

# Computational Solid Mechanics

## Assignment 1

Jordi Parra Porcar

Delivered in April 2018

## 1 Continuum Damage Models

The science of *Continuum Damage Mechanics* is used to model materials behaviour taking into account mathematically the change on its macroscopic mechanical behaviour due to changes that takes place at a microscopic scale, more precisely in the material micro-structure. Likewise, micro-cracks and micro-pores appears internally in the material which make the area of tension application smaller inside it, whereas at the macroscopic scale the area is considered to be the total one disregarding the before-mentioned effects. The response to damage of some material it may be through a loose of stiffness, as it occurs with rocks and concrete, on the other side, ductile materials like low-middle carbon content steels, by means of a visco-plastic dissipation whilst maintaining the magnitude of its stiffness modulus constant, in case an unload event would take place.

### 1.1 Description of the Isotropic Damage Model

A consideration between two different types of stresses has to be made in a continuum damage model: apparent stresses ( $\sigma$ ) and effective stresses ( $\bar{\sigma}$ ).

The apparent stresses are external and they act on the whole section ( $S$ ) whilst the effective stresses act on the undamaged regions ( $\underline{S}$ ).

Damage variable, denoted by  $d(r)$ , relates the apparent stresses and the effective stresses. Likewise, for a completely damaged section, the damage variable takes the value of 1, whereas for a not damaged section its value is equal to zero. The variable  $d(r)$  depends on internal variables and its magnitude never decreases, a fact that is explained trough the explanation of the second principle of thermodynamics which establish that the mechanical dissipation in a material is always greater or equal to zero. The constitutive equation for isotropic damage model is stated as:

$$\sigma = (1 - d(r))\mathbb{C} : \varepsilon \quad (1)$$

Where  $\mathbb{C}$  is the elastic isotropic tensor (4th order definite positive tensor).

It is necessary to define a spatial domain whose boundary represents the damage surface. This domain can be presented in two ways as:

$$f(\sigma, r) = \tau_\sigma - q(r) \quad (2)$$

$$g(\varepsilon, r) = \tau_\varepsilon - r \quad (3)$$

Where equation 2 is related to the stress space and equation 3 is related to the strain space. The value of  $\tau_\sigma$  in the equation 2 corresponds to the stress norm while the value of  $\tau_\varepsilon$  in the equation 3 corresponds to the strain norm.

$q(r)$  is the hardening/softening variable; a scalar function defined in the stress space which depends on the magnitude of stresses and it controls the size of the elastic domain. Likewise, when the stress state lies inside  $f$ , the material behaves elastically ( $f < 0$ ). The damage criterion requires that the current stress state must be on ( $f = 0$ ) or inside the damage surface, taking into account an inviscid model.

In the case of the stress space definition, the elastic domain increases or decreases accordingly to the change of  $q(r)$ . On the other side, the strain space domain it always increases or remains constant, this is due to the fact that physically a softening is directly related to a decrease in the strength of the material to resist stresses, this is why  $q(r)$  is also referred as *stress-like* variable. The hardening/softening parameter  $H$  which is the rate of change of the hardening/softening variable, is characterized by:

- Damage with Hardening  $H(r) > 0$
- Perfect Damage  $H(r) = 0$
- Damage with Softening  $H(r) < 0$

Linear Hardening/Softening Law assuming a linear variation of  $q$  with  $r$  is stated as:

$$q(r) = r_0 + H(r - r_0) \quad (4)$$

$$H = dq(r)/dr \quad (5)$$

The Exponential Hardening/Softening Law can be expressed as:

$$q(r) = q_\infty - (q_\infty - r_0)e^{A(1-r/r_0)} \quad (6)$$

$$H = dq(r)/dr = A \frac{(q_\infty - r_0)}{r_0} e^{A(1-r/r_0)} \quad (7)$$

## 2 Continuum Damage Code

In order to execute the simulation, certain material properties must be given as an input in Matlab routines, these properties are the following:

- Young Modulus ( $E$ )
- Poisson coefficient ( $\nu$ )
- Hardening/softening modulus ( $H$ )
- Uniaxial elastic limit ( $\sigma_u$ )
- Ratio compression/tension ( $n$ )
- Viscosity parameter ( $\eta$ )
- Alpha coefficient ( $\alpha$ )

Furthermore, the user can also choose in between the different options that exists within the code:

- Type of analysis (plane stress, plane strain or 3D)
- Type of damage surface (symmetric model, tensile-damage-only or non-symmetric model)
- Type of Hardening/Softening Law (Linear or Exponential)
- Viscous/Inviscid
- Length of the interval of time ( $T$ )

## 3 Assignment

### 3.1 Rate independent models

In this part two algorithms have to be implemented; the continuum isotropic damage “non-symmetric tension-compression damage” model and the “tension-only” damage model. The first model it is based on the fact that the domain of the elastic region is greater when compression occurs, and the transition between tension to compression zones is linearly controlled by means of two parameters which are the weight factor  $\theta$  and the ratio between compression and tension  $n$ . The second model it takes in account only tensile loads whereas the compressive loads are made 0, this is achieved by using the so-called *Macaulay Brackets*.

Once the previous models are implemented, Linear Law, equation 4, and Exponential Hardening/Softening Law ( $H < 0$  and  $H > 0$ ), equation 6, are needed. For the linear case the Hardening/Softening Parameter ( $H$ ) is constant value, while for the exponential case it depends of the internal variable (equation 7).

Likewise, it is proceeded by obtaining the path at the stress space and the stress-strain curve, corresponding to appropriate loading paths starting at the point  $\sigma_1 = 0 ; \sigma_2 = 0$  and described by three-segment paths in the strain space  $\Delta\varepsilon_1 \rightarrow \Delta\varepsilon_2 \rightarrow \Delta\varepsilon_3$ .

## Inviscid Model (Tension only damage model)

### Case 1

$\Delta\bar{\sigma}_1 = \alpha; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile loading*

$\Delta\bar{\sigma}_1 = -\beta; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile unloading/compressive loading*

$\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = 0$  *Uniaxial compressive unloading/tensile loading*

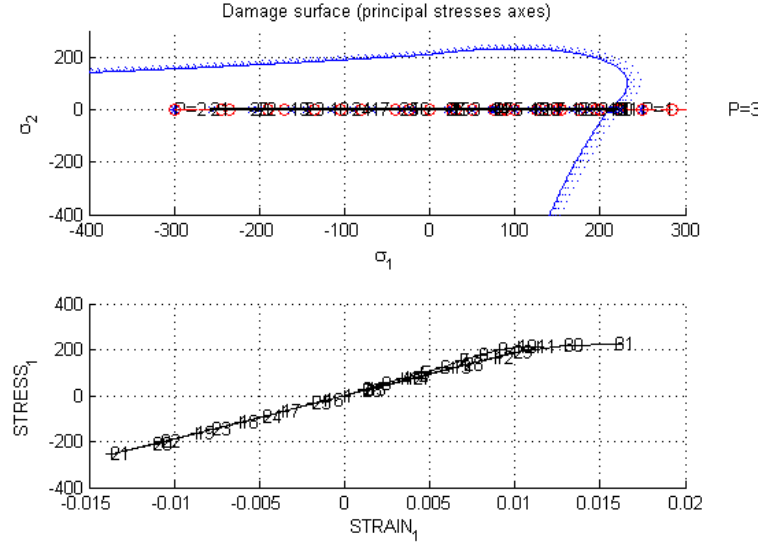


Figure 1: Inviscid Model - Tension only damage model (Case 1)

Figure 12 shows how the material behaves elastically as the tensile load is less than the yield stress, which is fixed up to  $100 \text{ N/m}^2$  for all the next simulations. As the limit is exceeded the elastic domain increases and the material experiences damage. During the next stage (compressive loading) the material experiences compression with no changes of the domain. For the last stress path the elastic domain is exceeded again.

