Assignment 1: Continum Damage Models

Computational Solid Mechanics

MNME

Carles Duñó Nosas

8 April 2016

Contents

1. Introduction.

Solid mechanics usually deals with continuum damage models, being these models strong tools for numerical simulations when the material presents micro fractures. Due to the importance of the theoretical knowledge and the numerical aspects for using, the aim is to implement correctly the algorithms and perform the numerical integration of continuum damage constitutive models as well. By means of numerical test, the current study is focus on prove the correctness of the code.

Given the mechanical characteristics of a material and the load states, the purpose of the code is to compute for a plane stress problem the strains, damage variable, and as a post process the effective stresses, and the tangent constitutive tensor.

In this section three models for the elastic domain are considered (symmetric, tensile only, and non-symmetric) which in function of the experimental results one of them could be used for model the elastic domain and the damage surface of the material studied.

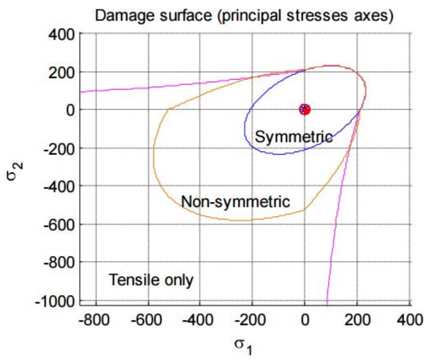

Figure 1: Models for the elastic domain.

2. PART I: Rate independent models

The rate independent model considers that the stress state (or strain state) does not depends on the velocity application of the load, so it can be computed without take into account the time.

2.1. Exponential hardening/softening ($H < 0$ and $H > 0$).

An exponential law needs two parameters, A which determines the velocity to achieve the asymptotic value of q_{inf}. This section has done the test with A=2.5 and $q_{inf} = 2$ for the hardening test, and $q_{inf} = 0.8$ for the softening. Values that ensures that the model is realistic.
In the following figures it can be appreciate that for hardening

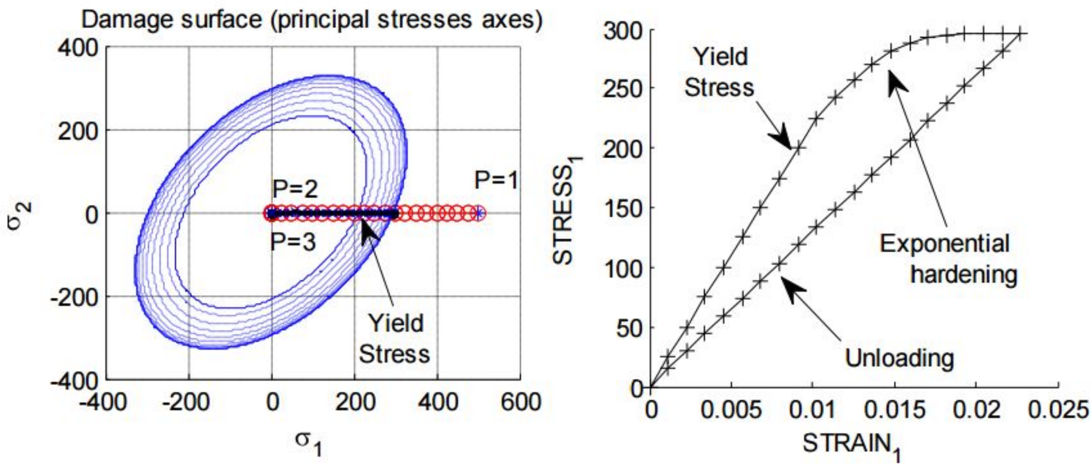

Figure 2: Damage surface (left) and σ_1/ε_1 plot of an exponential hardening test (right).

