

COMPUTATIONAL SOLID MECHANICS
ASSIGNMENT 1
Constitutive Damage Models

Author: Cristina García Albela
MsC in Computational Mechanics

Index

1. Introduction. Isotropic damage model
2. Rate independent model
 - 2.1 Introduction to the case
 - 2.2 Results and conclusions
 - 1) Only tension model - Uniaxial tensile loading/unloading
 - 2) Only tension model - Uniaxial tensile loading, biaxial tensile loading/unloading
 - 3) Only tension model - Biaxial tensile loading/unloading
 - 4) Non symmetric model - Uniaxial tensile loading/unloading
 - 5) Non symmetric model - Uniaxial tensile loading, biaxial tensile loading/unloading
 - 6) Non tension model - Biaxial tensile loading/unloading
3. Rate dependent model
 - 3.1 Introduction to the case
 - 3.2 Results and conclusions
 - 1) Stress space path - Biaxial tensile loading/unloading
 - 2) Viscosity effects - Uniaxial tensile loading/unloading
 - 3) Time effect - Uniaxial tensile loading/unloading
 - 4) Alpha effects - Constitutive operators

1. Introduction. Isotropic Damage Model

Along the years, Continuum Damage Mechanics has been used to model materials characterized by loss of stiffness and also to simulate materials which are characterized by having irreversible material degradation. This mentioned degradation of the mechanical material properties can be described as processes in which the initiation and propagation of micro defects take place. Since the later introduction of effective stress and other successful theories, CDM theory has become an important tool to solve different kinds of problems, based on irreversible thermodynamic processes. One of the typical examples in which CDM is implemented for simulation are materials whose mechanical properties are degrading because of small cracks that propagate along the material during loading.

This kind of materials are going to be studied under different stress conditions and simulated, taking into account the isotropic characteristic, meaning that mechanical and thermal properties are the same in all directions.

Main variables

Given a material point under a stress state in which micro cracks appear and taking into account the force balance on it, the following main variables must be defined in order to work with Damage model.

- Apparent (σ) and effective stress ($\bar{\sigma}$). Once a micro crack appears, effective stress can be defined as the stress acting on the remaining material section, whereas the apparent stress would be the total stress applied by the user.
- Damage variable (d). Denoted as the damaged section with respect to the total section. Having in mind the constitutive equation for stress in damage model, it is verified that once the material has suffered damage, it will be permanent, characterizing an irreversible process.
- Internal variable (r). In its undamaged state the material is characterized by the initial r_0 . Remark that internal variables are those non-observables by the user.
- Hardening/Softening variable (q). It is a function of r , considering that the relationship between both can be linear or exponential.
- Stress/Strain tensor energy norm ($\tau_\sigma, \tau_\varepsilon$). They are surface equations that characterize the stress state at the current point. Different norms will be defined to describe materials different behaviours. Each material starts to fail when the norm exceeds the value.

Continuum damage model algorithm

From a computational point of view continuum damage models are very attractive. They present simple algorithms that can be generally summarized in the following steps.

Having for a time t the variables ε_t, r_t and ε_{t+1}

- 1) Computation of effective stress ($\bar{\sigma}_{t+1}$) and strain energy norm ($\tau_{\varepsilon_{t+1}}$) at current time.
- 2) Comparison between current strain norm ($\tau_{\varepsilon_{t+1}}$) and previous internal variable (r_t) in order to identify the current state.
- 3) Compute the current internal variable (r_{t+1}), damage variable (d_{t+1}), stress state (σ_{t+1}) and tangent/algorithmic constitutive tangent/algorithmic operator ($\mathbb{C}_{tan,t+1}^{vd}, \mathbb{C}_{alg,t+1}^{vd}$).
- 4) Plot the new results.

2. Rate independent model

2.1 Introduction to the case

Defining the damage surface as the elastic material limit, the damage criterium for inviscid materials required that the current stress state must be always inside or on the damage surface. If the stress state is inside the damage surface, the material shows elastic behaviour (elastic loading or unloading). On the other hand, if it lies on the surface, inelastic damage state is defined with or without evolution of internal variable (pure loading and neutral loading/unloading respectably).

Different rate independent, plane strain case models are going to be studied and compared, in order to understand and check the correct behaviour of each of them. Starting at point $(\sigma_1, \sigma_2) = (0, 0)$, each example is going to be defined in terms of its corresponding effective stress increments.

Problem data and material properties

Material properties will be constant along each case $\begin{cases} E = 20000 \text{ N/m}^2 \\ \nu = 0.3 \\ \sigma_u = 200 \text{ N/m}^2 \end{cases}$

Initial internal variable $r_0 = \frac{\sigma_u}{\sqrt{E}} = 1.414$

Total time & Time steps/load state $t = 10; nsteps = 10$

2.2 Results and conclusions

1) Only tension model - Uniaxial tensile loading/unloading

Linear hardening variable $q \rightarrow H = 0.1$

Initial stress points, Case 1 $\begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (50, 0) \rightarrow \Delta\sigma_1^{(2)} = -10; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (400, 0) \rightarrow \Delta\sigma_1^{(3)} = 35; \Delta\sigma_2^{(3)} = 0 \end{cases}$

Initial stress points, Case 2 $\begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (250, 0) \rightarrow \Delta\sigma_1^{(1)} = 25; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (50, 0) \rightarrow \Delta\sigma_1^{(2)} = -20; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (400, 0) \rightarrow \Delta\sigma_1^{(3)} = 35; \Delta\sigma_2^{(3)} = 0 \end{cases}$

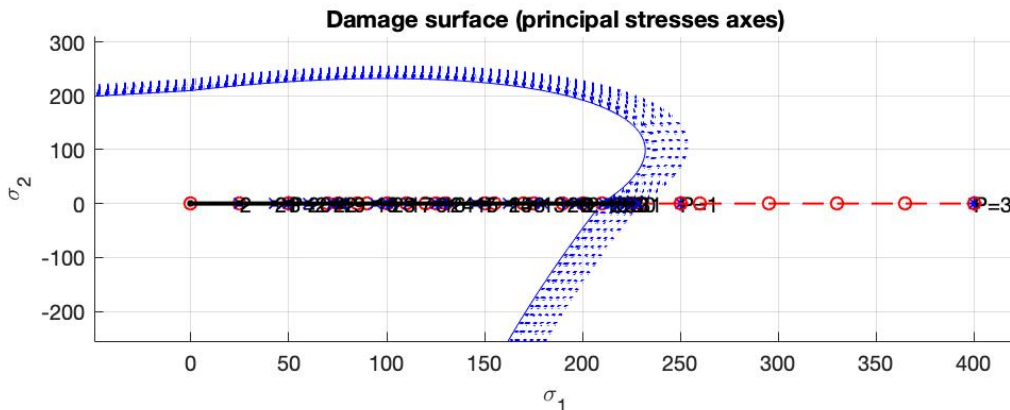


Figure 2. Stress space path – Case 2

At first look, both stress paths could be thought to be the same. Working with uniaxial tensile loading/unloading (just stress in first principal direction) it becomes difficult to appreciate each state point. Damage appears in both situations, meaning that at some point the strain norm will overcome the initial internal variable $r_0 = \sigma_u / \sqrt{E}$. For case 1 r_0 is not reached until $\sigma^{(3)}$ is applied, whereas for case 2 damage will appear under both stresses $\sigma^{(1)}$ and $\sigma^{(3)}$.

In front such a confused result, a tool to guaranteed, as theory states, that material behaviour is not the same will be damage variable/time graphic. For case 1 (Figure 3, black line), damage variable remains zero until step 25 whereas at case two (Figure 3, blue line), where $\sigma_1^{(1)} > \sigma_u$ a first increment on d appears between steps 9 and 10.

