Computational Solid Mechanics

Rate Independent & Rate Dependent Damage Model

Assignment No. 1

Adnan Ali Imran

MS-Computational Mechanics

Part 1 (Rate Independent Model)

a) Integration Algorithm

Two different types of rate independent damage models will be discussed here to some depth while the 3rd damage model (symmetric tension/ compression damage model) has already been implemented.

1. *Non Symmetric tension/ compression Damage Model*

Algorithm for Non-symmetric case has been implemented and damage surface is retried successfully while the corresponding MatLab code can be seen in Appendix A. The positive side of the stress space (tension) covers exactly the same space as it is being covered by symmetric case. While the negative stress space (compression) is for more away than the space covered in symmetric casein the same region. The amplification factor of 1/n is being multiply with the norm of the stresses (taking n = 3 in this case). Figure-1 shows the non-symmetric damage surface.

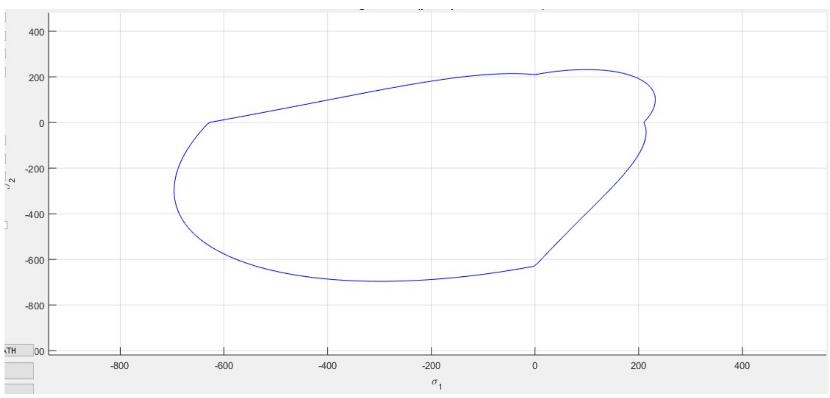


Figure-1 Non-Symmetric tension/ compression Damage Surface

2. *Tension Only Damage Model*

In order to create surface for Tension Only damage model in stress space, algorithm is implemented and MatLab code is presented in Appendix A.

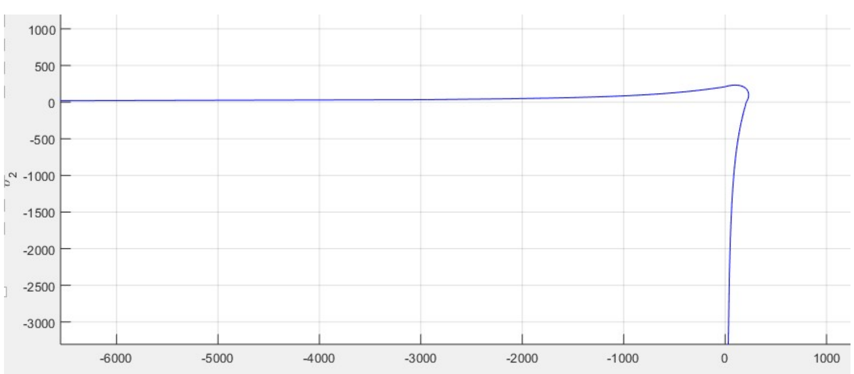


Figure-2 Tension only Damage Surface

It can be seen in figure-2 that the positive part of the stress space (Tension) have the same ellipsoidal surface as the symmetric damage model have. But in the negative region of the stress space (compression) the space is open to infinity. This is due to fact that Macaulay bracket present in the stress formula always extract the positive part so, in case of negative domain (compression), the norm of the stress returns zero value and hence r_0 is always negative and it never approaches zero. In other words, one point experiences 3 axial compression will arrive to elastic limit.

