UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

MSc. in Computational Mechanics
Computational Solid Mechanics (COSM)
Spring Semester 2017/2018

ASSINGMENT 1: CONTINUUM DAMAGE MODELS

Submitted By:
Luan Malikoski Vieira

March 23, 2018
Barcelona, Spain

Contents

1 Part I : Rate Independent Models

Before presenting the results for the proposed load/unloading cases, for *Rate Independent* models, a brief assessment of the implemented exponential hardening/softening model is done in section 1.1. Then, section 1.2 presents simulation results of each loading/unloading case employing different continuum damage models, named from here as follows: Non-Symmetric (N) and Tensile (T) model.

1.1 Assessment of Hardening/softening Exponential Model

The implementation of hardening/softening exponential law for $(q(r))$ was tested with an uniaxial loading case (only σ_1 component). The material property, load data, damage model and hardening parameters are given in table (1). The **H** parameter, for the exponential case, is set to be the slope $q'(r_o) = H = H_o$ at the beginning of analysis $r = r_o$, thus, matching the slope of the linear model.

$\sigma_{\mathbf{y}}$ [Pa]	E [Pa]	ν[-]	$\mathbf{\Delta\bar{\sigma}_1}^{(1)}$ [Pa]	$\mathbf{\Delta\bar{\sigma}_1}^{(2)}$ [Pa]	$\mathbf{\Delta\bar{\sigma}_1}^{(3)}$ [Pa]	Model	H [-]	$\mathbf{q}_\infty$
200	2.0E5	0.3	350	350	350	S	± 0.1	1.56/1.26

Table 1: Material properties and effective stress increments

Figure (1) (a) shows the evolution of the internal variable q with respect to the internal variable r for the cases of exponential softening (E-S), exponential hardening (E-H), linear softening (L-S) and linear Hardening (L-H). Also, in Figure (1) (b) the stress-strain path is provided for each case.
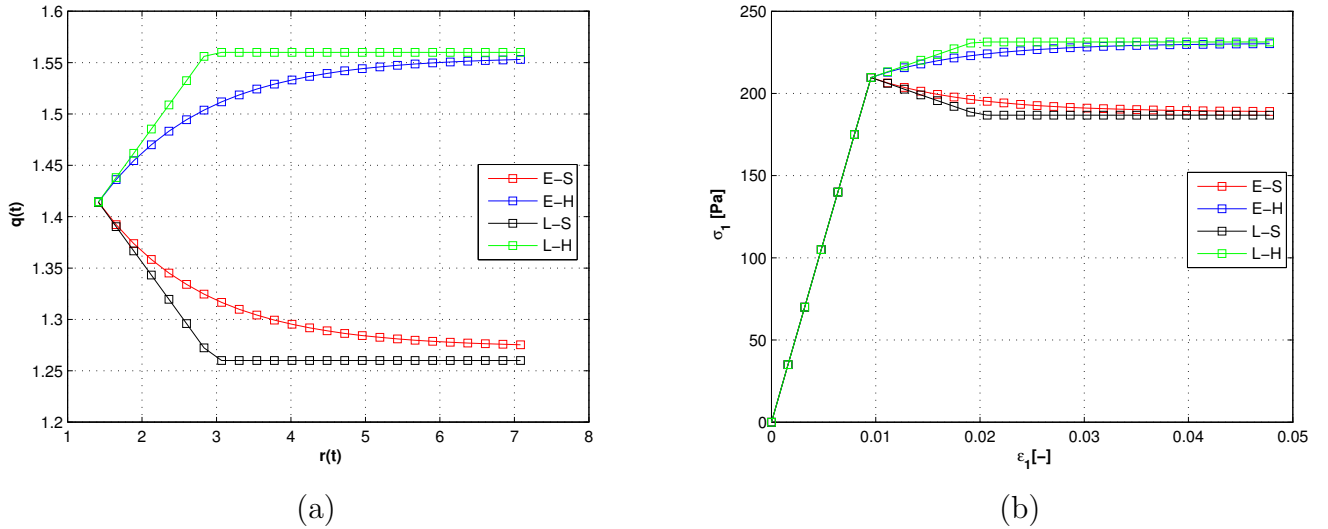

Figure 1: (a) Internal Variables;(b) Stress-Strain space

Those results shows the proper implementation of the exponential hardening/softening law. From here further analysis on each damage model, using these hardening laws, can be made.

1.2 Continuum Damage Models: Proposed Load Cases

The next subsections will focus on three load cases. Each case is composed of three load segments, starting from zero stress/strain, where effective stress increments is imposed. For each load case, the implemented damage models Non-Symmetric (N) and Tensile (T) will be evaluated. In order to shorten the analysis only linear softening model will be used as some conclusions can be made with no loss of generality regarding hardening law.

Also, the $\mathbf{q_\infty}$ parameter will be such that the variation of q will be practically unlimited ($q \geq 10^{-6}$). For all 3 cases the same material properties and hardening parameters will be employed and are summarized in table (2). Further, in all simulations in this section 15 discrete points and 4s for each effective stress increment are used.

$\sigma_{\mathbf{y}}$ [Pa]	E [Pa]	ν[-]	H [-]	$\mathbf{q_\infty}$	n
200	2.0E5	0.3	-0.1	10^{-6}	2

Table 2: Material properties

1.2.1 Load Case I

For case I, an uniaxial load, the selected effective stress increments are shown in table (3).

Increment	$\mathbf{\Delta\bar\sigma_1}$ [Pa]	$\mathbf{\Delta\bar\sigma_2}$ [Pa]
1	350	0
2	-950	0
3	1200	0

Table 3: Case I: Effective principal stresses increments.

Case I: Tension Only model (T)

The resultant evolution of damage surface ($f(\sigma, q)$) and damage variable q(t) for the Tension model (T) are shown in Figure (2).

Figure 2: Case I (T): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

The loading path at stress-strain space is shown in Figure (3). Regarding ϵ_2 for both T and N model, even though $\sigma_2 = 0$ for the whole time domain analysis, due to the poisson effect $\epsilon_2 = -\nu\epsilon_1$.
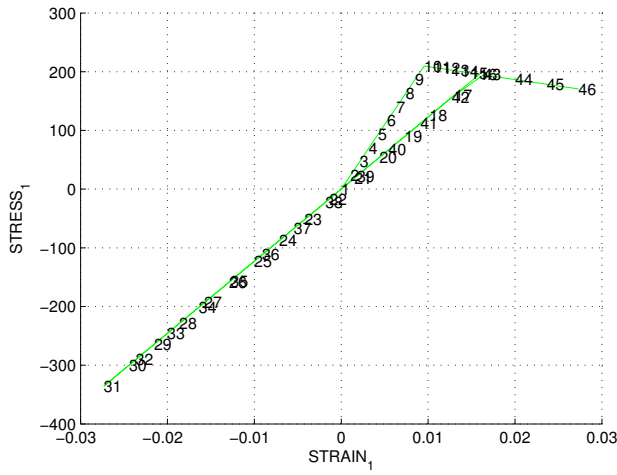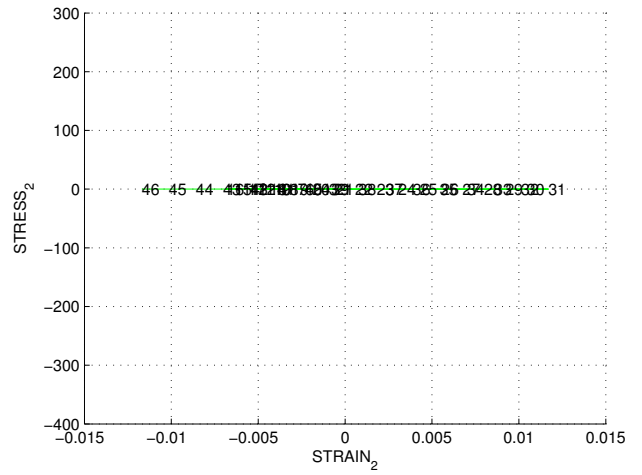
Figure 3: Case I (T): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

From the previous results it can be observed that the material, at gauss point level, undergoes a elastic load in the first increment $\Delta\bar\sigma_1{}^{(1)}$ until the yield stress σ_y is reached, as seen in Figure (3). From this point, around 2.2 s in Figure (2) (b), the damage variable starts to increase. The second increment $\Delta\bar\sigma_1{}^{(2)}$, uniaxial tensile unloading/compressive loading, is elastic in its whole range due to the damage model employed.