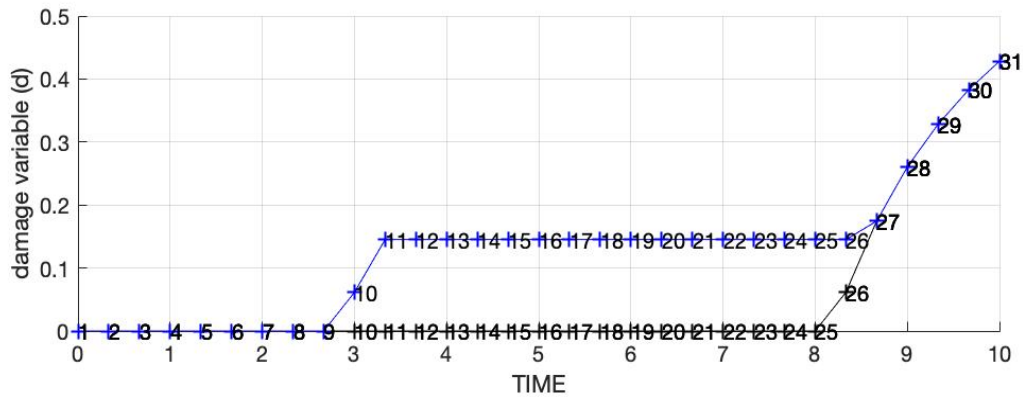


Figure 3. Damage variable vs Time graphic - Case 1

2) Only tension model - Uniaxial tensile loading, biaxial tensile loading/unloading

Linear hardening variable $q \rightarrow H = 0.1$

$$\text{Initial stress points, Case 1} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (250, 0) \rightarrow \Delta\sigma_1^{(1)} = 25; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (-200, -100) \rightarrow \Delta\sigma_1^{(2)} = -45; \Delta\sigma_2^{(2)} = -10 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (-100, 300) \rightarrow \Delta\sigma_1^{(3)} = 10; \Delta\sigma_2^{(3)} = 40 \end{cases}$$

$$\text{Initial stress points, Case 2} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (350, 0) \rightarrow \Delta\sigma_1^{(1)} = 35; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (-200, -100) \rightarrow \Delta\sigma_1^{(2)} = -55; \Delta\sigma_2^{(2)} = -10 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (-100, 300) \rightarrow \Delta\sigma_1^{(3)} = 10; \Delta\sigma_2^{(3)} = 40 \end{cases}$$

Here two similar loading paths, whose only difference appears in $\sigma^{(1)}$ value, are compared in order to better understand the fact that once damage appears in the material it is permanent and affects the following steps.

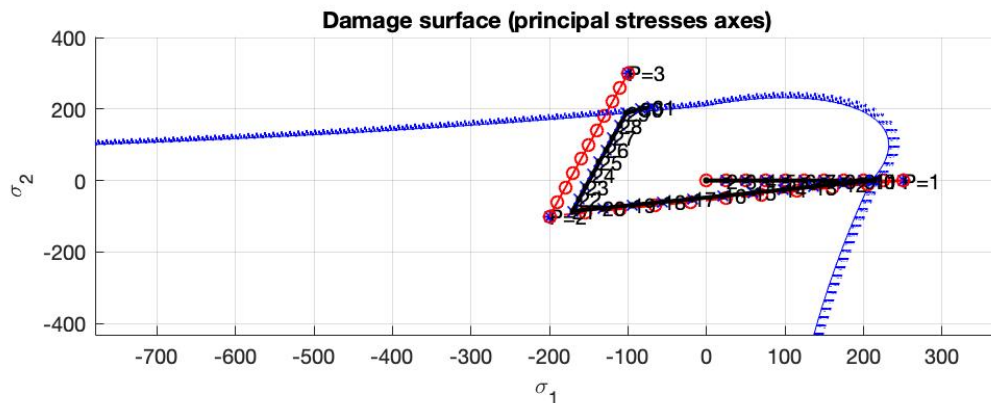


Figure 4. Stresses space path - Case 1

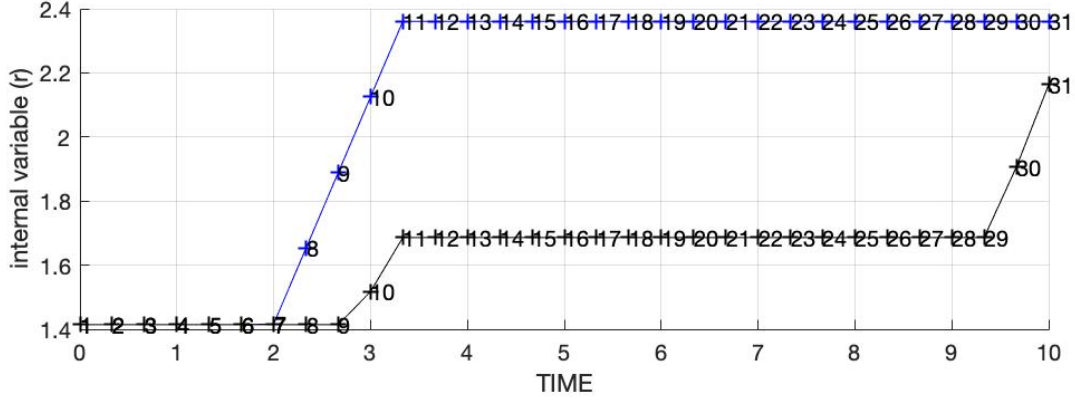


Figure 5. Internal variable evolution – Case 1 (black) and Case 2 (blue)

In both cases under $\sigma^{(1)}$ uniaxial tensile loading ($\sigma_1^{(1)} > \sigma_u$), the initial damage surface evolves (Figure 4, stress space path for Case 1 represented). Then, looking to r evolution (Figure 5) it is found that identical following $\sigma^{(2)}$ and $\sigma^{(3)}$ stresses do not produce the same effects on the material.

The highest $\sigma_1^{(1)}$ value at Case 2, affects to r, d and σ along the iterative process obtaining a larger damage surface at step 11. Once $\sigma^{(3)}$ is applied the strain norm never reaches the r_{11} value, remaining it constant until the end. At Case 1, damage appears at final time steps under the same applied $\sigma^{(3)}$ stress, due to the lower $\sigma^{(1)}$ and r_{11} values, so the strain norm reaches the internal variable once again.

3) Only tension model – Biaxial tensile loading/unloading

Case 1 - Linear hardening variable $q \rightarrow H = 0.1$

Case 2 - Exponential hardening variable $q \rightarrow q_{inf} = r_0 + (r_0 - q_{zeros})$; $A = 1$

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 150) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 15 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (100, 50) \rightarrow \Delta\sigma_1^{(2)} = -5; \Delta\sigma_2^{(2)} = -10 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (350, 100) \rightarrow \Delta\sigma_1^{(3)} = 25; \Delta\sigma_2^{(3)} = 5 \end{cases}$$

For same initial stress paths, exponential and linear hardening variable results are obtained. Obtaining stress space path (Figure 6) without the stress point evolution, the larger step increase on damage surface stands out compared with linear hardening case. Under inelastic loading, for q exponential relationship, $d_{\Delta t+1}$ decreases and $\sigma_{\Delta t+1}$ increases compared with the linear case, affecting next strain norm and so that next internal variable value.