b) Hardening/ Softening

1. Linear Law

Using data: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = 0.6**

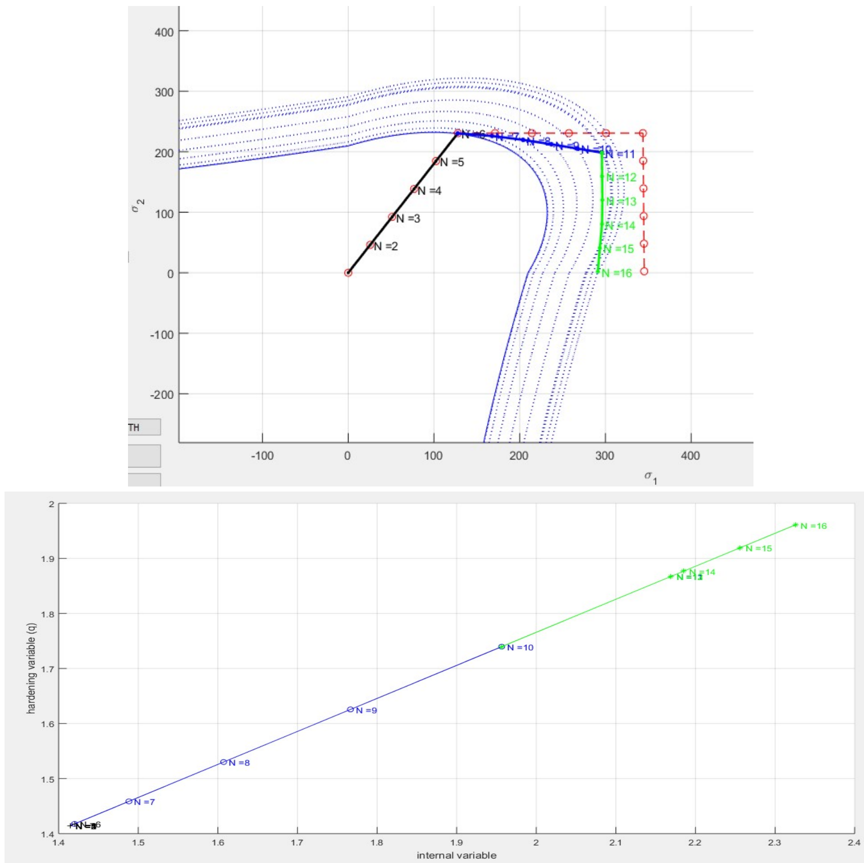


Figure-3 Linear hardening of Tension Only damage model

Using data: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = -0.6**

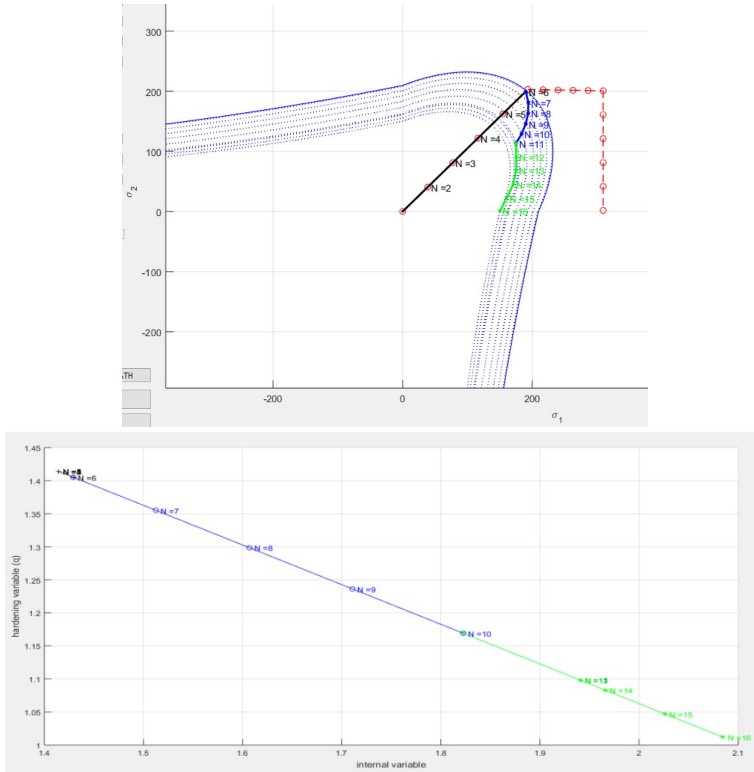


Figure-4 Linear softening of Tension Only damage model

Using data: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = 0.3**

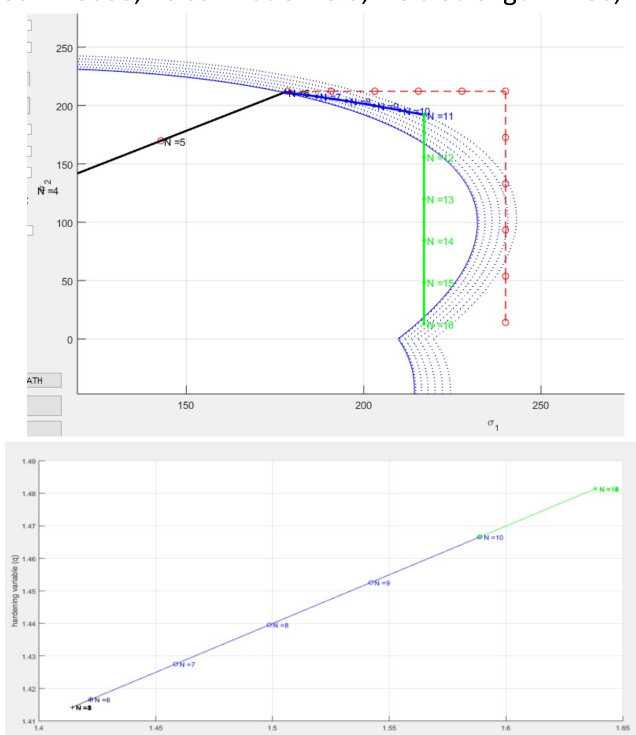


Figure-5 Linear hardening of Non-Symmetric damage model

Using data: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = -0.3**

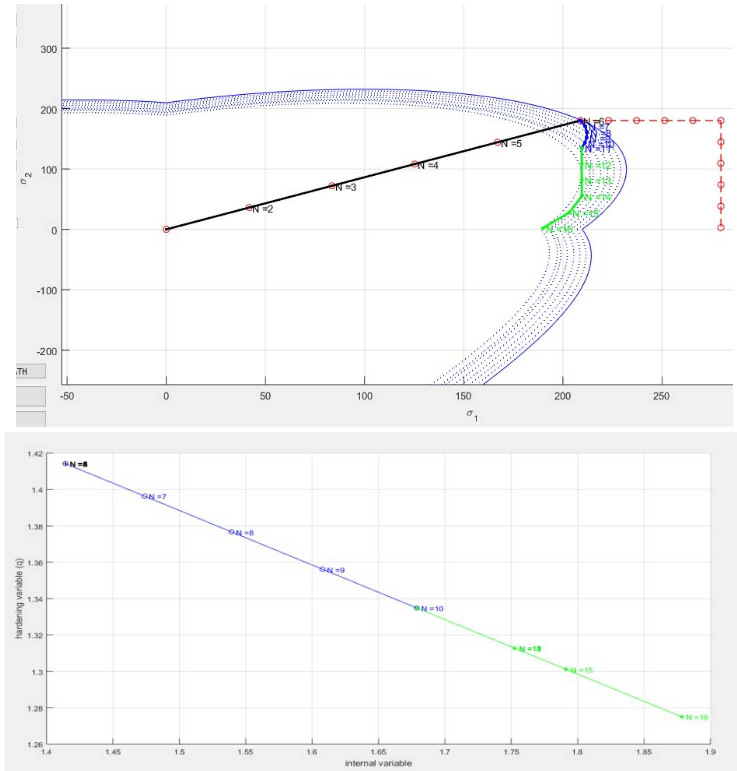
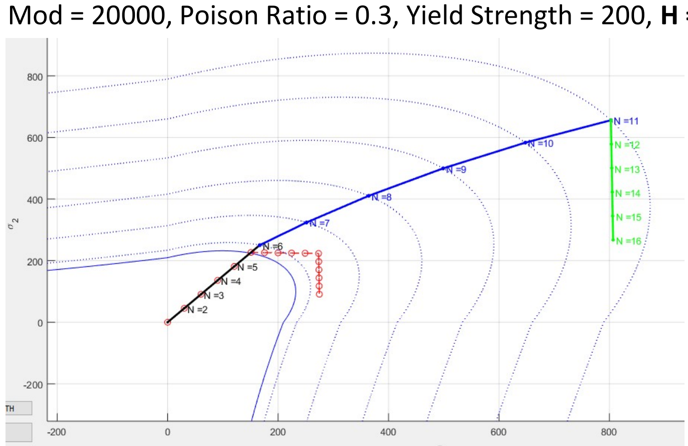


Figure-6 Linear softening of Non-Symmetric damage model

Figures 3 to 6 show the results of linear hardening and softening for both Tension Only and Non Symmetric damage models. It is clearly evident that within elastic domain there is no evolution of internal variable (r). Once the inelastic loading (damage surface) starts, the elastic domain shrinks or expands depending upon the value of H (H<0 is softening & H>0 is hardening) and internal variable (r) varies with linearly with respect to hardening variable (q).