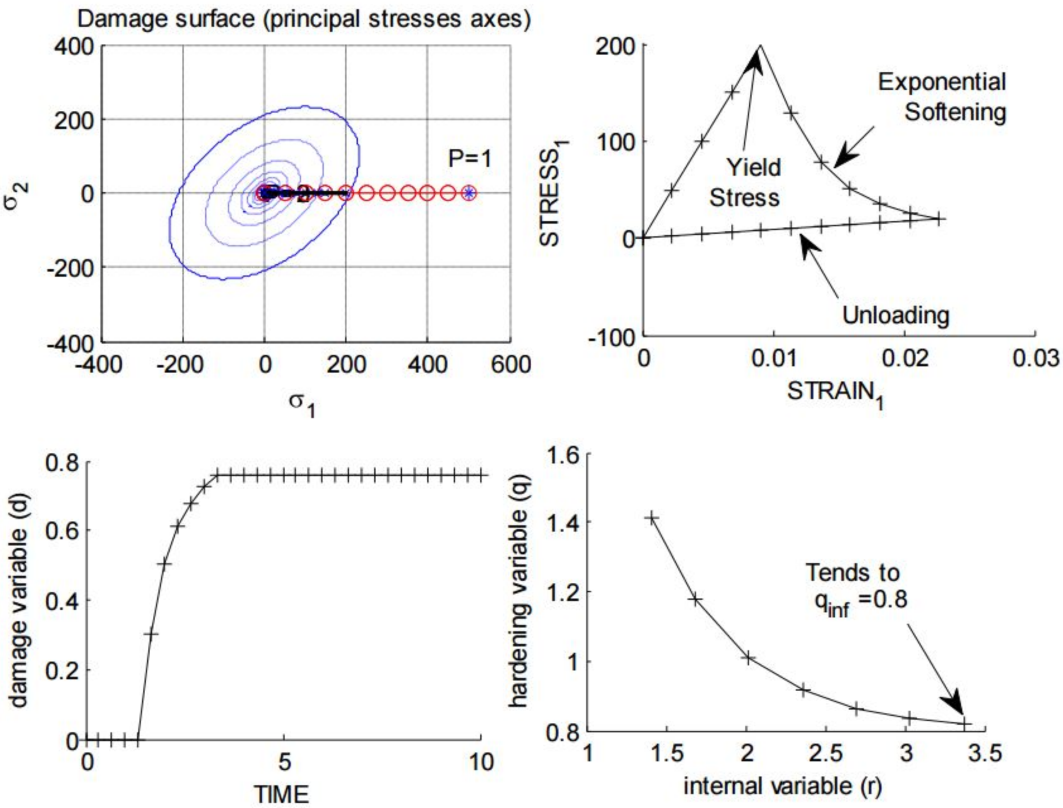

Figure 3: Damage surface (top-left), σ_1/ε_1 plot (top-right), damage variable evolution through the time (bottom-left) and the q/r plot of exponential softening test (bottom-right).

In the figures 2 and 3 it is shown that the damage surface expands, whereas, for the softening it reduces. In both cases the stress state remains always on the damage surface or in the elastic domain. The exponential laws begin when the yield stress is exceeded and tens to $q_i nf$ value, and finishes for the unloading process. The exponential law, the one implanted it is explained in $Annex^3$

2.2. Assessing the correctness of the implementation.

Three load cases will be carried out to check the implementation and each load case described by three-segment paths in the strain space. All the test cases for rate independent models will be done with the linear hardening law, elastic modulus $E = 20000Pa$, poison's ratio $\nu = 0.3$, hardening modulus $H = 0.6$, the yield stress $\sigma_y = 200Pa$ and $n = 3$. The loading parameters chosen are $\alpha = 400Pa$, $= 200Pa$ and $\gamma = 700Pa$.