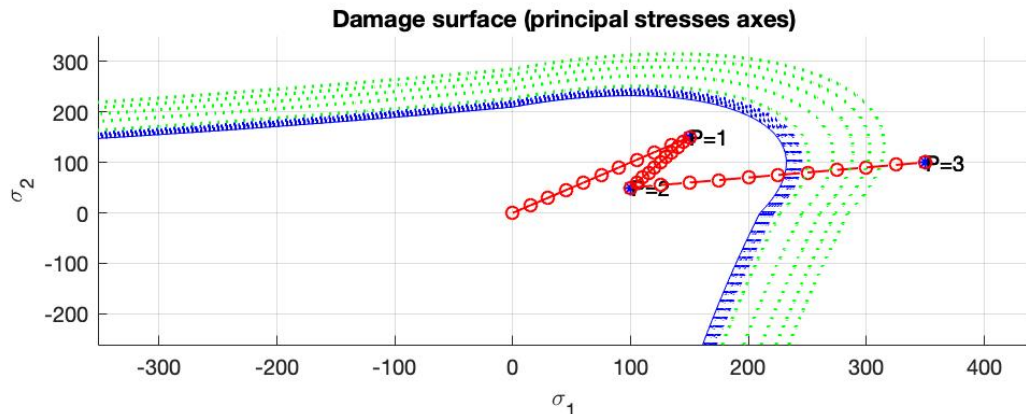


Figure 6. Stress space path – Linear q (blue) vs Exponential q (green)

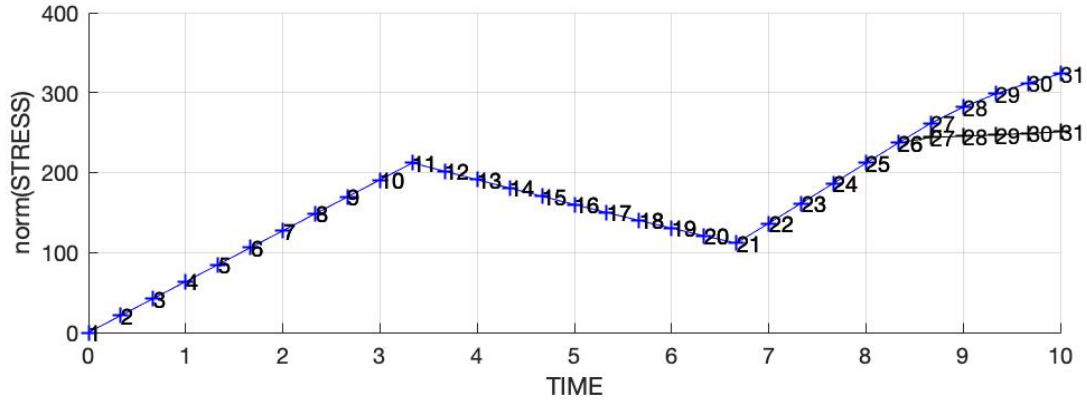


Figure 7. Stress norm evolution – Linear (black) and exponential (blue) hardening variable

The stress evolution plot for linear (black) and exponential (blue) cases, proved that the differences between the models just appear when inelastic loading starts, introducing the different formulation of q that affect next variables values under the iterative process.

4) Non symmetric model - Uniaxial tensile loading/unloading

Ratio tension strength $n = 3$

Case 1 - Linear hardening variable $q \rightarrow H = 0.1$

Case 2 - Linear softening variable $q \rightarrow H = -0.1$

$$\text{Initial stress points } \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (100, 0) \rightarrow \Delta\sigma_1^{(2)} = -5; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (350, 0) \rightarrow \Delta\sigma_1^{(3)} = 25; \Delta\sigma_2^{(3)} = 0 \end{cases}$$

Until now just hardening behaviour has been studied. Once the strain norm reached r value the current damage surface evolved to a larger one, due to hardening condition. Now, for non-symmetric material model, linear hardening and softening will be implemented under uniaxial tensile loading/unloading.

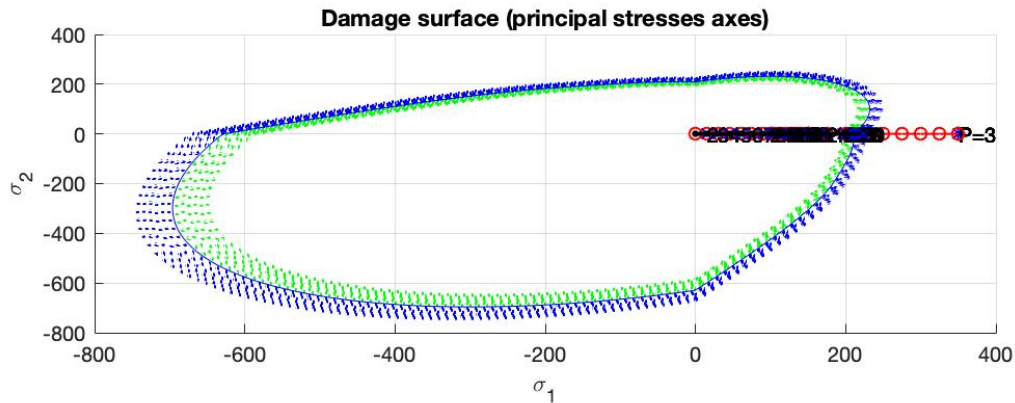


Figure 8. Stress space path – Linear hardening (blue) and linear softening (green)

Looking to the hole damage surface (Figure 8) and putting emphasis in the surface evolution, hardening and softening model can be easily differentiated. Softening modulus ($H < 0$) results in a compression of damage surface and the consequent decreasing stress results (Figure 9). Once damage appears, the $\Delta\sigma$ turns negative as $q_{\Delta t+1}/r_{\Delta t+1}$ relationship gets smaller and d increases compared with the hardening case at same time step.

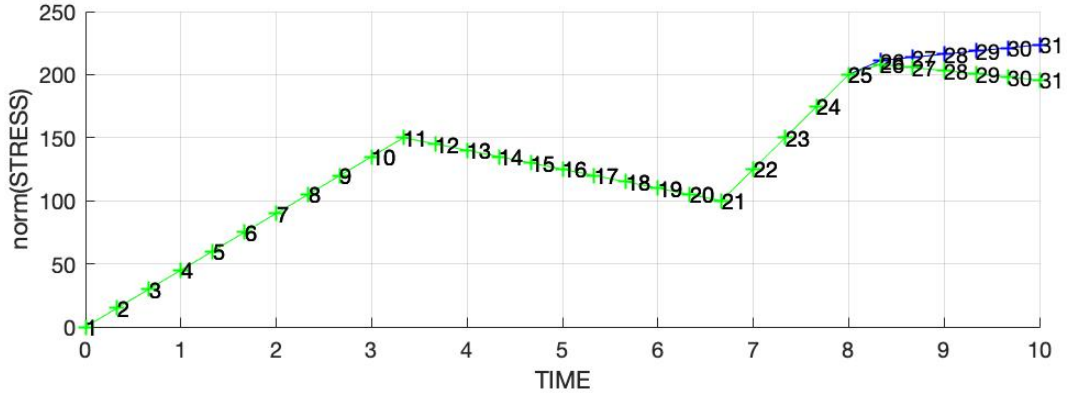


Figure 9. Stress norm evolution – Hardening (blue) and softening (green) models.

Verifying hardening vs softening for the same $|H|$ modulus, q behaviour is completely symmetric with respect to the horizontal line $q = q_0$ (Figure 10) as expected, being H the lines slope.