2. Exponential Law

Parameters: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = 0.5, $q_{+\infty}$ =10*r_0**



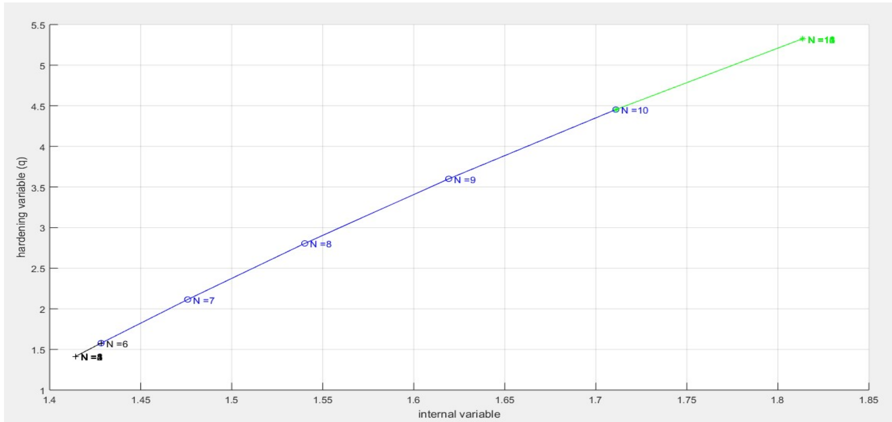
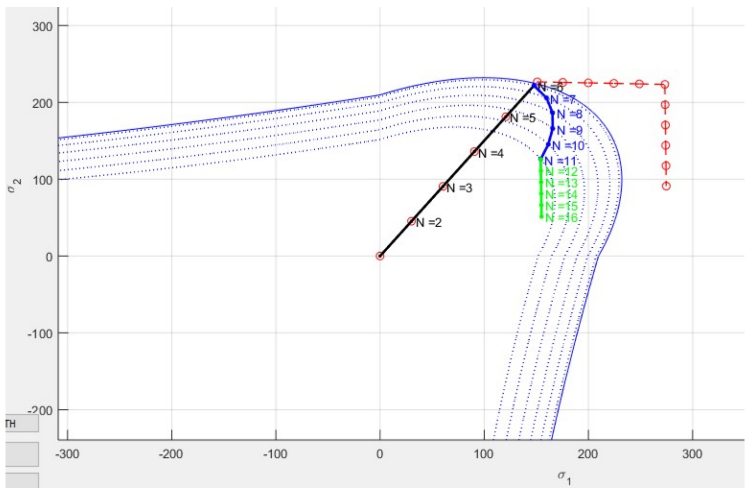


Figure-7 Exponential Hardening of Tension Only damage model

Parameters: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = -0.5, $q_{-\infty}$ =0.1*r_0**



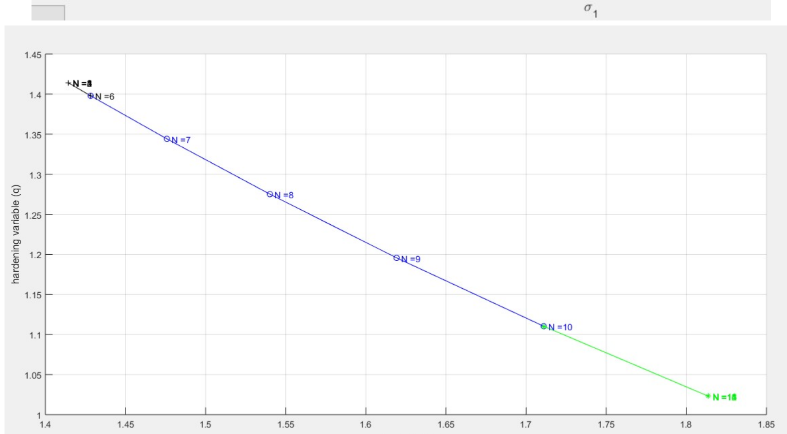


Figure-8 Exponential Softening of Tension Only damage model

Parameters: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = 0.5, $q_{+\infty}$ =5*r_0**

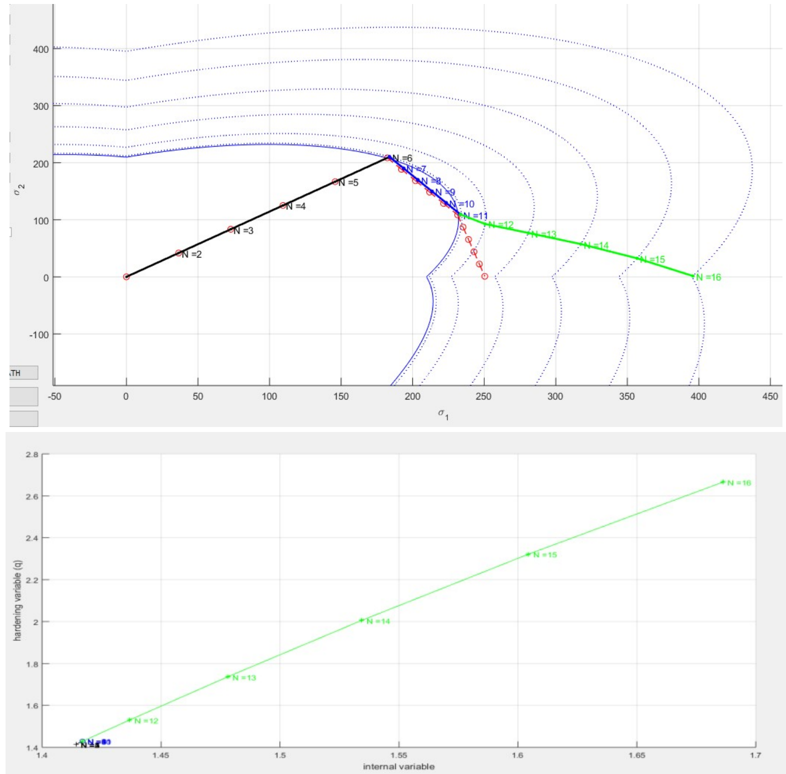


Figure-9 Exponential Hardening of Non-Symmetric damage model

Parameters: Young's Mod = 20000, Poison Ratio = 0.3, Yield Strength = 200, **H = 0.9, $q_{-\infty}$ =0.1*r0**