The third increment $\Delta\bar\sigma_1{}^{(3)}$, uniaxial compressive unloading/ tensile loading, is elastic until the strain is enough to reach the maximum inelastic strain of the first increment, which correspond to the yield surface in the strain space. At this point, damage increases again, around 11 s in Figure (2) (b). As this surface is equivalent as the stress space one, the stress surface is also reached and the principal stress decreases in a inelastic behavior (q reduces) while strains increases further (r increases).

Case I: Non-Symmetric Model (N)

The resultant evolution of damage surface ($f(\sigma, q)$) and damage variable q(t) for the Non-Symmetric model (N) are shown in Figure (4).
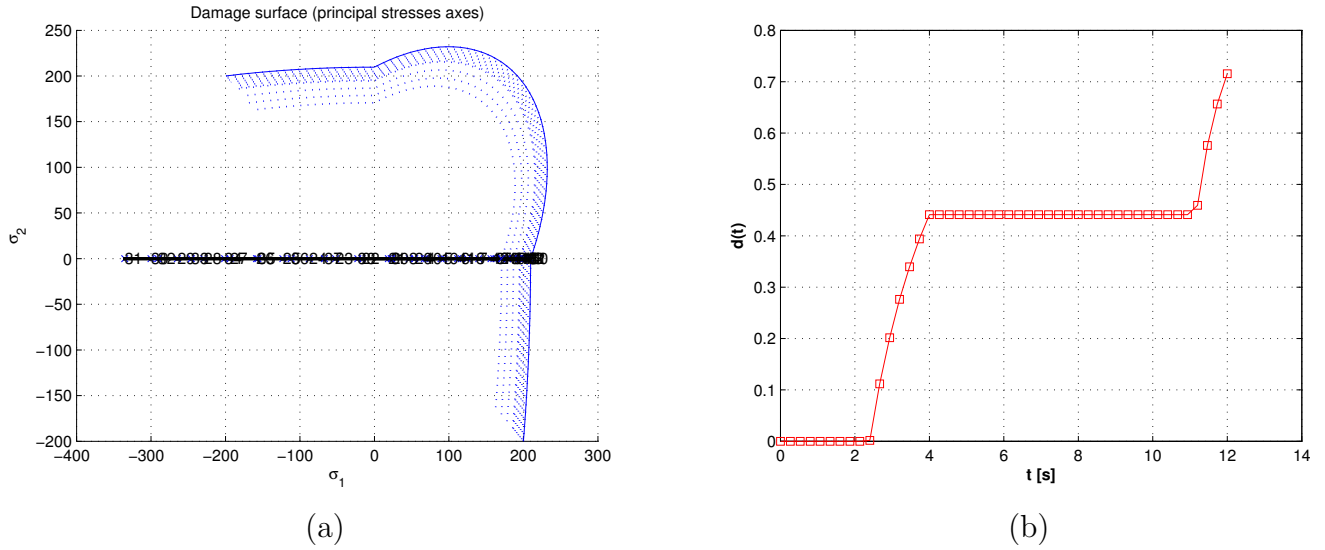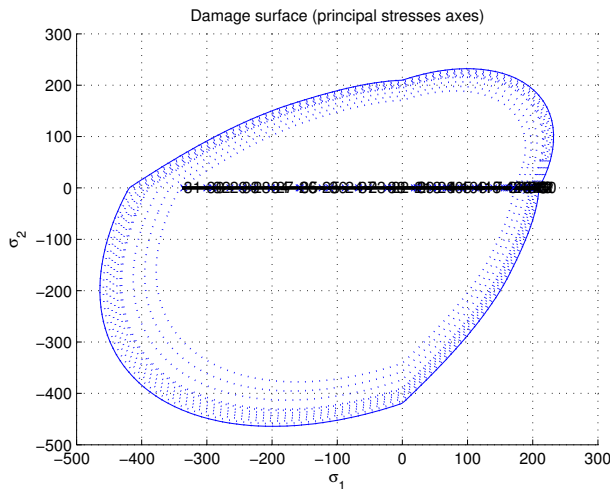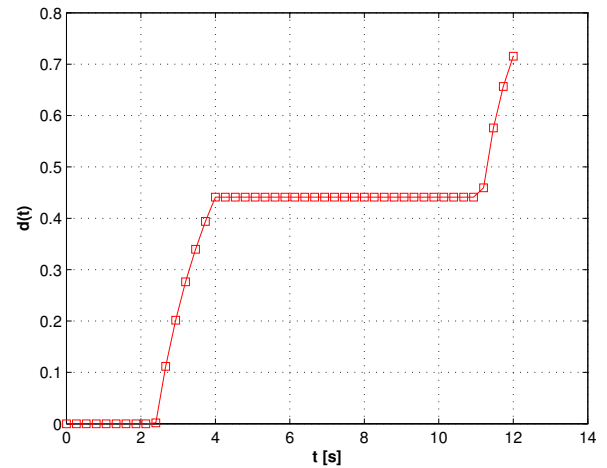

Figure 4: Case I (N): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

4

The loading path at stress-strain space is shown in Figure (5).

(a)   (b)
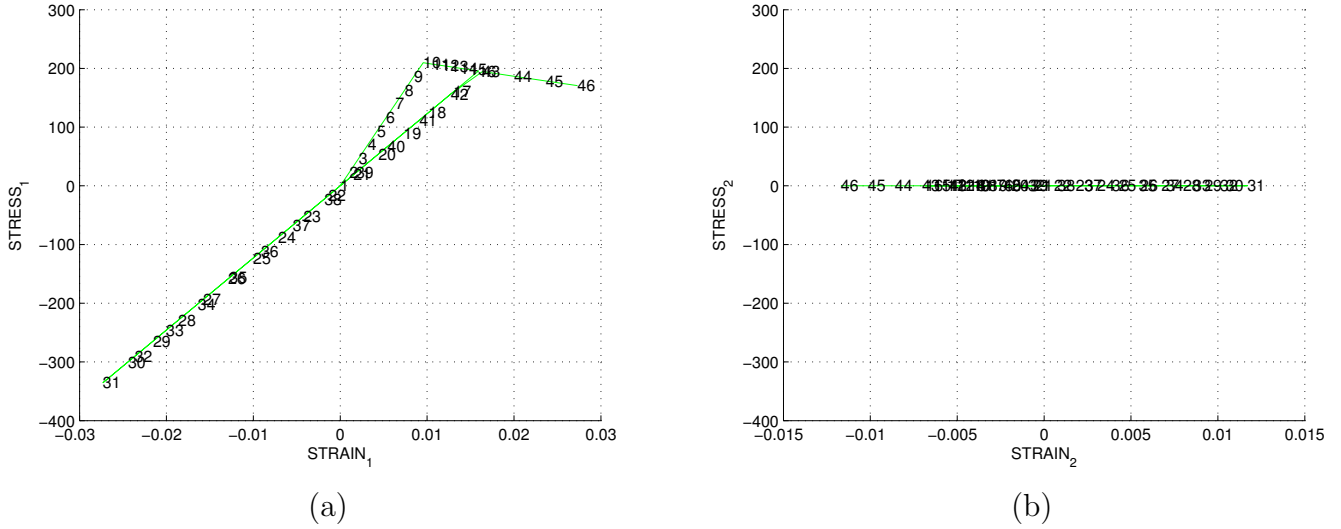
Figure 5: Case I (N): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

Exact same results as the Tension Only model (T) are obtained for Non-symmetric model (N) for this considered load case. Thus, same conclusions can be made. However, the same results were obtained because the compressive stress state, at increment $\Delta\bar{\sigma}_1^{(3)}$, was such that its norm was smaller than the boundary value at the compressive surface for this model $(n\sigma_y)$. Then, a smaller n value or higher imposed strains would lead to a different behavior.

1.2.2 Load Case II

For case II, an uniaxial load at first increment $\Delta\bar{\sigma}$ followed by two biaxial loads, the selected effective stress increments are shown in table (4).

Increment	$\Delta\bar{\sigma}_1$ [Pa]	$\Delta\bar{\sigma}_2$ [Pa]
1	350	0
2	-950	-950
3	1200	1200

Table 4: Case II: Effective principal stresses increments.

Case II: Tension Only model (T)

The resultant evolution of damage surface ($f(\sigma, q)$) and damage variable q(t) for the Tension model (T) are shown in Figure (6). In this case, both σ_1 and σ_2 are changing, thus both stress strain space should be analyzed. Here Poisson effects takes place, and effects can be observed in the stress-strain state.