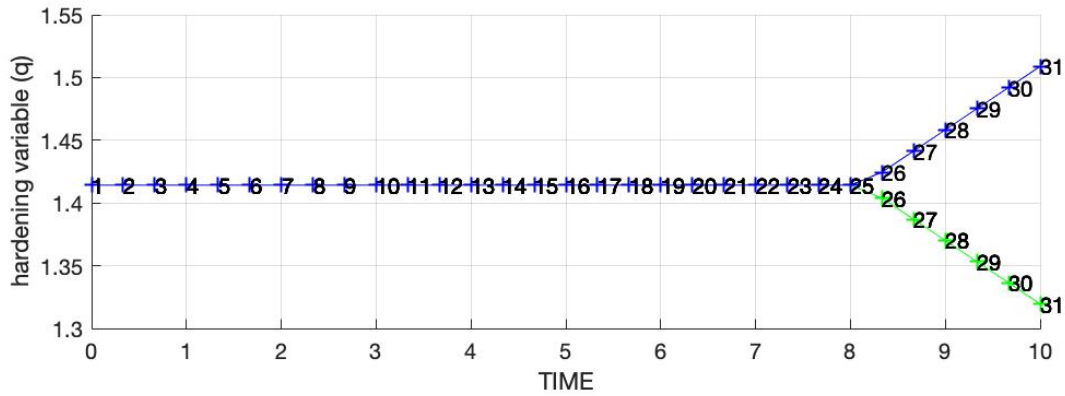


Figure 10. Hardening (blue) and softening (green) linear variable

5) Non symmetric model - Uniaxial tensile loading, biaxial tensile loading/unloading

Case 1 - Ratio compression strength $n = 3$

Case 2 - Ratio compression strength $n = 2$

Linear hardening variable $q \rightarrow H = 0.1$

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (-800, -600) \rightarrow \Delta\sigma_1^{(2)} = -95; \Delta\sigma_2^{(2)} = -60 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (0, 0) \rightarrow \Delta\sigma_1^{(3)} = 80; \Delta\sigma_2^{(3)} = 60 \end{cases}$$

Work with non - symmetric damage model is useful for simulating materials whose tension domain differs with respect to compression. Two different n are implemented under a larger $\sigma^{(2)}$ stress, in order to better understand how this parameter affects.

Having the same tension domain, failure by compression appears on Case 2 (Figure 12), while at Case 1 (Figure 11) damage surface remains constant, conserving the material its initial state working just under elastic conditions (loading/unloading) during the test. This leads us to see, the importance of understand

correctly the material response under any condition before use it, since two materials with equal behaviour at tension region could be completely different under compression regime.

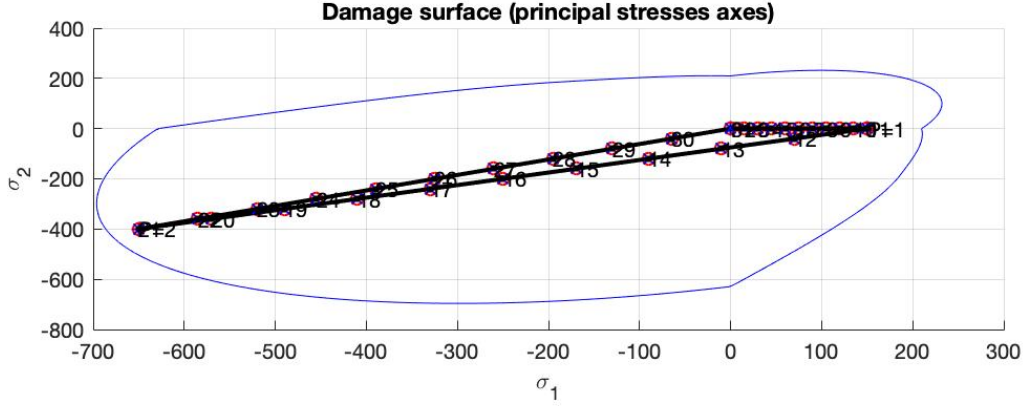


Figure 11. Stress space path – Case 1, $n=3$

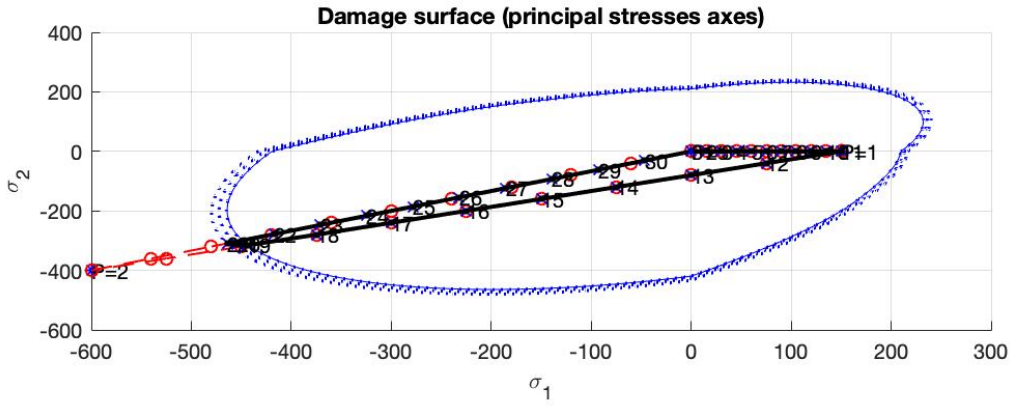


Figure 12. Stress space path – Case 2, $n=2$

6) Non tension model - Biaxial tensile loading/unloading

Ratio tension strength $n = 2$

Case 1 - Exponential hardening variable $q \rightarrow q_{inf} = r_0 + (r_0 - q_{zeros})$; $A = 1$

Case 2 - Exponential softening variable $q \rightarrow q_{inf} = q_{zeros}$; $A = 1$

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 100) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (-800, -600) \rightarrow \Delta\sigma_1^{(2)} = -95; \Delta\sigma_2^{(2)} = -60 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (0, 0) \rightarrow \Delta\sigma_1^{(3)} = 80; \Delta\sigma_2^{(3)} = 60 \end{cases}$$

For the last case of independent rate models, the effects of exponential hardening vs exponential softening will be compared. With a similar stress path as previous example, it is easier to check as in Example 3, that under inelastic loading the exponential q expands or compress much more the surface at each step (Figure 13). Once again, q vs t plot is symmetric with respect to the horizontal line $q = r_0$, being the slop H no longer a constant, since it depends on q value at each iteration (Figure 14).

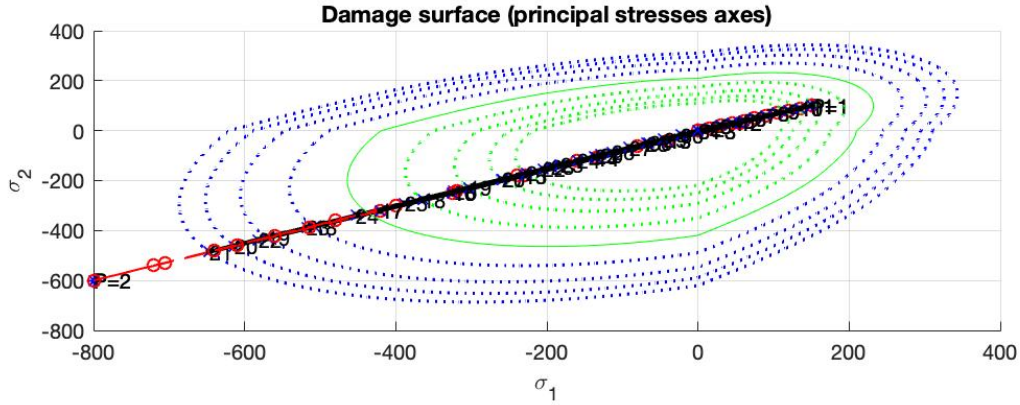


Figure 13. Stress space path – Hardening variable case (blue) vs Softening variable case (green)

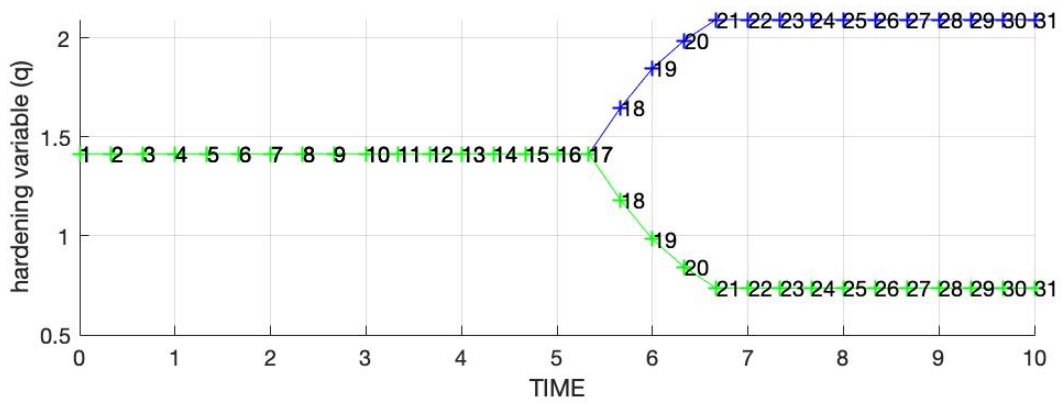


Figure 14. Hardening (blue) vs Softening (green) variable evolution

3. Rate dependent model

3.1 Introduction to the case

The rate dependent model is derived from the rate independent simply by introducing some changes in the damage evolution equation. Working with the alpha method, at the interval $[t_{n+1}, t_n]$ the strain norm will be identified at $t_{n+\alpha}$ and compared with the previous r_n to decide which is the current state. Finally, r_{n+1} is computed applying the appropriated new rate dependent formulation. To choose α value, pros and cons should be taken into account.