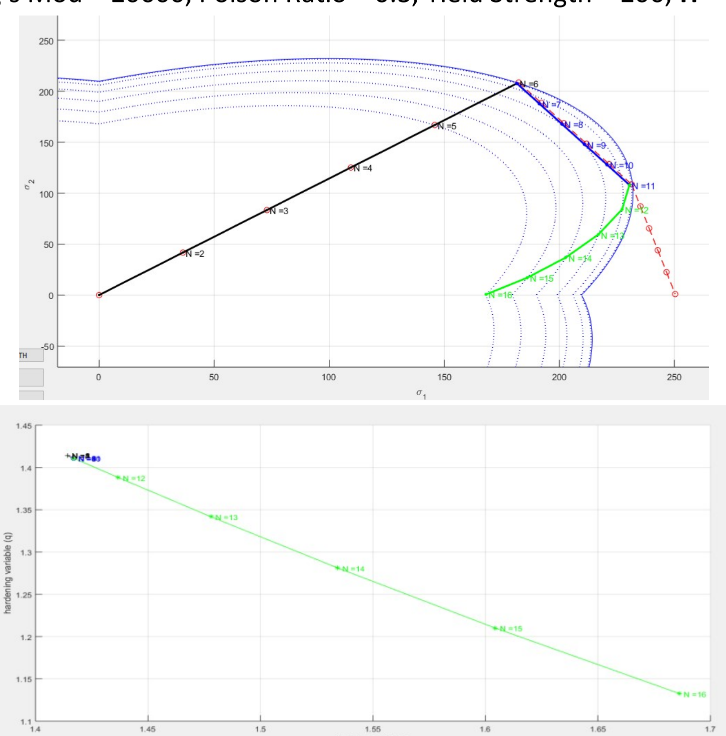


Figure-10 Exponential Softening of Non-Symmetric damage model

Figures 7 to 10 show the results of exponential hardening and softening for both Tension Only and Non Symmetric damage models. The graphs show that when point start to escape the damage surface the elastic domain expands or shrinks according to H value and internal variable have an exponential variations with respect to hardening variable. And the expansion of elastic domain depends upon the value of **q+∞,** more the value greater the expansion of elastic domain.

c) Effective Stress Increment

Parameters: **H = 0.2, α = 380, β = -180, γ = 200**, Linear Hardening/ Softening law and using default values for other parameters all three cases

1. *Case-1*

The simulation curves for the loop start with Uniaxial tension loading to uniaxial tension unloading/ compressive loading to uniaxial compressive unloading and then uniaxial tension loading again.

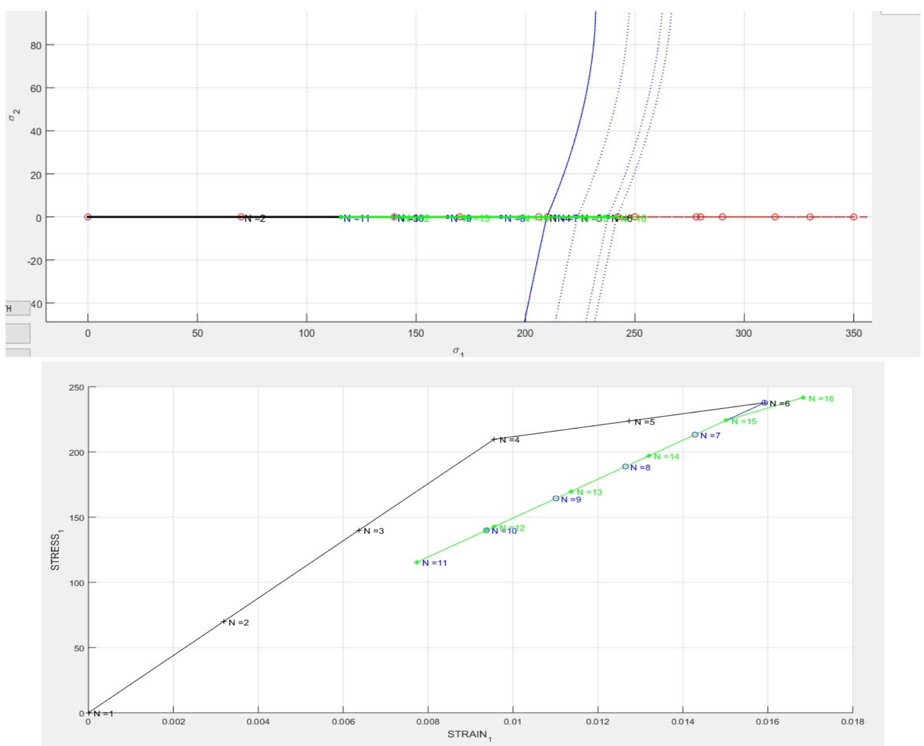


Figure-11 Loading path in Stress Space (SS) and Stress-Strain response (curve) of Case-1 for Tension Only damage model

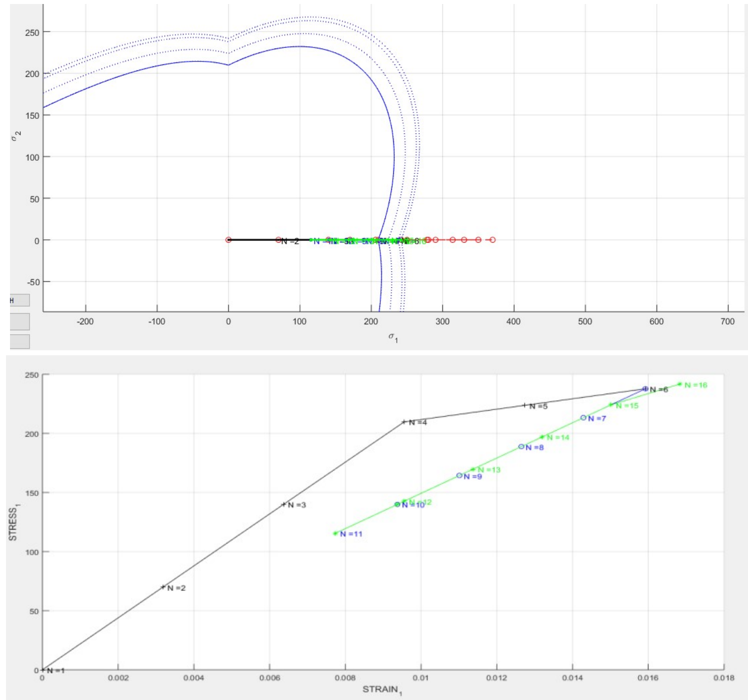


Figure-12 Loading path in SS and Stress-Strain curve Case-1 for Non-Symmetric damage model