2.2.1. Loading path 1.

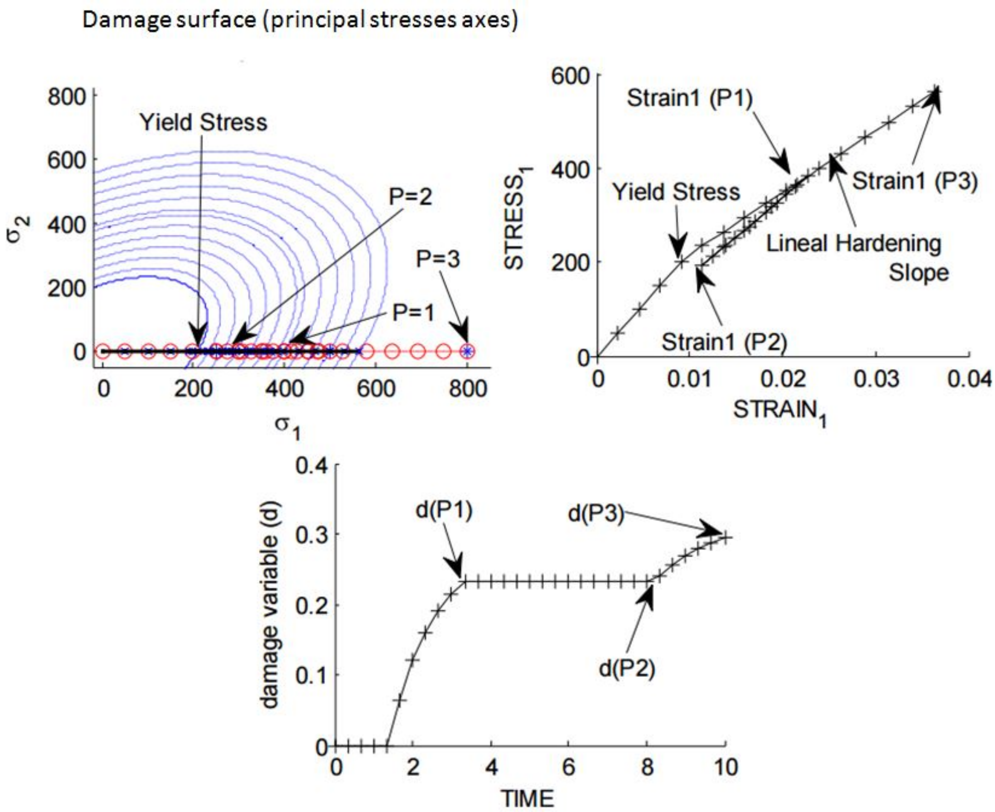

Figure 4: Damage surface (top-left), σ_1/ε_1 plot (top-right), damage variable evolution for loading case 1 (bottom).

In the Figure 4 is observed that in the first segment, once it reaches the yielding stress the slope changes and follows the lineal hardening slope. During the second segment (uni-axial tensile unloading and compressive loading) the slope it tends to the get back part of the elastic strain, but if we load again, it takes the same unloading slope until it achieves the maximum damage surface, where it follows again the lineal hardening slope. And the damage variable starts in 0 and just increases when there is a damage loading, and never decreases which is expected.

2.2.2. Loading path 2.

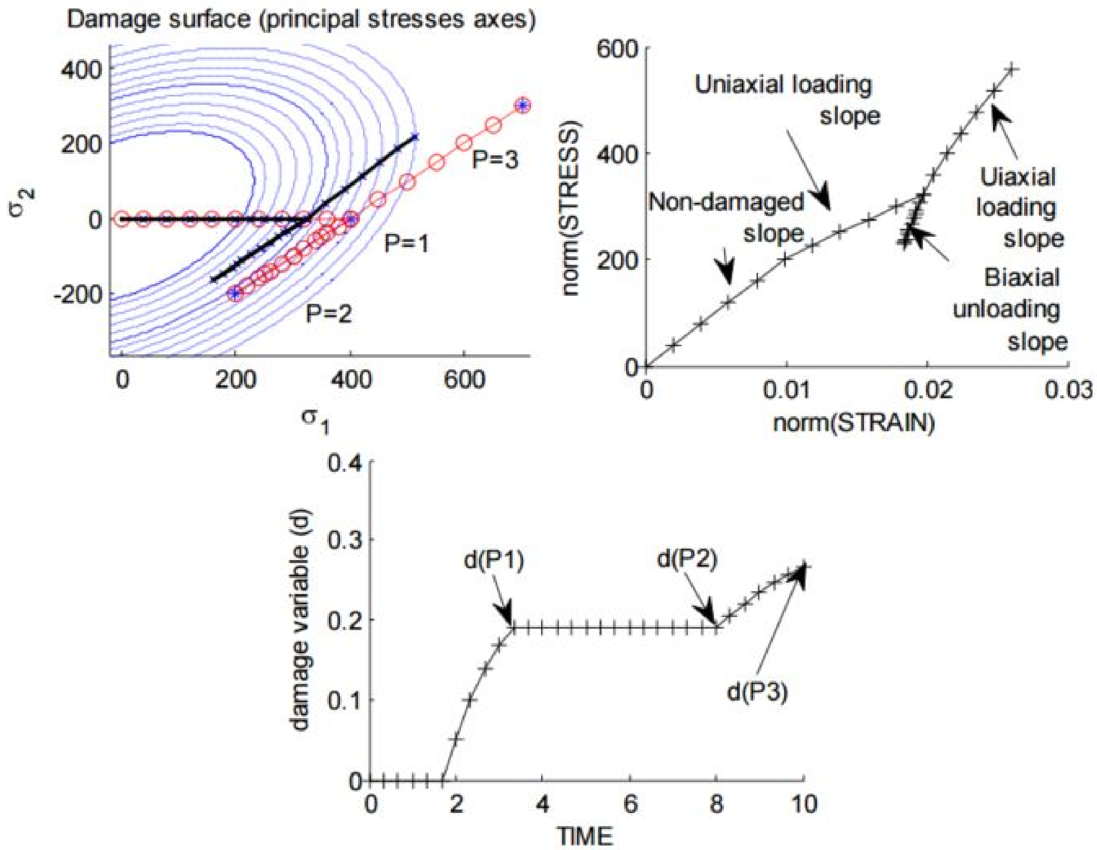

Figure 5: Damage surface (top-left), $norm(\sigma)/norm(\varepsilon)$ (top-right), and damage variable evolution for the loading case 2 (bottom).