Being time t an independent variable and not just a parameter (rate independent formulation), two main differences will be found between the rate dependent and independent models: stress points can be outside the elastic domain and having a constant strain along the time the does not mean that stress is constant too.

Different rate dependent, plane strain case models are going to studied and compared, in order to understand and check their correctly behaviour. Starting at point $(\sigma_1, \sigma_2) = (0, 0)$, each example is going to be defined in terms of its corresponding effective stress increments.

Problem data and material properties

Material properties will be constant along each case $\begin{cases} E = 20000 \text{ N/m}^2 \\ \nu = 0.3 \\ \sigma_u = 200 \text{ N/m}^2 \end{cases}$

Initial internal variable $r_0 = \frac{\sigma_u}{\sqrt{E}} = 1.414$

Total time & Time steps/load state $t = 10; nsteps = 10$

Symmetric model

3.2 Results and conclusions

1) Stress space path - Biaxial tensile loading/unloading

Linear hardening variable, Case 1 $q \rightarrow H = 0.1$

Linear softening variable, Case 1 $q \rightarrow H = -0.1$

Viscosity $\eta = 0.5$

Alpha coefficient $\alpha = 0.5$

Initial stress points $\begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 100) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 10 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (400, 0) \rightarrow \Delta\sigma_1^{(2)} = 25; \Delta\sigma_2^{(2)} = -10 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (0, 0) \rightarrow \Delta\sigma_1^{(3)} = -40; \Delta\sigma_2^{(3)} = 0 \end{cases}$

The possibility to have stress state points outside the damage surface is one of the main differences that viscosity introduces to the model. Now obtaining a $\tau_{\epsilon_{n+1}} > r_n$ will no longer mean $r_{n+1} = \tau_{\epsilon_{n+1}}$.

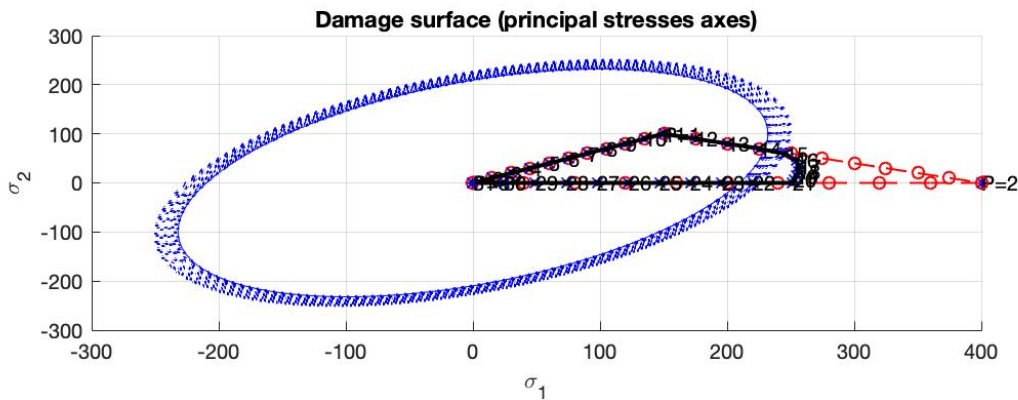


Figure 15. Stress space path – Case 1, hardening q

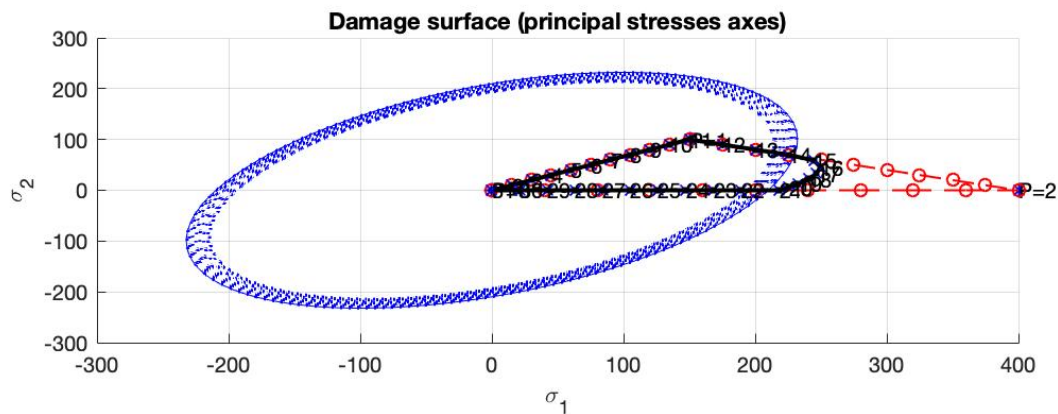


Figure 16. Stress space path – Case 2, softening q

Figures 15 and 16 show how phenomenon for hardening and softening variable respectively. Like in independent rate, the effect of the positive/negative H modulus is clearly verified at stress/strain plot (Figure 17). Once damage appears, the same strain corresponds to a lower stress in softening behaviour. Look that as the stress goes back to zero the strain also does, since an elastic material formulation is applied.

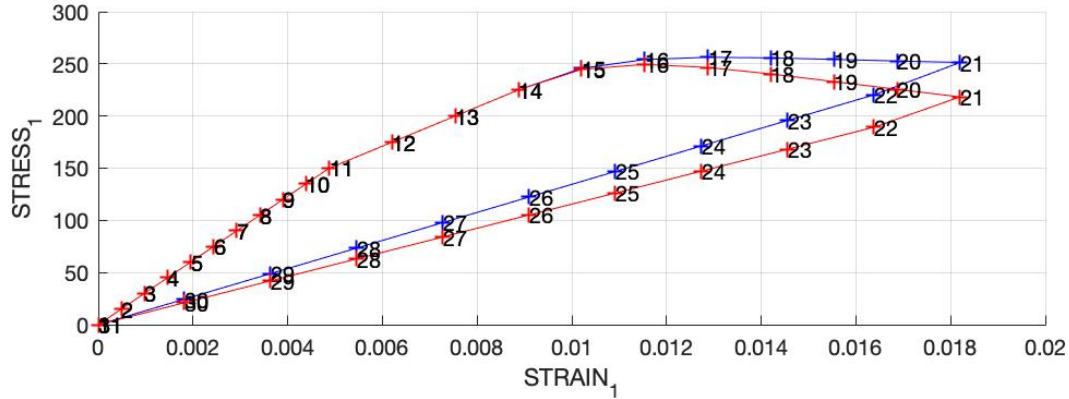


Figure 17. Principal stress vs principal strain – Case 1 (blue), Case 2 (red)

2) Viscosity effects - Uniaxial tensile loading/unloading

Linear hardening variable $q \rightarrow H = 0.1$

Alpha coefficient $\alpha = 0.5$

Viscosity $\rightarrow \eta_1 = 0.3$ (green); $\eta_2 = 0.6$ (red); $\eta_3 = 0.9$ (blue)

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (300, 0) \rightarrow \Delta\sigma_1^{(2)} = 15; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (200, 0) \rightarrow \Delta\sigma_1^{(3)} = -10; \Delta\sigma_2^{(3)} = 0 \end{cases}$$

Under uniaxial tensile loading/unloading path, the effect of viscosity is studied. Before looking the results, it is worth noting that they will be exactly the same until r_0 is reached, as in the independent rate model. Once $\tau_{\varepsilon_{n+1}} > r_0$, the new variables as viscosity or alpha come into play along inelastic loading state.