2. *Case-2*

The simulation curves for the loop start with Uniaxial tension loading to biaxial tension unloading/ compressive loading to biaxial compressive unloading and then biaxial tension loading again

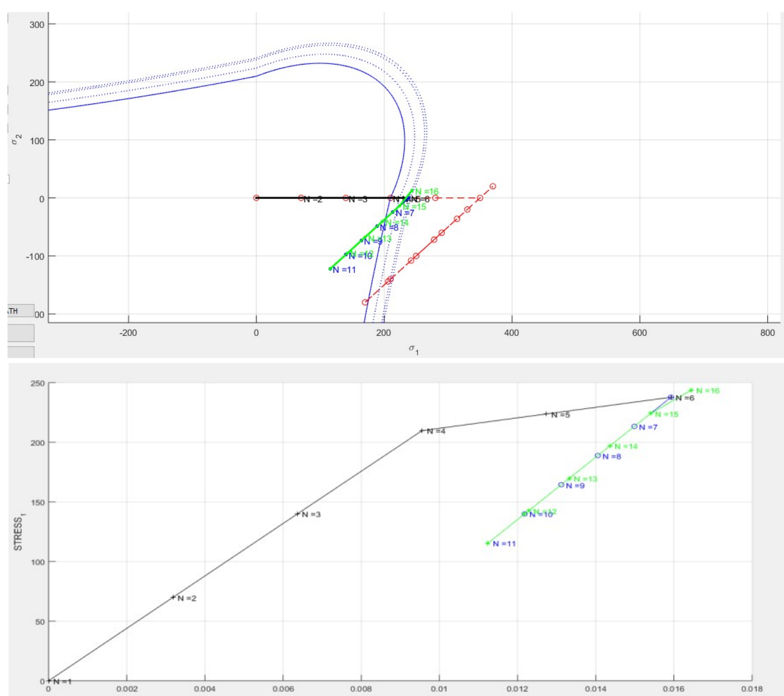


Figure-13 Loading path in SS and Stress-Strain curve of Case-2 for Tension Only damage model

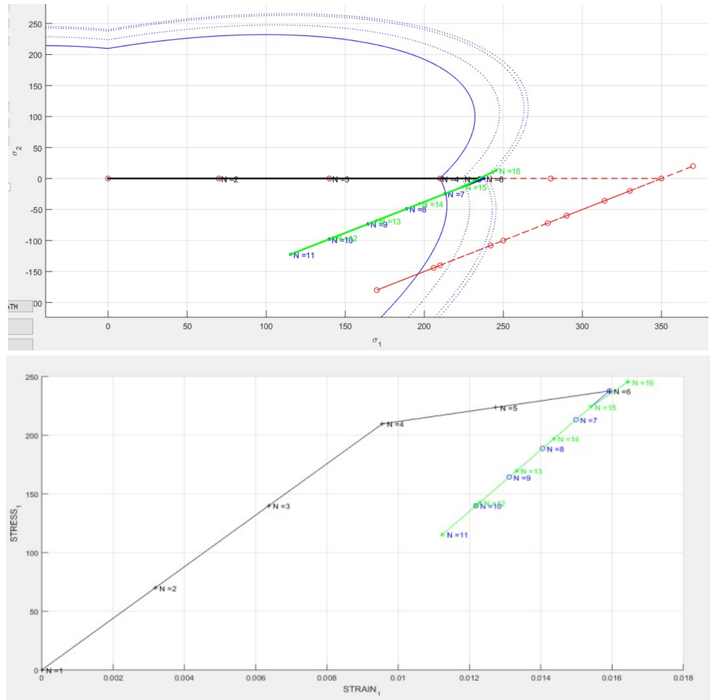


Figure-14 Loading path in SS and Stress-Strain curve of Case-2 for Non-Symmetric damage model

3. *Case-3*

The simulation curves for the loop start with biaxial tension loading to biaxial tension unloading/ compressive loading to biaxial compressive unloading and then biaxial tension loading again

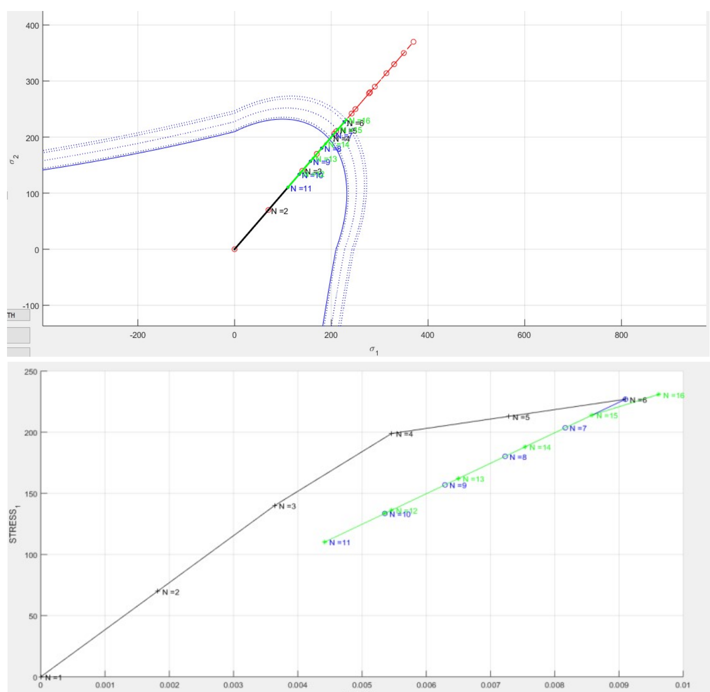


Figure-15 Loading path in SS and Stress-Strain curve of Case-3 for Tension Only damage model

