UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER IN COMPUTATION MECHANICS AND NUMERICAL METHODS IN ENGINEERING

COMPUTATIONAL SOLID MECHANICS

ASSIGNMENT 1

by

Renan Alessio

Barcelona, March of 2020

INDEX

1- Introduction

The first goal of the assignment is to implement and apply tension-only and non-symmetric tension-compression damage models with exponential hardening/softening law. Both damage models were evaluated under inviscid conditions considering three different stress paths. The second goal of the assignment was to implement and evaluate the viscous damage model (rate dependent). Such viscous damage model was evaluated with a symmetric elastic domain and linear hardening/softening law. Parameters such as viscosity, strain rate and the coefficient alpha were varied to analyze the response of the viscous damage model. An evaluation of the evolution of component C11 from tangential and algorithmic constitutive tensors along time was also considered for different values of the coefficient alpha.

2 – PART I (Rate Independent Model)

For the first part of the assignment, the tensile and the non-symmetric tension-compression damage models were required to be implemented as options for the characterization of the elastic domain (Appendix A). Figure 1 depicts both implemented elastic domains considering a Young Modulus of 20000 MPa and a Yield Stress of 200 MPa.

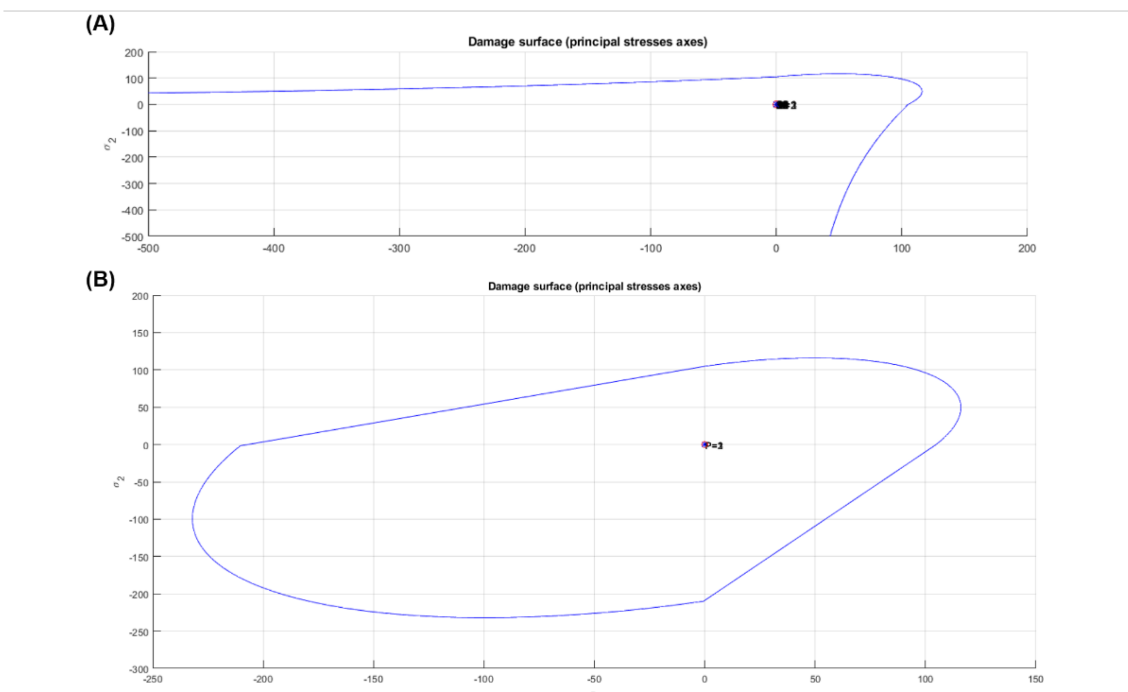

Figure 1. Elastic domain for (A) tensile damage model and (B) non-symmetric tension-compression damage model.

The exponential law to characterize the hardening/softening behavior was also required to be implemented. To present the correctness of such implementation along with the implemented elastic domains, three different stress paths will be considered. The stress paths were defined according to what was specified in the assignment [1]. The following three sub-sections will present and evaluate the stress paths.

2.1 – Stress Path 1

The chosen elastic domain and hardening law behavior parameters for Stress Path 1 are presented in Table 1. The values for the stress path parameters α, β and γ for such case are presented in Table 2.

Table 1. Parameters of elastic domain and hardening law behavior for Stress Path 1.

Young Modulus [MPa]	**Yield Stress [MPa]**	**Elastic Domain**	**Hardening Law**	**Parameter A**	**Parameter q_∞**	**Behavior**
20000	200	Tensile damage model	Exponential	1	0.1	Softening

Table 2. Parameters α, β and γ for Stress Path 1.

Stress Path Parameter α [MPa]	**Stress Path Parameter β [MPa]**	**Stress Path Parameter γ [MPa]**
250	950	1100

Considering the data from Tables 1 and 2, Figure 2 presents the stress path in stress space, stress-strain curve and damage evolution along time obtained for Stress Path 1.

