

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER OF SCIENCE IN COMPUTATIONAL MECHANICS

COMPUTATIONAL SOLID MECHANICS

Assignment 1

Damage Models

Author:

Carlos Eduardo Ribeiro Santa Cruz Mendoza

April 10, 2019



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**

1. Introduction

Damage mechanics is the field inside continuum mechanics where degradation of materials, namely decrease in stiffness through development of micro-defects under loading, is assessed. Damage models, such as the rate independent and the rate dependent, make an attempt on predicting the behaviour of material properties when subject to different loads and strain rates. They provide simple integration algorithms that, once implemented, allow for a quick understanding of the mechanical response of a damaged material [1].

A fully functioning MATLAB program for modelling damage was provided to aid the visualization of such mechanical responses. The already implemented rate-independent model allows the assessment of materials with a symmetric elastic domain and a linear hardening/softening law. The model is 2D, considering the *plane strain* hypothesis. Once the material properties, the loading path and the strain rate are given, several properties can be analysed via graphical representations such as stress-strain ($\sigma - \varepsilon$) curves, damage (d) over time and hardening variable (q) as a function of the internal variable (r).

The objective was to expand the provided code by implementing the possibility to assess materials which suffer damage only by tensile loads or which have a non-symmetric elastic domain (with a larger yield stress for compression, for example). Furthermore, the exponential hardening/softening law was implemented, together with the visodamage model, which includes materials that respond differently depending on the strain rate.

The material properties considered throughout the analysis is given in Table 1.1. The loading path is given by three segments $\Delta\bar{\sigma}^{(1)}$, $\Delta\bar{\sigma}^{(2)}$ and $\Delta\bar{\sigma}^{(3)}$ which, starting from $\sigma = [0 \ 0]$, define the inputs $\sigma^{(1)}$, $\sigma^{(2)}$ and $\sigma^{(3)}$.

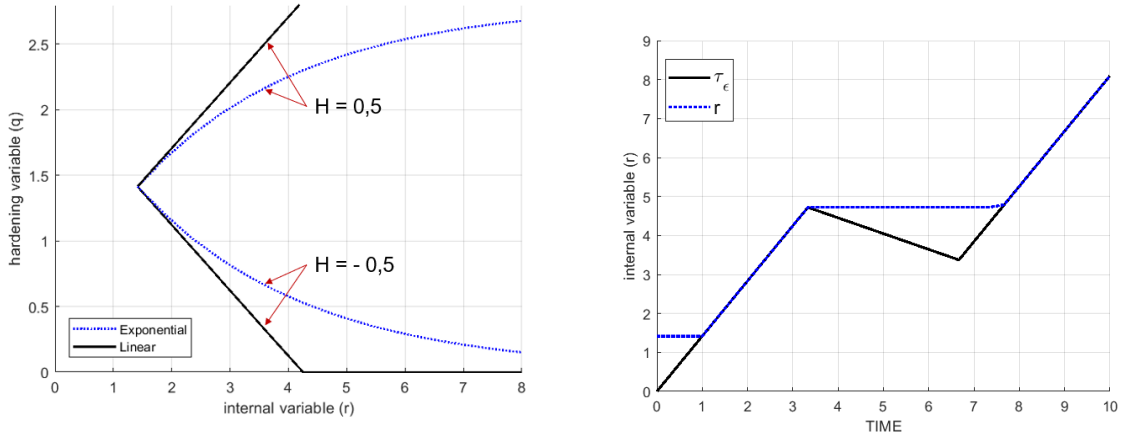
Young's modulus (E)	20 000 Pa
Poisson's ratio (ν)	0.3
Yield stress (σ_y)	200 Pa

2. Rate independent models

2.1. Hardening/Softening law implementation

The first implemented routine was the exponential hardening/softening law. As opposed to the linear law, it provides a smoother variation of the hardening variable as a function of the internal variable until it finally tends asymptotically to a defined value q_∞ , which, for the softening case, is a number just over zero that represents a complete damage. The MATLAB code can be seen on Appendix A.

To assess the correctness, load paths and properties were chosen specifically to yield large damages and, thus, allow the hardening/softening law to be evaluated on a wide range. Figure 2.1 depicts the behaviour of the implemented law against the linear as well as the progress of the internal variable. On Figure 2.1a we see that the hardening variable starts with r_0 and varies in accordance with the literature ([1]) for the exponential case. Specifically for the softening ($H=-0,5$) we note the convergence of the hardening as previously described. The absence of healing when unloading must also hold, therefore Figure 2.1b is presented showing that, as expected, r starts with r_0 and only varies with pure loading, not following the path of the effective strain τ_ϵ .