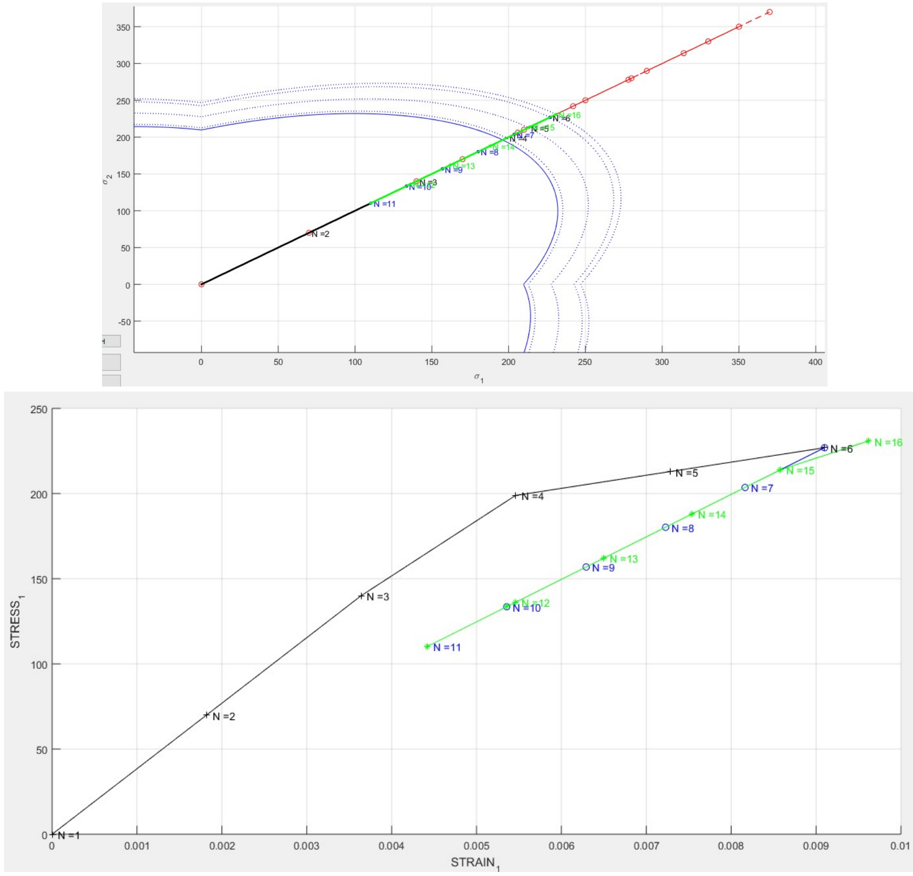


Figure-16 Loading path in Stress Space and Stress-Strain response of Case-3 for Non-Symmetric damage model

Figures 11 to 16 shows response for all 3 cases for both damage models (Tension Only and No-Symmetric).Under tensile loading the response is elastic and when the stress reaches yield point the hardening start and elastic domain expands until maximum stress and then compression start during unloading and curve start to return elastically to the value that is specified. Almost similar behavior can be seen in all three cases. And stress strain curves show the response as per expectations.

Part 2 (Rate dependent Model)

d) Plain strain case for viscous damage model

The algorithm for Rate dependent damage model is implemented and Matlab code is attached in Appendix A. The code is run to check the response of symmetric tension/ compression damage model. And simulation results are presented below.

Parameters: **H = 0.6, η = 0.3**, Linear Hard/ Soft law and using default values for other parameters.

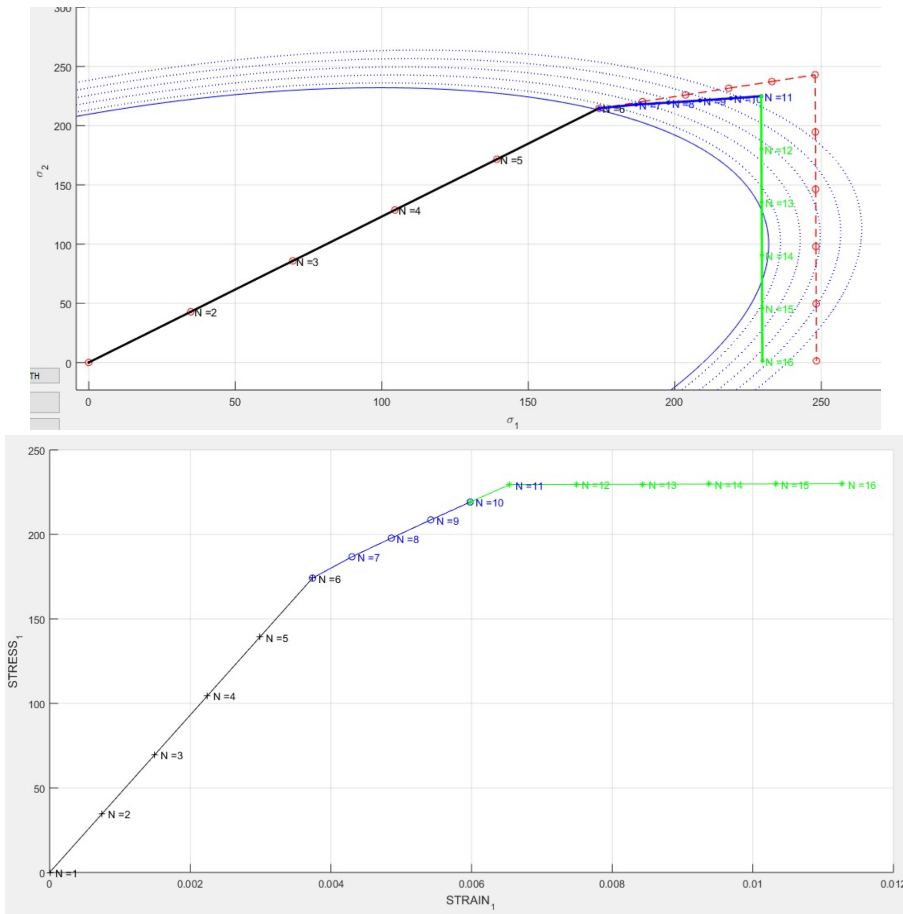


Figure-17 Viscous Symmetric Tension/ Compression damage model

It can be seen that when viscosity comes into play the stress point on damage surface escapes from elastic domain to inelastic domain and after a certain limit return back to expanded elastic domain. The point shows such trend due to presence of viscosity.

e) Correctness of Code with Linear Hardening/ Softening Law.

1. Case-1

Considering Linear Hard/ Soft law with **H =0.3** and keeping other parameters at default value. Variations in stress-strain curves are observed at different values of viscosity coefficient and results are presented below.