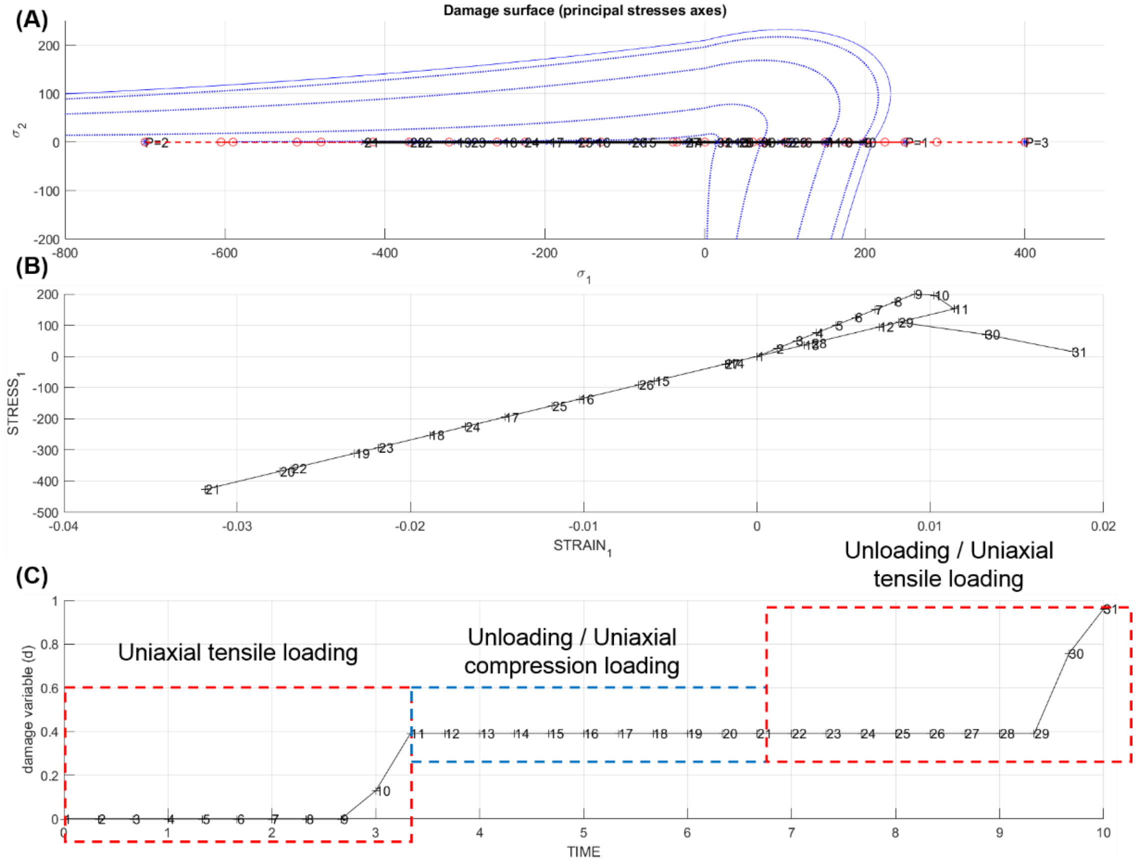

Figure 2. (A) Damage surface, (B) stress-strain curve and (C) damage evolution along time for Stress Path 1.

According to Figure 2(A), 2(B) and 2(C), the tension only damage model and exponential law for softening were implemented correctly. Such observation is based on the reduction of the elastic domain (softening) and reduction of the Young Modulus (uniaxial loading) as damage evolved (reduction of the slope when unloading phase started). Also, during the unloading/uniaxial compression loading increment (between points 11 and 21), there was no evolution of damage and no change in the Young Modulus. Both observations are also in agreement with the model supposed to be implemented.

2.2 – Stress Path 2

The chosen elastic domain and hardening law behavior parameters for Stress Path 2 are presented in Table 3. The values for the stress path parameters α, β and γ for such case are presented in Table 4.

Table 3. Parameters of elastic domain and hardening law behavior for Stress Path 2.

Young Modulus [MPa]	Yield Stress [MPa]	Elastic Domain	Ratio n	Hardening Law	Parameter A	Parameter q_∞	Behavior
20000	100	Non-symmetric	2	Exponential	0.5	2	Hardening

Table 4. Parameters α, β and γ for Stress Path 2.

Stress Path Parameter α [MPa]	Stress Path Parameter β [MPa]	Stress Path Parameter γ [MPa]
300	1000	1500

Considering the data from Tables 3 and 4, Figure 3 presents the stress path in stress space, stress-strain curve and damage evolution along time obtained for Stress Path 2.