- (a) Linear and exponential hardening/softening laws with the corresponding hardening/softening parameter H under uniaxial tensile loading $\sigma^{(1)} = [700 \ 0]$, $\sigma^{(2)} = [1200 \ 0]$, $\sigma^{(3)} = [1700 \ 0]$
- (b) Behaviour of r and τ_ϵ over time using the exponential softening law ($H = -0.5$) under uniaxial tensile loading $\sigma^{(1)} = [700 \ 0]$, $\sigma^{(2)} = [500 \ 0]$, $\sigma^{(3)} = [1200 \ 0]$

Figure 2.1.: Assessment of hardening/softening laws using the symmetric damage model

2.2. Damage model implementations

Subsequently, the Tension-Only and the Non-symmetrical damage models were implemented (codes available on Appendix A). To assess the correctness of the implementation, three test cases were settled in terms of the parameters $\alpha = 250$ N, $\beta = 600$ N and $\gamma = 450$ N. The profile of each test case is presented on Table 2.1. Moreover, the softening parameter was set to $H=-0,2$, the exponential softening law was chosen and the ratio of the compression elastic limit and the tensile elastic limit was set to $n=2$.

Table 2.1.: Test cases

	Case 1			Case 2			Case 3		
	$\Delta\bar{\sigma}$	σ	Load	$\Delta\bar{\sigma}$	σ	Load	$\Delta\bar{\sigma}$	σ	Load
(1)	$[\alpha \ 0]$	$[250 \ 0]$	--- ---	$[\alpha \ 0]$	$[250 \ 0]$	--- ---	$[\alpha \ \alpha]$	$[250 \ 250]$	--- ---
(2)	$[-\beta \ 0]$	$[-350 \ 0]$	--- ---	$[-\beta \ -\beta]$	$[-350 \ -600]$	--- ---	$[-\beta \ -\beta]$	$[-350 \ -350]$	--- ---
(3)	$[\gamma \ 0]$	$[100 \ 0]$	--- ---	$[\gamma \ \gamma]$	$[100 \ -150]$	--- ---	$[\gamma \ \gamma]$	$[100 \ 100]$	--- ---

(*i*): Load segment *i*, Tensile: (—), Compressive: (—), Elastic: (—)

Case 1 was used initially to test the correctness of the Tension-Only model. As perceived on Figure 2.2, there's a clear distinction on the σ - ε curves obtained from the symmetric model and the tension-only model. It's clear on Figure 2.2a that the Elastic modulus changes due to both kinds of loading (tension and compression) on the symmetric model, whereas on Figure 2.2b we notice a change only after tensile loading.

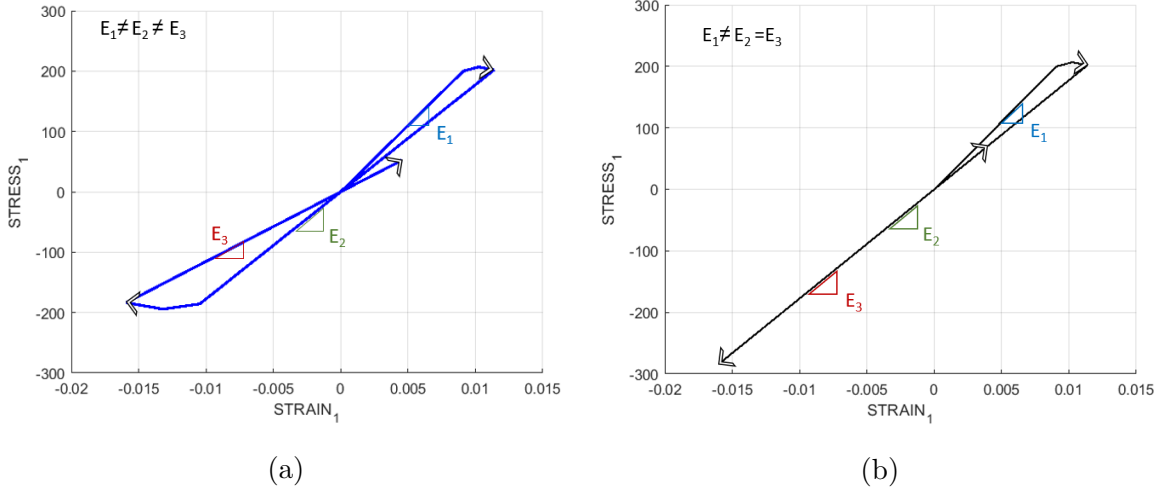


Figure 2.2.: Comparison of the σ - ε curves obtained with the Symmetric (a) and Tension-only (b) damage models

The differences are more evident when we analyse Figure 2.3. We notice on the stress space depicted on Figure 2.3a that the load segment ending on $P = 2$ leaves the elastic domain only on the symmetric case, given that the elastic domain of the tension-only model has no boundaries for compression. Therefore, the model works as expected yielding, thus, the smaller damage shown on Figure 2.3b.