### Case 2

$\Delta\bar{\sigma}_1 = \alpha; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile loading*

$\Delta\bar{\sigma}_1 = -\beta; \Delta\bar{\sigma}_2 = -\beta$  *Biaxial tensile unloading/compressive loading*

$\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = \gamma$  *Biaxial compressive unloading/tensile loading*

### Case 3

$\Delta\bar{\sigma}_1 = \alpha; \Delta\bar{\sigma}_2 = \alpha$  *Biaxial tensile loading*

$\Delta\bar{\sigma}_1 = -\beta; \Delta\bar{\sigma}_2 = -\beta$  *Biaxial tensile unloading/compressive loading*

$\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = \gamma$  *Biaxial compressive unloading/tensile loading*

Where for all cases  $\alpha=250$ ,  $\beta=300$  and  $\gamma=350$ , with linear Hardening Modulus of 0.1.

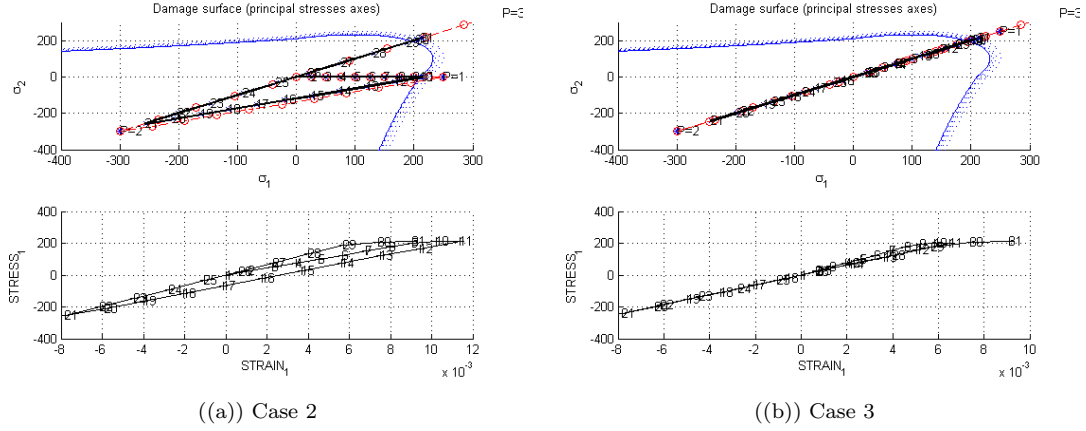


Figure 2: Inviscid Model - Tension only damage model

Figure 2(a) and Figure 2(b) show the same behaviour for all stages, so that the only stress path where no damage occurs is when compression takes place.

**Inviscid Model (Non Symmetric tension compression damage model)**

**Case 1**

- $\Delta\bar{\sigma}_1 = -\alpha; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile loading*
- $\Delta\bar{\sigma}_1 = \beta; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile unloading/compressive loading*
- $\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = 0$  *Uniaxial compressive unloading/tensile loading*

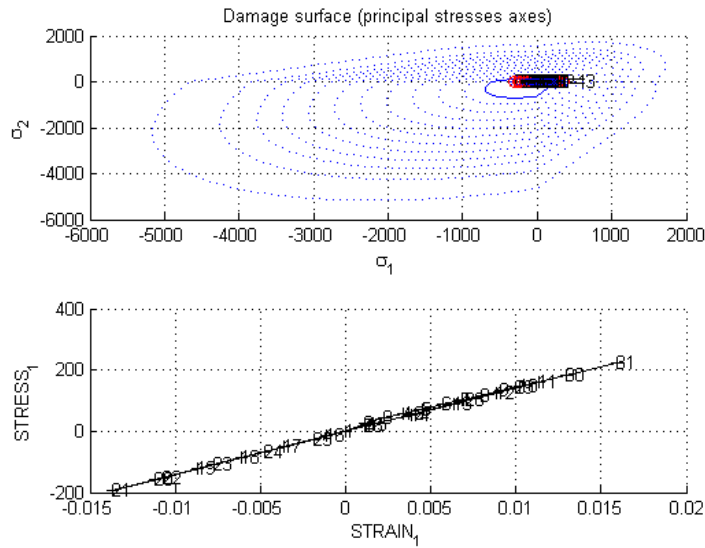


Figure 3: Inviscid Model - Non Symmetric tension compression damage model (Case 1)

Figure 3 shows as for the Case 1 (Tension only damage model) that the material behaves elastically as the tensile load is less then the yield stress. As the the limit is exceed the elastic domain increases and the material experient deformation. During the next stage (compressive loading) the material experience compression with no domain size changes and at the last stage elastic domain grows again because yield stress is exceeded again.

