Technical University of Catalonia

COMPUTATIONAL SOLID MECHANICS

DAMAGE MODELS

ASSIGNMENT 1

M.Sc. Computational Mechanics – CIMNE

Mohammad Mohsen Zadehkamand

8 April 2017

Table of Contents

1. Introduction 1

2. Code Algorithm and Modifications 2

3. Elastic Domain Limits in Stress Space 3

4. Effect of Loading Path selection 4

❖ *Rate Independent Models [INVISCID]* 6

5. Only-tension / Non-Symmetric tension compression Models 6

6. Effect of Hardening/Softening Variable Type 9

7. Effect of Hardening/Softening Modulus Value (H) 10

❖ *Rate Dependent Models [VISCOUS]* 11

8. Effect of Viscosity Parameter (η) 11

9. Effect of Strain Rate 12

10. Effect of Time Integration Parameter - Alpha (α) 13

❖ *Appendix* I

1. Appendix1: dibujar_criterion_dano1 I

2. Appendix2: damage_main III

3. Appendix3: rmap_dano1 IV

4. Appendix4: modelos-de_dano1 VI

1. Introduction

This is the report for the final project of the course "Computational Solid Mechanics" [Part 1] which deals with *Continuum Damage Constitutive Models*. In this project, using the supplied MATLAB code and doing some modifications and changes, the algorithmic structure underlying the *Numerical Integration of Continuum Damage Constitutive Models* are studied in the local constitutive domain. Actually, in reality what happens is that for the overall structural response, *Standard Finite Element* programs would solve the problem using the geometry, boundary/load conditions and the constitutive models, but in this project the local strain path (at a point) is prescribed by the user as a data.

Finally, the modified Matlab code should be able to execute all conditions detailed in ***chart1***, and as it is obvious there could be so many cases to study, but the correctness of program and also the behavior of continuum damage models, would be elaborated using 3 different loading paths [**chapter 4**].

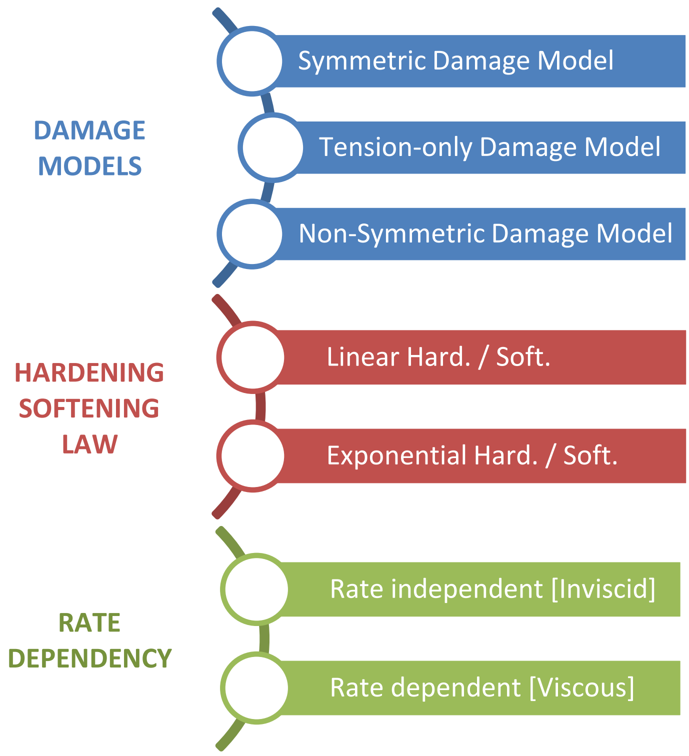


Chart1. Model types in the code.

The material chosen for this study is acting like steel, but with a difference in n parameter. Here n≠1 is chosen in order to be able to study the non-symmetric tension compression method behavior. Material properties are provided in **table1**.

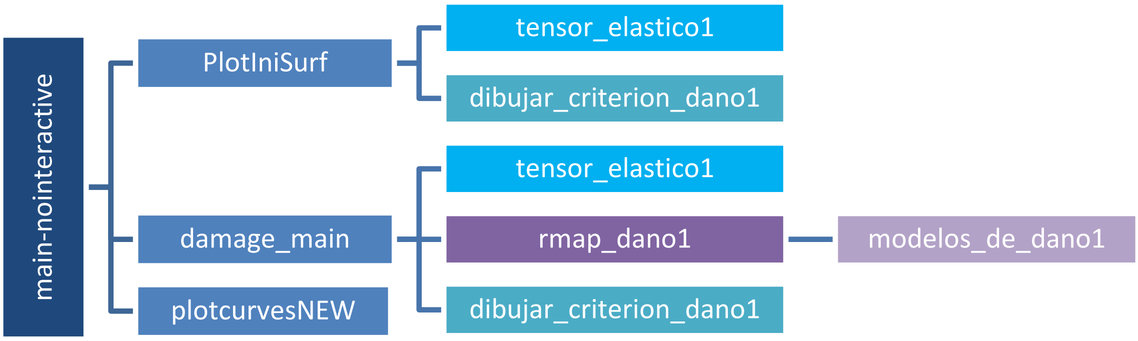
Following in **chapter1** the Code algorithm and the modifications done in the code are briefly explained. Then in **chapter3** we try to capture the desired stress-space domain limits in the program and compare them with theory ones. In **chapter4** a sensitivity study on loading would be conducted and the final loading paths would be selected. Following the **chapters5, 6 and 7** would be on features of model that are not related to time. Finally, in **chapter8, 9 and 10** parameters which are related to time would be studied by some samples.

At the end, the modified or added routines in the Matlab code are also provided in the **appendix**.

	Material Parameters	
E	Young's modulus [MPa]	200000
σ_y	Yield stress [MPa]	200
ϑ	Poisson's coefficient	0.3
n	compression/tension strength	2
H	Hardening/Softening modulus	±0.2

Table1. Definition of Material Properties

2. Code Algorithm and Modifications



❖ Subroutine Brief Explanation:

➢ **main_nointeractive**

[Elemental gauss point level Program for modeling isotropic damage constitutive model]

- **PlotIniSurf**
  [Plots elastic domain boundary limits in stress-domain and also plots the stress path]
  - **Tensor_elastico1**
    [Elastic constitutive tensor calculation]
  - **Dibujar_criterion-dano1**
    [Plots damage surface criterion - non damaged phase]

- **damage_main**
  [Returns the evolution of the Cauchy stress]
  - **tensor_elastico1**
  - **rmap_dano1**
    [Implements the Integration algorithm for isotropic damage model]
    - **modelos_de_dano1**
      [Defines damage criterion surface]
  - **dibujar_criterion_dano1**
    [Plots hardened/softened damage surface]

- **plotcurvesNEW**
  [Plots 2nd figure mainly stress-strain relationship]

❖ Applied Changes and Modifications in Code:

❖ ***main_nointeractive*** changes for plotting options in ***plotcurvesNEW.***
❖ ***plotIniSurf*** changes for plotting options in ***plotpathNI.***
❖ ***dibujar_criterion_dano1*** added damage surface for only-tension and non-symmetric case, in Polar coordinate. (**Appendix1**)
❖ ***damage_main*** importing "step-n strains" and "dt" to ***rmap_dano1.*** (**Appendix2**)
❖ ***rmap_dano1*** adding viscous case and exponential hardening. (**Appendix3**)
❖ ***modelos-de_dano1*** added damage criterium for only-tension and non-symmetric case, adding McAuley Bracket feature to model. (**Appendix4**)

3. *Elastic Domain Limits in Stress Space*

For the first step we will try to capture elastic domain limit boundaries as it is supposed in theory for non-damaged phase. This should be done adding related formulations in polar coordinate routines in the *dibujar_criterion_dano1* function. Code modification for this part is available in (**Appendix 1**). The figures are exactly the same with reference figures in Lecture4 in course slide.