5
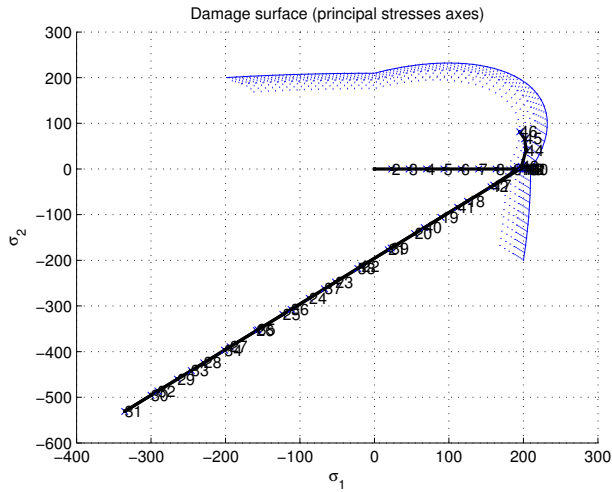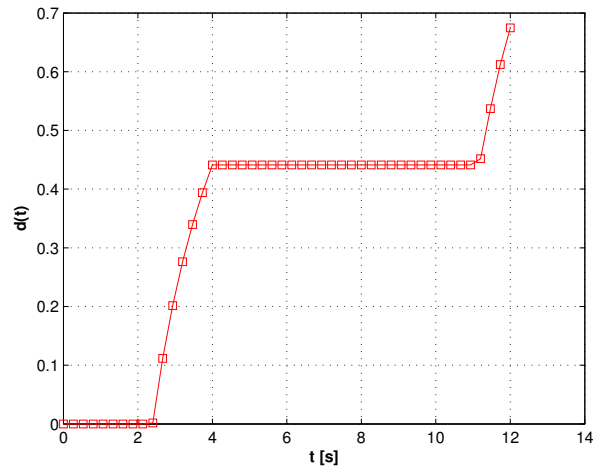
(a)   (b)

Figure 6: Case II (T): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

The loading path at stress-strain space is shown in Figure (7).
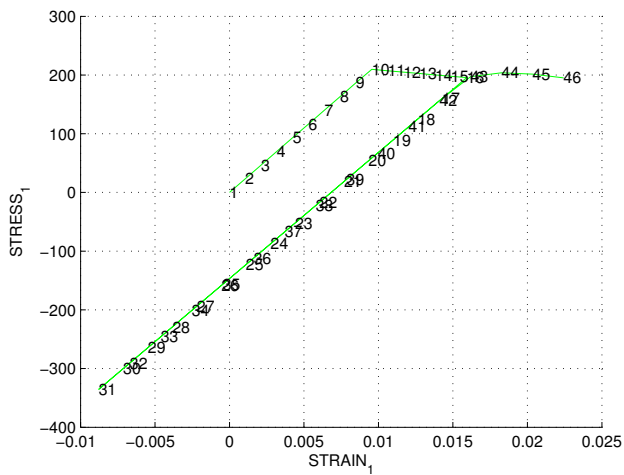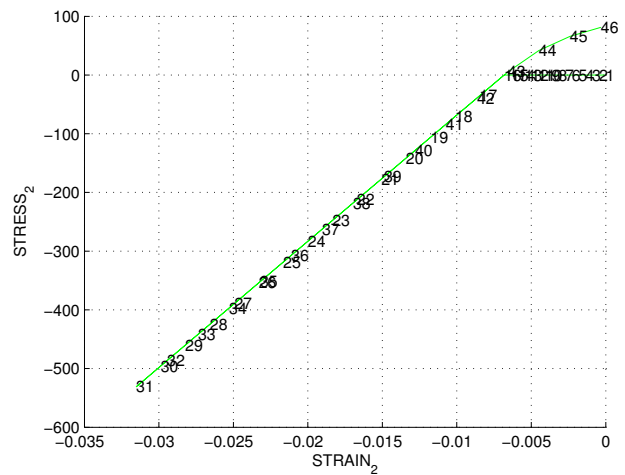
(a)   (b)

Figure 7: Case II (T): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

As in load Case I (for both T and N models), the material undergoes a elastic load in the first increment $\Delta\bar{\sigma}_1^{(1)}$ until the yield stress σ_y is reached, as seen in Figure (7) (a). Until this point σ_2 is zero, as seen in Figure (7) (b). From this point the material undergoes a biaxial tensile unloading/compressive loading ($\Delta\bar{\sigma}_1^{(2)}$,$\Delta\bar{\sigma}_2^{(2)}$) in a elastic way, as the norm never reaches the yield surface border.

The Poisson effect is noticed in the ϵ_1, which is smaller than the value found in load Case I, Figures (3) and (5). As σ_1 and σ_2 are both compressive, the ϵ_1 is constrained in the negative direction as, due to the nonzero ν value, σ_2 tends to cause a elongation in the positve $\hat{e}_1$ direction . The opposite is also true for ϵ_2 and σ_1.

The third increment is a biaxial compressive unloading/tensile loading ($\Delta\bar{\sigma}_1^{(3)}$,$\Delta\bar{\sigma}_2^{(3)}$) and elastic until the yield surface is reached again, and damage is developed, as seen in Figure (6) (a) and (b) respectively.

6

Case II: Non-Symmetric Model (N)

The resultant evolution of damage surface ($f(\sigma, q)$) and damage variable q(t) for the Non-Symmetric model (N) are shown in Figure (8).

For this case, the Non-Symetric Model (N) provides a limit for the compressive stresses norm, differently form the tension model (T). Thus , the difference in the bahavior here, lies in the fact that during the end of the second increment, a biaxial tensile unloading/compressive loading ($\Delta\bar{\sigma}_1^{(2)}$,$\Delta\bar{\sigma}_2^{(2)}$), the stresses norm reaches the yield surface, and damage is devloped, as seen in Figures (8) and (9).

Again, the third increment a biaxial compressive unloading/tensile loading ($\Delta\bar{\sigma}_1^{(3)}$,$\Delta\bar{\sigma}_2^{(3)}$) is elastic until the yield surface is reached, as in previous case.

Figure 8: Case II (N): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

The loading path at stress-strain space is shown in Figure (9).
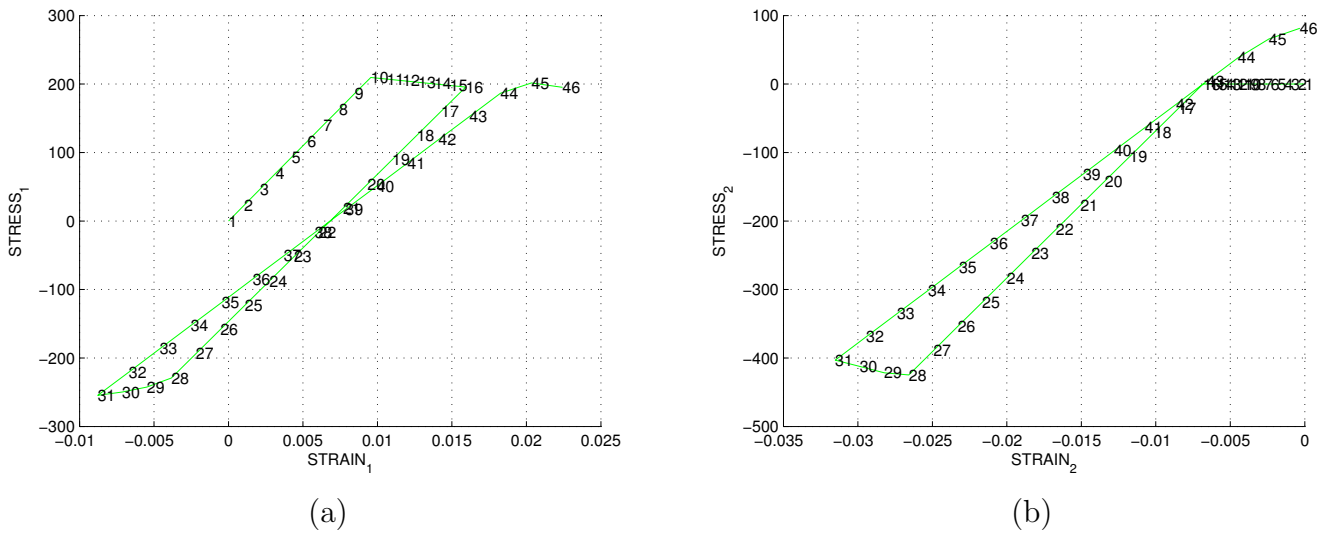
Figure 9: Case II (N): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

7

1.2.3 Load Case III

For case III, an biaxial uniform load with same effective stress increments as Case I, as shown in table (5).

| Increment	$\Delta\bar{\sigma}_1$ [Pa]	$\Delta\bar{\sigma}_2$ [Pa]
1	350	350
2	-950	-950
3	1200	1200

Table 5: Effective principal stresses increments

This load case is a especial one. Besides presenting biaxial as load Case II, here, the effective loads $\Delta\bar{\sigma}_1$ and $\Delta\bar{\sigma}_2$ are symmetric, in the sense that they have the same values over the entire analysis. This implies the load path at the stress space is located in a 45° axis in the stress plane.

