n1'; % Elastic
sigma_v{i} = [sigma_n1(1) sigma_n1(3) 0;sigma_n1(3) sigma_n1(2) 0 ; 0
0 sigma_n1(4)];
c11_t = ce(1,1); %the c11 component of the tangent constitutive
operator
c11_ta = c11_t; %the c11 component of the tangent algorithmic
operator

nplot = 3 ;
vartoplot = cell(1,totalstep+1) ;
vartoplot{i}(1) = hvar_n(6) ; % Hardening variable (q)
vartoplot{i}(2) = hvar_n(5) ; % Internal variable (r)
vartoplot{i}(3) = 1-hvar_n(6)/hvar_n(5) ; % Damage variable (d)

for iload = 1:length(istep)
    % Load states
    for iloc = 1:istep(iload)
        i = i + 1 ;
        TIMEVECTOR(i) = TIMEVECTOR(i-1)+ delta_t(iload) ;
        % Total strain at step "i"
        % -----
        eps_n1 = strain(i,:);
        eps_np = strain(i-1,:); %the strain vector for the previous
time step

%*****
%*****
%*          DAMAGE MODEL
%
%*****
%*****
%

```

```

    [sigma_n1,hvar_n,aux_var,c11_tan,c11_tan_alg] =
rmap_dano1_visc(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_np,delta_t);
    % PLOTTING DAMAGE SURFACE
    if(aux_var(1)>0)
        hplotSURF(i) = dibujar_criterio_dano1(ce, nu, hvar_n(6),
'r:',MDtype,n );
        %set(hplotSURF(i), 'Color',[0 0 1], 'LineWidth',1);
        set(hplotSURF(i), 'Color','blue', 'LineWidth',1);
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%*****
*

    % GLOBAL VARIABLES
    % *****
    % Stress
    % -----
    m_sigma=[sigma_n1(1)  sigma_n1(3)  0;sigma_n1(3)  sigma_n1(2)  0
; 0 0  sigma_n1(4)];
    sigma_v{i} = m_sigma ;
    c11_t(i) = c11_tan;
    c11_ta(i) = c11_tan_alg;

    % VARIABLES TO PLOT (set label on cell array LABELPLOT)
    % -----
    vartoplot{i}(1) = hvar_n(6) ; % Hardening variable (q)
    vartoplot{i}(2) = hvar_n(5) ; % Internal variable (r)
    vartoplot{i}(3) = 1-hvar_n(6)/hvar_n(5) ; % Damage variable
(d)
end
end

```

## rmap\_dano1\_visc.m

This function implements the integration algorithm for both the inviscid and viscous model.

```

function [sigma_n1,hvar_n1,aux_var,c11_tan,c11_tan_alg] =
rmap_dano1_visc (eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_np,delta_t)

%*****
%*****
%*
%*
%*          *
%*          Integration Algorithm for a isotropic damage model
%*
%*
%*
%*          [sigma_n1,hvar_n1,aux_var] = rmap_dano1
(eps_n1,hvar_n,Eprop,ce)          *
%*
%*
%*
%* INPUTS          eps_n1(4)    strain (almansi)    step n+1
%*
%*          vector R4    (exx eyy exy ezz)
%*
%*          hvar_n(6)    internal variables , step n
%*
%*

```

```

%*                               hvar_n(1:4) (empty)
*
%*                               hvar_n(5) = r ; hvar_n(6)=q
*
%*                               Eprop(:)   Material parameters
*
%*                               ce(4,4)   Constitutive elastic tensor
*
%*
%* OUTPUTS:                       sigma_n1(4) Cauchy stress , step n+1
*
%*                               hvar_n(6)   Internal variables , step n+1
*
%*                               aux_var(3)  Auxiliar variables for computing
const. tangent tensor *
%*****
*****

hvar_n1 = hvar_n;
r_n     = hvar_n(5);
q_n     = hvar_n(6);
E       = Eprop(1);
nu      = Eprop(2);
H       = Eprop(3);
sigma_u = Eprop(4);
hard_type = Eprop(5) ;
visc    = Eprop(6);
eta     = Eprop(7);
alpha   = Eprop(8);
%*****
*****

%*****
*****
%*           initializing
%*
r0 = sigma_u/sqrt(E);
zero_q=1.d-6*r0;
% if(r_n<=0.d0)
%   r_n=r0;
%   q_n=r0;
% end
%*****
*****

%*****
*****
%*           Damage surface
%*
[rtrial,rtrialp] = Modelos_de_dano1_visc (MDtype,ce,eps_n1,n,eps_np);

rtr = (1-alpha)*rtrialp + alpha*rtrial; %strain norm at n+alpha
%*****
*****

```