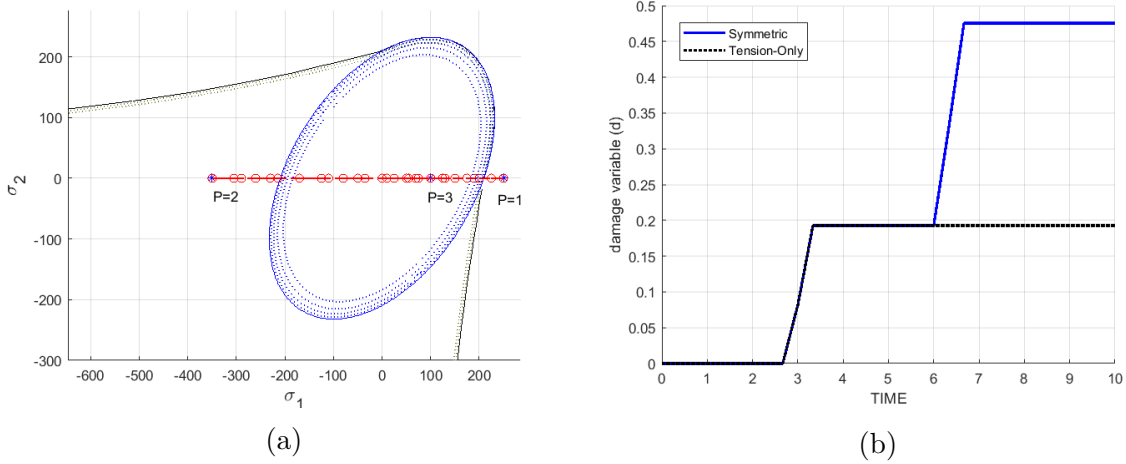


Figure 2.3.: Comparison of the elastic domain and its corresponding damage surface on the stress space (a) and of the damage evolution (b) for the Symmetric (—) and Tension-Only (—) damage models

Similarly, Case 2 was served as input to compare all three damage models. By observing the stress space obtained for the three models on Figure 2.4 we see, firstly, that the elastic domain of the non-symmetric model is in accordance with the literature and with the ratio of the compression elastic limit and the tensile elastic limit ($n=2$). Furthermore, we see that while on the first load segment there are virtually no differences between the effective stress paths, when approaching the load point $P=2$ the models start to diverge. This highlights the correct implementation that captures the differences of each model to deal with compression, leading to an increasing damaged surface from the symmetric, through the non-symmetric to the tension-only models.

These distinctions can also be visualized on Figure 2.5, where the final inclination of each σ - ε curve on Figure 2.5a, i. e. the elastic modulus, is different due to the different damages inflicted during compression, as seen on Figure 2.5b.

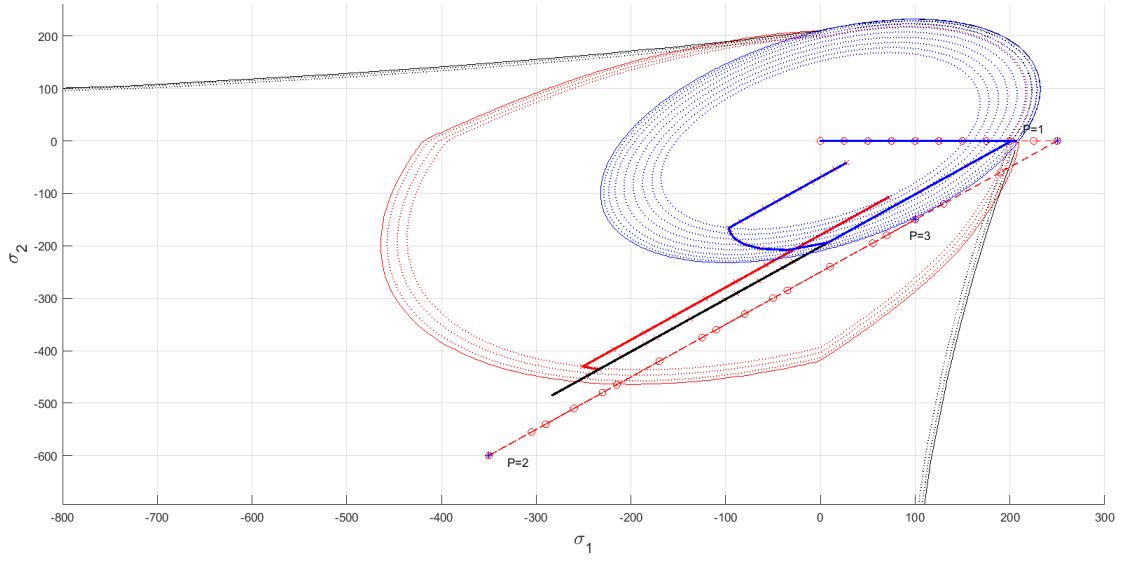


Figure 2.4.: Comparison of the elastic domain and its corresponding damage surface on the stress space for the Symmetric (—), Tension-Only (—) and Non-symmetric (—) damage models.