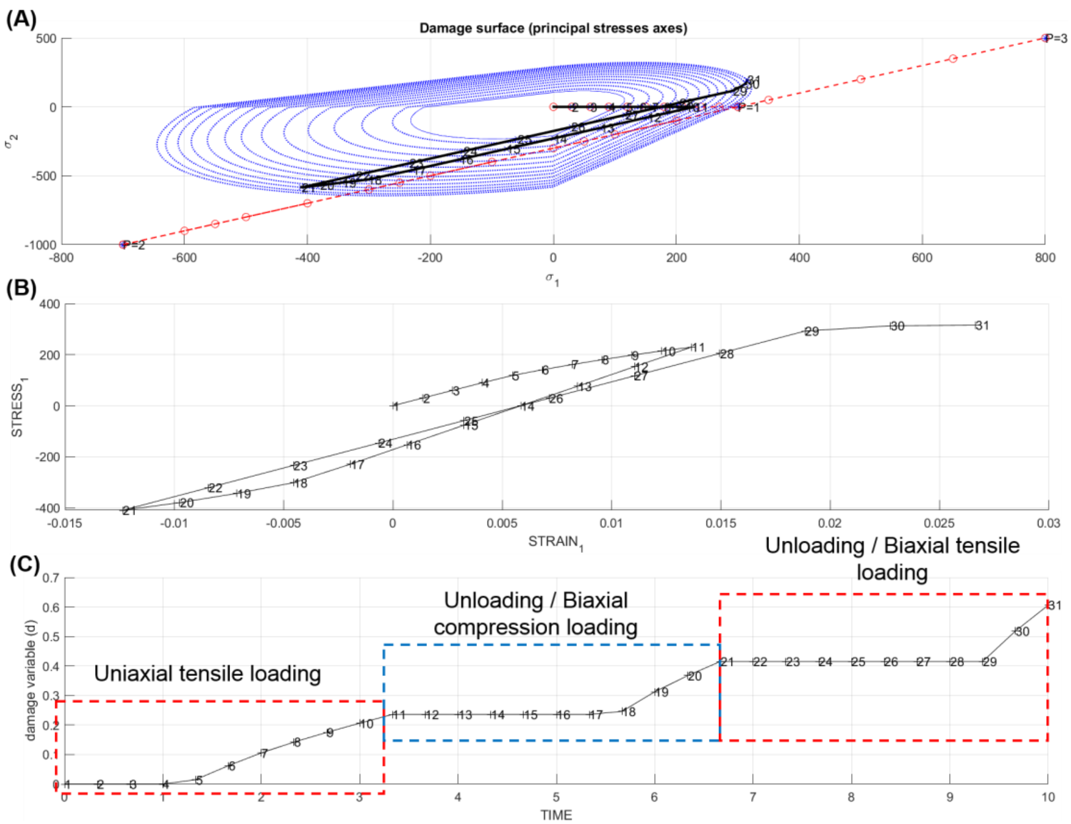

Figure 3.(A) Damage surface, (B) stress-strain curve and (C) damage evolution along time for Stress Path 2.

According to Figure 3(A), 3(B) and 3(C), the non-symmetric tension-compression damage model together with the exponential law for hardening were implemented correctly. Such observation is based on the enlargement of the elastic domain (hardening) and reduction of the stiffness of the material (biaxial loading) as the damage grew along time. Also, during the unloading/biaxial compression loading increment (between points 18 and 21 in Figures 3 (C)), there was an increase of the damage since the compression stress reached the damage surface. As expected, the material underwent hardening under compression, as it is depicted in Figure 2(B) between points 9 and 11. It is also worth mentioning the hardening behavior in Figure 2(B) between points 15-16. Between such points, the internal variable q(r) is converging to the value q_∞ and the increase in stress is greatly reduced while the strain grows rapidly. In such case, the stress will converge to a certain value related to q_∞ while the strain will continue to grow until the damage parameter d reaches the value 1.

2.3 – Stress Path 3

The chosen elastic domain and hardening law behavior parameters for Stress Path 3 are presented in Table 5. The values for the stress path parameters α, β and γ for such case are presented in Table 6.

Table 5. Parameters of elastic domain and hardening law behavior for Stress Path 2.

Young Modulus [MPa]	**Yield Stress [MPa]**	**Elastic Domain**	**Ratio n**	**Hardening Law**	**Parameter A**	**Parameter q_∞**	**Behavior**
20000	100	Non-symmetric	2	Exponential	0.5	0.1	Softening

Table 6. Parameters α, β and γ for Stress Path 2.

Stress Path Parameter α [MPa]	**Stress Path Parameter β [MPa]**	**Stress Path Parameter γ [MPa]**
50	350	700

Considering the data from Tables 5 and 6, the stress path in stress space, stress-strain curve and damage evolution along time obtained for Stress Path 3 are presented in Figure 4.

According to Figure 4(A), 4(B) and 4(C), the non-symmetric tension-compression damage model together with the exponential law for softening were implemented correctly. Such observation is enforced by the reduction of the elastic domain (softening) and the reduction in stiffness of the material when damage started. In the stress increment of biaxial tensile loading, the damage surface was not reached, therefore there was no damage, as it is depicted in Figure 4(C) between points 1 and 6. The second stress increment, which accounted for unloading/biaxial compression loading, the compression stresses reached the damage surface, causing the softening of the material. For the last stress increment, unloading/biaxial tensile loading, the tensile stresses also reached the current damage surface, causing an increase in damage (Figure 4(C)) and the softening the material. It is also worth mentioning the asymptotic behavior depicted in Figure 4(B) between point 14-16. Such behavior characterizes the exponential law for softening and indicates that the internal variable q(r) is reaching its value q_∞.