Even though the stress norm for this load are higher than in load Case I, the Poisson effects generates an elliptical surface with major axis in the 45° direction, which compensates this higher stress, leading Case III to a similar behavior as Case II for both damage models T and N, where only elastic behavior is present in the compressive loads.

However, for a Poisson $\nu = 0$ the damage surface would be a circle and consequently, for this case load, damage would happen during the compressive increment load $(\Delta\bar{\sigma}_1^{(2)}, \Delta\bar{\sigma}_2^{(2)})$.

As in load Case one, the behavior for both models T and N are similar, as seen in Figures (10) to (13). Further, again the Poisson effect is noticed, as it can be seen in Figures (11) and (12), the strains ϵ_1 and ϵ_2, at the first increment $(\Delta\bar{\sigma}_1^{(1)}, \Delta\bar{\sigma}_2^{(1)})$, are around half the value of the strain in load Case I, Figures (3) and (5), for the same yield tension σ_1 and σ_2. This is a apparent *"stiffer"* behavior which is in accordance with the balance of internal strain energy.

Case III: Tension Only model (T)

The resultant evolution of damage surface $(f(\sigma, q))$ and damage variable q(t) for the Tension model (T) are shown in Figure (10).
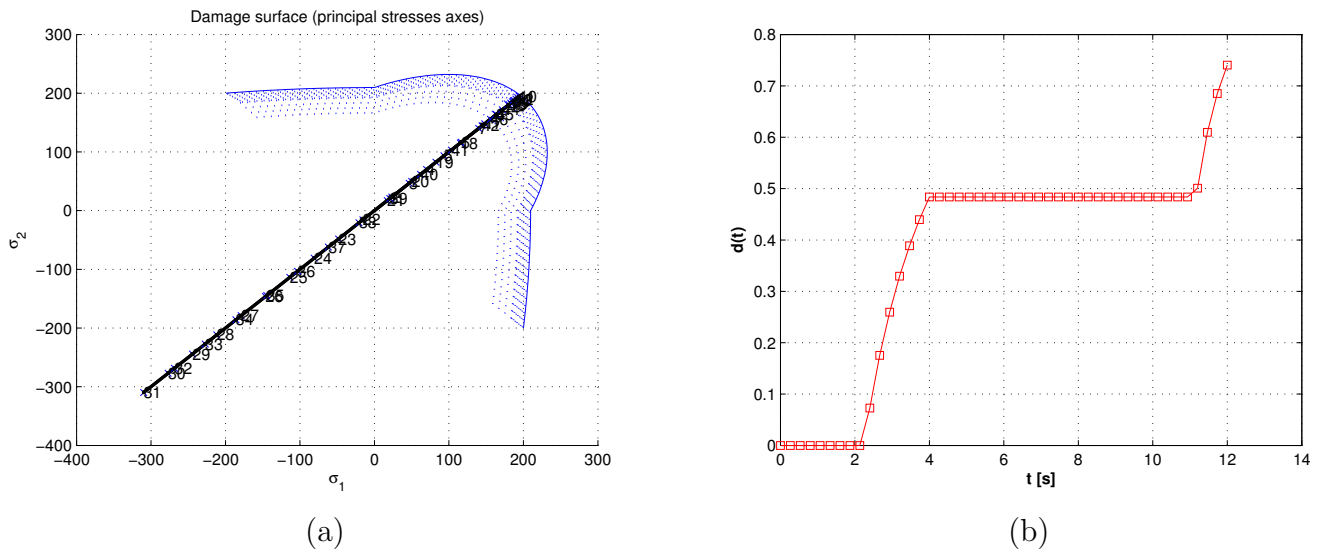
Figure 10: Case III (T): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

The loading path at stress-strain space is shown in Figure (11).

8
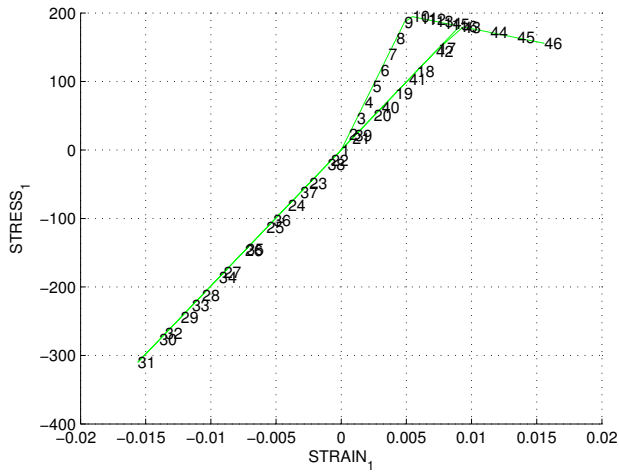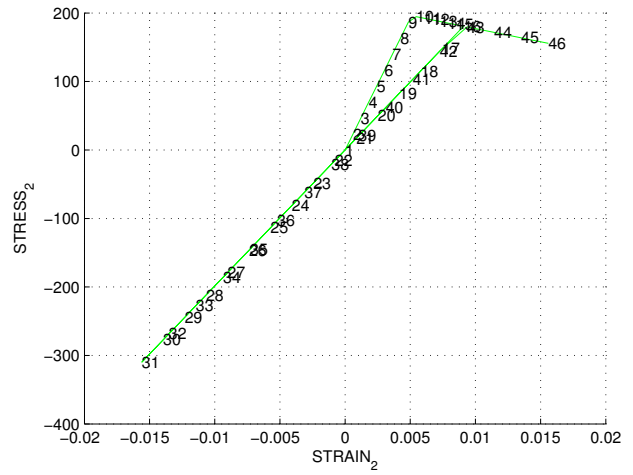
Figure 11: Case III (T): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

Case III: Non-Symmetric Model (N)