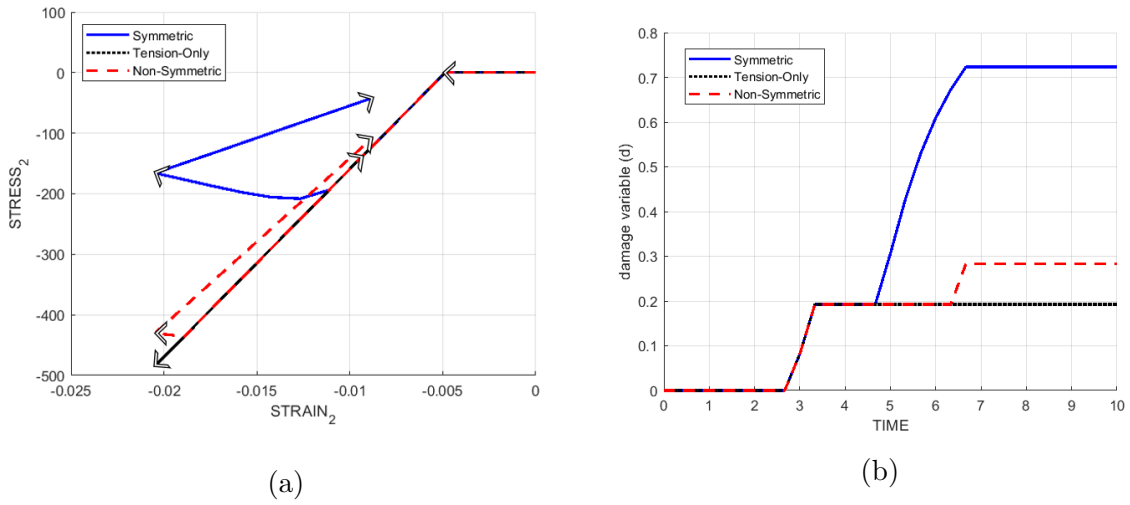


Figure 2.5.: Comparison of the σ_2 - ε_2 curves (a) and of the damage evolution (b) for the Symmetric, Tension-Only and Non-symmetric damage models.

As a final verification, Case 3 is tested. Observing the stress space on Figure 2.6 we note that the second load segment never crosses the elastic domain of the non-symmetric model. Therefore, the tension-only and the non-symmetrical models must respond the same when it comes to damage. As expected, Figure 2.7 displays the curves of both models over one another, indicating the same damage to the applied load.

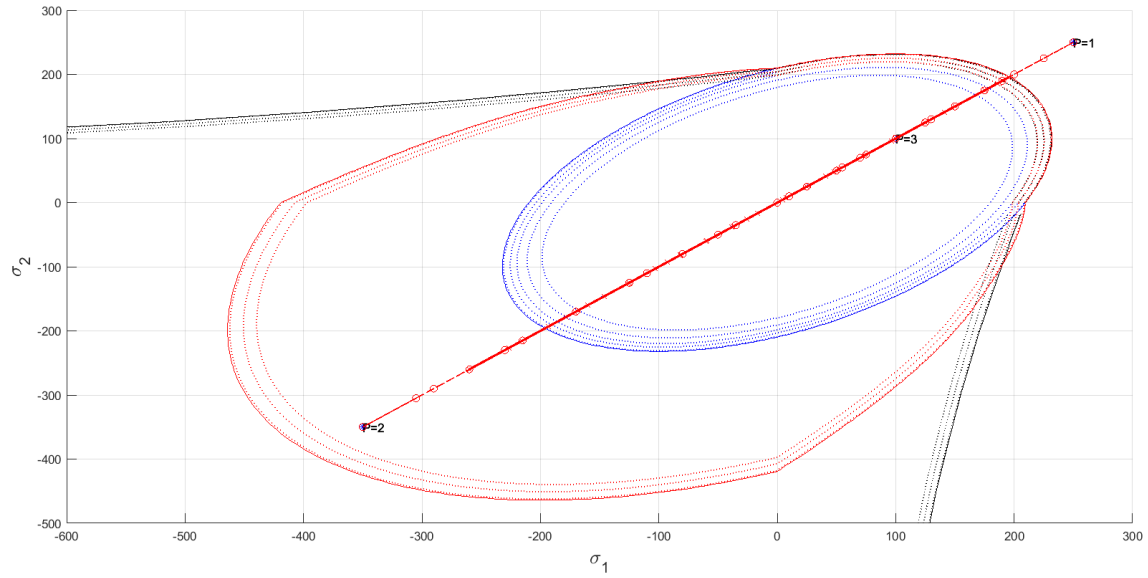


Figure 2.6.: Comparison of the elastic domain and its corresponding damage surface on the stress space for the Symmetric (—), Tension-Only (—) and Non-symmetric (—) damage models.