**Case 2**

$\Delta\bar{\sigma}_1 = \alpha; \Delta\bar{\sigma}_2 = 0$  *Uniaxial tensile loading*  
 $\Delta\bar{\sigma}_1 = -\beta; \Delta\bar{\sigma}_2 = -\beta$  *Biaxial tensile unloading/compressive loading*  
 $\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = \gamma$  *Biaxial compressive unloading/tensile loading*

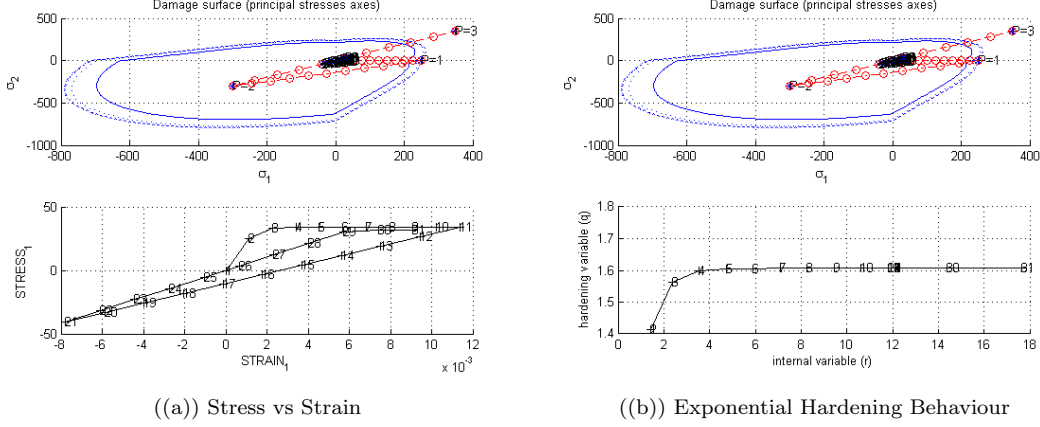


Figure 4: Inviscid Model - Non Symmetric tension compression damage model (Case 2)

Figure 4(a) shows a deformation of the domain during the first stage (Uniaxial tensile loading and also during the last stage (Biaxial tensile loading). Figure 4(b) shows an Exponential Hardening Behaviour which relates the Damage variable (q) with the Internal Variable (r). We can see that the function is bounded as the Internal Variable goes to infinity.

**Case 3**

$\Delta\bar{\sigma}_1 = \alpha; \Delta\bar{\sigma}_2 = \alpha$  *Biaxial tensile loading*  
 $\Delta\bar{\sigma}_1 = -\beta; \Delta\bar{\sigma}_2 = -\beta$  *Biaxial tensile unloading/compressive loading*  
 $\Delta\bar{\sigma}_1 = \gamma; \Delta\bar{\sigma}_2 = \gamma$  *Biaxial compressive unloading/tensile loading*