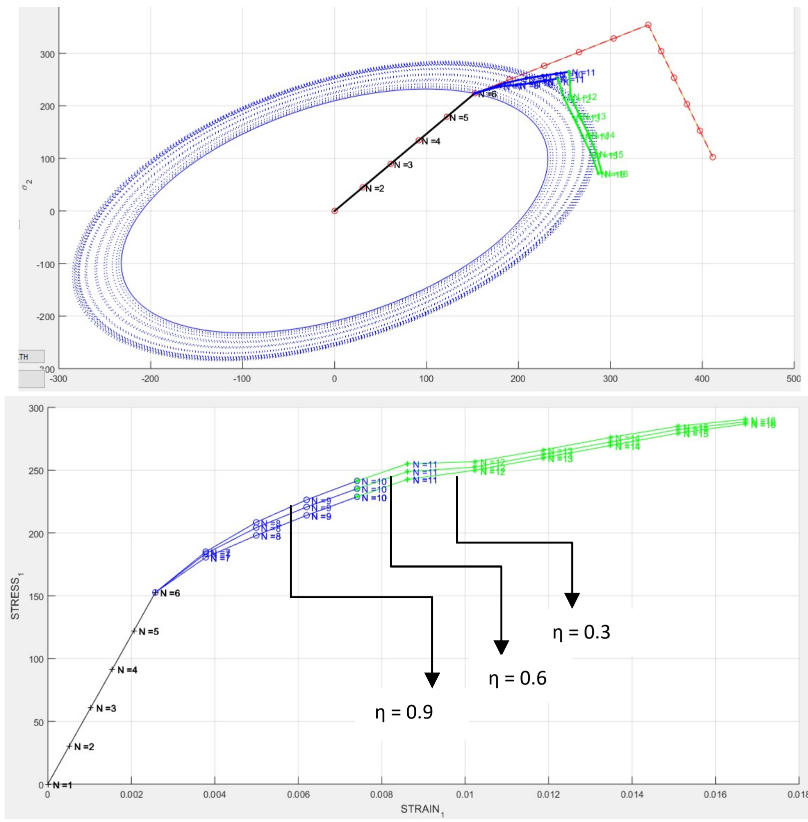


Figure-18 Stress-Strain curves at different values of viscosity

It can be seen that higher the viscosity value, higher the stress value. In elastic domain viscosity does not play any role while in inelastic domain for same value of strain, stress have different values against different viscosity.

2. Case-2

Considering Linear Hard/ Soft law with **H = 0.3, η = 0.3, α = 1.0** and keeping other parameters at default value. Variations in stress-strain curves are observed at different values total time because varying total time varies strain rate έ and results are presented in figure 19.
It can be seen that decreasing the total time, increases strain rate which increases the corresponding stress. Another thing is that in elastics domain the change of strain rate does not affect stress values while in inelastic domain higher the strain rate, higher the stresses are.
The stress strain curves are plotted at total time of **3, 7 & 10 seconds**.

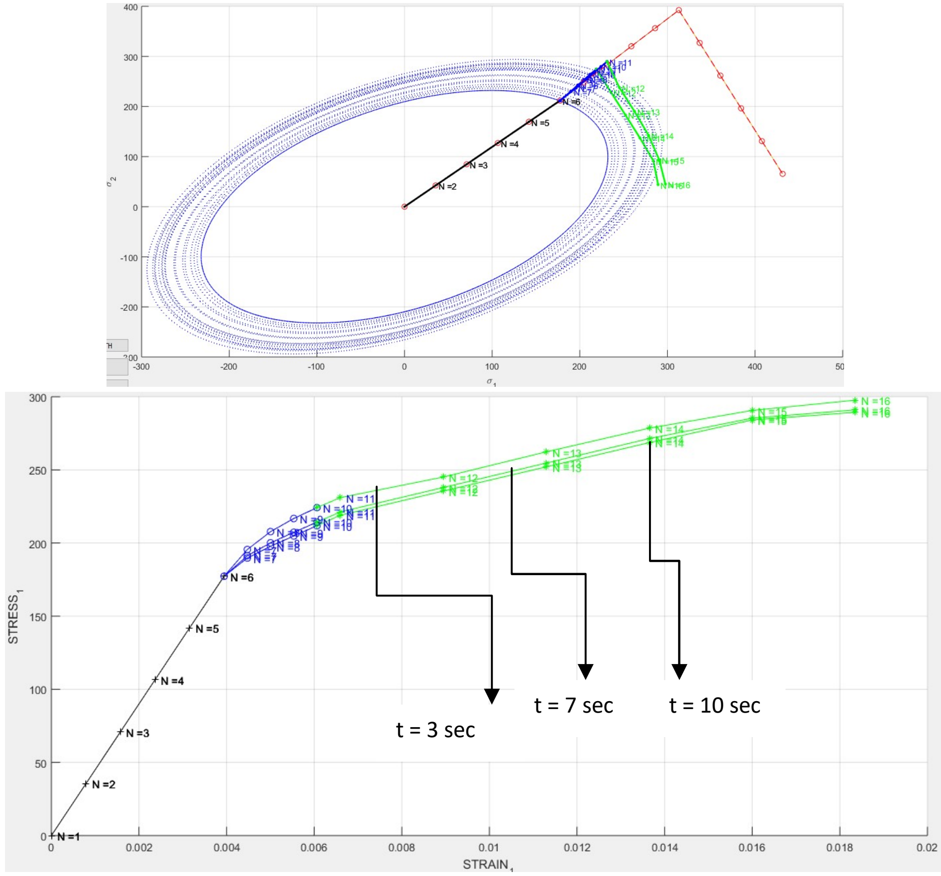
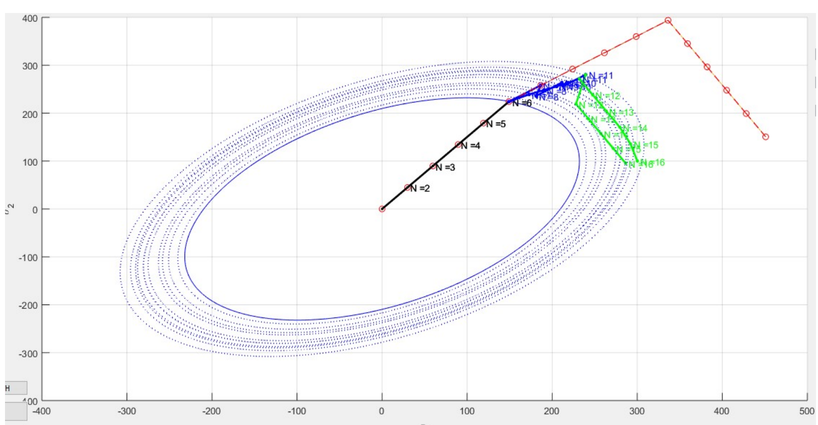


Figure-19 Stress-Strain curves at different values of total time (Strain rate)

3. Case-3

For Alpha time integration method, considering Linear Hard/ Soft law with **H = 0.3, η = 0.3**and keeping other parameters at default value. Variations in stress-strain curves are observed at different values of Alpha α and results are presented below.