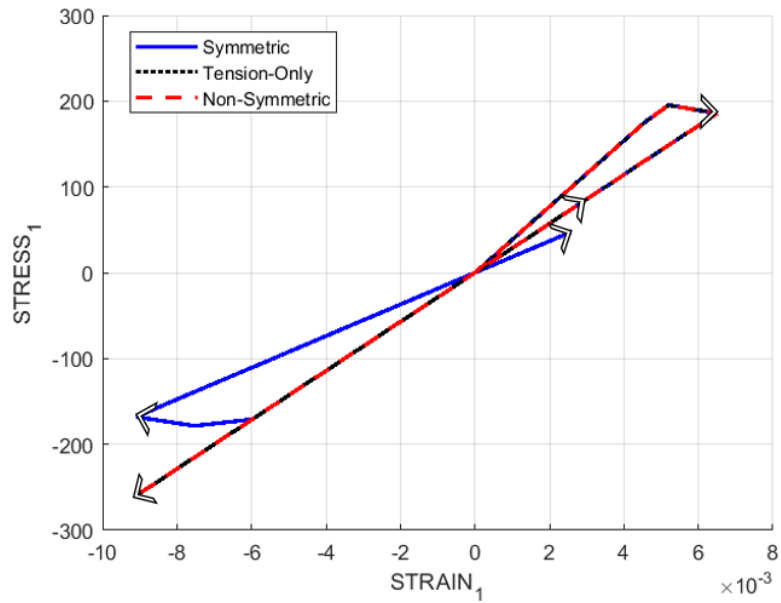


Figure 2.7.: Comparison of the σ_1 - ϵ_1 curves for the Symmetric, Tension-Only and Non-symmetric damage models.

3. Rate dependent models

Finally, the rate dependent model was implemented using the code available on Appendix B. The also called viscous model introduces the effect of strain rate, which, in practice, provides a smoother transition from a undamaged to a damaged material. The viscosity η specifies how smoothly this transition will happen (the rate independent models can be seen as having $\eta=0$) and the strain rate ($\dot{\epsilon}$) can be interpreted as velocity the load paths will be traveled with. The type of integration of the damage variables for the viscous case depend of the parameter α , typical values are $\alpha=0$ (explicit), $\alpha=1$ (implicitly) and $\alpha=0,5$ (Crank-Nicolson).

In practice, a material with higher viscosity will yield a lower damage comparatively if applying the same strain rate. Figure 3.1 shows exactly that, confirming the correct implementation. As the viscosity grows, a higher stress is needed for the same strain (after the yield stress), since the damage is not inflicted with the same velocity.

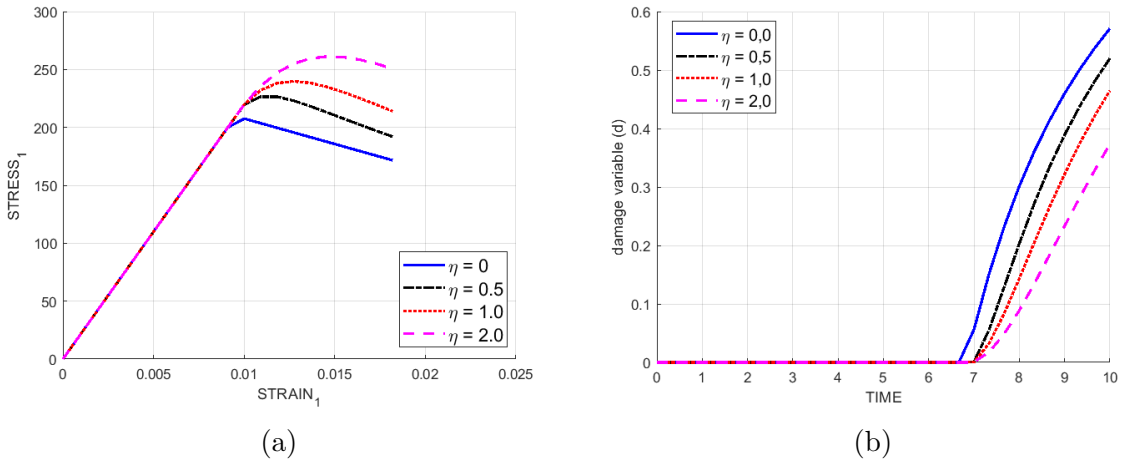


Figure 3.1.: Comparison of the σ_1 - ϵ_1 curves (a) and of the damage evolution (b) for different viscosities assuming the symmetric model with a uniaxial tensile loading of $\boldsymbol{\sigma}^{(1)} = [100 \ 0]$, $\boldsymbol{\sigma}^{(2)} = [200 \ 0]$, $\boldsymbol{\sigma}^{(3)} = [400 \ 0]$, $H = -0,2$, exponential softening law, $\alpha = 0.5$ and $\dot{\epsilon} = 3,3 \cdot 10^{-3}$

The effect of viscosity is also evident on the stress space presented on Figure 3.2. We note that the effective stress path surpasses the elastic boundary for the viscous case, “accepting” a higher stress with a lower damage surface.