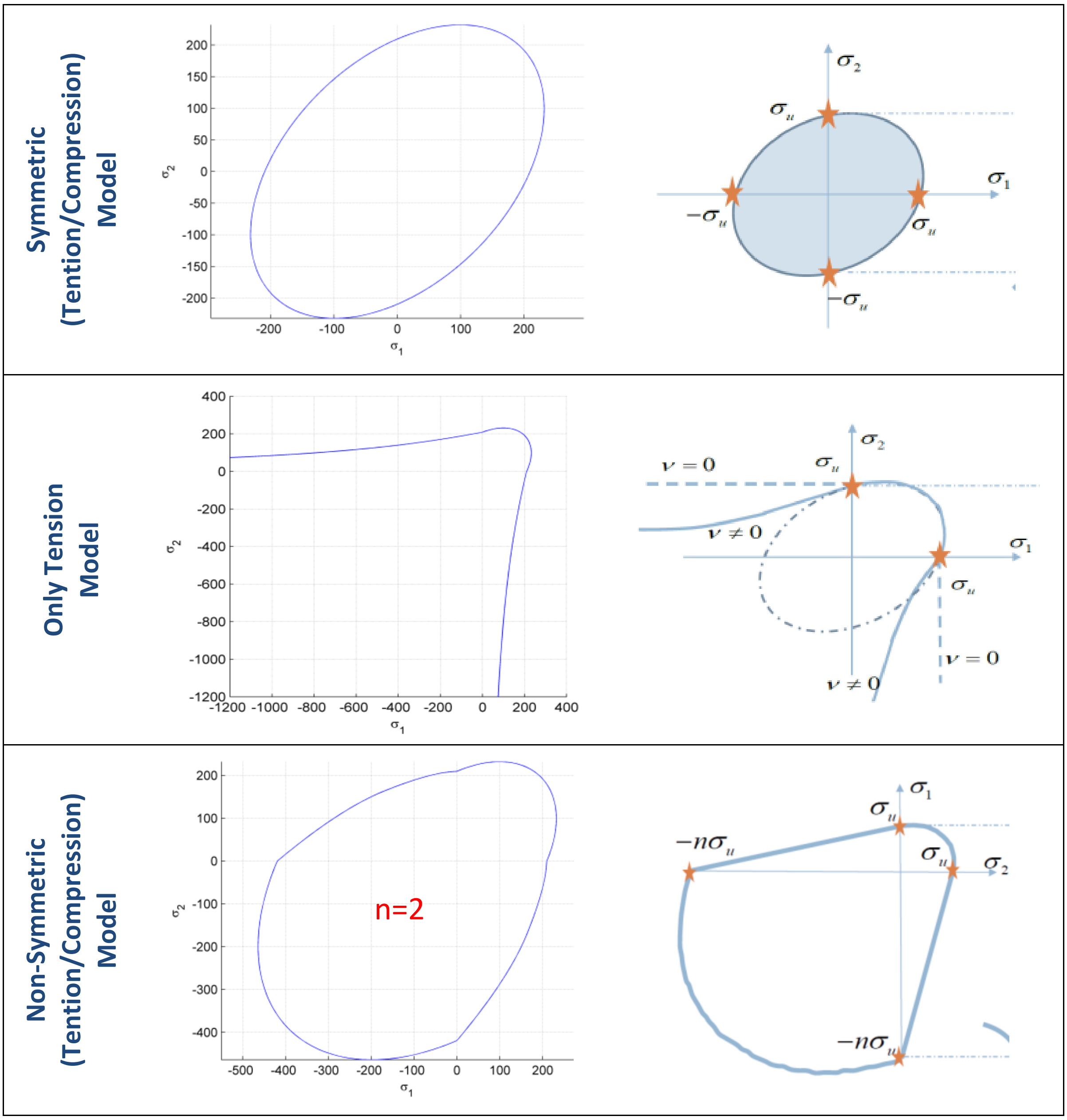


Figure1. damage model elastic limit boundary (non-damaged phase)

As it appears, the curves cuts X and Y axis in $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma}_2$ when nu=0. And we should note here that for damaged phase, the program calculates the new [rtrial] and plots expanded domain in hardening (positive H) or contracted domain in softening (negative H) type.

4. *Effect of Loading Path selection*

For a comprehensive study on the behavior of new added models, three loading paths should be selected, all of which starting from (0,0) in (σ_1, σ_2) domain. We would introduce three-segment paths in the strain-space in terms of their corresponding effective stress increment in ***table2***, but before defining those stress points, some considerations should be implemented.

As an experience in working with this code, it is observed that the selection of 1st segment of path has a crucial effect on the whole loading procedure. Since if in the 1st segment material undergoes lots of damage, it would have different behavior for the same 2nd segment if it was not suffered that much in the 1st segment. So, in order to choose an appropriate path of loading for the rest of report that enables us to fully discuss the behavior of models both in tension and compression, a case sensitivity analysis was done on the 1st segment of loading path.

For this case study, only non-symmetric model is used because we want the material to fail in both tension and compression in order to check its behavior in both extends. Two loading paths are selected; namely, **case-a** in which the first segment starts with +200 MPa and **case-b** with +800 MPa of uniaxial tension at the beginning. The rest of loading is the same for both cases.

case	$\Delta\sigma_1$	$\Delta\sigma_2$
a	+200	0
	-1600	-1600
	+400	+400

case	$\Delta\sigma_1$	$\Delta\sigma_2$
b	+800	0
	-1600	-1600
	+400	+400

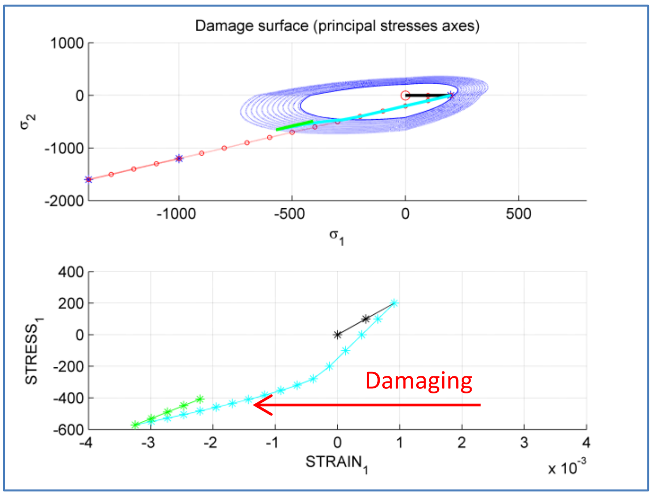


Figure2a. non-sym model - hardening - loading1

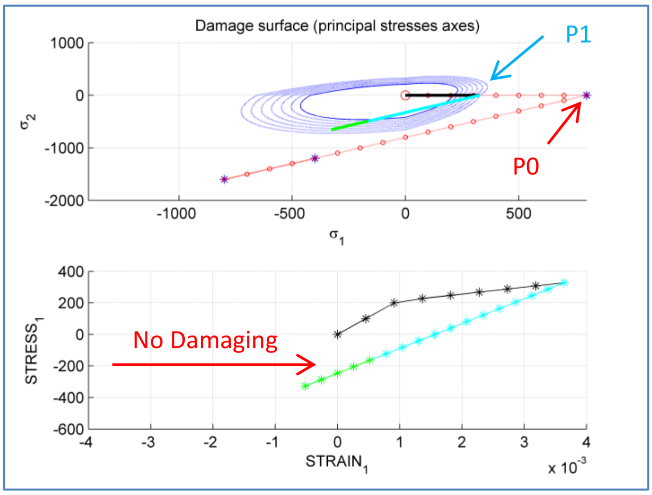


Figure2b. non-sym model - hardening - loading2

As it is clearly visible in **Figure2a** and **2b**, since in **case-b** material goes far beyond the elastic domain, model starts the hardening modification procedure and makes some relative expansion on the elastic limit boundary. In this procedure material suffers a lot from damages and the point is that the continuation of stress imposition from 1st to 2nd segment of load path shall be done from the last point of 1st path, where material can stand exactly on the hardened elastic domain. This point has a stress around +300 MPa (P1 cyan sign) and is far less than previously supposed +800 MPa (P0 red sign). So all this way from 800 to 300 would be expended in unloading phase and no hardening would be added to model. Consequently 2nd segment should be translated to the left and start its way from +300 MPa. This would yield in late hardening (or even no need for extra hardening) in the remaining part of the 2nd segment, because In this loading situation 1st segment is predominant from hardening/ softening point of view.

On the other hand if we choose a 1st segment stress of +200 MPa which is actually equal to yield stress, then material does not damage in 1st segment and compared to previous case there would be a considerable threshold and ability of hardening for the 2nd and even 3rd segments of loading path. This fact is shown in **Figure2a** that the hardening starts in the middle of 2nd segment after just 5 steps (dots on the loading segment lines are representing time steps).

The same behavior is captured for the softening mechanism in **Figure 3a** and **3b**. In this figure also the modified loading paths in stress-strain domain are schematically introduced. These colors would be used in the rest of report also.

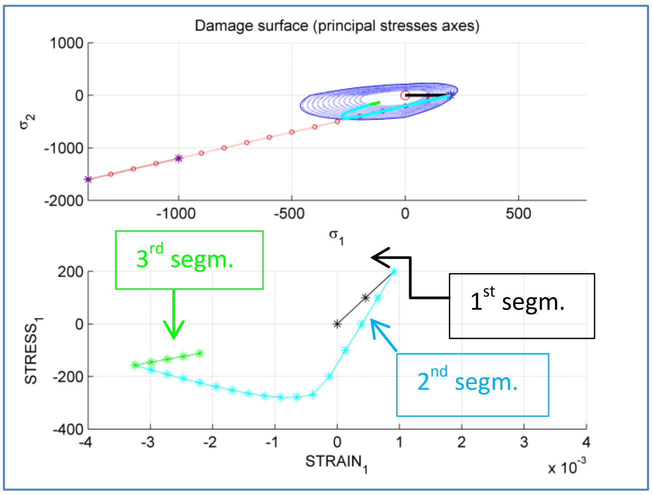


Figure3a. non-sym model - softening - loading1

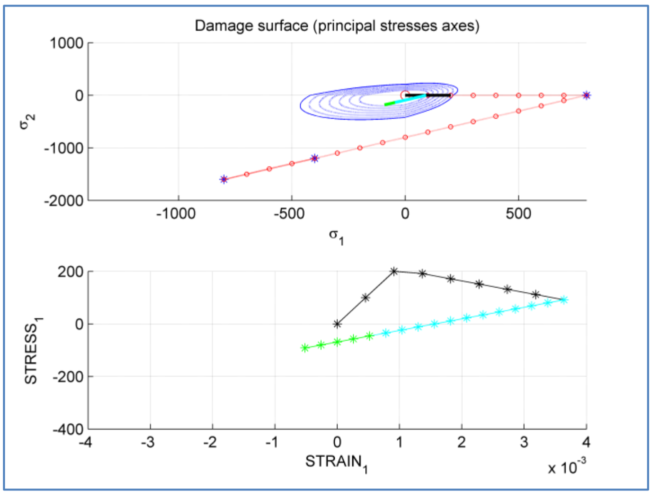


Figure3b. non-sym model - softening - loading2

By this sensitivity analysis, considering the fact that we are willing in this study to witness all behaviors of material for continuum damage models, we may conclude that for the 1st segment of loading path it is convenient to choose a stress more than the yield stress but not as much that it would completely affect 2nd and 3rd steps. So 400 MPa is chosen.