```

%*****
%*****
%*   Ver el Estado de Carga
%*
%*   ----->   fload=0 : elastic unload
%*
%*   ----->   fload=1 : damage (compute algorithmic constitutive
tensor)          %*
fload=0;

if (visc == 1) %implementing the viscous case

    if(rtr > r_n)
        %*   Loading

        fload=1;
        r_n1 = ((eta-delta_t*(1-alpha))/(eta + alpha*delta_t))*r_n
+ (delta_t/(eta+alpha*delta_t))*rtr;
        delta_r=r_n1-r_n;

        if hard_type == 0
            % Linear
            q_n1= q_n+ H*delta_r;
        else
            %exponential
            if H>0
                q_inf = r0+(r0-zero_q); %hardening
            else
                q_inf = zero_q; %softening
            end
            A = abs(H);
            q_n1= q_inf-(q_inf-q_n)*exp(A*(1-r_n1/r_n));

        end

        if(q_n1<zero_q)
            q_n1=zero_q;
        end

    else

        %*   Elastic load/unload
        fload=0;
        r_n1= r_n ;
        q_n1= q_n ;

    end

else %implementing the inviscid model

    if(rtrial > r_n)
        %*   Loading

        fload=1;
        delta_r=rtrial-r_n;
        r_n1= rtrial ;
        if hard_type == 0

```

```

    % Linear
    q_n1= q_n+ H*delta_r;
else
    %exponential
    if H>0
        q_inf = r0+(r0-zero_q); %hardening
    else
        q_inf = zero_q; %softening
    end
    A = abs(H);
    q_n1= q_inf-(q_inf-q_n)*exp(A*(1-r_n1/r_n));

end

if(q_n1<zero_q)
    q_n1=zero_q;
end

else

    %*      Elastic load/unload
    fload=0;
    r_n1= r_n ;
    q_n1= q_n ;

end

end

% Damage variable
% -----
dano_n1 = 1.d0-(q_n1/r_n1);
% Computing stress
% *****
sigma_n1 =(1.d0-dano_n1)*ce*eps_n1';
%hold on
%plot(sigma_n1(1),sigma_n1(2), 'bx')

%*****
%*****

%*****
%*****

%* Updating historic variables
%*
% hvar_n1(1:4) = eps_n1p;
hvar_n1(5)= r_n1 ;
hvar_n1(6)= q_n1 ;
%*****
%*****

%*****
%*****

```

```

%* Auxiliar variables
%*
aux_var(1) = fload;
aux_var(2) = q_n1/r_n1;
aux_var(3) = (q_n1-H*r_n1)/r_n1^3;

if fload ==0
    C_tan_alg = (1-dano_n1)*ce; % C_tan_alg is the algorithmic
constitutive operator
else
    if visc == 0
        C_tan_alg = (1-dano_n1)*ce - aux_var(3)*(sigma_n1*sigma_n1');
    else
        C_tan_alg = (1-dano_n1)*ce +
((alpha*delta_t)/(eta+alpha*delta_t))*(1/rtrial)*((H*r_n1-
q_n1)/r_n1^2)*(sigma_n1*sigma_n1');
    end
end
c11_tan_alg = C_tan_alg(1,1);
C_tan = (1-dano_n1)*ce; % C_tan is the tangent constitutive
operator
c11_tan = C_tan(1,1);
end
%*****
%*****

```

## dibujar\_criterio\_dano1.m

This function gives the damage surface for all the MDtypes

```

function hplot = dibujar_criterio_dano1(ce,nu,q,tipo_linea,MDtype,n)
%*****
%*****
%*
%* PLOT DAMAGE SURFACE CRITERIUM: ISOTROPIC MODEL
%*
%*
%*
%* function [ce] = tensor_elastico (Eprop, ntype)
%*
%*
%* INPUTS
%*
%*
%* Eprop(4) vector de propiedades de material
%*
%* Eprop(1)= E----->modulo de
Young %*
%* Eprop(2)= nu----->modulo de
Poisson %*
%* Eprop(3)= H----->modulo de
Softening/hard. %*
%* Eprop(4)=sigma_u-----
>tensii; %n i; %ltima %*
%* ntype %*
%* ntype=1 plane stress
%*
%* ntype=2 plane strain
%*

```



```

%*                               ntype=3  3D
%*
%*                               ce(4,4)   Constitutive elastic tensor  (PLANE
S.          )   %*
%*                               ce(6,6)   ( 3D)
%*
%*****
%*****