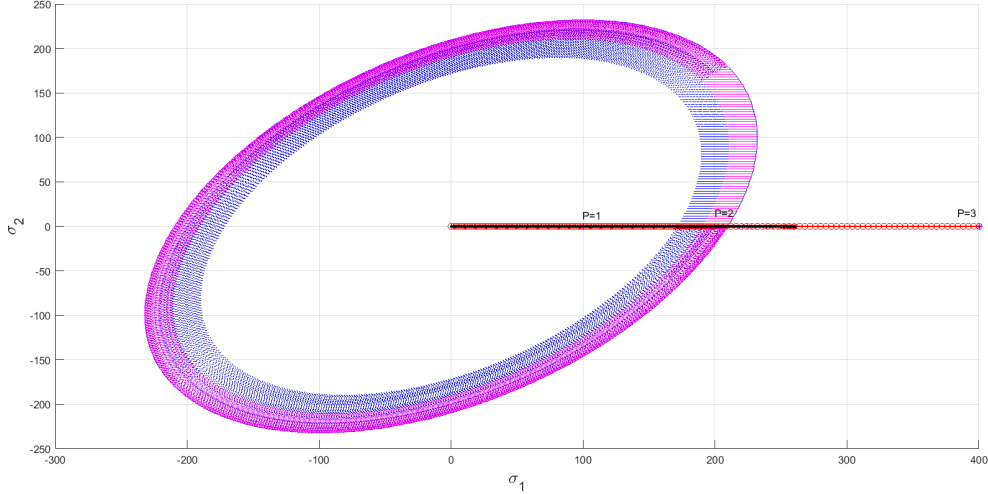


Figure 3.2.: Comparison of the damage surface on the stress space for a rate independent model (—) and a rate dependent model with $\eta = 2$ (---).

The strain rate imposes a similar relationship to the σ - ε curve. As documented on the literature ([2, 3]), a high strain rate yields to a elevation of the yield stress due to the lack of time for the dislocations to move, not allowing damage/plastic deformation and maximizing the elastic behaviour. Accordingly, the model translates this behaviour by pushing the curve upwards after the yield stress, implying also, a lower damage. The strain rate effect was, thus, well reproduced by the code as shown on both Figure 3.3 and Table 3.1.

Table 3.1.: Effect of $\dot{\varepsilon}$ on damage variable and constitutive tensor component

	$\dot{\varepsilon} = 1,1 \cdot 10^{-3}$	$\dot{\varepsilon} = 1,1 \cdot 10^{-2}$	$\dot{\varepsilon} = 1,1 \cdot 10^{-1}$
C_{11}	$7,3 \cdot 10^3$	$1,8 \cdot 10^4$	$2,5 \cdot 10^4$
d	0,54	0,29	0,05
Original: $C_{11} = 2,7 \cdot 10^4$, $d = 0$			

As we've seen, the the stiffness of a material diminishes once damage is inflicted. The constitutive elastic tensor, thus, varies with time for the rate-dependent case. On the other hand, the integration algorithm can result to be unstable, specially for high time-steps [4] (a high strain-rate without the appropriate total time of the simulation might result in high time steps). The stability depends moreover on the α parameter, being unconditionally stable for values of $\alpha > 0,5$. The Figure 3.4 shows that, in fact, there's a risk of instability in the calculus of the constitutive tensor for $\alpha < 0,5$. Both the algorithmic (Figure 3.4a) and the tangent (Figure 3.4b) elastic tensor displayed oscillations for $\alpha = 0,25$, while the other values converged smoothly. Even smaller oscillations like the one obtained for the tangent moduli might impose great problems for

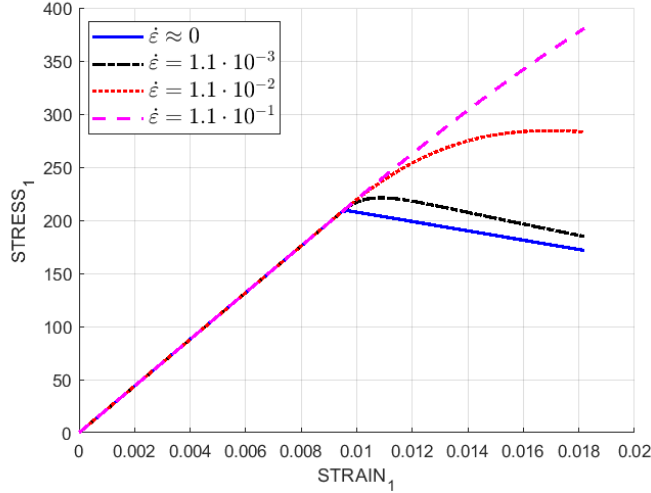


Figure 3.3.: Comparison of the σ_1 - ϵ_1 curves for different strain rates assuming the symmetric model with a uniaxial tensile loading of $\boldsymbol{\sigma}^{(1)} = [100 \ 0]$, $\boldsymbol{\sigma}^{(2)} = [200 \ 0]$, $\boldsymbol{\sigma}^{(3)} = [400 \ 0]$, $H = -0,2$, exponential softening law, $\alpha = 0.5$ and $\eta = 1$.

a simulation, specially when considering them being added over thousands of elements in a Finite Element simulation. Hence, when bigger time steps are required due to computational cost limitations, the Crank-Nicolson or the Implicit method are suggested.