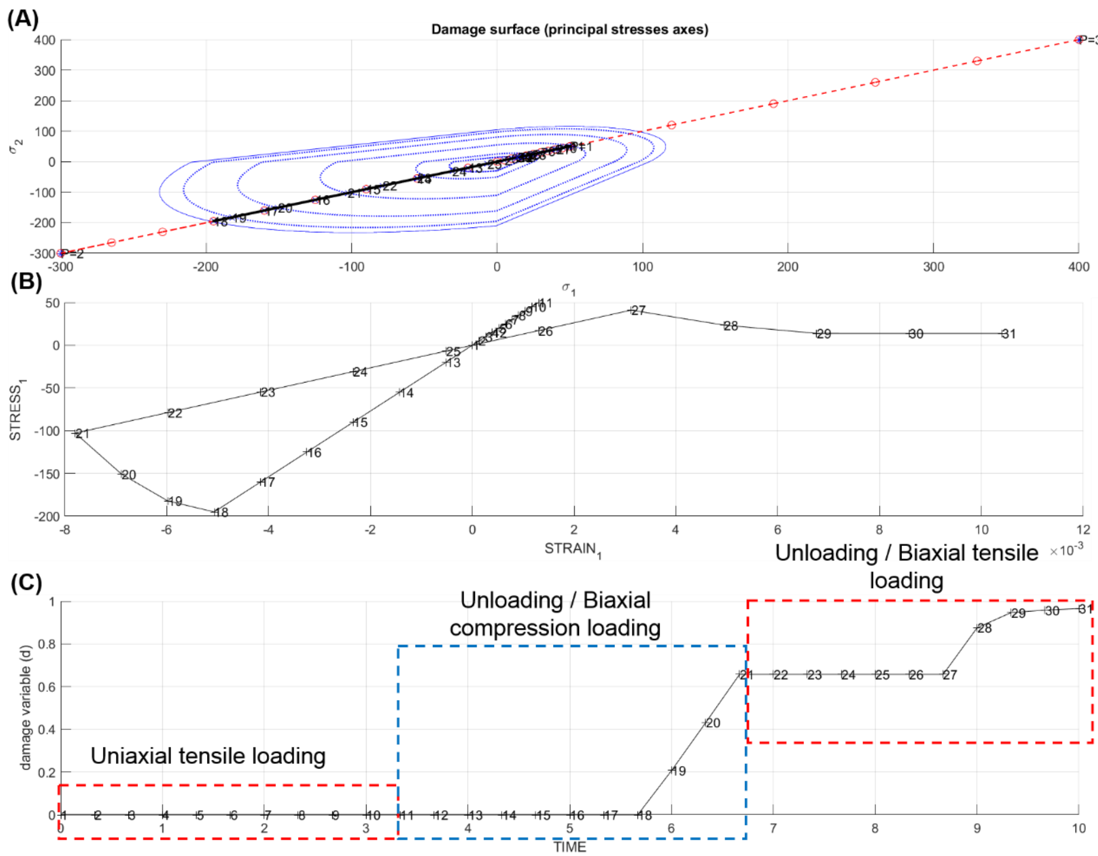

Figure 4. (A) Damage surface, (B) stress-strain curve and (C) damage evolution along time for Stress Path 3.

3. PART II (Rate Dependent Model)

For the second part of the assignment, the implementation of the rate dependent model (Appendix B) for a symmetric elastic domain with linear hardening law was required. To assure correctness of the implementation, three parameters (viscosity η, strain rate and alpha coefficient) were varied in order to evaluate the response of model. The effect on stress-strain curves for each parameter is presented. Also, an evaluation of the evolution of component C11 of both tangent and algorithmic constitutive tensors along time was performed.

3.1 – Effect of Viscosity η

For such evaluation, three different values of viscosity η were chosen: 0.3, 2 and 5. The other input parameters were maintained constant, as well as, the stress path. Table 7 presents the values chosen for the input parameters and Table 8 presents the values of the stress path.

Table 7. Input parameters for evaluation of viscosity η

Young Modulus [MPa]	**Yield Stress [MPa]**	**Hardening Modulus H**	**Behavior**	**Alpha Coefficient**	**Total Time**	**Time Increment**
20000	200	0.1	Hardening	1	10	0.66667

Table 8. Stress path chosen for evaluation of viscosity η

	Point 1 in Stress Space	**Point 2 in Stress Space**	**Point 3 in Stress Space**
σ_1	200	500	300
σ_2	200	500	300
Case	Biaxial tensile loading	Biaxial tensile loading	Biaxial tensile unloading

Considering the chosen values of viscosity η and the data from Table 7 and 8, three stress-strain curves were obtained. Figures 5 (A) and (B) depict the stress-strain curves, as well as the damage evolution along time for each value of viscosity η.