The resultant evolution of damage surface ($f(\sigma, q)$) and damage variable q(t) for the Non-Symmetric model (N) are shown in Figure (12).
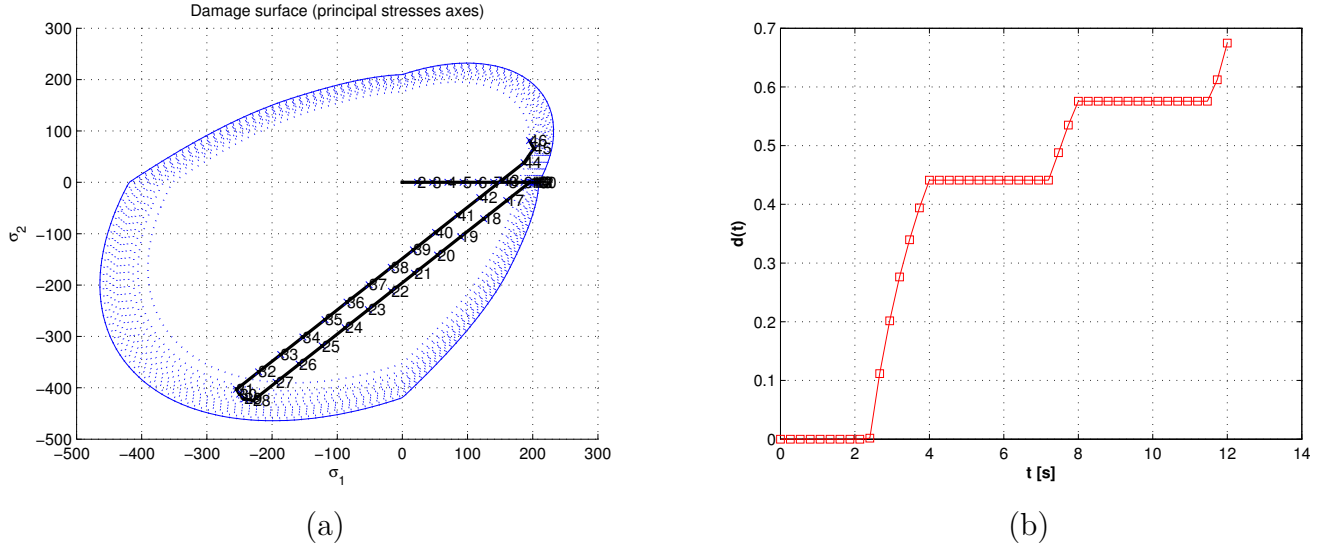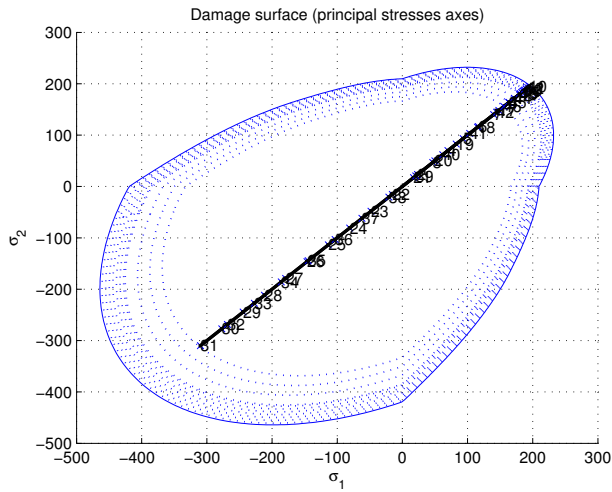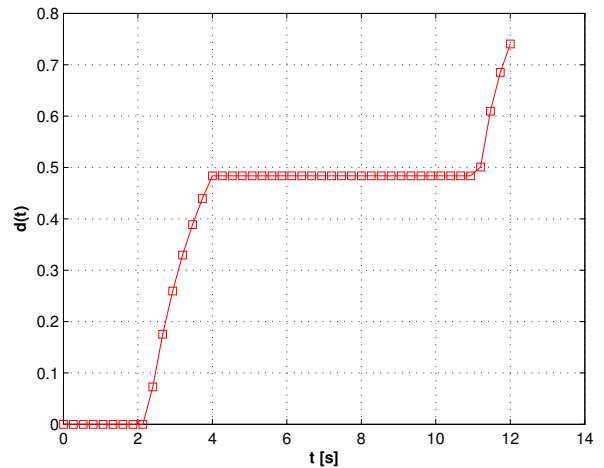

Figure 12: Case III (N): (a) Damage Surface Evolution;(b) Damage Variable Evolution.

The loading path at stress-strain space is shown in Figure (13).

9

(a)

(b)
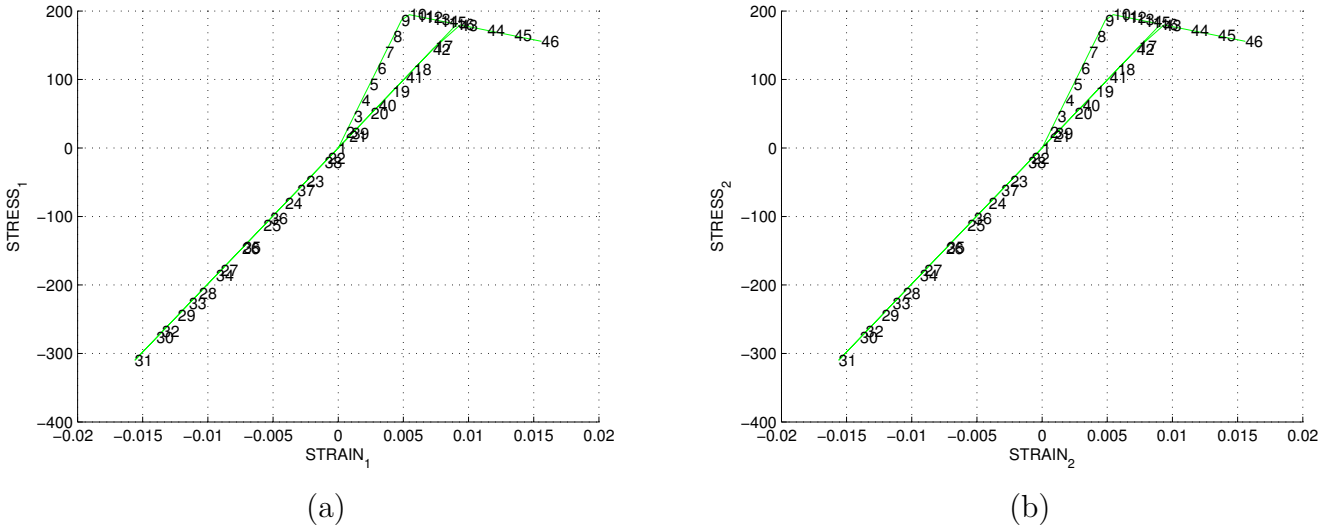
Figure 13: Case III (N): Variation of principal stress in terms of principal strain: (a) σ_1;(b) σ_2.

2 Part II : Rate Dependent Models

In this section the effects of the viscosity (η) and strain rate ($\dot{\epsilon}$) in the stress-strain response of rate dependent model is evaluated. Also, the effect of α coefficient, used in time integration numerical method, on stress-strain response and C_{11} component of tangent and algorithmic constitutive operators is discussed. Only linear softening model combined with Symmetric Damage Model (S) and uniaxial load will be considered for all tests. The following parameters in Table 6 will be constant for all analysis from here.

$\sigma_{\mathbf{y}}$ [Pa]	$\mathbf{E}$ [Pa]	ν[-]	$\mathbf{H}$ [-]	$\mathbf{q}_\infty$
200	2.0E5	0.3	-0.1	10^{-6}

Table 6: Material properties and effective stress increments

Same effective stress increment $\mathbf{\Delta\bar{\sigma}_1}$ and time length will be used for each increment segment in order to ensure a constant strain rate $\dot{\epsilon}$ over the analysis time domain.

2.1 Viscosity Effect

For this test, apart from the data in table (4), the effective stress increments will be set as $\mathbf{\Delta\bar{\sigma}_1}^1 = \mathbf{\Delta\bar{\sigma}_1}^2 = \mathbf{\Delta\bar{\sigma}_1}^3 = 500$ [Pa]. The total time is set to 12s with 30 integration points, leading to $\mathbf{\Delta t} = 0.4s$ and $\dot{\epsilon} = 0.0057s^{-1}$. Also the constant alpha will be $\alpha = 0.5$ (Crank-Nicholson), except for $\eta = 0$ where $\alpha = 1$ (Implicit).

As expected the higher the viscosity value, the lower is the rate of change in the internal variable $\dot{r}$ and consequently damage will increase in a lower rate, thus, the higher will be the maximum stress which can be located outside of the yield surface.
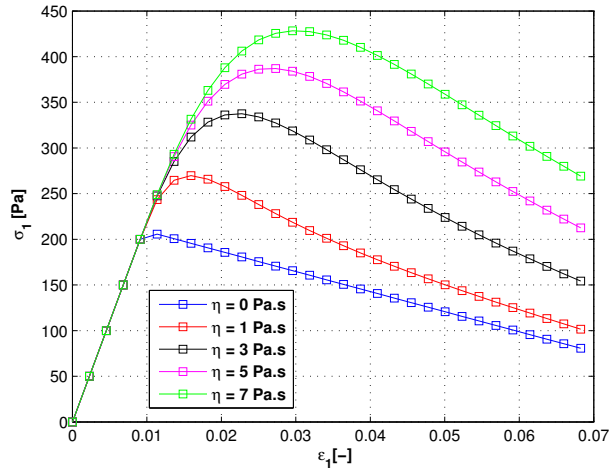
Figure 14: Effect of viscosity (η) on stress-strain response.

2.2 Strain Rate Effect

For this test, apart from the data in table (4), the effective stress increments will also be set as $\boldsymbol{\Delta\bar{\sigma}_1}^{\mathbf{1}} = \boldsymbol{\Delta\bar{\sigma}_1}^{\mathbf{2}} = \boldsymbol{\Delta\bar{\sigma}_1}^{\mathbf{3}} = 500$ [Pa] and 30 integration points will be used. However, here the total time will be changed in order to change the strain rate ($\dot{\epsilon}$). Also the constant alfa will be $\alpha = 0.5$ (Crank-Nicholson). The viscosity will be $\eta = 3$ [Pa.s] for all cases. The total time (T) and correspondent strain rate ($\dot{\epsilon}$) for each case is shown in table (7).