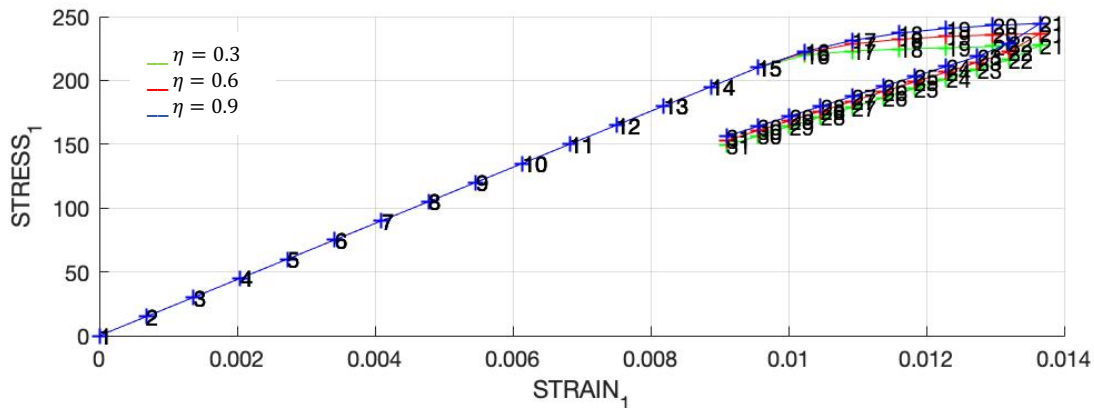


Figure 18. Principal Stress vs Principal strain plot for different viscosities

The figure above shows the principal stress vs principal strain results for different viscosities. Until the initial damage surface is reached ($\sigma^{(2)}$ stress), stress/strain points have the same values for all the cases. Then, high stress for same strain is obtained as viscosity grows (See Figure 19 for more detail).

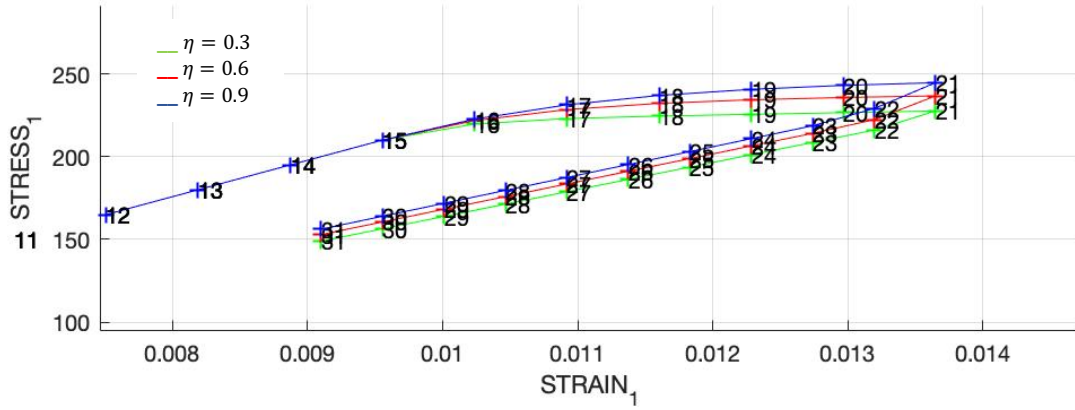


Figure 19.. Principal Stress vs Principal strain plot for different viscosities (DETAIL)

Knowing the dependency between parameters q , r , d and σ under rate dependent conditions, for a same alpha and time step, as viscosity increases the relationship q/r decreases obtaining higher values of d (Figure 20). Then, the current stress state will be smaller.

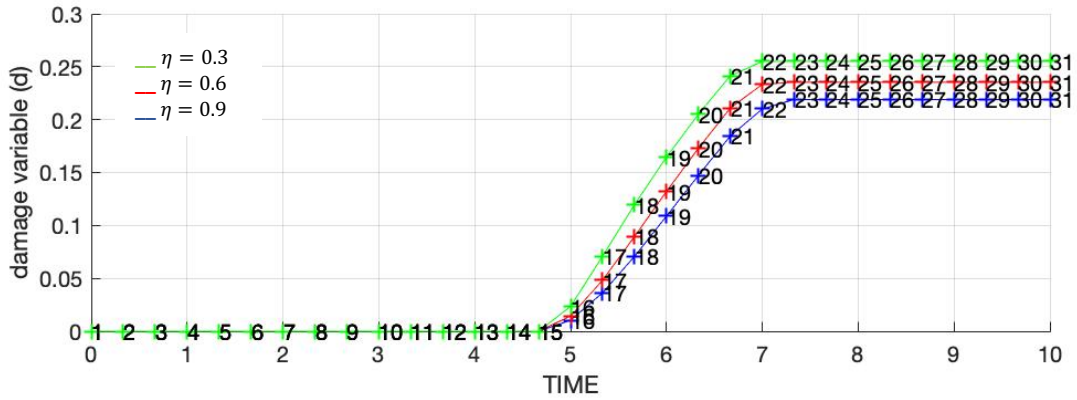


Figure 20. Damage variable evolution for each viscosity case

3) Time effect - Uniaxial tensile loading/unloading

Linear hardening variable $q \rightarrow H = 0.1$

Alpha coefficient $\alpha = 0.5$

Viscosity $\eta = 0.5$

Total time $\rightarrow t_1 = 10$ (green); $t_2 = 5$ (red); $t_3 = 1$ (blue)

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (300, 0) \rightarrow \Delta\sigma_1^{(2)} = 15; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (200, 0) \rightarrow \Delta\sigma_1^{(3)} = -10; \Delta\sigma_2^{(3)} = 0 \end{cases}$$

Fixing now the viscosity total time will change now, affecting directly to strain rate $\dot{\epsilon}$ and, consequently, to the stress/strain relationship. The results (Figure 21) reveal that material stress/strain relationship shows the same behaviour under $\dot{\epsilon}$ variation as viscosity variation as expected.

A shorter time (higher $\dot{\epsilon}$) means that the same stress has to be applied more abruptly over the material. Achieving same principal strain values regardless time, once again the difference appears around the stress field. As higher the $\dot{\epsilon}$, higher the principal stress for a same strain. Moreover, that difference is more pronounced as shorter the time.