Due to the fact that this loading case (Figure 5)has a bi-axial loading it have no sense to make the plot σ_1/ε_1 so it is depicted as the $norm(\sigma)/norm(\varepsilon)$. In the figure it can be seen that the slopes for the uniaxial and the biaxial loading are different, which makes sense. And the evolution of the damage variable shows reasonable results.

2.2.3. Loading path 3.

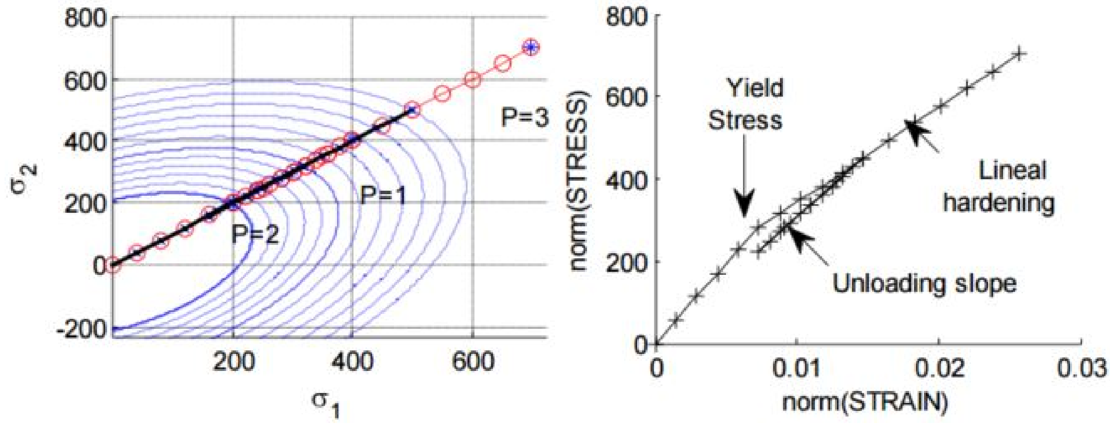

Figure 6: Damage surface (left) and $norm(\sigma)/norm(\varepsilon)$ (right).

The third test is much similar than the first one, because the rate between the σ_1/ε_1, thus the slope of the curves are equal during the test. Note that the difference is that the value for damaging the material is not σ_u any more, due to the poisson effect which increases the initial elastic domain in oblique direction.

After analyze this three numerical experiments we have assessed the implementation of the rate independent code, and everything is what was expected, so the code is well implemented.

3. PART II: Rate dependent part

The rate-dependent model required a more involved modification of the MATLAB code provided, that could be find in $Annex^5$. It is interesting to check that it is possible to recall the rate independent results with the rate dependent code when the viscous effects are small enough.

The new parameters for the viscous model are the viscosity parameter η, and the T time of the load application. However, two more parameters: ΔT which discretizes in time, and α which defines the integration type, both variables are needed to perform the numerical integration of the constitutive equation. We must take care about the influence of ΔT and α on the results, in order to not introduce numerical errors or instabilities on the computations. For analyze the variables effects some uniaxial test have been carrying out, and are shown in this section.

The mechanical values for the computation are the same used in the rate independent test, and the ones which are collected in Table 1.

Load	σ_1	σ_2
P_1	500	0
P_2	250	0
P_3	800	0

Table 1: Load state for the test

3.1. Testing the time:

Just for start one simple test varying the load time application, with some viscosity and taking $\alpha = 1/2$ for be second order accurate.

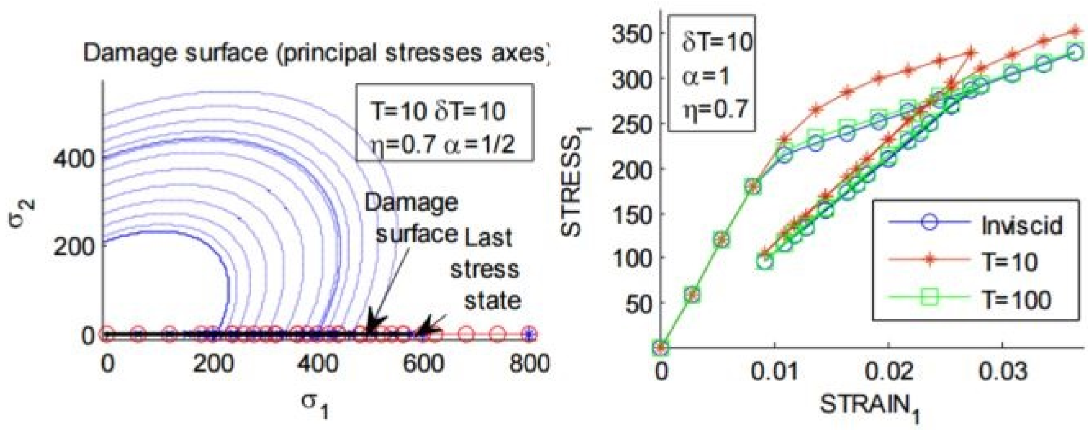

Figure 7: Damage surface (left) and testing the time effect ploting σ_1/ε_1 (right).