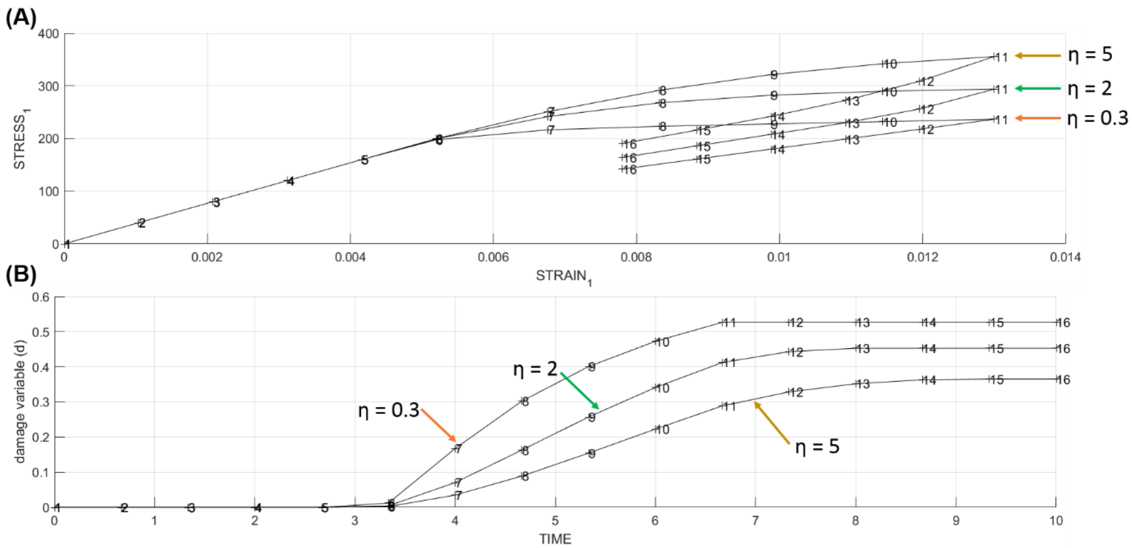

Figure 5. (A) Stress-strain curves and (B) damage evolution along time for different values of viscosity η.

Figure 5 (A) shows that with increasing values of viscosity η, the stress values also increase for the same value of inelastic strain. Such observation is in agreement with the literature [2], since the stress in the inelastic region is proportional to the value of viscosity η. Also, the damage evolution along time is faster as viscosity η decreases (Figure 5 (B)), which agrees with the smaller capability of carrying load in the inelastic regime presented in Figure 5 (A).

3.2 – Effect of Strain Rate

For such evaluation, four different values of total time (ToT) were applied to the model: 1, 5, 10 and 20. Varying the total time of simulation automatically varies the strain rate applied to the model considering the same stress path. Table 9 presents the values of the input parameters which were maintained constant and Table 10 presents the stress path considered for such study-case.

Table 9. Input parameters for evaluation of strain rate.

Young Modulus [MPa]	Yield Stress [MPa]	Hardening Modulus H	Behavior	Alpha Coefficient	Viscosity η
2000	200	-0.2	Softening	1	0.8

Table 10. Stress path chosen for evaluation of strain rate.

	Point 1 in Stress Space	Point 2 in Stress Space	Point 3 in Stress Space
σ_1	300	600	1000
σ_2	0	0	0
Case	Axial tensile loading	Axial tensile loading	Axial tensile loading

Considering the chosen values of total time (ToT) and the data from Tables 9 and 10, four stress-strain curves were obtained. Figures 6 depict the stress-strain curves obtained in such study-case.

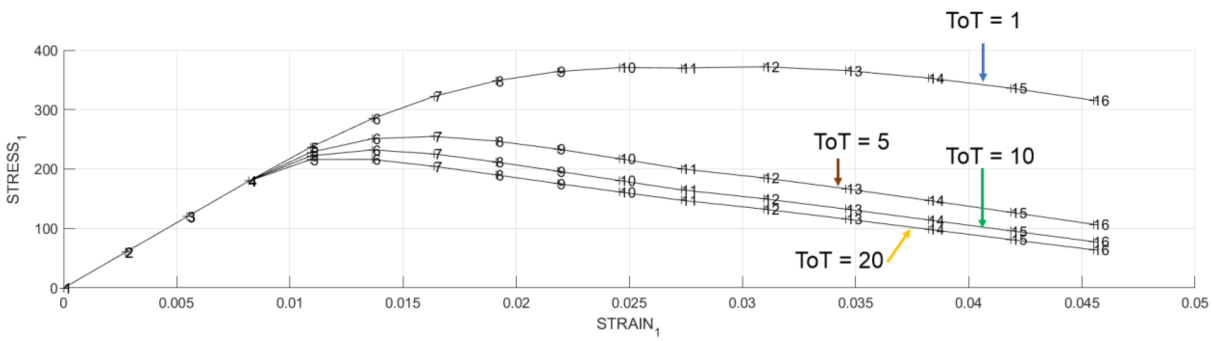

Figure 6. Stress-strain curves considering different values of Total Time (ToT) (different strain rates).