On the other hand, for the 2nd and 3rd segment we need a relatively large compressive and tensile stress, respectively, insuring that the material would enter compressive inelastic phase in 2nd segment for non-symmetric model and tensile hardening in 3rd segment for both models. These stresses are so high for this material but are chosen just for academic purposes, as discussed.

case	$\Delta\sigma_1$	$\Delta\sigma_2$	Tensile.	Compress.	(σ_1, σ_2)
1	+400	0	Loading	-	(400,0)
	-1600	0	Unloading	Loading	(-1200,0)
	+2400	0	Loading	Unloading	(1200,0)
2	+400	0	Loading	-	(400,0)
	-1600	-1600	Unloading	Loading	(-1200,-1600)
	+2400	+2400	Loading	Unloading	(1200,800)
3	+400	+400	Loading	-	(400,400)
	-1600	-1600	Unloading	Loading	(-1200,-1200)
	+2400	+2400	Loading	Unloading	(1200,1200)

Table2. Definition of Load cases

❖ *Rate Independent Models [INVISCID]*

5. Only-tension / Non-Symmetric tension compression Models

In this part the behavior of two added damage models in the code, namely only-tension and non-symmetric are provided for various load cases.

❖ **Loadcase1.**

The hardening and softening behavior of only-tension and non-symmetric models are portrayed for loadcase1 in **Figure4** and **5**, respectively. This load case is completely **uniaxial** and one may check that in 2nd path and after tensile unloading, material comes back exactly to the stress initiation point, namely (0,0) and this is the characteristic of purely damage models. In **Plastic Damage Models** which are widely used for **concrete modeling** one may detect that doing the same test, the tensile unloading path (cyan line) does not coincide tensile loading path (black line) in (0,0) due to the residual plastic strains.

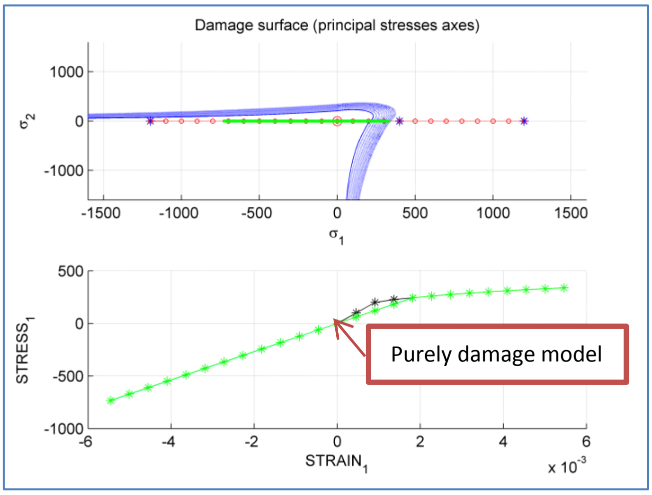


Figure4a. only-tension – exp hardening - loadcase1

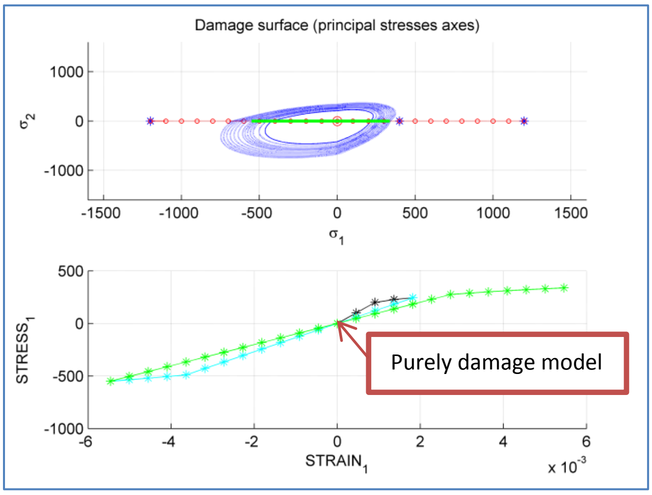


Figure4b. non-symmetric – exp hardening - loadcase1

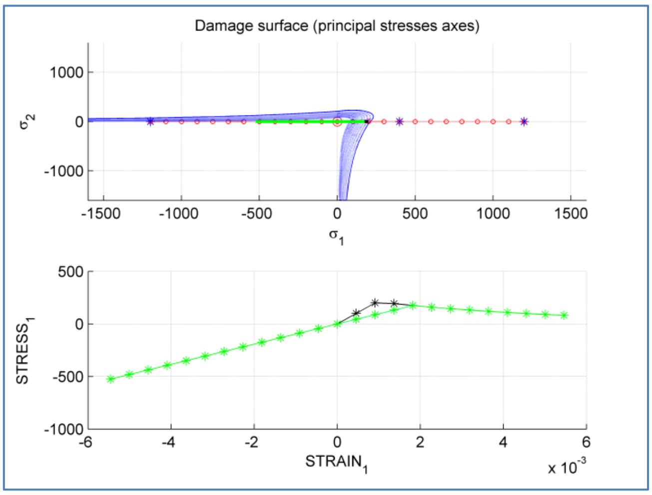


Figure5a. only-tension – exp softening - loadcase1

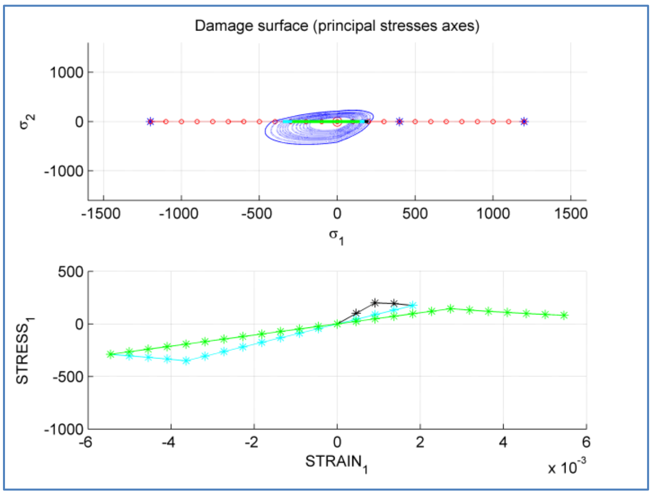


Figure5b. non-symmetric – exp softening - loadcase1

- **Loadcase2.**

The hardening and softening behavior of only-tension and non-symmetric models are portrayed for loadcase2 in **Figure6** and **7**, respectively. This load case is a mixed uniaxial and biaxial loading and one may check that in 2nd path and after tensile unloading, material does not come back exactly to the stress initiation point (0,0). This is not related to the residual strain theory but it is based on the biaxial nature of 2nd segment of loading path which does not allow the material to come back exactly to its origin point at (0,0).