As it is shown in Figure 7, the effective stresses not match on the damage surface, which it is explained because the rate dependent model the Karush-Kuhn-Tuker conditions are not necessarily fulfilled. And about the time effect, the viscous solutions tends to the inviscid one when time is larger enough, which means that the load application velocity is small.

3.2. Testing α and η:

The results are shown in terms of the σ_1/ε_1 for different η Fig 8-9 and the first component of the tangent constitutive operators are depicted in Fig 10 as well.

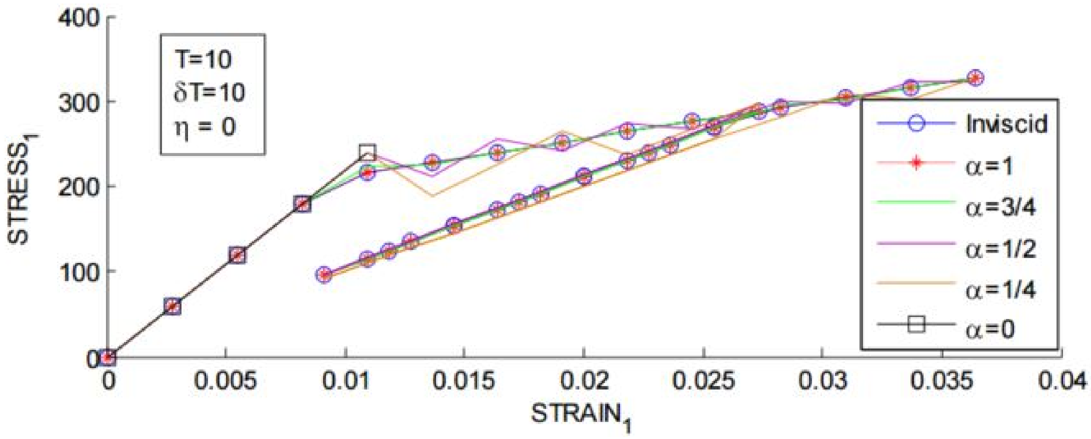

Figure 8: Plot of σ_1/ε_1 with $\eta = 0$ changing α

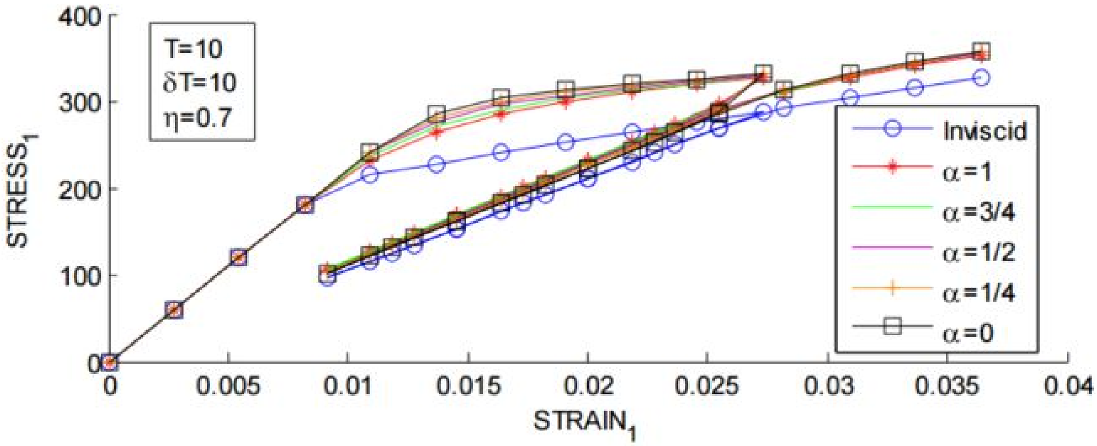

Figure 9: Plot of σ_1/ε_1 with $\eta = 0.7$ changing α

Firstly note that α has strong influence on numerical instabilities, as it is shown in Figure 8, just appears during the damage loading. This behaviour is larger for small α, but also with $\alpha = 1/2$ there are some instabilities. This was not expected form a theoretical point of view, because it is well known that alpha integration method works properly in the interval [1/2, 1]. But according to the theory lectures, for $\alpha = 1/2$ and $\eta = 0.7$ the solution is exactly the same than the inviscid results, this test ensures that the algorithm is well implemented.

Comparing the Figures 8 and 9 which the only difference is that there is some viscosity for the second one, we can conclude that the instability effect appears for small η. There is still one observation more for the viscous case, which is that the effective stresses are greater than the inviscid case, and it is because of the non-fulfilling the Karush-Kuhn-Tucker conditions.