	1	**2**	**3**	**4**	**5**
$\mathbf{T}$[s]	48	24	12	6	3
$\dot{\epsilon}[s^{-1}]$	0.0014	0.0028	0.0057	0.0114	0.0227

Table 7: Total time and correspondent strain rates.
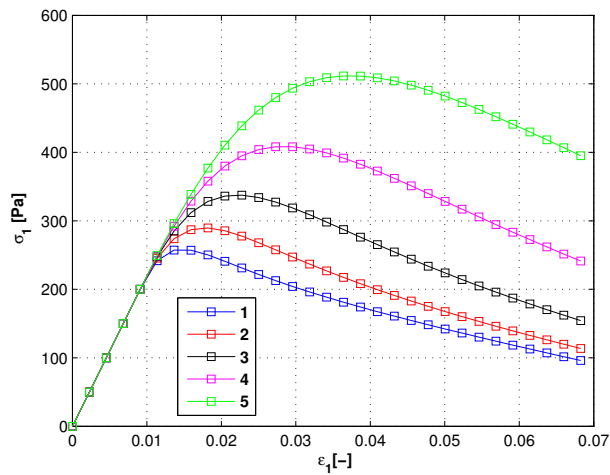
Figure 15: Effect of strain rate ($\dot{\epsilon}[s^{-1}]$) on stress-strain response.

The strain rate effect is similar to the effect of the viscosity (η), as can be observed in Figure (15). Here the effect relates to the fact that as a result of the modeling, the stress tensor σ has a explicit

dependence on time, then when strain rate $\dot{\epsilon}$ is changed not only the $\dot{\sigma}$ changes directly but also σ. Thus, when the rate is higher, higher stresses are reached (higher the peak) before the yield surface reaches the current stress state.

2.3 Integration Parameter Effect

The effect of α in the numerical integration result will be evaluated for $\mathbf{\Delta t} = 0.4s$ and $\mathbf{\Delta t} = 0.08s$. The value of $\mathbf{\Delta t}$ here is changed by changing the total number of integration points from 30 to 150. Effective stress increments will be set as $\mathbf{\Delta\bar{\sigma}_1}^1 = \mathbf{\Delta\bar{\sigma}_1}^2 = \mathbf{\Delta\bar{\sigma}_1}^3 = 500$ [Pa] and 12s as total time. The viscosity will be $\eta = 3$ [Pa.s] for all cases. For other parameter, refer again to Table (6).
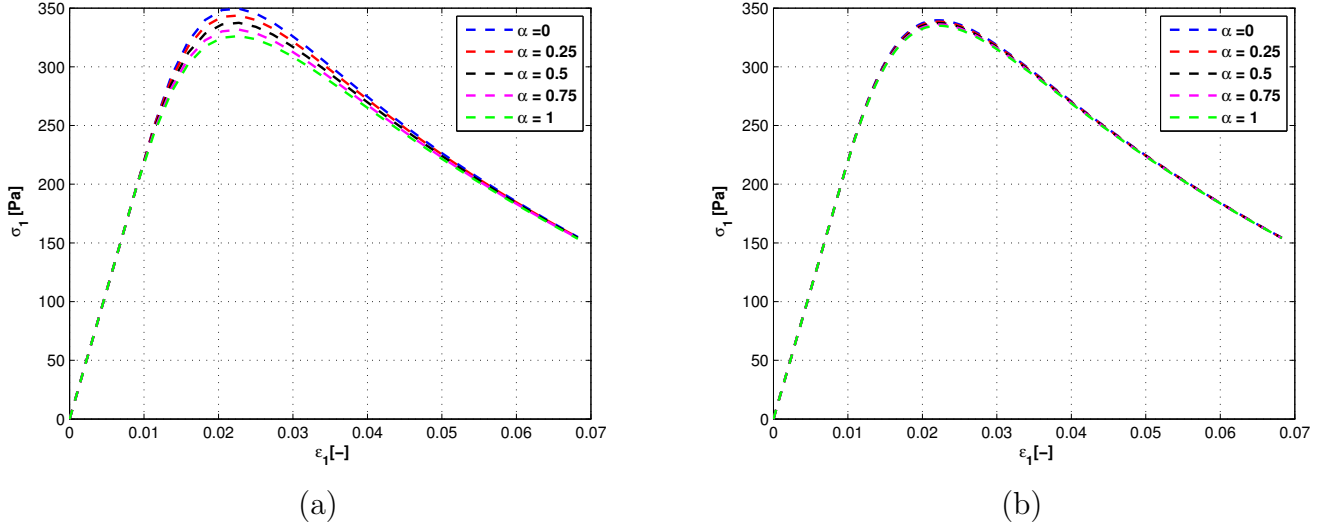
Figure 16: Effect of α on stress-strain response: (a) $\Delta t = 0.4s$; (b) $\Delta t = 0.08s$.

It can be observed that as $\Delta t \rightarrow 0$, the solution for all values of α converges to the same one. Where the solution for 4 $\alpha = 0.5$ is the second order ($\mathcal{O}(\Delta t^2)$) accurate one.

The effect of the α in the first component C_{11} of the constitutive tensors *Analytic* (C_{11}^{anl}) and *Algorithmic* (C_{11}^{alg}) are given in Figure (17) (a) and (b).
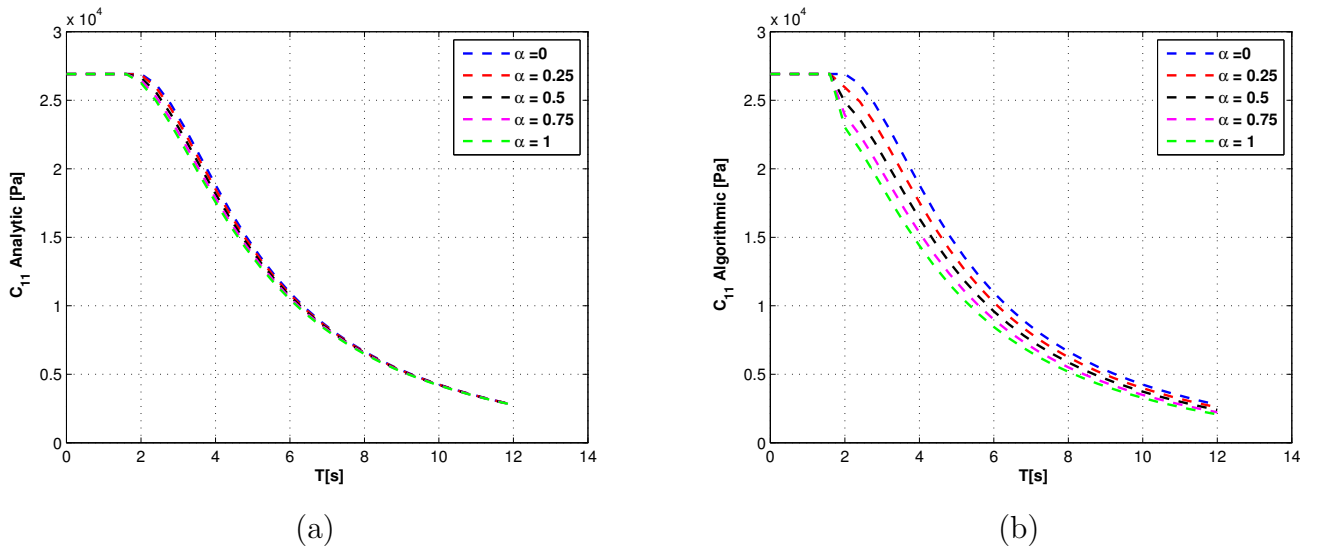
Figure 17: Effect of α on C_{11} component: (a) Analytic; (b) Algorithmic.