It's also worth noticing that, as expected, the algorithmic and tangent moduli are the same for $\alpha = 0$

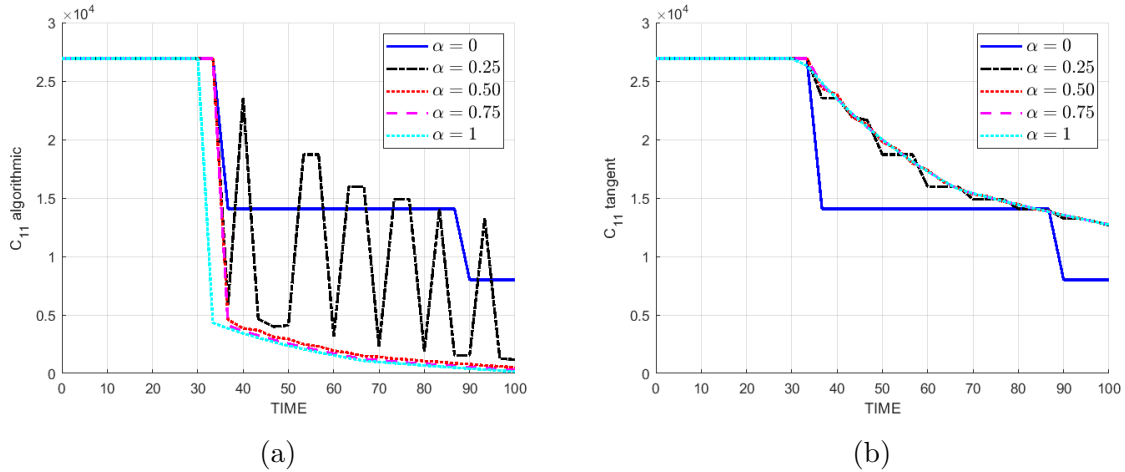


Figure 3.4.: Comparison of the algorithmic (a) and tangent (b) C_{11} components of the constitutive elastic tensor for different α values assuming the symmetric model with a biaxial tensile loading $\boldsymbol{\sigma}^{(1)} = [200 \ 200]$, $\boldsymbol{\sigma}^{(2)} = [300 \ 300]$, $\boldsymbol{\sigma}^{(3)} = [350 \ 350]$, $H = -0,2$, exponential softening law, $\eta = 0.1$ and $\dot{\epsilon} = 3, 3 \cdot 10^{-4}$

Bibliography

- [1] E.W.V. Chaves. *Notes on Continuum Mechanics*. Lecture Notes on Numerical Methods in Engineering and Sciences. Springer Netherlands, 2013.
- [2] T. Belytschko, W.K. Liu, B. Moran, and K. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. No Longer used. Wiley, 2013.
- [3] W.D. Callister and D.G. Rethwisch. *Fundamentals of Materials Science and Engineering: An Integrated Approach*. Wiley, 2012.
- [4] J. Lubliner. *Plasticity Theory*. Dover books on engineering. Dover Publications, 2008.

Appendix A.

Implemented damage routines

Exponential implementation

```
1 r0 = sigma_u/sqrt(E);
2 zero_q=1.d-6*r0;
3 % hardening/softening parameter H is user defined
4 A=abs(H);
5 q_inf=r0+sign(H)*0.99*r0;
6 % user defined variable to choose hardening/softening law
7 if hard_type == 0
8     % Linear
9     q_n1= q_n+ H*delta_r;
10 else
11     % Exponential
12     q_n1= q_inf-(q_inf-r0)*exp(A*(1-r_n1/r0));
13 end
```