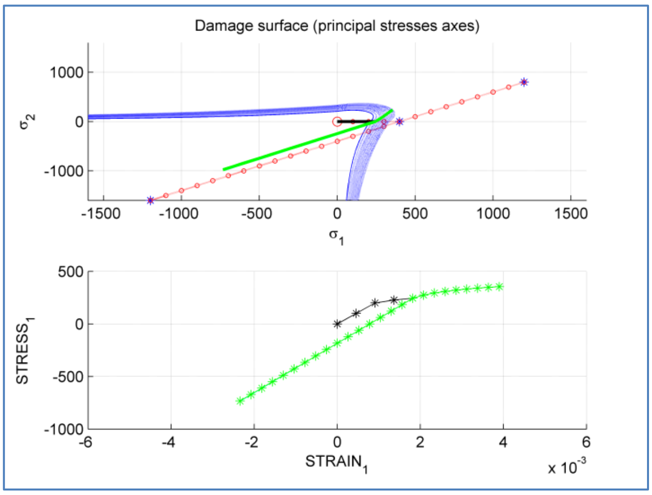


Figure6a. only-tension model – exp hardening - loading2

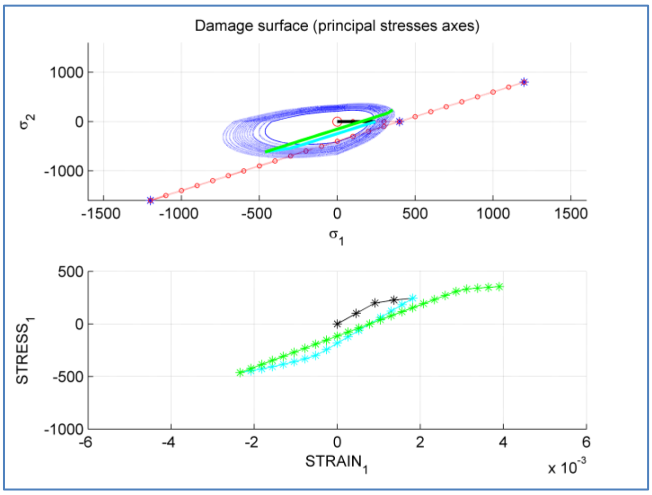


Figure6b. non-sym model – exp hardening - loading2

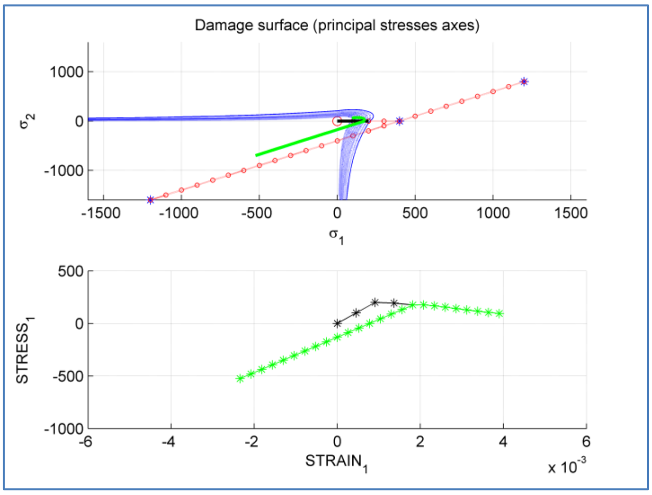


Figure7a. only-tension model – exp softening - loading2

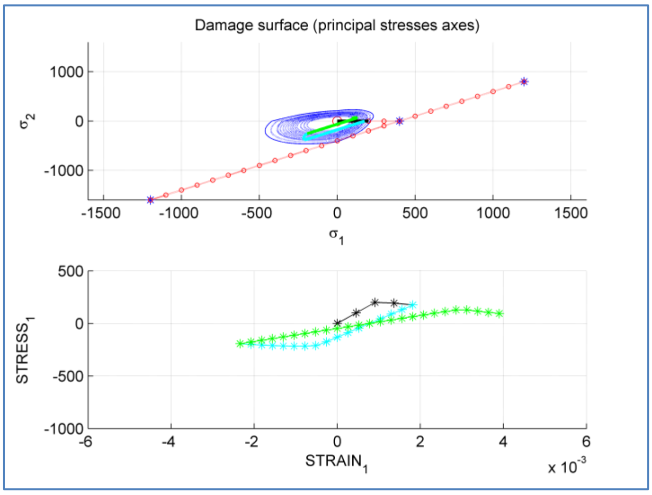


Figure7b. non-sym model – exp softening - loading2

As it was supposed, the compression hardening/softening occurs only for non-symmetric model (cyan line, **Figure6b-7b**) and the only-tension model just do the hardening/softening when the tensile stress value goes beyond elastic limit, not for the compression. It let the material to tolerate compression even to the infinity (green line, **Figure6a-7a).**

❖ Loadcase3.

The hardening and softening behavior of only-tension and non-symmetric models are portrayed for loadcase3 in **Figure8** and **9**, respectively. Like the previous case, this load case is a mixed uniaxial and biaxial loading and one may check that in 2^{nd} path and after tensile unloading, material does not come back exactly to the stress initiation point (0,0), due to the biaxial nature of 2^{nd} segment of loading path.

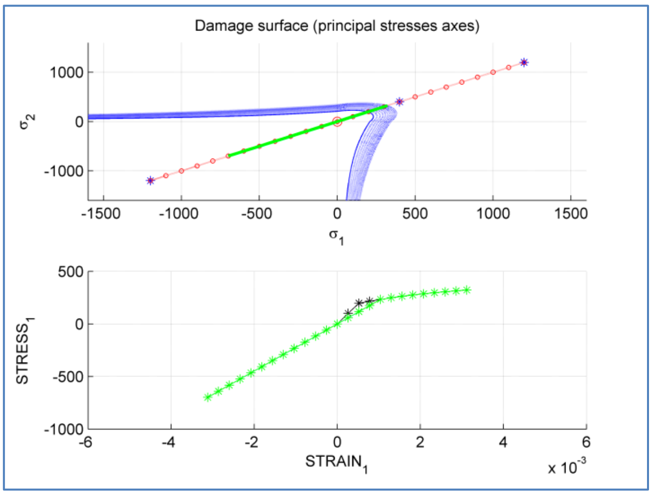


Figure8a. only-tension model – exp hardening - loading3

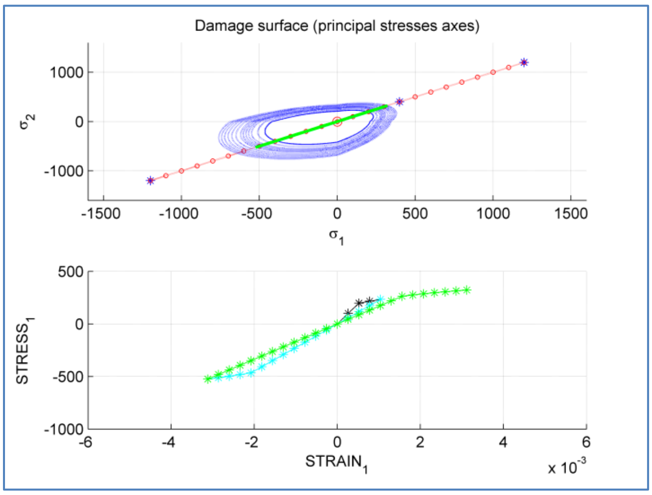


Figure8b. non-sym model – exp hardening - loading3

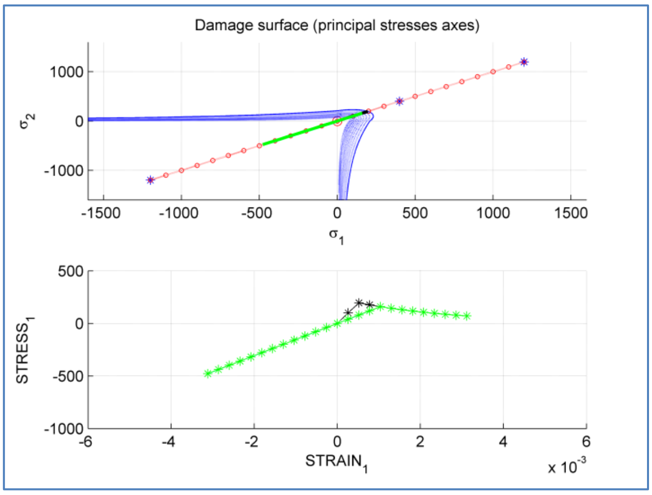


Figure9a. only-tension model – exp softening - loading3

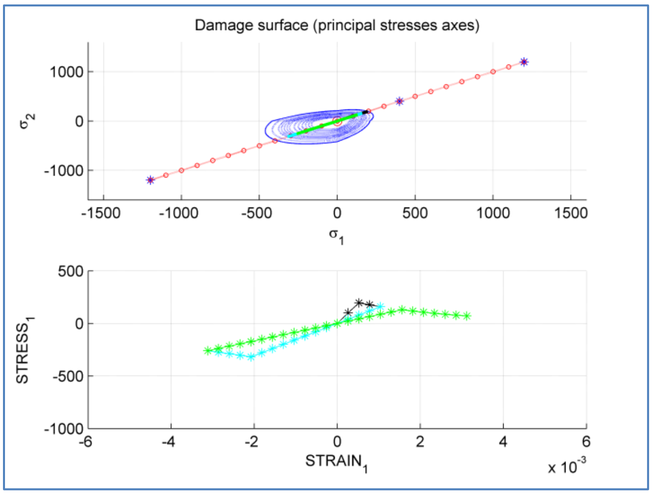


Figure9b. non-sym model – exp softening - loading3

6. Effect of Hardening/Softening Variable Type

As it is provided in **Figure10**, there are two main differences between linear and exponential hardening/softening laws. The linear one reaches to its limits faster and it also fixes the hardening internal variable (q) after some specified value r1 (as a function of yield stress, elasticity modulus, upper and lower bound of hardening variable), but the exponential law reaches to the upper and lower limits so gradually and converges to them at the infinity. So, the linear law would act more strictly when it enters to inelastic limit. These facts are shown in **Figure11** and **12** for the linear hard./soft. case:

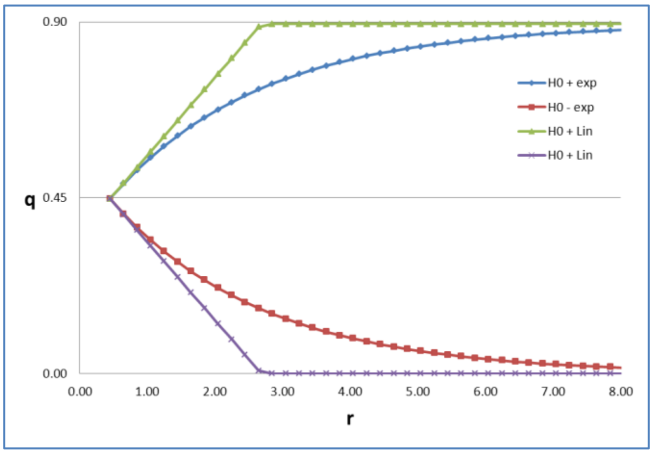


Figure10. Linear and Exponential Hardening/Softening

- The slope of stress-strain curve would be steeper.
- The hardened/softened limits in stress space would be more exaggerated.