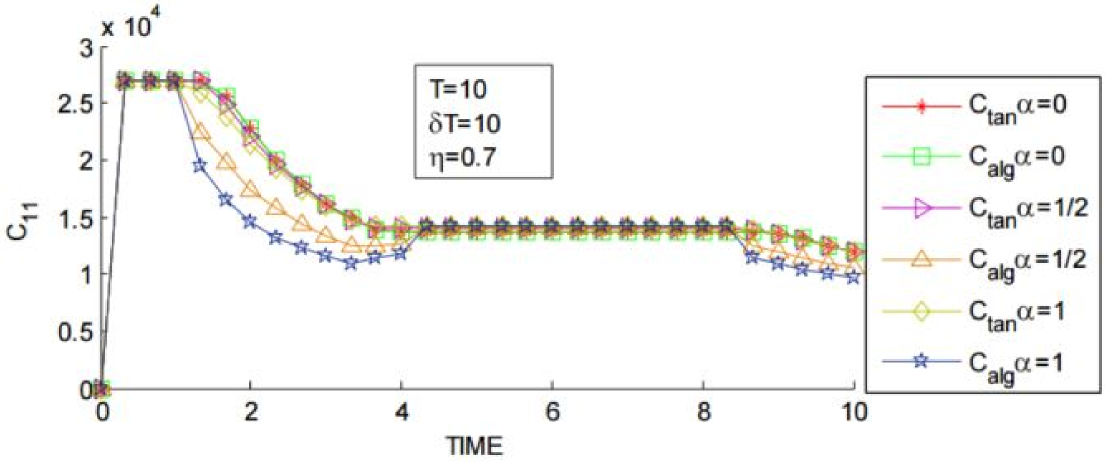

Figure 10: Plot of the first component of the constitutive operators changing α

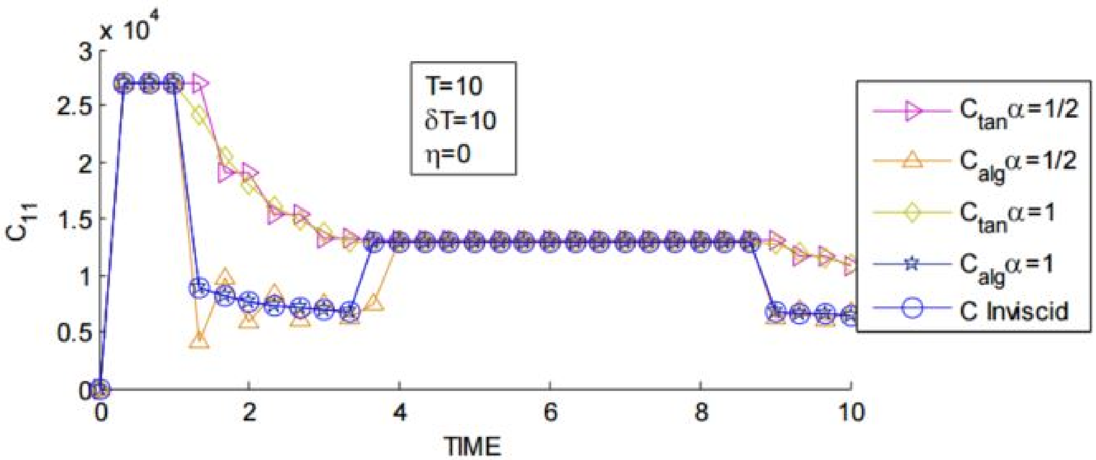

Figure 11: Plot of the first component of the constitutive operators changing α without viscosity

3.3. Testing the ΔT effect:

As it is shown in the figure 8 for a test with $\alpha = 1/2$ and $\eta = 0$ the results are not stable, so this parameters are chosen for the next test, where we want to analyze how the instabilities can be reduced varying the time discretization. As it was said before the parameter ΔT is the number of times which is the load state divided