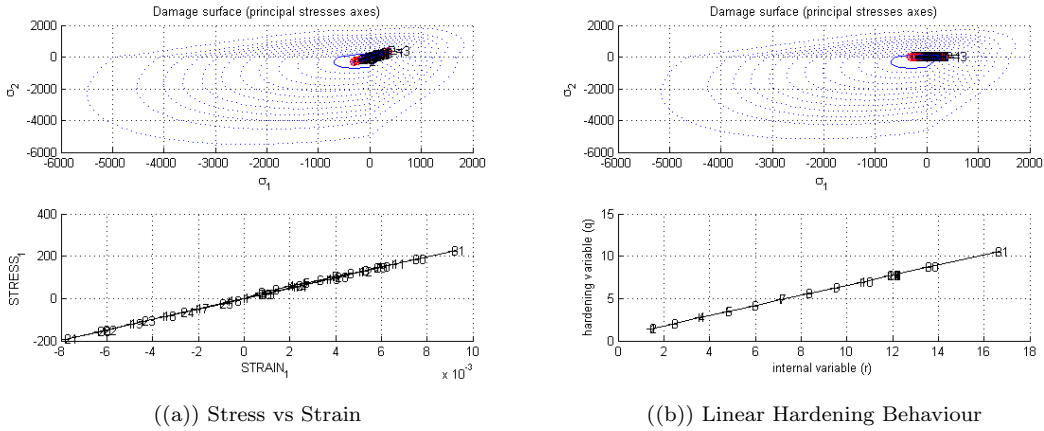


Figure 5: Inviscid Model - Non Symmetric tension compression damage model (Case 3)

Figure 5(a) and Figure 5(b) show a Linear Hardening Behaviour which relates the Damage vari-

able (q) with the Internal Variable (r), which means that the material experiment hardening and the domain increase.

Where for the all cases  $\alpha=250$ ,  $\beta=300$  and  $\gamma=350$ , with linear Hardening Modulus of 0.5.

### 3.2 Part 2 (rate dependent models)

The second part of the assignment include the integration algorithm (plane strain case) for the continuum isotropic visco-damage “symmetric tension-compression” model. Once the previous implementation is done the following cases (for a specific given Poisson ratio and linear hardening/softening parameter) are considered:

- Different viscosity parameters  $\eta$ : [0 ,0.8]
- Different strain rate  $\dot{\epsilon}$
- Different  $\alpha$  values:  $\alpha = [0, 1/4, 1/2, 3/4, 1]$  (for the time-integration method)

And the next results are obtained:

- The effects of the previous values on the obtained stress-strain curves in appropriate loading paths
- The effects of the  $\alpha$  values, on the evolution along time of the  $C_{11}$  component of the tangent and algorithmic constitutive operators

### Results

For the viscosity parameters  $\eta=0$  and for different values of  $\alpha$  we get:

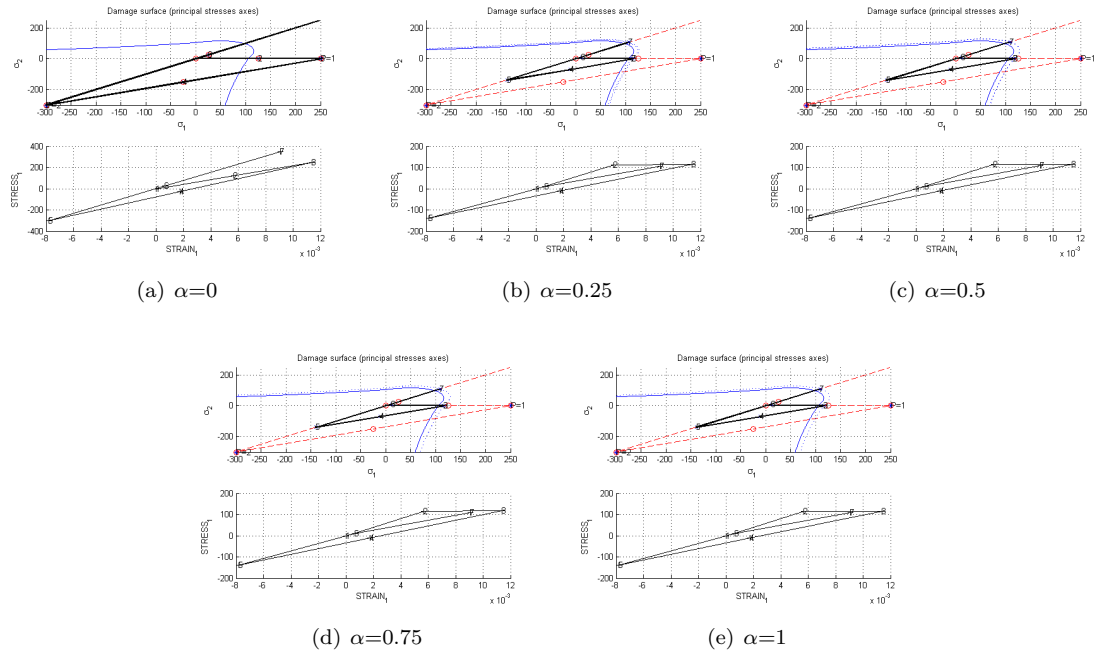


Figure 6: Results for different values  $\alpha$  coefficient and  $\eta=0$

And for the viscosity parameters  $\eta=0.8$  and for different values of  $\alpha$  we get:

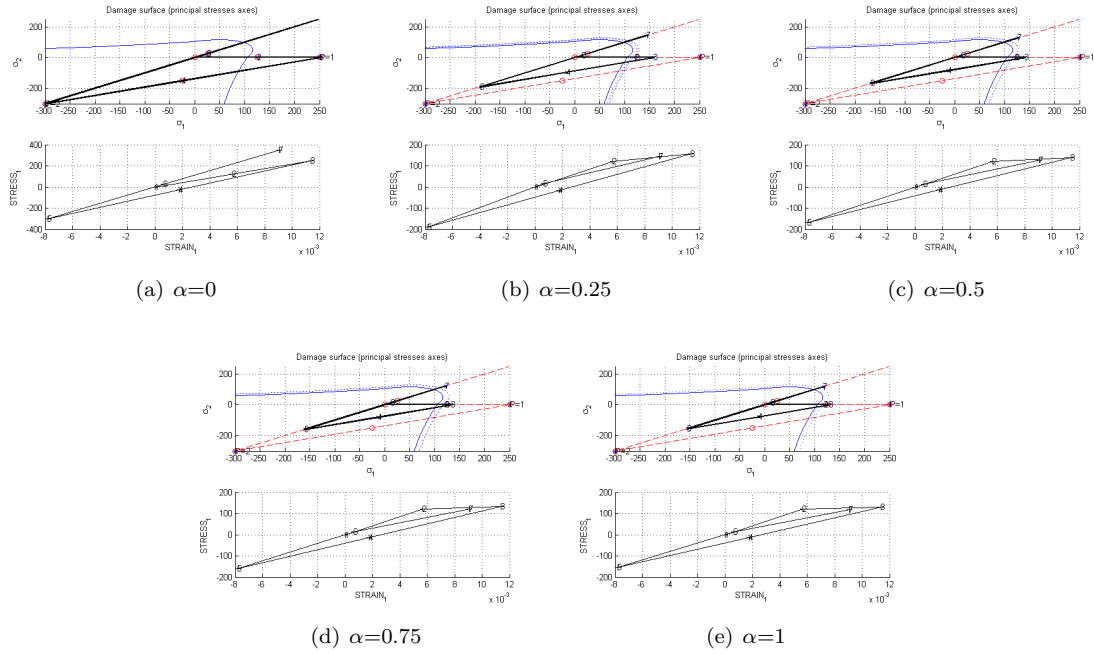


Figure 7: Results for different values  $\alpha$  coefficient and  $\eta=0.8$

Figure 6 and Figure 7, compared with the inviscid case, shown how  $\eta$  and  $\alpha$  determine the behaviour of the numerical simulation. As we know from the theory  $\alpha$  is important in order to prevent instabilities. For  $\alpha=[1/2,1]$  the solution should not produce instability. Small values  $\eta$  also produce instability. For  $\eta=0$  and  $\alpha=1$  the inviscid case is recovered with implicit integration.

In order to obtain results for different strain rate values we just need to change the total time in the main code. The simulations has be done for the next total time values (s): [20,50,100].