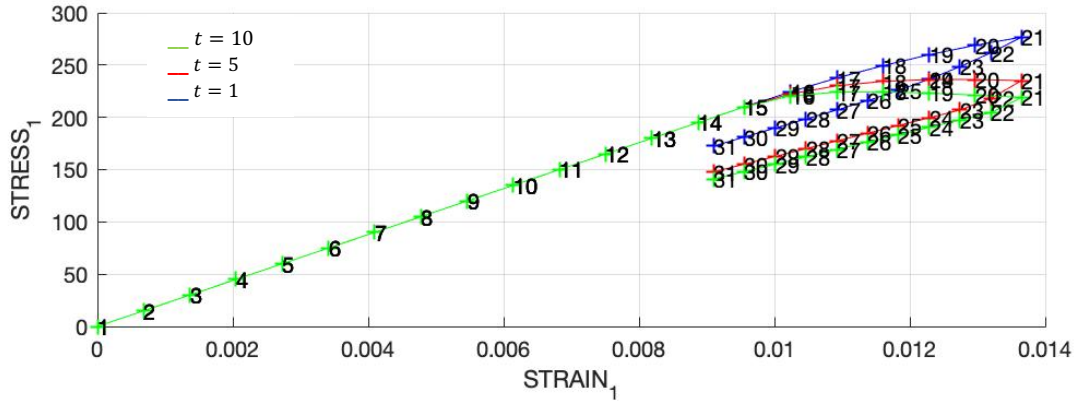


Figure 21. Principal Stress vs Principal strain plot for different strain rates

4) Alpha effects - Constitutive operators

Linear hardening variable $q \rightarrow H = 0.1$

Alpha coefficient $\alpha = 0 - 0.25 - 0.5 - 0.75 - 1$

Viscosity $\eta = 0.5$

Total time $t = 10 s$

$$\text{Initial stress points} \begin{cases} (\sigma_1^{(1)}, \sigma_2^{(1)}) = (150, 0) \rightarrow \Delta\sigma_1^{(1)} = 15; \Delta\sigma_2^{(1)} = 0 \\ (\sigma_1^{(2)}, \sigma_2^{(2)}) = (300, 0) \rightarrow \Delta\sigma_1^{(2)} = 15; \Delta\sigma_2^{(2)} = 0 \\ (\sigma_1^{(3)}, \sigma_2^{(3)}) = (200, 0) \rightarrow \Delta\sigma_1^{(3)} = -10; \Delta\sigma_2^{(3)} = 0 \end{cases}$$

Keeping in mind the algorithmic and analytical tangent operators' definitions, following results are obtained working from different alpha parameters. Further attention will be in analyse the effect that alpha has over both operators, taking as reference C_{11} element.

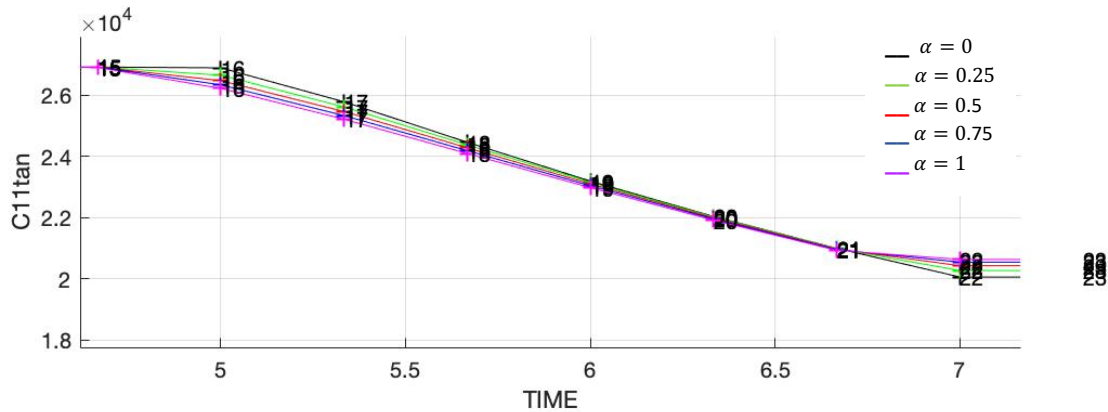


Figure 22. Tangent operator evolution - Inelastic loading region

Starting with the tangent operator, only under inelastic loading differences between the 5 problems appears. Same values are obtained until damage stars (step 15), having impact on d , and hence, over C_{tan} too.

C_{11} values decrease (Figure 22), getting larger initial values to lower α , This trend change as the applied stress goes higher, and, since unloading starts (Figure 23) C_{11} remains constant again but, affected by the previous generated damage.

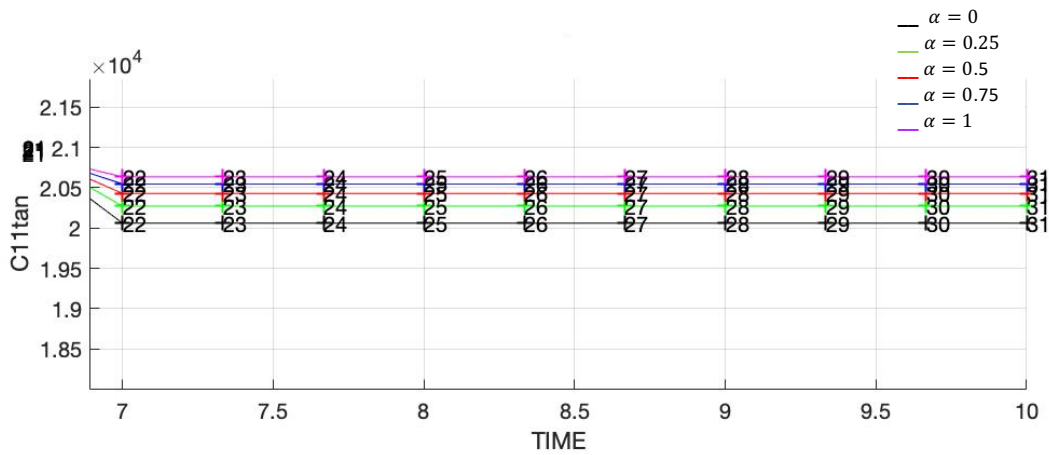


Figure 23. Tangent operator evolution – Unloading region

Looking algorithmic operator results (Figure 24), once again damage introduces alpha effects. Same evolution tendency is obtained being smaller the values and clearer the C_{11} differences. Finally, after step 23 same results as in Figure 23 are obtained, verifying that $C_{tan,11} = C_{alg,11}$ unless under inelastic loading state.

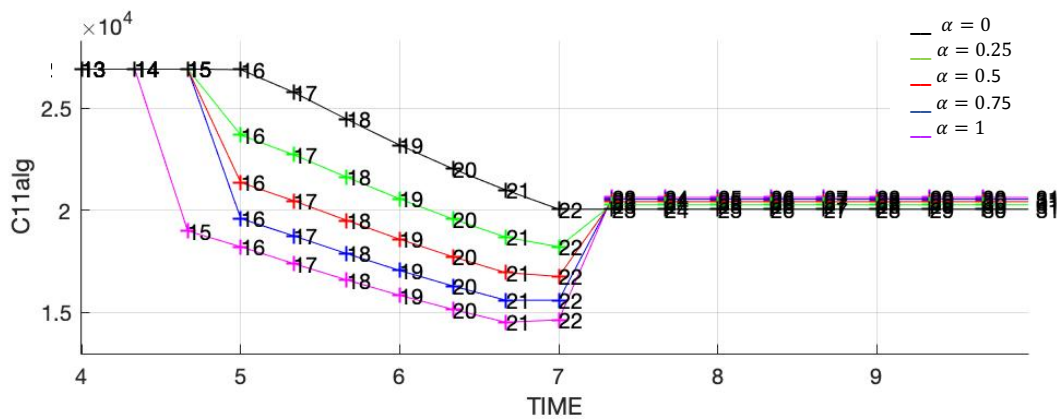


Figure 24. Algorithmic operator evolution – Inelastic loading and unloading regions

As final comment remarks the fact that for $\alpha = 0$ the same plots are obtained for tangent and algorithmic operators. Dividing C_{alg} equation as the sum of C_{tan} and a function of alpha f that depends directly on alpha, this case computation will be reduced to $C_{alg} = C_{tan}$ (Figure 25).