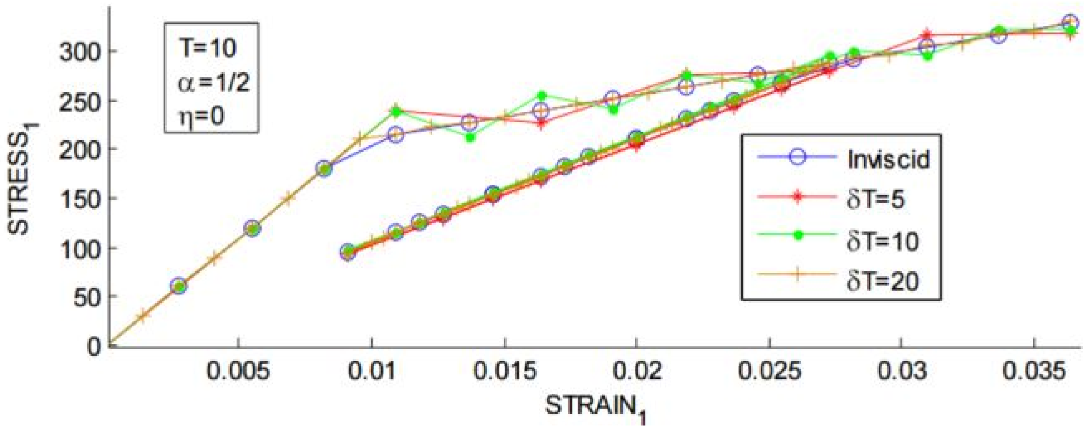

Figure 12: Plot of σ_1/ε_1 changing ΔT

As can be seen in Fig 12, the instabilities decreases while time discretization is much finer ($\Delta T = 20$) and are larger for court discretization ($\Delta T = 5$). This result is quite obvious, but proving that in our code it works, is important for asses the correctness of the implementation.

4. Conclusions.

The implementation was tested against several cases, and in every case, it met the results expected from the theory. However, the field of Continuum Damage Model is very complex and requires a lot of experience and care from the user when applying it to the real world.

For alpha greater than one half the results does not show instabilities, and larger ones for small values of alpha. Note that this instability appears just for low viscosity parameter. That remark is the key to match the theory which says that for alpha equal to one half second order error is achieved, but there are not instabilities.

As a final conclusion, stress out that even with just the rate dependent code we are able to compute for a non-viscous material it is better to keep both codes, because of the rate dependent introduce some errors while computing the integration of the constitutive equation of the damage model.

5. Annex.

c1) Tensile-damage-only model

To implement the tensile-damage only model the following expression is coded,

$$\tau_{\varepsilon_{t+\Delta t}} = \sqrt{\langle \varepsilon_{t+\Delta t}\colon \mathbb{C}\rangle\colon \varepsilon_{t+\Delta t}}$$

modelos_de_dano1.m (line 19-20)

```
elseif (MDtype==2)  %* Only tension
rtrial=sqrt(fmacaulay(eps_n1*ce)*eps_n1');
```

where the *fmacaulay* is the Macaulay function, which given a number, it does nothing if it is positive otherwise returns the zero value.

fmacaulay.m (line 1-3)

```
function [m] = fmacaulay(v)
m=(abs(v)+v)/2;
end
```

For plotting the damage surface for the tensile model the whole structure of the symmetric model is copied and just the expression of the radius is changed for the following one,

$$r = \frac{q}{\sqrt{\langle \sigma^T \rangle\colon \mathbb{C}^{-1}\colon \sigma}}$$

dibujar_criterio_dano1.m (line 73-75)

```
%%% vm and v are vectors collecting the components of the
stresses, just for make more easy at the implementation level
vm=[fmacaulay(m1(i)) fmacaulay(m2(i)) 0 …
fmacaulay(nu*(m1(i)+m2(i)))];
v=[m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
radio(i)= q/((vm*ce_inv*v')^0.5);
```

c2) Non-symmetric Model

In the non-symmetric model we proceed as before where now the expression is the following one,

$$\theta = \frac{\sum_1^3 \langle \bar{\sigma} \rangle}{\sum_1^3 |\sigma|}$$

$$\tau_{\varepsilon_{t+\Delta t}} = \left(\theta + \frac{1-\theta}{n}\right)\sqrt{\varepsilon_{t+\Delta t}\colon \mathbb{C}\colon \varepsilon_{t+\Delta t}}$$

modelos_de_dano1.m (line 23-24)

```
tetha=sum(fmacaulay(eps_n1*ce))/sum(eps_n1*ce);
rtrial=(tetha+((1-tetha)/n))*sqrt(eps_n1*ce*eps_n1');
```

and for plotting the damage surface,

dibujar_criterio_dano1.m (line 96-99)

```
%%% tita interpolates the radio when one of the components is
under traction
v=[m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
tita=sum(fmacaulay(v))/sum(abs(v));
radio(i)=q/((tita+((1-tita)/n))*(sqrt(v*ce_inv*v')));
```

c3) Exponential hardening/softening law

Firstly the parameters A and q_{inf} are stored as components of vector properties,

main_noninteractive.m (line 108-109)

```
%%% The A_exp, and q_inf are added in the Eprop vector
Eprop   = [E nu HARDSOFT_MOD sigma_u hard_type viscpr eta
ALPHA_COEFF, A_exp, q_inf];
```