Plot damage surface criterion

```
1 % MDtype is a user defined variable to switch between:
2 % 1 - Symmetric ; 2 - Tension-only ; 3 - Non-symmetric
3 tetha=[0:0.01:2*pi];
4 D=size(tetha);
5 m1=cos(tetha);
6 m2=sin(tetha);
7 Contador=D(1,2);
8 radio = zeros(1,Contador) ;
9 s1     = zeros(1,Contador) ;
10 s2    = zeros(1,Contador) ;
11
12 if MDtype==1 % Symmetric
13     for i=1:Contador
14         radio(i)= q/sqrt([m1(i) m2(i) 0 nu*(m1(i)+m2(i))]*ce_inv ...
15             *[m1(i) m2(i) 0 nu*(m1(i)+m2(i))]' );
16         s1(i)=radio(i)*m1(i);
17         s2(i)=radio(i)*m2(i);
18     end
```

```

19     hplot =plot(s1,s2,tipa_linea);
20 elseif MDtype==2      % Tension-only
21     m1_p = zeros(1,Contador) ; % admits only positive values
22     m2_p = zeros(1,Contador) ; % admits only positive values
23     for i=1:Contador
24         m1_p(i)=m1(i);
25         m2_p(i)=m2(i);
26         if m1(i)<0
27             m1_p(i)=0;
28         end
29         if m2(i)<0
30             m2_p(i)=0;
31         end
32         radio(i)= q/sqrt([m1_p(i) m2_p(i) 0 nu*(m1_p(i)+ ...
33         m2_p(i))]*ce_inv*[m1(i) m2(i) 0 nu*(m1(i)+m2(i))] ');
34         s1(i)=radio(i)*m1(i);
35         s2(i)=radio(i)*m2(i);
36     end
37     hplot =plot(s1,s2,tipa_linea);
38 elseif MDtype==3      % Non-symmetric
39     for i=1:Contador
40         sum_stress=0;
41         vec_stress=[m1(i) m2(i) nu*(m1(i)+m2(i))];
42         sum_stress_abs=sum(abs(vec_stress));
43         for j=1:3
44             if vec_stress(j)<0
45                 vec_stress(j)=0;
46             end
47             sum_stress=sum_stress+vec_stress(j);
48         end
49         theta_stress=sum_stress/sum_stress_abs;
50         coeft=theta_stress+(1-theta_stress)/n;
51
52         radio(i)= q/(coeft*(sqrt([m1(i) m2(i) 0 nu*(m1(i)+ ...
53         m2(i))]*ce_inv*[m1(i) m2(i) 0 nu*(m1(i)+m2(i))] ')));
54         s1(i)=radio(i)*m1(i);
55         s2(i)=radio(i)*m2(i);
56     end
57     hplot =plot(s1,s2,tipa_linea);
58 end

```

Defining damage criterion surface

```
1 if (MDtype==1)      /* Symmetric
2   rtrial= sqrt(eps_n1*ce*eps_n1');
3 elseif (MDtype==2) /* Only tension
4   sigma_n1 =ce*eps_n1';
5   for j=1:4
6     if sigma_n1(j)<0
7       sigma_n1(j)=0;
8     end
9   end
10  cei=inv(ce);
11  eps_n1m=(cei*sigma_n1)';
12  rtrial= sqrt(eps_n1m*ce*eps_n1m');
13 elseif (MDtype==3) /*Non-symmetric
14  sigma_n1 =ce*eps_n1';
15  sum_stress=0;
16  vec_stress=[sigma_n1(1) sigma_n1(2) sigma_n1(4)];
17  sum_stress_abs=sum(abs(vec_stress));
18  for j=1:3
19    if vec_stress(j)<0
20      vec_stress(j)=0;
21    end
22    sum_stress=sum_stress+vec_stress(j);
23  end
24  theta_stress=sum_stress/sum_stress_abs;
25  coeft=theta_stress+(1-theta_stress)/n;
26  rtrial= coeft*sqrt(eps_n1*ce*eps_n1');
27 end
```

Appendix B.

Implemented visco-damage routines

Integration algorithm

```
1  Damage surface
2  [rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);
3  [tau_eps_n]=Modelos_de_dano1(MDtype ,ce ,eps_n ,n);
4  [tau_eps_n1]=Modelos_de_dano1(MDtype ,ce ,eps_n1 ,n);
5  [rtrial]=(1-alpha)*tau_eps_n+alpha*tau_eps_n1;
6  Ver el Estado de Carga
7  -----> fload=0 : elastic unload
8  -----> fload=1 : damage
9  fload=0;
10 if(rtrial > r_n)
11     Loading
12     fload=1;
13     r_n1= ((eta-delta_t*(1-alpha))/(eta+alpha*delta_t))* ...
14     r_n+(delta_t/(eta+alpha*delta_t))*rtrial;
15     delta_r=r_n1-r_n;
16
17     ...
18
19
20 else
21     Elastic load/unload
22     fload=0;
23     r_n1= r_n ;
24     q_n1= q_n ;
25 end
26 Damage variable
27 dano_n1 = 1.d0-(q_n1/r_n1);
28 Computing stress
29 sigma_n1 =(1.d0-dano_n1)*ce*eps_n1 ';
```

Tangent and Algorithmic constitutive tensors

```
1 aux_var(1) = fload; % 1 for pure loading, 0 otherwise
2 aux_var(2) = q_n1/r_n1;
3 aux_var(3) = (q_n1-H*r_n1)/r_n1^3;
4 Ce_tan=(1-dano_n1)*ce;
5 Ce_alg=Ce_tan-aux_var(1)*((alpha*delta_t)/(eta+alpha ...
6 *delta_t)*aux_var(3)*r_n1/(sqrt(sigma_n1'*(inv(ce)) ...
7 *sigma_n1)/(1-dano_n1))*((ce*eps_n1')*(ce*eps_n1')));
```