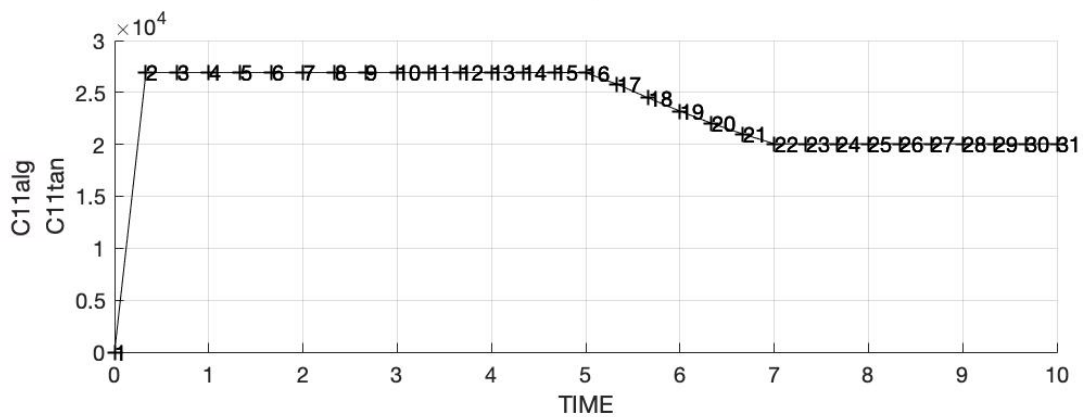


Figure 25. C_{tan} vs C_{alg} for $\alpha = 0$

APPENDIX

Listing of modified routines

A.1 Function “dibujar_criterio_dano1.m”

```
function hplot = dibujar_criterio_dano1(ce,nu,q,tipo_linea,MDtype,n)

%*          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);
rat = n;
cont = 1;
cont2 = 1;

% POLAR COORDINATES
%SYMMETRIC CASE

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

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

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

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

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

hplot =plot(s1,s2,tipo_linea);

% ONLY TENSION CASE

elseif MDtype==2

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

%* RADIUS
D=size(tetha);           %* Range
```



```

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

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

for i=1:Contador

    %Define positive stress vector
    s_pos = [max(m1(i),0) max(m2(i),0) 0 max(nu*(m1(i)+m2(i)),0)];

    radio(i)= q/sqrt(s_pos*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);

%NON-SYMMETRIC CASE

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

        %Compute tetha as function of stresses
        s_pos = sum([max(m1(i),0) max(m2(i),0) 0
max(nu*(m1(i)+m2(i)),0)]);
        s_abs = sum(abs([m1(i) m2(i) 0 nu*(m1(i)+m2(i))]);
        tta = s_pos/s_abs;

        radio(i)= q/((tta+(1-tta)/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);

end
return

```

A.2 Function “damage_main.m”

```

function[sigma_v,vartoplot,LABELPLOT,TIMEVECTOR]=
damage_main(Eprop,ntype,istep,strain,MDtype,n,TimeTotal,ce)
global hplotSURF

% 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);
eta = Eprop(7);
ALPHA_COEFF = 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

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_elasticol (Eprop, ntype);
% Initz.
% -----
% Strain vector
% -----
eps_n1 = zeros(mstrain,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 = 5 ;
vartoplot = cell(1,totalstep+1) ;
vartoplot{i}(1) = hvar_n(6) ; % Hardening variable (q)

```

```

vartoplot{i}(2) = hvar_n(5) ; % Internal variable (r)
vartoplot{i}(3) = 1-hvar_n(6)/hvar_n(5) ; % Damage variable (d)

for iload = 1:length(istep)
    % Load states
    for iloc = 1:istep(iload)
        i = i + 1 ;
        TIMEVECTOR(i) = TIMEVECTOR(i-1)+ delta_t(iload) ;
        % Total strain at step "i"
        % -----

        eps_n = eps_n1; %Strain step n (needed for viscous case)

        eps_n1 = strain(i,:) ; %strain step n+1

%*****
%*****
        %*          DAMAGE MODEL
        %
%*****
%*****
        [sigma_n1,hvar_n,aux_var,C_tan,C_alg] =
rmap_dano1(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_n,viscpr,delta_t);
        % PLOTTING DAMAGE SURFACE
        if(aux_var(1)>0)
            hplotSURF(i) = dibujar_criterio_dano1(ce, nu, hvar_n(6),
'r:',MDtype,n );
            set(hplotSURF(i),'Color',[0 0 1],'LineWidth',1)
;
            end
            C1ltan = C_tan(1,1);
            C1lalg = C_alg(1,1);
            vartoplot{i}(4) = C1ltan; %C1l tangent constitutive tensor
            vartoplot{i}(5) = C1lalg; %C1l algorith constitutive tensor

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

%*****
%*****
        % 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 ;

        % 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

```

A.3 Function “rmap_danol.m”

```

function [sigma_n1,hvar_n1,aux_var,C_tan,C_alg] = rmap_danol
(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_n,viscpr,delta_t)

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_COEFF = Eprop(8);
%*****
%*****

%*      initializing
%*
r0 = sigma_u/sqrt(E);
zero_q=1.d-6*r0;

A = 1; %For exponential hard_case

% if(r_n<=0.d0)
%   r_n=r0;
%   q_n=r0;
% end
%*****
%*****

%*      Damage surface      %*
if viscpr == 0
[rtrial] = Modelos_de_danol (MDtype,ce,eps_n1,n);
elseif viscpr ==1
    rtrial_n = sqrt(eps_n*ce*eps_n');
    rtrial_n1 = sqrt(eps_n1*ce*eps_n1');
    rtrial = (1 - ALPHA_COEFF)*rtrial_n + ALPHA_COEFF*rtrial_n1;
end

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

if(rtrial > r_n) % INELASTIC LOADING

    %FOR VISCOUS CASE
    if viscpr ==1
        fload=1;

        r_n1 = ((eta - delta_t*(1-ALPHA_COEFF))/(eta +
ALPHA_COEFF*delta_t))*r_n + ...
(delta_t/(eta+ALPHA_COEFF*delta_t))*rtrial;

```

```

        delta_r=r_n1-r_n;

%FOR NON VISCOUS CASE
elseif viscpr == 0
    fload=1;
    delta_r=rtrial-r_n;
    r_n1= rtrial ;
end

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

%Limit to the hardening variable. It can be lower than that.
if(q_n1<zero_q)
    q_n1=zero_q;
end

else

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

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

%*****
%*****
%CONSTITUTIVE OPERATORS

if viscpr ==1

    if (rtrial > r_n)
        C_tan = (1-dano_n1)*ce;
        C_alg = (C_tan) +
        (((ALPHA_COEFF*delta_t)/(eta+ALPHA_COEFF*delta_t))*(inv(rtrial_n1))*..
        .
        (H*r_n1-q_n1)/(r_n1)^2)*(sigma_ef*sigma_ef');
    else
        C_tan = (1-dano_n1)*ce
        C_alg = C_tan;
end

```

```

else
    if (rtrial > r_n)
        C_tan = (1-dano_n1)*ce
        C_alg = C_tan-(q_n1-H*r_n1)/r_n1^3*(sigma_ef*sigma_ef');
    else
        C_tan = (1-dano_n1)*ce
    end
end
end
%*****
%*****
%* Updating historic variables
%*
% hvar_n1(1:4) = eps_nlp;
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;
%*****
%*****

```

A.3 Function “Modelos_de_dano1.m”

```
function [rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n)

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

elseif (MDtype==2) %* Only tension

s_n1 = ce*eps_n1';
s_nlp = [max(s_n1(1),0) max(s_n1(2),0) max(s_n1(3),0) max(s_n1(4),0)];
rtrial =sqrt(eps_n1*s_nlp');

elseif (MDtype==3) %*Non-symmetric

s_n1 = ce*eps_n1';
s_nlabs = abs(s_n1);
s_nlp = [max(s_n1(1),0) max(s_n1(2),0) max(s_n1(3),0) max(s_n1(4),0)];

tta = (sum(s_nlp))/(sum(s_nlabs));

rtrial= (tta + (1 - tta)/n) * sqrt(eps_n1*ce*eps_n1');
end
%*****
%*****
return
```