%*****
%*****
%*          Inverse ce   %*

ce_inv=inv(ce);
c11=ce_inv(1,1);
c22=ce_inv(2,2);
c12=ce_inv(1,2);
c21=c12;
c14=ce_inv(1,4);
c24=ce_inv(2,4);
%*****
%*****

%*****
%*****
% POLAR COORDINATES
if MDtype==1
    tetha=[0:0.01:2*pi];

%*****
%*****
%* RADIUS
D=size(tetha);           %* Range
m1=cos(tetha);          %*
m2=sin(tetha);          %*
Contador=D(1,2);        %*

radio = zeros(1,Contador) ;
s1     = zeros(1,Contador) ;
s2     = zeros(1,Contador) ;

for i=1:Contador
    radio(i)= q/sqrt([m1(i) m2(i) 0
nu*(m1(i)+m2(i))] *ce_inv*[m1(i) m2(i) 0 ...
nu*(m1(i)+m2(i))] ');

    s1(i)=radio(i)*m1(i);
    s2(i)=radio(i)*m2(i);

```

```

end
hplot =plot(s1,s2,tipo_linea);

elseif MDtype==2
tetha=[0:0.01:2*pi];

%*****
%*****
%* RADIUS
D=size(tetha);           %* Range
m1=cos(tetha);          %*
m2=sin(tetha);          %*
Contador=D(1,2);        %*

radio = zeros(1,Contador) ;
s1     = zeros(1,Contador) ;
s2     = zeros(1,Contador) ;
thet_part = zeros(Contador,4) ;
for i=1:Contador
    thet_part(i,:) = [m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
    radio(i)=
q/sqrt((thet_part(i,:).* (thet_part(i,:)>0))*ce_inv*[m1(i) m2(i) 0 ...
nu*(m1(i)+m2(i))]');

    s1(i)=radio(i)*m1(i);
    s2(i)=radio(i)*m2(i);

end
hplot =plot(s1,s2,tipo_linea);

elseif MDtype==3

tetha=[0:0.01:2*pi];

%*****
%*****
%* RADIUS
D=size(tetha);           %* Range
m1=cos(tetha);          %*
m2=sin(tetha);          %*
Contador=D(1,2);        %*

radio = zeros(1,Contador) ;
s1     = zeros(1,Contador) ;
s2     = zeros(1,Contador) ;
ratio = zeros(1,Contador) ;
for i=1:Contador
    ratio(i) =
(m1(i)*(m1(i)>0)+m2(i)*(m2(i)>0))/(abs(m1(i))+abs(m2(i)));
    %ratio(i) =
(m1(i)*(m1(i)>0)+m2(i)*(m2(i)>0))/(abs(m1(i))+abs(m2(i)));

    radio(i)= q/(sqrt([m1(i) m2(i) 0
nu*(m1(i)+m2(i))] *ce_inv*[m1(i) m2(i) 0 ...

```

```

nu*(m1(i)+m2(i)]')*(ratio(i)+(1-ratio(i))/n));

s1(i)=radio(i)*m1(i);
s2(i)=radio(i)*m2(i);

end
hplot =plot(s1,s2,tipo_linea);

end
%*****
%*****

%*****
%*****

return

```

## modelos\_de\_dano1\_visc.m

This function calculates the strain norm for all the three MDtypes.

```

function [rtrial,rtrialp] = Modelos_de_dano1_visc
(MDtype,ce,eps_n1,n,eps_np)
%*****
%*****
%*           Defining damage criterion surface
%*
%*
%*
%*           MDtype= 1           : SYMMETRIC
%*
%*           MDtype= 2           : ONLY TENSION
%*
%*           MDtype= 3           : NON-SYMMETRIC
%*
%*
%*
%*
%* OUTPUT:
%*
%*           rtrial
%*
%*****
%*****

%*****
%*****
if (MDtype==1)      %* Symmetric
rtrial= sqrt(eps_n1*ce*eps_n1');
rtrialp= sqrt(eps_np*ce*eps_np');    %strain norm for previous time
step
elseif (MDtype==2) %* Only tension
rtrial = sqrt(eps_n1.*(eps_n1>0)*ce*eps_n1');

```

```
rtrialp =0;
elseif (MDtype==3)  %*Non-symmetric
    sig_n1 = eps_n1*ce;
    ratio =
    (sig_n1(1)*(sig_n1(1)>0)+sig_n1(2)*(sig_n1(2)>0))/(abs(sig_n1(1))+abs(
    sig_n1(2)));
    % rtrial= sqrt(eps_n1*ce*eps_n1');
    rtrial = (ratio+((1-ratio)/n))*sqrt(eps_n1*ce*eps_n1');
    rtrialp =0;

end
%*****
%*****
return
```