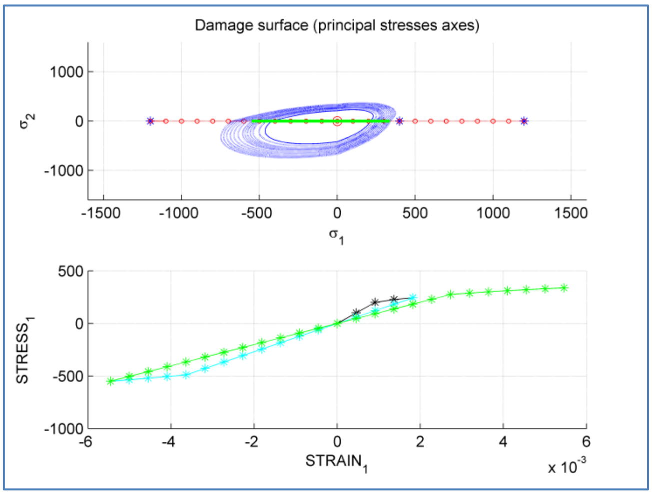


Figure11a. non-sym model – exp hardening - loading1

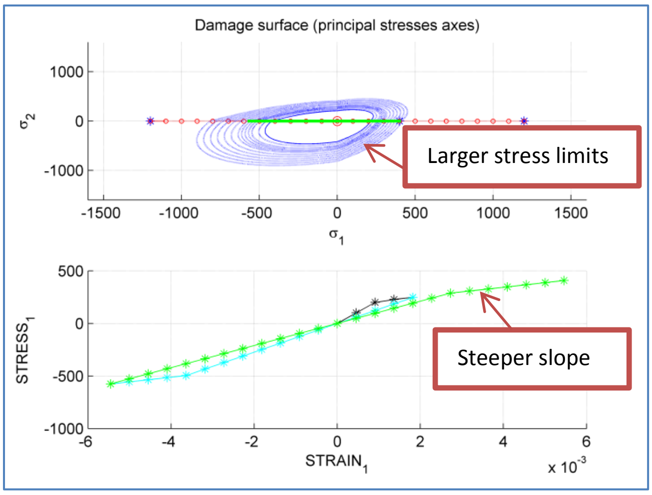


Figure11b. non-sym model – lin hardening - loading1

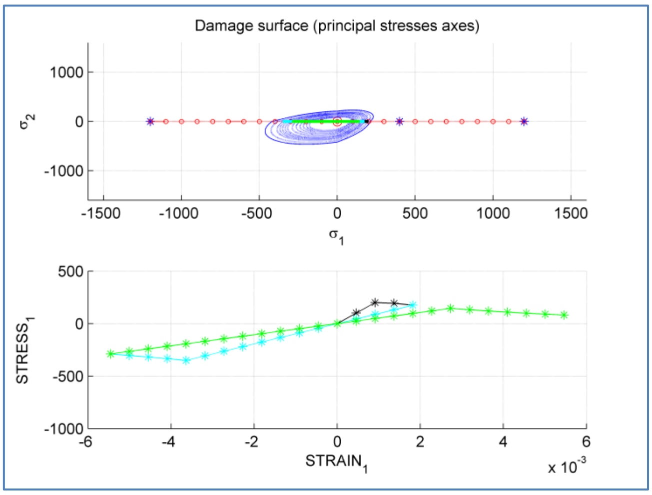


Figure12a. non-sym model – exp softening - loading1

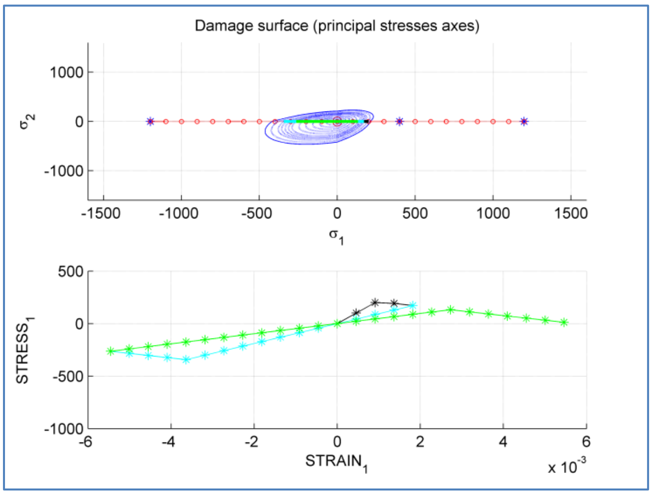


Figure12b. non-sym model – lin softening - loading1

7. *Effect of Hardening/Softening Modulus Value (H)*

One may be interested to study the effect of hardening/softening initial slope on the behavior of material. For this purpose a sensitivity study is carried out on the linear Hardening/softening law as a sample for non-symmetric model under the loadcase1. The H0 value for low value is considered ±0.1 and for high value as ±0.8. The results for the hardening case are portrayed in **Figure13**. It is clear that the higher the H0, the faster material reaches to its hardening limits.

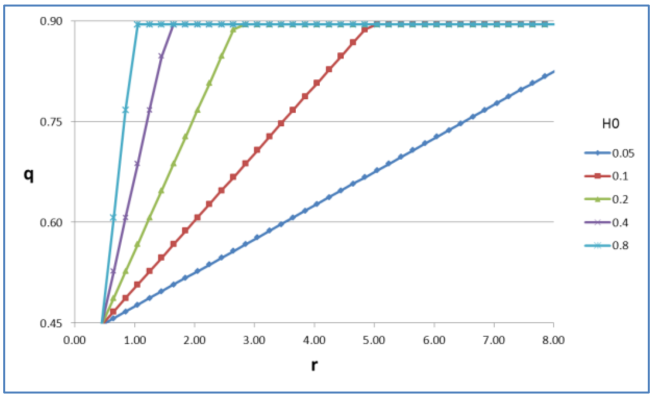


Figure13. effect of H0 value on Linear Hardening

Also, as it is shown in **Figure14** and **15**, it may be concluded that by increasing the initial value of the slope of hardening/softening formula, one may expect:

- A higher value for the slope of stress-strain curve.
- A more exaggerated hardened/softened limit in stress space domain.