Figure (18) shows the effect of α in the difference $\Delta C_{11} = C_{11}^{anl} - C_{11}^{alg}$ for the whole time domain range.
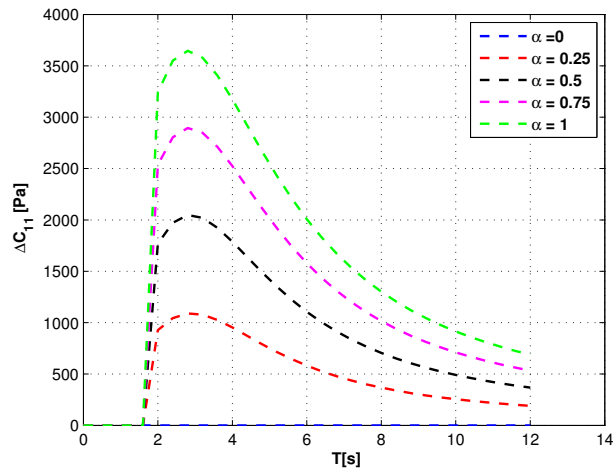
Figure 18: Effect of α on the difference in C_{11} component

It can be noticed that $\Delta C_{11} \to 0$ as $\alpha \to 0$. Also the difference is reduced as $t \to \infty$, as expected.