The curves presented in Figure 6 show correctness in the implementation of the viscous model since the different values of strain rate result in different stress-strain curves. Also, the stress carried out in the inelastic region decreases as the strain rate decreases (increase of total time). Such behavior is in agreement with the literature [2], since the stress in the inelastic regime is proportional to the strain rate. Therefore, the viscous model implemented presented the expected behavior.

3.3 – Effect of alpha

For such evaluation, five different values of alpha were predetermined: 0, 0.25, 0.5, 0.75 and 1. The other input parameters were maintained constant, as well as, the stress path. Table 11 presents the values chosen for the input parameters and Table 12 presents the values of the stress path.

Table 11. Input parameters for evaluation of coefficient alpha (α).

Young Modulus [MPa]	**Yield Stress [MPa]**	**Hardening Modulus H**	**Behavior**	**Viscosity η**
20000	200	-0.2	Softening	0.8

Table 12. Stress path chosen for evaluation of coefficient alpha (α).

	Point 1 in Stress Space	**Point 2 in Stress Space**	**Point 3 in Stress Space**
σ_1	-300	-600	-300
σ_2	0	0	0
Case	Uniaxial compres. load.	Uniaxial compres. load.	Uniaxial compres.unload.

Considering the determined values of coefficient alpha and the data from Table 11 and 12, five stress-strain curves were obtained. Figures 7 (A) depict the complete curves while Figure 7 (B) depicts a highlighted zone of the curves.

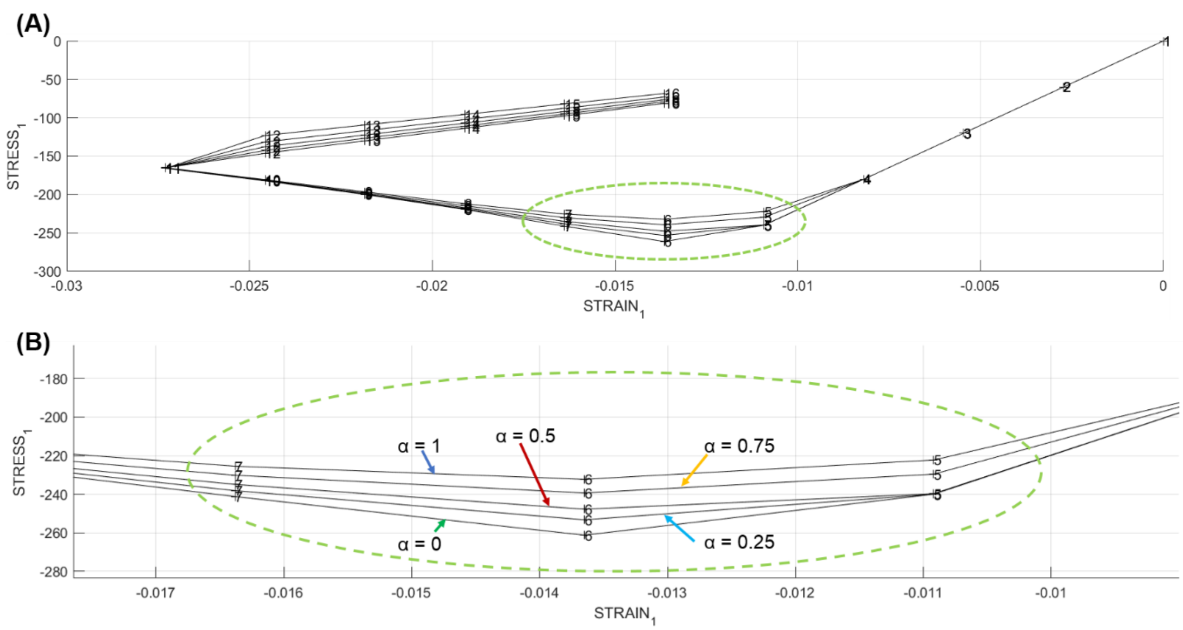

Figure 7. (A) Complete stress-strain curves and (B) highlighted portion of the curves for different values of coefficient alpha α.

According to Figure 7(B), the lower the value of coefficient alpha, higher is the stress carried out (in absolute value). It indicates that the damage evolves faster, at first, for higher values of alpha. Nevertheless, according to Figure 7 (A), the curves converge to the point where the unloading phase starts, indicating a change of damage rate for each value of alpha. Therefore, the value of alpha influences when damage starts and how fast it will develop itself throughout loading. Such observation points out the relationship between the coefficient alpha, the time integration of the constitutive equation and especially the result of the integration.

3.4 – Evaluation of C11 component

To evaluate the behavior of component C11 of the tangential and algorithmic constitutive tensors along time, five different values for the coefficient alpha were considered: 0, 1/4, 1/2, 3/4 and 1. The other input parameters were kept constant and the same stress path was considered throughout the entire study-case. Table 14 presents the values of the input parameters and Table 15 presents the stress path considered for the current study.

Table 14. Input parameters for evaluation of component C11.

Young Modulus [MPa]	**Yield Stress [MPa]**	**Hardening Modulus H**	**Behavior**	**Viscosity η**	**Total Time**	**Time Increment**
20000	200	0.2	Hardening	1	10	0.66667

Table 15. Stress path chosen for evaluation of component C11.