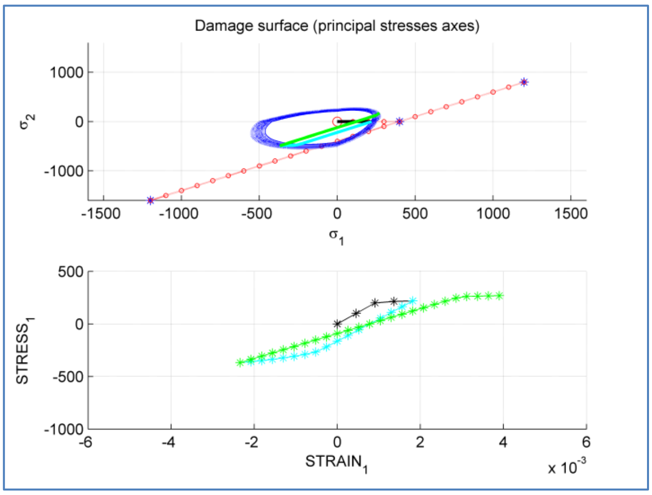


Figure14a. non-sym model – lin hardening - low value

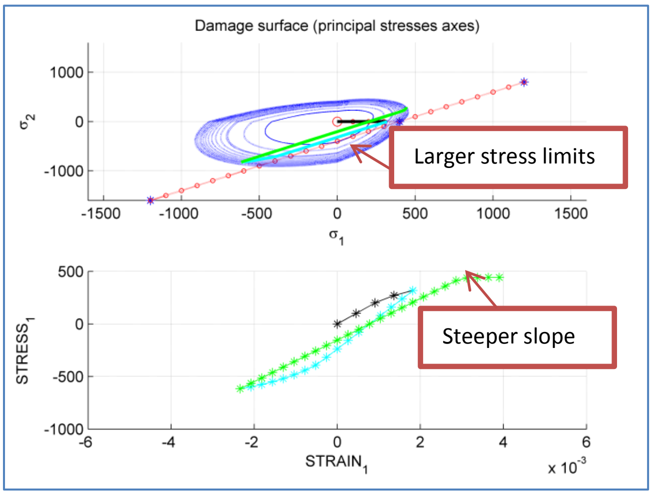


Figure14b. non-sym model – lin hardening - high value

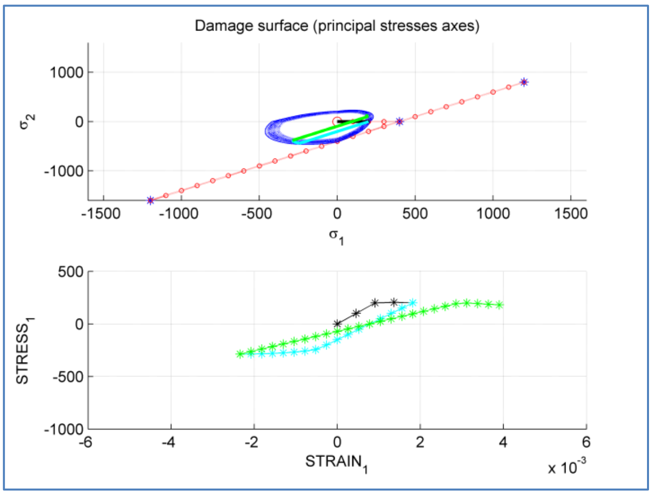


Figure15a. non-sym model – lin softening - low value

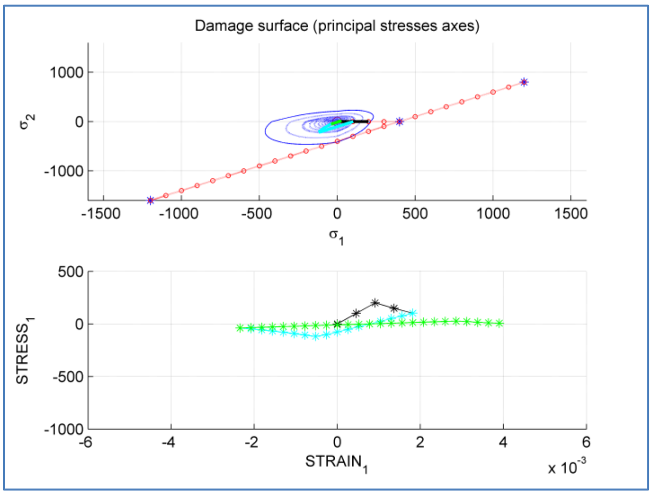


Figure15b. non-sym model – lin softening - high value

❖ *Rate Dependent Models [VISCOUS]*

The main difference between viscous and inviscid models is the role of time. Inviscid models are not sensitive to time and the rate of strain is not considered in the formulation. So they can only model cases that strains are applied to the structure so gradually and in a long period of time. In this chapter we will discuss the effect of involving time in the equations.

8. *Effect of Viscosity Parameter (η)*

As theory say, if viscosity parameter (η) tends to zero, the sensitivity of viscous model to the rate of strain decreases and finally it would converge to inviscid model results. We may study this effect by doing a sensitivity analysis on the value of η.

Figure16a to 16e are related to a continuum damage symmetric model under a uniaxial loading 3-segment path in order to follow the typical tensile cyclic stress-strain curve provided in all references for softening behavior of materials. This loading test starts with a uniaxial tension (start point to 1st point). All the tensile load would be compensated by the same amount of tensile unloading (1st point to 2nd point) and then again a larger tensile loading would cause the material to step up exactly on the previously unloading path and continue its way from the point where 1st tensile segment were finished (1st point). We may see that the material would again start to follow the same damage degradation procedure when it comes back all along the unloading path and enters again into the loading path (2nd point to 3rd point).

In regular solids No Healing is expected. So internal (r) and damage (d) variable should only increase along time and this fact is clearly shown in **Figure16d** and **16e** which

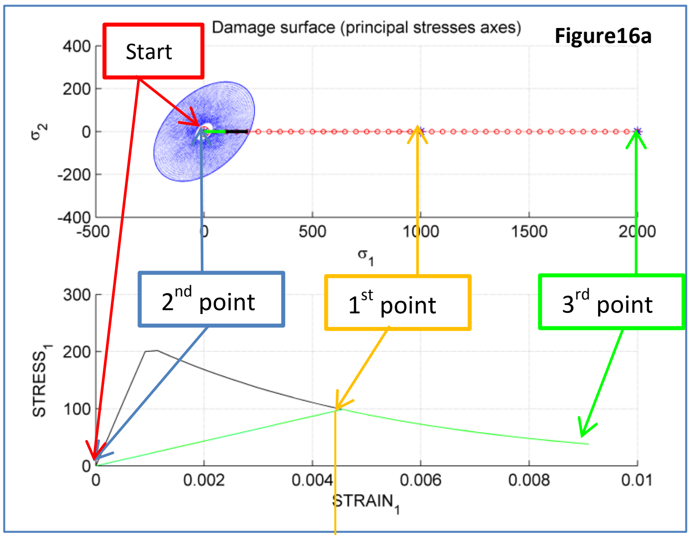


Figure16a. Stress softgening space and stress-strain curve

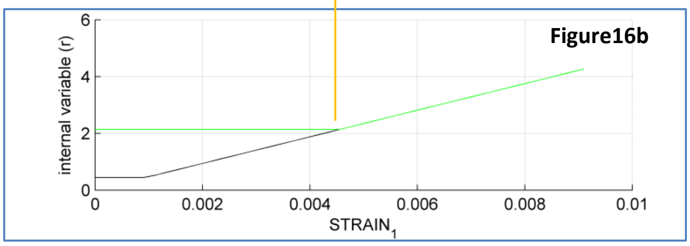


Figure16b. internal variable with respect to strain evolution

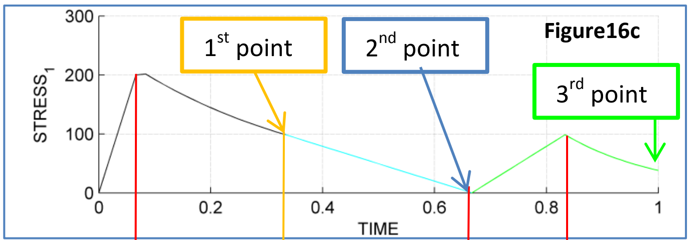


Figure16c. stress evolution with respect to time

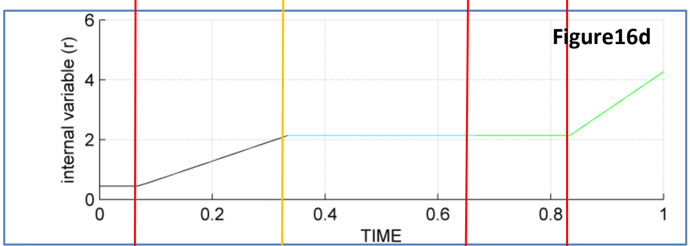


Figure16d. internal variable evolution with respect to time

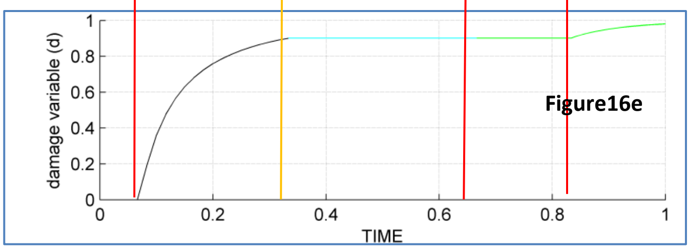
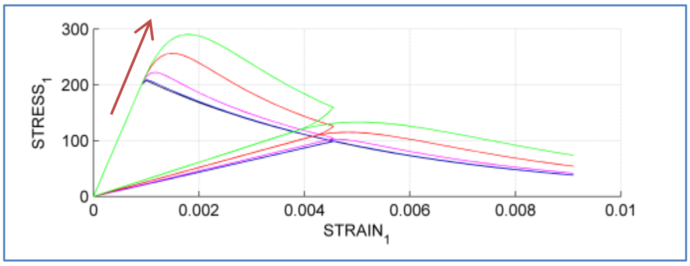


Figure16e. damage variable evolution with respect to time

draw the evolution of those variables along time and also in **Figure16b** along strain evolution.

On the other hand it can be seen **Figure16c** that due to the softening nature of material (H=-0.2) going forward with strain evolution, the stresses decrease as the first moment in which the elastic domain is exceeded and subsequently there has been a need for softening algorithm. But this decrease in stress is concurrent with increase in damage, only in loading part. In unloading part the internal variable and damage parameters are constant but the material is forfeiting stress due to unloading.

For studying the effect of viscosity parameter we may compare different η parameters with each other and with inviscid case. As it was expected by increasing the η, no difference appears in elastic part but the inelastic part expands and the max inelastic threshold goes up. On the other hand, if we converge the η to zero the results would exactly fit to the rate independent (inviscid) case, which was widely discussed in previous part.