and the expression for the exponential hardening/softening looks like,

$$q = q_{inf} - (q_{inf} - r_0)e^{\left(A - \frac{Ar}{r_0}\right)}$$

moreover the derivative H will be needed for compute the C damage tensor,

$$H = A\left(\frac{q_{inf} - r_0}{r_0}\right)e^{\left(A - \frac{Ar}{r_0}\right)}$$

rmap_dano1.m (line 68-71)

```
elseif hard_type==1
    % Exponential
    q_n1 = q_inf-(q_inf-r0)*exp(A_exp*(1-r_n1/r0));
    H=A_exp*((q_inf-r0)/r0)*exp(A_exp*(1-r_n1/r0));
```

c4) Constitutive tangent operator for the rate independent code

This tensor give information about the evolution of the C which keeps the mechanical behaviour of the material, thus when there is damage, this C changes, and it is called C damage, and has the following expression,

$$\mathbb{C}_{tan}^{d} = \begin{cases} (1-d)\mathbb{C}, & Elastic/Unloading \\ (1-d)\mathbb{C} - \dfrac{q-Hr}{r^3}\bar{\sigma}\otimes\bar{\sigma}, & Loading \end{cases}$$

```
rmap_dano1.m (line 96-100)

      %%% Computing the constitutive tangent tenso
      Ctang_n1=(1-dano_n1)*ce;
        if fload == 1
            Ctang_n1=(1-dano_n1)*ce-((q_n1-
            H*r_n1)/r_n1^3)*((ce*eps_n1')*(ce*eps_n1')');
        end
```

for plotting the first component of the tensor,

```
rmap_dano1.m (line 118)

      aux_var(4)=Ctang_n1(1,1);
```

```
damage_main.m

(line 130)
      vartoplot{i}(4)=ce(1,1);

(line 205)
      vartoplot{i}(4) =aux_var(4);
```

c5) Rate Dependent (viscous case)

The rate dependent algorithm is a little bit more complex than the inviscid code, because the constitutive equation must be integrated. Thus, and in order to simplify the implementation is decided to create a new function *rmap_dano0.m* which is the simile than the *rmap_dano1.m* which is exclusive for the rate independent case. This function has the *DeltaT* and *eps_n0* as new inputs, and note that the *viscosity* and the *alpha* parameter are coming by the 7th and 8th components of the vector *Eprop*.

```
rmap_dano0.m

      function [sigma_n1,hvar_n1,aux_var,Ctang_n1,Calg_n1] =
      rmap_dano0 (eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_n0,delta_t)

(Line 33-35)
      %%% the eta is bring as the viscosity parameter, and alpha is
      the integration coefficient
      eta_v=Eprop(7);
      alpha=Eprop(8);

(Line 53-55)
      [rtrial0] = Modelos_de_dano1 (MDtype,ce,eps_n0,n);
      [rtrial1] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);
```

```
      rtrial= rtrial0*(1-alpha)+alpha*rtrial1;

(Line 62-66)
      if(rtrial > r_n)
      %*   Loading
      fload=1;
      r_n1=((eta_v-delta_t*(1-alpha))/(eta_v+alpha*delta_t))*r_n+
      +(delta_t/(eta_v+alpha*delta_t))*rtrial;
      delta_r=r_n1-r_n;
```

c6) Constitutive tangent and algorithmic operators for the rate dependent case

The C algorithmic tensor gives information about how the mechanical properties of the material is evolution along the process, and for computing we implement the next equation,

$$\mathbb{C}_{alg}^{vd} = \begin{cases} (1-d)\mathbb{C}, & \textit{Elastic/Unloading} \\ (1-d)\mathbb{C} - \dfrac{\alpha\Delta t(q-Hr)}{(\eta+\alpha\Delta t)\tau_{\varepsilon} * (r^2)}\bar{\sigma}\otimes\bar{\sigma}, & \textit{Loading} \end{cases}$$

rmap_dano0.m (Line 94-101)

```
      %%% Computing the Constitutive tangent and the algorithmic
      operators
      Ctang_n1 = (1-dano_n1)*ce;
      Calg_n1 = (1-dano_n1)*ce;
      if fload==1
            Calg_n1 = Calg_n1+(alpha*delta_t/(eta_v+alpha*delta_t))
            *((H*r_n1-q_n1)/(r_n1^3))*(ce*eps_n1')*(ce*eps_n1')';
      end
```

and for displaying the first tensor component,

main_noninteractive.m (line 85)

```
      LABELPLOT = {'hardening variable (q)','internal variable
      (r)','damage variable (d)','C_t_a_n_g11','C_a_l_g11'};
```

rmap_dano0.m (Line 120-121)

```
      aux_var(4)=Ctang_n1(1,1);
      aux_var(5)=Calg_n1(1,1);
```