3 – Annex – MatLab Modified Routines

```matlab
clc
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program for modelling damage model
% (Elemental gauss point level)
% -----------------
% Developed by J.A. Hdez Ortega
% 20-May-2007, Universidad Politécnica de Cataluña
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%profile on
% ------------------------
% ***************
% INPUTS
% ***************

% YOUNG's MODULUS
% ---------------
YOUNG_M = 20000 ;
% Poisson's coefficient
% ----------------------
POISSON = 0.3 ;
% Hardening/softening modulus
% ----------------------------
HARDSOFT_MOD =  -0.1; %FOR EXPONETIAL CASE H is q'(r0)= Ho = H
% Yield stress
% ------------
YIELD_STRESS = 200 ;
% Problem type  TP = {'PLANE STRESS','PLANE STRAIN','3D'}
% ----------------------- = 1            =2          =3
% ------------
ntype= 2 ;
% Model    PTC = {'SYMMETRIC','TENSION','NON-SYMMETRIC'} ;
%                   = 1          = 2          = 3
% ---------------------------------------------------
MDtype = 1;
% Ratio compression strength / tension strength
% ---------------------------------------------
n = 2 ;
% SOFTENING/HARDENING TYPE
% -----------------------
HARDTYPE = 'LINEAR' ; %{LINEAR,EXPONENTIAL}
% VISCOUS/INVISCID
% -----------------------
VISCOUS = 'YES' ;
% Viscous coefficient ----
% ------------------------
eta = 3;
% TimeTotal (initial = 0) ----
% ------------------------
TimeTotal = 12;
% Integration coefficient ALPHA
% ------------------------
ALPHA_COEFF = 1;
% Points --------------------------
% ---------------------------------
nloadstates = 3 ;
SIGMAP = zeros(nloadstates,2) ;
SIGMAP(1,:) =[500 0];
SIGMAP(2,:) =[1000 0];
```

```matlab
SIGMAP(3,:) =[1500 0];
% Number of time increments for each load state
% --------------------------------------
istep = 10*ones(1,nloadstates) ;
% VARIABLES TO PLOT
vpx = 'STRAIN_1' ; % AVAILABLE OPTIONS: 'STRAIN_1', 'STRAIN_2'
%                   '|STRAIN_1|', '|STRAIN_2|'
% 'norm(STRAIN)', 'TIME'
vpy = 'STRESS_1';          % AVAILABLE OPTIONS: 'STRESS_1', 'STRESS_2'
%                   '|STRESS_1|', '|STRESS_2|'
% 'norm(STRESS)', 'TIME', 'DAMAGE VAR.','hardening variable (q)','damage variable (d)'
%  'internal variable (r)'

%  3) LABELPLOT{ivar}            --> Cell array with the label string for
%                                 variables of "varplot"
%
LABELPLOT = {'hardening variable (q)','internal variable (r)','damage variable (d)'};


%%%%%%%%%%%%%%%%%%%%55 END INPUTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Plot Initial Damage Surface and effective stress path
strain_history = PlotIniSurf(YOUNG_M,POISSON,YIELD_STRESS,SIGMAP,ntype,MDtype,n,istep);


E       = YOUNG_M       ;
nu      = POISSON       ;
sigma_u = YIELD_STRESS ;

switch  HARDTYPE
    case 'LINEAR'
        hard_type = 0  ;
    otherwise
        hard_type = 1  ;
end
switch VISCOUS
    case 'YES'
        viscpr = 1      ;
    otherwise
        viscpr = 0      ;
end

Eprop   = [E nu HARDSOFT_MOD sigma_u hard_type viscpr eta ALPHA_COEFF]             ;
% DAMAGE MODEL
% ------------
[sigma_v,vartoplot,LABELPLOT_out,TIMEVECTOR,C11t,C11a]=damage_main(Eprop,ntype,istep,strain_
history,MDtype,n,TimeTotal);

try; LABELPLOT;catch;LABELPLOT = LABELPLOT_out ; end ;
% PLOTTING
% -------
ncolores = 3 ;
colores =  ColoresMatrix(ncolores);
markers = MarkerMatrix(ncolores) ;
hplotLLL = [] ;
for i = 2:length(sigma_v)
    stress_eig  = sigma_v{i} ; %eigs(sigma_v{i}) ;
    tstress_eig = sigma_v{i-1}; %eigs(sigma_v{i-1}) ;
    hplotLLL(end+1)    =    plot([tstress_eig(1,1)    stress_eig(1,1)     ],[tstress_eig(2,2)
stress_eig(2,2)],'LineWidth',2,'color',colores(1,:),'Marker',markers{1},'MarkerSize',2);
    plot(stress_eig(1,1),stress_eig(2,2),'bx')
```

```matlab
        text(stress_eig(1,1),stress_eig(2,2),num2str(i))
end
DATA.sigma_v    = sigma_v     ;
DATA.vartoplot  = vartoplot   ;
DATA.LABELPLOT  = LABELPLOT    ;
DATA.TIMEVECTOR = TIMEVECTOR   ;
DATA.strain = strain_history ;


plotcurvesNEW(DATA,vpx,vpy,LABELPLOT,vartoplot) ;
```

```matlab
function
[sigma_v,vartoplot,LABELPLOT,TIMEVECTOR,C11t,C11a]=damage_main(Eprop,ntype,istep,strain,MDty
pe,n,TimeTotal)
global hplotSURF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CONTINUUM DAMAGE MODEL
% ----------------------
% Given the almansi 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
%
%
% n           = Ratio compression/tension strength (dummy if MDtype is different from 3)
%
```

```
% TimeTotal  = Interval length
%
%   OUTPUTS <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
%   ----------------------------------------------------------------
%   1) sigma_v{itime}(icomp,jcomp)   --> Component (icomp,jcomp) of the cauchy
%                                        stress tensor at step "itime"
%                                        REMARK: sigma_v is a type of
%                                        variable called "cell array".
%
%
%   2) vartoplot{itime}                 --> Cell array containing variables one wishes to plot
%                                          --------------------------------------
%    vartoplot{itime}(1) =   Hardening variable (q)
%    vartoplot{itime}(2) =   Internal variable (r)%
%
%
%   3) LABELPLOT{ivar}                --> Cell array with the label string for
%                                        variables of "varplot"
%
%          LABELPLOT{1} => 'hardening variable (q)'
%          LABELPLOT{2} => 'internal variable'
%
%
%   4) TIME VECTOR  - >
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% SET LABEL OF "vartoplot" variables  (it may be defined also outside this function)
% --------------------------------
 LABELPLOT = {'hardening variable (q)','internal variable'};

E = Eprop(1) ;
nu = Eprop(2) ;
viscpr = Eprop(6) ;
sigma_u = Eprop(4);
alpha = Eprop(8);

if ntype == 1
    menu('PLANE STRESS has not been implemented yet','STOP');
    error('OPTION NOT AVAILABLE')
elseif ntype == 3
    menu('3-DIMENSIONAL PROBLEM has not been implemented yet','STOP');
    error('OPTION NOT AVAILABLE')
else
    mstrain = 4    ;
    mhist   = 6    ;
end
%SETTING VISCOSITY TO ZERO FOR RATE INDEPENDENT ANALISYS
if viscpr ~= 1
    Eprop(7) = 0;
else
end
totalstep = sum(istep) ;
% INITIALIZING GLOBAL CELL ARRAYS
% ------------------------------
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);
```

```matlab
% Initz.
% -----
% Strain vector
% -------------
eps_n  = zeros(mstrain,1); %STRAIN STATE AT t_n
eps_n1  = zeros(mstrain,1);%STRAIN STATE AT t_n+1
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)];
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)
%INITIALIZATION C11 COMPONENT OF TANGENT AND ALGORITHIM CONSTITUTIVE TENSOR
C11a =  ce(1,1);
C11t = ce(1,1);


for  iload = 1:length(istep)
    % Load states
    for iloc = 1:istep(iload)
        i = i + 1 ;
        dt = delta_t(iload);
        TIMEVECTOR(i) = TIMEVECTOR(i-1)+ dt;
        % Total strain at step "i"
        % -----------------------
        eps_n = strain(i-1,:);                %STRAIN STATE at t_n
        eps_n1 = strain(i,:);                 %STRAIN STATE at t_n+1
        %***********************************************************************
        %*       DAMAGE MODEL
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        [sigma_n1,hvar_n,aux_var,Ct,Ca]                                     =
rmap_dano1(eps_n,eps_n1,hvar_n,Eprop,ce,MDtype,n,dt);
        % 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)                       ;
        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 COMPONENT OF TANGENT AND ALGORITHIM CONSTITUTIVE TENSOR
        C11a(i) = Ca(1,1);
        C11t(i) = Ct(1,1);
        % 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)
```

```matlab
        vartoplot{i}(3) = 1-hvar_n(6)/hvar_n(5)  ; %  Damage variable (d)
    end
end


function
 [sigma_n1,hvar_n1,aux_var,Ct,Ca] =rmap_dano1(eps_n,eps_n1,hvar_n,Eprop,ce,MDtype,n,dt)


%***********************************************************************************
%*                                          *
%*          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);
eta = Eprop(7);
alpha = Eprop(8);
%***********************************************************************************
%***********************************************************************************
%*        initializing                                              %*
 r0 = sigma_u/sqrt(E);
 zero_q= 1.d-6*r0;
 %SUPERIOR LIMIT FOR q infinity
 inf_q = 1.d6;
 %EXPONETIAL MODEL CONSTANT EXPOENT
 A = 1;
% if(r_n<=0.d0)
%      r_n=r0;
%      q_n=r0;
% end
%*****************************************************************************
%*****************************************************************************
%*        Damage surface
[rtrialn rtrialn1] = Modelos_de_dano1 (MDtype,ce,eps_n,eps_n1,n);
rtrial = alpha*rtrialn1+(1-alpha)*rtrialn; %STRAIN RADIUS AT t_n+alpha
%*****************************************************************************
%*****************************************************************************
%*   Ver el Estado de Carga
%*   ------->    fload=0 : elastic unload
```

```matlab
%*   ------->    fload=1 : damage (compute algorithmic constitutive tensor)
fload=0;

if(rtrial > r_n)
    %*   Loading
    fload=1;
    %THETA INTEGRATION METHOD
    r_n1 = (r_n*(eta-dt*(1-alpha))+dt*rtrial)/(eta+alpha*dt);
    delta_r = r_n1-r_n;
    if hard_type == 0 %LINEAR SOFTENING/HARDENING LAW
        q_n1= q_n+ H*delta_r;
        if(q_n1<zero_q) %SOFTENING
        q_n1=zero_q;
        elseif (q_n1 > inf_q)%HARDENING
        q_n1=inf_q;
        end
        Hn1 = H; %H PARAMETER at t_n+1
    else %EXPONETIAL SOFTENING/HARDENING LAW
        Ho = H; %H  PARAMETER at t_0
        inf_q = (Ho/A+1)*r0; %
        q_n1 = q_n +(inf_q-r0)*(exp(A*(1-r_n/r0))-exp(A*(1-r_n1/r0)));
        Hn1 = A*((inf_q-r0)/r0)*exp(A*(1-rn1/r0)); %H PARAMETER at t_n+1
    end
    dano_n1   = 1.d0-(q_n1/r_n1);
    %CONSTITUTIVE TENSORS
    Ca = (1.d0-dano_n1)*ce + ((alpha*dt)/(eta+alpha*dt))*(1/rtrialn1)*...
        ((Hn1*r_n1-q_n1)/(r_n1^2))*((ce*eps_n1')*(ce*eps_n1')'));
    Ct = (1.d0-dano_n1)*ce;
else
    %*     Elastic load/unload
    fload=0;
    r_n1= r_n  ;
    q_n1= q_n  ;
    dano_n1   = 1.d0-(q_n1/r_n1);
    %CONSTITUTIVE TENSORS
    Ca = (1.d0-dano_n1)*ce;
    Ct = (1.d0-dano_n1)*ce;
end
%  Computing stress
%  ***************
sigma_n1  = (1.d0-dano_n1)*ce*eps_n1';
%TANGENT CONSTITUTIVE TENSOR
%***************************************************************************
%***************************************************************************
%* 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;
%***************************************************************************
```

```matlab
function [rtrialn rtrialn1] = Modelos_de_dano1 (MDtype,ce,eps_n,eps_n1,n)
%******************************************************************************
%*          Defining damage criterion surface                             %*
%*                                                                        %*
%*
%*                          MDtype=  1      : SYMMETRIC                    %*
%*                          MDtype=  2      : ONLY TENSION                 %*
%*                          MDtype=  3      : NON-SYMMETRIC                %*
%*                                                                        %*
%*                                                                        %*
%* OUTPUT:                                                                 %*
%*                          rtrial                                        %*
%******************************************************************************
%******************************************************************************
if (MDtype==1)        %SYMMETRIC
rtrialn = sqrt(eps_n*ce*eps_n');       %STRAIN RADIUS at t_n
rtrialn1 = sqrt(eps_n1*ce*eps_n1'); %STRAIN RADIUS at t_n+1

elseif (MDtype==2)   %ONLY TENSION
% ELASTIC STRESS VECOTOR
sigma_n =ce*eps_n';
sigma_n1 =ce*eps_n1';
% MACAULLEY BRACKET
sigma_n = sigma_n.*(sigma_n > 0);
sigma_n1 = sigma_n1.*(sigma_n1 > 0);
%DAMAGE SURFACE (TRIAL STATES)
rtrialn = sqrt(sigma_n'*eps_n');       %STRAIN RADIUS at t_n
rtrialn1 = sqrt(sigma_n1'*eps_n1'); %STRAIN RADIUS at t_n+1

elseif (MDtype==3)   %NON-SYMMETRIC
% ELASTIC STRESS VECTOR
sigma_n =ce*eps_n';
sigma_n1 =ce*eps_n1';
%THETA DEFINITION
%macaulley bracket
Mn = sigma_n.*(sigma_n > 0);
Mn1 = sigma_n1.*(sigma_n1 >0);
Smn = sum(Mn);
Smn1 = sum(Mn1);
Sn = sum(abs(sigma_n));
Sn1 = sum(abs(sigma_n1));
Thetan = Smn/Sn;
Thetan1 = Smn1/Sn1;
%DAMAGE SURFACE (TRIAL STATES)
Kn = Thetan+(1-Thetan)/n;
Kn1 = Thetan1+(1-Thetan1)/n;
rtrialn = Kn*sqrt(eps_n*ce*eps_n');     %STRAIN RADIUS at t_n
rtrialn1 = Kn1*sqrt(eps_n1*ce*eps_n1');%STRAIN RADIUS at t_n+1
end
%******************************************************************************
return
```

```matlab
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----->tensiï¿½n ï¿½ltima       %*
%*                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  % SYMMETRIC 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))]*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  % COMPRESSION ONLY (TENSION) MODEL
    tetha=[-0.5*pi/2:0.01:0.75*pi];
    %*************************************************************************
    %* RADIUS
    D=size(tetha);                          %*  Range
    m1=cos(tetha);                          %*
    m2=sin(tetha);                          %*
    Contador=D(1,2);                        %*
```

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

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

    for i=1:Contador
        S = [m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
        %MACAULLEY BRACKET
        M = S.*(S>0);
        %SUMMATTION NUMERATOR
        Sm = sum(M);
        %SUMMATTION DENOMINTOR
        Smm = sum(abs(S));
        %THETA
        Theta = Sm/Smm;
        K = Theta+(1-Theta)/n;
        radio(i)= q/(K*sqrt(S*ce_inv*S'));
        s1(i)=radio(i)*m1(i);
        s2(i)=radio(i)*m2(i);
    end
    hplot =plot(s1,s2,tipo_linea);
end
%**********************************************************************************
%**********************************************************************************
return
```