—	Inviscid
—	η=0.0001
—	η=0.01
—	η=0.04
—	η=0.08

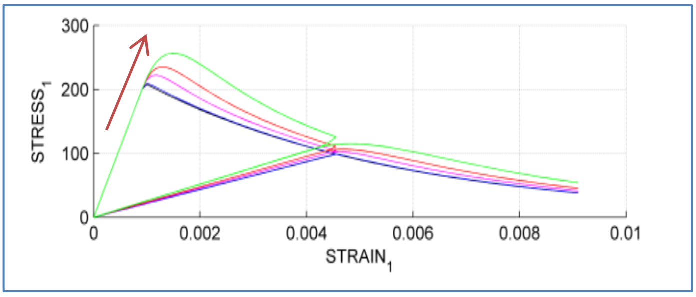
Total Time = 1
α = 1
H = -0.2

Figure17. effect of viscousity parameter (η) in Viscous models

9. *Effect of Strain Rate*

For the rate independent models no sensitivity of time is included in the model, But for the viscous models as much as the same amount of load is applied faster to the same material, the material would react stiffer and the elastic limit would goes higher. This is due to the viscosity effect like dampers in which they absorb high velocity loads but let the low velocity loads to apply.

In **Figure18** it may be checked that increasing the strain rate $\dot{\varepsilon}$ (which means decreasing the total time of applying load) would cause to a higher initial elastic limit and a stiffer response of material. On the other hand, decreasing the rate of strain would mean deleting the time effect and as it is shown in our test if we apply the load in more than 10 seconds it would act like inviscid models.



—	Inviscid
—	Time=10
—	Time=1
—	Time=0.5
—	Time=0.25

Eta (η) = 0.01
α = 1
H = -0.2

Figure18. effect of strain rate in Viscous models

10. Effect of Time Integration Parameter - Alpha (α)

We are using Alpha method for numerical time integration in rate dependent (viscous) models and the effect of α parameter should be studied. As we know for α=[0,1] the method is accurate (first order) and for α=0.5 it is 2nd order accurate. From stability point of view, method is stable for α=[0,0.5]. so the best option is to use α=0.5 which leads to 2nd order Crank-Nicholson method.

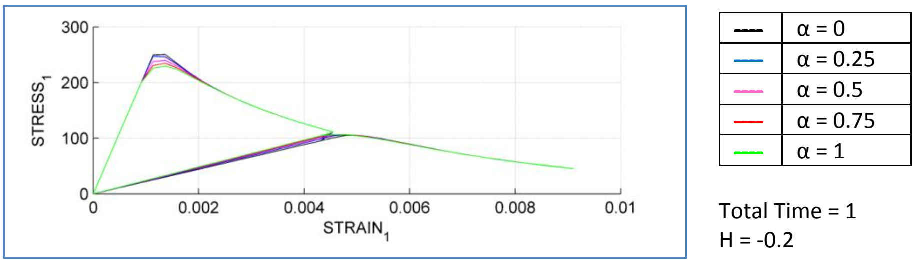


Figure19. effect of viscousity parameter (η) in Viscous models

Figure 19 shows the effect of α for the model which was analyzed in previous part. We should notice that in this figure a high time step number is used to insure to accuracy and stability of methods in order to be able to compare them together.

In **Figure20** the stability of methods is discussed. As we know if we increase time steps all alpha methods would converge to the accurate answer but with a high cost of computation. So as it is observed in **Figure20a** for α=0 which is less than 0.5 for few amount of time steps, method is doing oscillations which is the sign of instability. On the other hand α=0.5 and α=1 even for low number of time steps have no oscillation and the lack of accuracy is just related to low number of time steps.

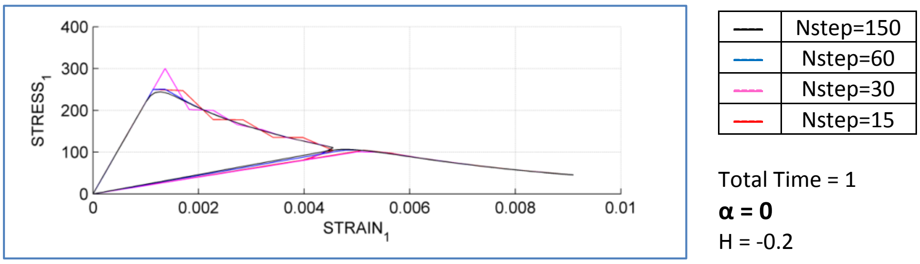


Figure20a. effect of (α) parameter in Viscous models

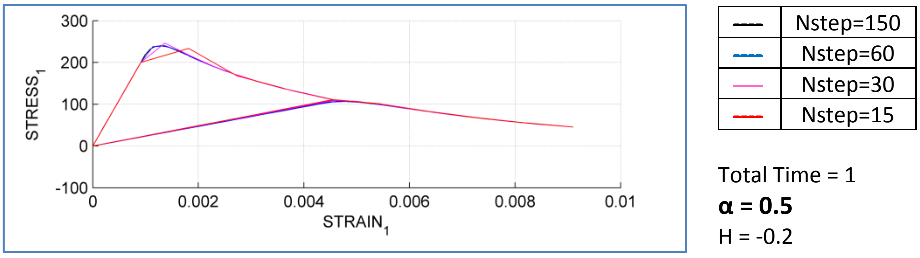


Figure20b. effect of (α) parameter in Viscous models

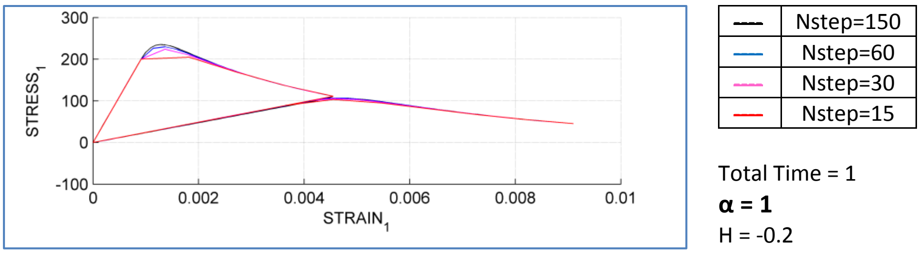


Figure20c. effect of (α) parameter in Viscous models

Finally the effect of alpha on the evolution along time of the C11 component of the tangential and algorithmic constitutive tensors is studies. As it is obvious in **Figure21** for α=0 these two tensors are exactly the same, as it was expected.

Figure22a develops the evolution along time of the C11 component of the tangential constitutive tensors.

Figure22b develops the evolution along time of the C11 component of the tangential constitutive tensors

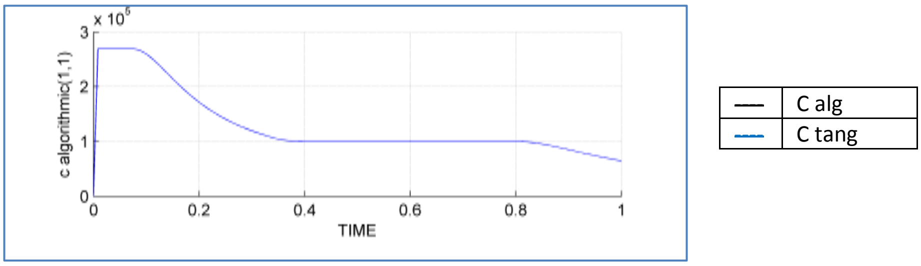


Figure21. C11 component of the tangential and algorithmic constitutive tensors for α=0

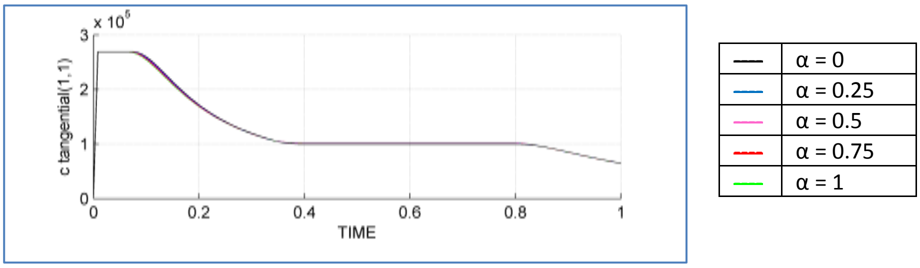


Figure22a. evolution along time of the C11 component of the tangential constitutive tensors

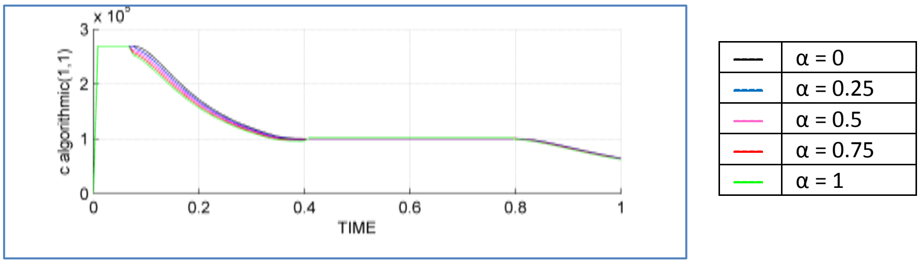


Figure22b. evolution along time of the C11 component of the algorithmic constitutive tensors