	Point 1 in Stress Space	**Point 2 in Stress Space**	**Point 3 in Stress Space**
σ_1	200	400	-700
σ_2	200	400	-700
Case	Biaxial tensile loading	Biaxial tensile loading	Unloading / Biaxial compression loading

The evolution of component C11 from the tangential and algorithmic constitutive tensor along time for different values of coefficient alpha is depicted in Figure 8.

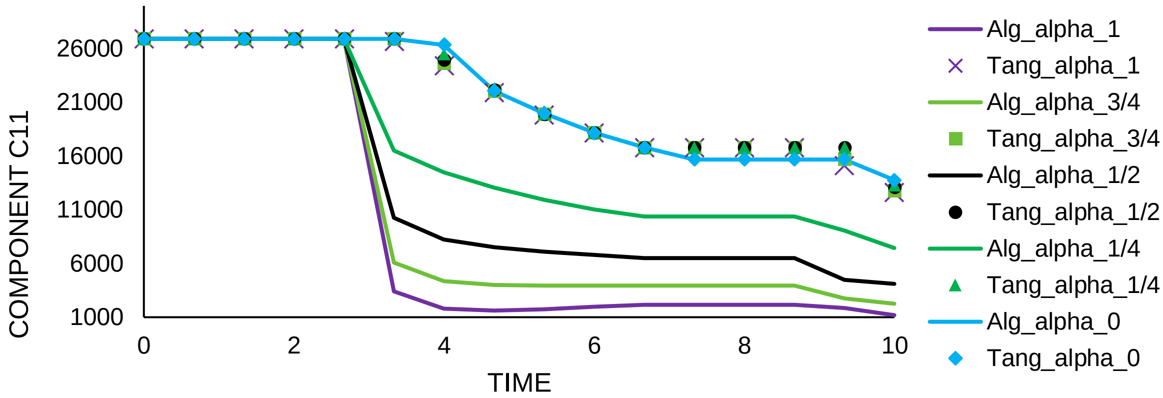

Figure 8. Evolution of component C11 of tangential and algorithmic constitutive tensors for different values of coefficient alpha.

According to Figure 8, the different values of alpha had small impact on the values of component C11 from the tangential constitutive tensor (markers in Figure 8) . Such observation is reasonable since the tangential constitutive tensor does not depend directly on the coefficient alpha. Nevertheless, different values of the coefficient alpha had noticeable impact on the values of component C11 from the algorithmic constitutive tensor (solid lines in Figure 8). Such impact is due to the algorithmic constitutive tensor direct dependence on the value of alpha. Also, there is a noticeable difference between component C11 of both tensors for the same value of the coefficient alpha. Such difference between the two constitutive tensors may be seen as the deviation caused by the numerical integration of the constitutive equation, which is strongly related to the dependence of the algorithmic constitutive tensor on the coefficient alpha. It is also worth mentioning that for the coefficient alpha equal to zero, the evolution of component C11 from both tensors coincide, which agrees with the literature [2]. However, it is important to point out that such outcome does not guarantee better results for alpha equal to zero, because such value for alpha has stability issues.

4 – Conclusions

The tension-only and non-symmetric tension-compression damage models implemented in the code presented coherent results. Coupled with the implemented exponential hardening/softening law, such models provided the expected behavior according to the determined stress paths. During damage evolution, there was enlargement of the elastic domain when exponential hardening law was applied and shrinkage of the elastic domain when exponential softening law was considered. Also, whenever a subsequent stress increment would cause unloading, the slope of the curve produced by such increment would be smaller if the damage variable was already different than zero, indicating a decrease in the material stiffness due to the damage. Such observations lead to correctness in the implemented codes. The viscous damage model implemented in the code also presented coherent results. As the viscosity and the strain rate were increased, the stress carried out throughout the stress paths also increased. When different values of the coefficient alpha were considered, different stress-strain curves were obtained, indicating different damage rates for each value of alpha. Such observation is coherent, since the coefficient alpha is related to the numerical time integration and its accuracy. The coefficient alpha also presented a noticeable influence on the evolution of component C11 from the algorithmic constitutive tensor along time. Such influence is related to the numerical time integration of the constitutive equation and it is responsible for the considerable difference between the tangent and algorithmic constitutive tensors' behavior along time in cases where alpha is different than zero.

5 – References

[1] – Assignment #1, Computational Solid Mechanics, Master of Science in Computational Mechanics, 2020.