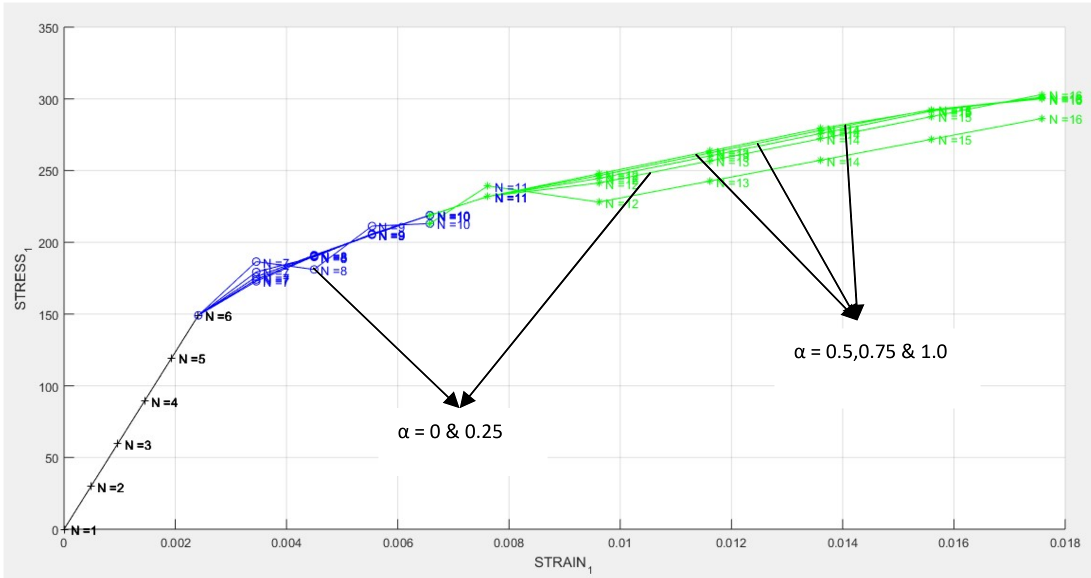


Figure-20 Stress-Strain curves at different values of Alpha

It can be seen that at α < 0.5 (at 0 & 0.25), the solution oscillate at nodal points due to fact that its conditionally stable (Explicit Euler scheme). While at 0.5 ≤ α ≤ 1, the stress-strain is stable because at these values of alpha, the computation scheme is un-conditionally stable (α = 0.5 Crank Nicholson & α = 1.0 Implicit Euler).

Appendix A

a) Matlab Code for Tension Only damage model

Modelos_de_dano1.m

```
elseif (MDtype==2)   %* Only tension
rtrial = sqrt(eps_nl.*(eps_nl>0)*ce*eps_nl');
```

dibujar_criterio_dano1.m

```
elseif MDtype==2                          %Tension only damage model
   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))]).*(([m1(i) m2(i) 0 nu*(m1(i)+m2(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);
```

b) Matlab Code for Non-Symmetric damage model

Modelos_de_dano1.m

```
sigmal =eps_nl*ce;
thet = ((sigmal(1).*(sigmal(1)>0))+(sigmal(2).*(sigmal(2)>0)))/(abs(sigmal(1))+abs(sigmal(2)));
rtrial = (thet+(1-thet)/n).*sqrt(eps_nl*ce*eps_nl');
```

dibujar_criterio_dano1.m

```
elseif MDtype==3
  tetha=[0:0.01:2*pi];                    % No symmetric damage model
    %*****************************************************************************************
    %* 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
        thet(i) = ((m1(i).*(m1(i)>0))+(m2(i).*(m2(i)>0)))/(abs(m1(i))+abs(m2(i)));
         radio(i)= q/((thet(i)+(1-thet(i))/n).*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);
```

c) Matlab Code for Exponential Hardening/ Softening Law

rmap_dano1.m

```
  A = abs(H);
if H>0
 q_infP = r0+r0-zero_q;
 q_n1 = q_infP-(q_infP-q_n)*exp(A*(1-r_n1/r_n)) ;
elseif H<0
 q_infN = zero_q;
 q_n1 = q_infN-(q_infN-q_n)*exp(A*(1-r_n1/r_n)) ;
end
```

d) Viscous damage model

rmap_dano1.m

```
Eta     = Eprop(7); %viscus modelling
Alpha = Eprop(8);   %viscus modelling
```

```
%*       Damage surface
[rtrial,rtrial_n] = Modelos_de_dano1 (MDtype,ce,eps_n1,n,eps);
%*************************************************************************

rtrial_alpha= ((1-Alpha)*rtrial_n) + (Alpha*rtrial);  %viscus modelling
```

```
fload=0;

if(rtrial_alpha > r_n)
    %*   Loading
    fload=1;
    r_n1= (((Eta-delta_t.*(1-Alpha))/(Eta+Alpha*delta_t)).*r_n)+((delta_t/(Eta+Alpha*delta_t)).*rtrial_alpha);
    delta_r=r_n1-r_n;
    if hard_type == 0
        %  Linear
        q_n1= q_n+ H*delta_r;
    else
          A = abs(H);
          if H>0
         q_infP = r0+r0-zero_q;
         q_n1 = q_infP-(q_infP-q_n)*exp(A*(1-r_n1/r_n)) ;
          elseif H<0
          q_infN = zero_q;
          q_n1 = q_infN-(q_infN-q_n)*exp(A*(1-r_n1/r_n)) ;
          end
    end
    if(q_n1<zero_q)
        q_n1=zero_q;
    end
```

```
sigma_n1  =(1.d0-dano_n1)*ce*eps_n1';
sigmabar=ce*eps_n1';    %Viscous Modeling
```

Modelos_de_dano1.m

```
if (MDtype==1)       %* Symmetric
rtrial= sqrt(eps_n1*ce*eps_n1');
rtrial_n = sqrt(eps_n*ce*eps_n');
```

damage_main.m

```
%*      DAMAGE MODEL
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[sigma_n1,hvar_n,aux_var] = rmap_dano1(eps_n1,hvar_n,Eprop,ce,MDtype,n,delta_t,eps);
```

```
% if viscpr == 1
%    % Comment/delete lines below once you have implemented this case
%    % *******************************************************
%    menu({'Viscous model has not been implemented yet. '; ...
%         'Modify files "damage_main.m","rmap_dano1" ' ; ...
%         'to include this option'},  ...
%         'STOP');
%    error('OPTION NOT AVAILABLE')
% else
% end
```