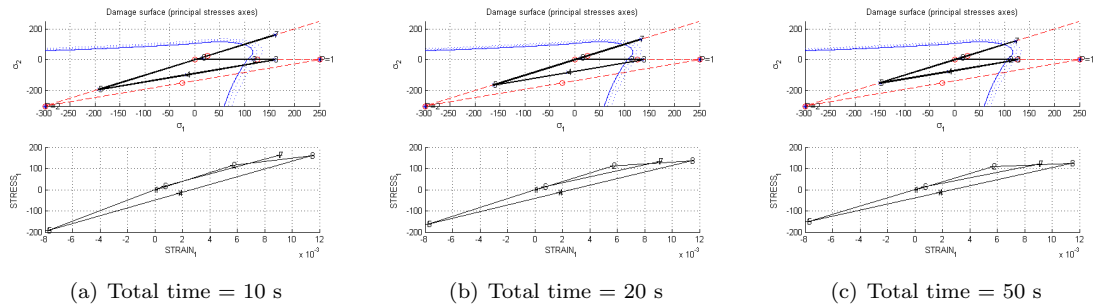


Figure 8: Results for  $\alpha=0.5$  coefficient and  $\eta=0.4$

Figure 8, compared with the inviscid case, shows that as the total time increases, the viscous solution tends to the inviscid one, which means the small velocity of the applied load.



### 3.3 Annexes

Implementation of the continuum isotropic damage “non-symmetric tension-compression damage” model and the “tension-only” damage model.

```
stress = ce * eps_n1';
stress_plus = stress;
%positive counterpart of a stress tensor
stress_plus(stress_plus < 0) = 0;
%inverse of ce
ce_inv=inv(ce);

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

elseif (MDtype==2) %* Only tension
    rtrial= sqrt(eps_n1*stress_plus);

elseif (MDtype==3) %*Non-symmetric
    num = sum(stress_plus)-stress_plus(4);
    den = sum(abs(stress))- abs(stress(4));
    theta = num/den;
    rtrial = (theta + (1-theta)/n) * sqrt(eps_n1 * ce*eps_n1');
end
```

Figure 9

Implementation of the linear and Exponential Hardening/Softening Law ( $H < 0$  and  $H > 0$ )

```
if viscopr == 0
    if(rtrial > r_n)
        %* Loading
        fload=1;
        delta_r=rtrial-r_n;
        r_n1= rtrial ;

    if hard_type == 0
        % Linear
        q_n1= q_n+ H*delta_r;
    else
        if H>0
            q_n1 = 2*r0-(2*r0-r0)*exp(1-rtrial/r0);
        else
            q_n1 = zero_q-(zero_q-r0)*exp(1-rtrial/r0);
        end
    end
end
```

Figure 10

## Implementation of the Viscous case

```
else % VISCOUS cases| ('viscrp == 1')
    rtrial=(1-alpha)*r_n+alpha*rtrial;

    if(rtrial > r_n)
        %* Loading

        fload=1;
        delta_r=rtrial-r_n;
        r_n1= ((eta-delta_t*(1-alpha))/(eta+alpha*delta_t))*r_n + ....
            ((delta_t/(eta+alpha*delta_t))*rtrial);
        if hard_type == 0
            % Linear
            q_n1= q_n+ H*delta_r;
        else
            q_infi=r0*1.3;
            H_plus= (A*(q_infi-r0)*exp((1-rtrial/r0)))/r0;
            q_n1= q_n + H_plus*delta_r;
        end
    end
```

Figure 11

## Implementation of the Consistent (algorithmic) tangent operator:

```
dano_n1 = 1.d0-(q_n1/r_n1);
if viscrp == 0
    ce_alg = (1 - dano_n1)*ce;
else

    H_plus= (A*(q_infi-r0)*exp(A*(1-rtrial/r0)))/r0;
    KR = kron(sigma_n1,sigma_n1');
    ce_alg= (1-dano_n1)*ce + alpha*delta_t/(Eprop(7)+alpha*delta_t)...
        *1/rtrial*(H_plus * r_n1-q_n1/(r_n1)^2 * KR);

end
```

Figure 12