❖ *Appendix*

1. Appendix1: dibujar_criterion_dano1

```
tel = 0.0  ;
te2 = 2*pi ;
dte = 0.01 ;

axeq=0;
%------------------------------------------Inverse ce
ce_inv=inv(ce);

%------------------------------------------Polar Coordinates
if MDtype==1

    tetha=[te1:dte:te2] ;
    D=size(tetha)       ;
    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
        vec_sin  = [m1(i) m2(i) 0 nu*(m1(i)+m2(i))] ;
        radio(i) = q/sqrt(vec_sin*ce_inv*vec_sin')  ;

        s1(i)  = radio(i)*m1(i) ;
        s2(i)  = radio(i)*m2(i) ;
    end
    hplot = plot(s1,s2,tipo_linea);

elseif MDtype==2

    tetha=[te1:dte:te2] ;
    D=size(tetha)       ;
    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
        vec_sin     = [m1(i) m2(i) 0 nu*(m1(i)+m2(i))]     ;
        vec_sin_pos = (vec_sin+abs(vec_sin))/2.0           ;
        radio(i)    = q/sqrt(vec_sin_pos*ce_inv*vec_sin') ;

        s1(i)  = radio(i)*m1(i) ;
        s2(i)  = radio(i)*m2(i) ;
    end
    hplot = plot(s1,s2,tipo_linea);

elseif MDtype==3

    tetha=[te1:dte:te2] ;
    D=size(tetha)       ;
    m1=cos(tetha)       ;
```

Symmetric model polar coordinate definition for elastic stress limits

Only tension model polar coordinate definition for elastic stress limits

Non symmetric model
polar coordinate definition
for elastic stress limits

```
    m2=sin(tetha)         ;
    Contador=D(1,2)       ;

    radio = zeros(1,Contador) ;
    s1    = zeros(1,Contador) ;
    s2    = zeros(1,Contador) ;
    for i=1:Contador
        vec_sin     = [m1(i) m2(i)  0 nu*(m1(i)+m2(i))]       ;
        vec_sin_pos = (vec_sin+abs(vec_sin))/2.0              ;
        angt        = (sum(vec_sin_pos))/(sum(abs(vec_sin))) ;
        radio(i)    = q/sqrt(vec_sin*ce_inv*vec_sin')         ;
        radio(i)    = radio(i)/(angt+(1.0-angt)/n)            ;

        s1(i)=radio(i)*m1(i);
        s2(i)=radio(i)*m2(i);
    end
    hplot =plot(s1,s2,tipo_linea);

end

    if axeq==1
        axis equal
    else
    end

return
```

2. Appendix2: damage_main

```
%-------------------------------------------Initializing Global Cell Arrays
totalstep  = sum(istep)                    ;
sigma_v    = cell(totalstep+1,1)           ;
TIMEVECTOR = zeros(totalstep+1,1)          ;
delta_t    = TimeTotal./istep/length(istep)  ;

%-------------------------------------------Elastic Constitutive Tensor
[ce]       = tensor_elastico1 (Eprop, ntype) ;

%-------------------------------------------Historic Variables
r0         = sigma_u/sqrt(E)               ;
hvar_n     = zeros(mhist,1)                ;  % empty
hvar_n(5)  = r0                            ;  % Internal variable (r)
hvar_n(6)  = r0                            ;  % Hardening variable (q)
eps_n1     = zeros(mstrain,1)              ;  % Strain vector
i          = 1                             ;
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)];

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)
%-------------------------------------------Core Loop
for  iload = 1:length(istep)                  % Stress point #
    for iloc = 1:istep(iload)                 % Load state # in each path
        i = i + 1 ;
        TIMEVECTOR(i) = TIMEVECTOR(i-1)+ delta_t(iload) ;
        dt = delta_t(iload) ;
        %-------------------------------------
        eps_n  = strain(i-1,:) ;
        eps_n1 = strain(i,:)   ;

        %-------------------------------------
        [sigma_n1,hvar_n,aux_var,c_tang,c_algo] =
rmap_dano1(eps_n,eps_n1,hvar_n,Eprop,ce,MDtype,n,dt);
        c11=c_tang(1,1);
        c22=c_algo(1,1);

        %-------------------------------------Plotting Inelastic damage surface
        if(aux_var(1)>0)                         % (in case: rtrial > r_n)
            hplotSURF(i) = dibujar_criterio_dano1(ce, nu, hvar_n(6),
'r:',MDtype,n);
            set(hplotSURF(i),'Color',[0 0 1],'LineWidth',1);
        end
        %------------------------------------------Glabal Var. - 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 ;

        %------------------------------------------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
```

Defining eps_n and dt

Importing them to the rmap_dano1

3. Appendix3: rmap_dano1

```
%-------------------------------------------Definition
hvar_n1    = hvar_n         ;
r_n        = hvar_n(5)      ;
q_n        = hvar_n(6)      ;

E          = Eprop(1)       ;
nu         = Eprop(2)       ;
H0         = Eprop(3)       ;
sigma_u    = Eprop(4)       ;
hard_type  = Eprop(5)       ;
viscpr     = Eprop(6)       ;
eta        = Eprop(7)       ;
ALPHA      = Eprop(8)       ;

 r0        = sigma_u/sqrt(E) ; % r0     [initial] Internal strain-like Var.
 q_min     = (10.^(-6))*r0   ; % q min  [minimum] Internal Stress-like Var.
 q_max     = r0+(r0-q_min)   ; % q max [infinity] Internal Stress-like Var.

 if H0>0
     q_ult = q_max;
 else
     q_ult = q_min;
 end
```

Viscous case definition

For damage surface criterion

```
%-------------------------------------------Damage surface
if viscpr==0    %Rate-independent |inviscid
    [rtrial]     = Modelos_de_dano1 (MDtype,ce,eps_n1,n) ;
else            %Rate-dependent |viscous
    [rtrial_n]   = Modelos_de_dano1 (MDtype,ce,eps_n,n)   ;
    [rtrial_n1]  = Modelos_de_dano1 (MDtype,ce,eps_n1,n) ;
    [rtrial]     = (1.0-ALPHA)*rtrial_n+ALPHA*rtrial_n1   ;
end

%-------------------------------------------Loading Status
fload=0;
if(rtrial > r_n)          % Inelastic load --> Hardening (compute Algorithmic
Constitutive Tensor)
    fload = 1 ;

    %---------------------------------------Viscousity Type
    if viscpr ==0                     % Inviscid.
        r_n1 = rtrial;
    else                              % Viscous.
        r_n1 =((eta-dt*(1.0-ALPHA))*r_n + dt*rtrial)/(eta+ALPHA*dt);
    end

    %---------------------------------------Hardening Type
    if hard_type == 0                 % >> Linear.
        r1 = r0+(q_ult-r0)/H0 ;
        if r_n1<r1
            H_n1 = H0 ;
            q_n1 = q_n+ H0*(r_n1 - r_n) ;
%           q_n1 = r0 + H0*(r_n1 - r_0) ;                    %  ??????
        else
            H_n1 = 0.0   ;
            q_n1 = q_ult ;
        end
```

```
    else                                % >> Exponential.
        A1    = H0*r0/(q_ult-r0) ;
        q_n1 = q_ult-(q_ult-r0)*exp(A1*(1-r_n1/r0))   ;
        H_n1 = A1*((q_ult-r0)/r0)*exp(A1*(1-r_n1/r0));
    end
    %--------------------------------------

else                          % Elastic load / unload --> No Hardening
    fload = 0    ;
    r_n1  = r_n ;
    q_n1  = q_n ;
end

if(q_n1 < q_min)
    q_n1 = q_min ;
end
if(q_n1 > q_max)
    q_n1 = q_max ;
end

%------------------------------------------Damage variable
dano_n1 = 1.d0-(q_n1/r_n1) ;

%------------------------------------------stress Tensor
sigma_bar_n1 = ce*eps_n1'                   ;
sigma_n1     = (1.d0-dano_n1)*sigma_bar_n1 ;

%------------------------------------------Tangential Constitutive Tensor
c_tang=zeros(4,4);
c_algo=zeros(4,4);

if fload==0
    if viscpr==0
        c_tang = (1-dano_n1)*ce ;
    else
        c_tang = (1-dano_n1)*ce ;
        c_algo = c_tang         ;
    end
else
    if viscpr==0
        c_tang = (1-dano_n1)*ce-((q_n1-
H_n1*r_n1)/(r_n1.^3))*(sigma_bar_n1*sigma_bar_n1') ;
    else
        c_tang = (1-dano_n1)*ce ;
        c_algo = c_tang -(ALPHA*dt/(eta+ALPHA*dt)*(1.0/rtrial_n1)*(q_n1-
H_n1*r_n1)/(r_n1.^2))*(sigma_bar_n1*sigma_bar_n1');
    end
end
```

Exponential hardening softening law

Tangential and algorithmic constitutive tensor calculation

4. Appendix4: modelos-de_dano1

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

elseif (MDtype==2)                               % Only tension
    vec_sig      = eps_n1*ce                                    ;
    vec_sig_pos  = (vec_sig+abs(vec_sig))/2.0                   ;
    rtrial       = sqrt(vec_sig_pos*eps_n1')                    ;

elseif (MDtype==3)                              % Non-symmetric
    vec_sig      = eps_n1*ce                                    ;
    vec_sig_pos  = (vec_sig+abs(vec_sig))/2.0                   ;
    angt         = sum(vec_sig_pos)/sum(abs(vec_sig))           ;
    rtrial       = sqrt(eps_n1*ce*eps_n1')                      ;
    rtrial       = (angt+(1.0-angt)/n)*rtrial                   ;
end

return
```