[2] – Presentation "Lecture 5", Computational Solid Mechanics, Master of Science in Computational Mechanics, 202

Appendix A – Implemented codes for PART I

```
% TENSION-ONLY DAMAGE MODEL
elseif (MDtype==2)
      stress = ce*eps_n1';
      a = stress;
       if stress(1) < 0
            stress(1)=0;
       end
       if stress(2)<0
            stress(2) = 0;
       end
       stress_plus = stress;
       rtrial = sqrt(stress_plus'*inv(ce)*a);

% NON-SYMMETRIC TENSION-COMPRESSION DAMAGE MODEL
elseif (MDtype==3)  %*Non-symmetric
    stress = ce*eps_n1';
        a = sum(stress);
        b = abs(stress(1)) + abs(stress(2)) +  abs(stress(3))+...
            +abs(stress(4));
        h = a/b;
        if h==1
           rtrial = sqrt(stress'*inv(ce)*stress);
        end
        if  stress(1) & stress(2) <0
            rtrial = (sqrt(stress'*inv(ce)*stress))/n;
        end
        if stress(1)>0 & stress(2)<0
            stress(2)=0;
            rtrial = sqrt(stress'*inv(ce)*stress);
        end
        if stress(2)>0 & stress(1)<0
            stress(1)=0;
           rtrial = sqrt(stress'*inv(ce)*stress);
        end

%EXPONENTIAL HARDENING-SOFTENING LAW
            A=.5;
            q_inf = 0.1
            q_n1 = q_inf -(q_inf - q_n)*exp(A*(1 - (rtrial/q_n)))
        end
```

Appendix B – Implemented codes for PART II

```
% MODIFICATION IN CODE DAMAGE_MAIN TO ALLOW VISCOUS CASE
if viscpr == 1
    totalstep = sum(istep) ;
% INITIALIZING GLOBAL CELL ARRAYS
% -------------------------------
sigma_v = cell(totalstep+1,1) ;
TIMEVECTOR = zeros(totalstep+1,1) ;
delta_t = TimeTotal./istep/length(istep);
% Elastic constitutive tensor
% ----------------------------
[ce]    = tensor_elastico1 (Eprop, ntype);
% Initz.
% -----
% Strain vector
% -------------
eps_n1  = zeros(mstrain,1);
% Historic variables
% hvar_n(1:4) --> empty
% hvar_n(5)  = q --> Hardening variable
% hvar_n(6)  = r --> Internal variable
hvar_n   = zeros(mhist,1)   ;
% INITIALIZING  (i = 1) !!!!
% ***********i*
i = 1 ;
r0 = sigma_u/sqrt(E);
hvar_n(5) = r0; % r_n
hvar_n(6) = r0; % q_n
eps_n1 = strain(i,:) ;
sigma_n1 =ce*eps_n1'; % Elastic
sigma_v{i} = [sigma_n1(1)  sigma_n1(3) 0;sigma_n1(3) sigma_n1(2) 0 ; 0 0
sigma_n1(4)];
nplot = 3 ;
vartoplot = cell(1,totalstep+1) ;
vartoplot{i}(1) = hvar_n(6) ; % Hardening variable (q)
vartoplot{i}(2) = hvar_n(5) ; % Internal variable (r)
vartoplot{i}(3) = 1-hvar_n(6)/hvar_n(5)  ; %  Damage variable (d)
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 = strain(i-1,:);   %VECTOR TO STORE STRAIN AT PAST TIME
        eps_n1 = strain(i,:) ;

%****************************************************************************
*********
        %*      DAMAGE MODEL
        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        [sigma_n1,hvar_n,aux_var] =
rmap_dano1(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps,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
        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 ;
        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

else
% IMPLEMENTATION OF VISCOUS CASE IN CODE RMAP_DANO
%-----------------------viscous modelling-----------------------------
    alpha =0;
    visc = .5;
    tau_eps_0 = sqrt(eps*ce*eps') ;
    tau_eps_1 = sqrt(eps_n1*ce*eps_n1');
    p = (visc - delta_t*(1-alpha))/(visc + alpha*delta_t);
    tau_eps_alpha = (1-alpha)*tau_eps_0 + alpha*rtrial;
    if  tau_eps_alpha> r_n     % loading
        fload=1;
         r_n1 = [(visc - delta_t*(1-alpha))/(visc + alpha*delta_t)]*r_n +...
        + [(delta_t)/(visc + alpha*delta_t)]*tau_eps_alpha;
        if hard_type == 0
         %Linear
            q_n1= q_n+ H*(r_n1-r_n);
        else
            A=.5;
            q_inf =  1;
              %hardening
            q_n1 = q_inf -(q_inf - q_n)*exp(A*(1 - (r_n1/q_n)));
        end
        if(q_n1<zero_q)
        q_n1=zero_q;
        end
    else     % elastic loading
        fload=0;
        r_n1= r_n  ;
        q_n1= q_n ;
   end

% IMPLEMENTATION TO EVALUATE COMPONENT C11

ct = (1.d0-dano_n1)*ce; % TANGENT CONSTITUTIVE TENSOR
ct_1 = ct(1,1);

c_al = 0; % ALGORITHMIC CONSTITUTIVE TENSOR
sig_eff = ce*eps_n1';
sig = sig_eff'*sig_eff;
f = ((alpha*delta_t)/(visc+alpha*delta_t));
g = (1/tau_eps_1)*((q_n1 - 0.1*r_n1)/(r_n1^2));
```

```
if  rtrial> r_n  % loading
    c_al  = (1.d0-dano_n1)*ce -  f*g*sig;
else
    c_al = (1.d0-dano_n1)*ce;
end
 c_al(1,1);
```