Màster en Mètodes Numèrics

Computational Solid Mechanics

Assignment 1. Continuum Damage Models

Aitor Bazán Escoda
12-4-2019
Universitat Politècnica de Catalunya

CONTENTS OUTLINE

1. PART I. RATE INDEPENDENT MODELS

1.2. Continuum damage models. Introduction.

Figures shown below depict the continuum damage models used for the current report-cases by means of stress-strain curves. Particularly, *Fig 1.* lights up the symmetric damage model in which both the elastic tension area and the elastic compression region are equivalent in terms of absolute values. On the other hand and with regard to the *Fig.2,* the only-tension damage model remarks an infinite elastic compression region and a comparatively smaller elastic traction region.
On the last point, *Fig. 3,* in which a non-symmetric damage model is shown, it can be observed how the elastic compression area is larger than the elastic traction region. This fact is driven by a parameter "*n"* which specifically relates the ratio of compression elastic limit to the tension elastic limit.

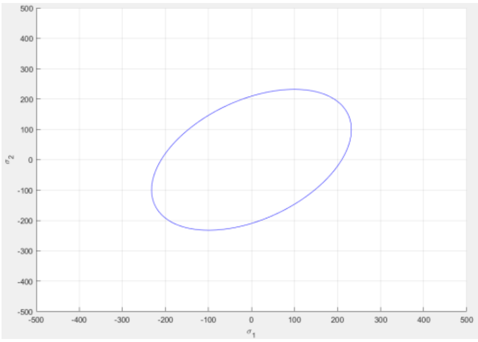

Figure 1. Symmetric damage model.

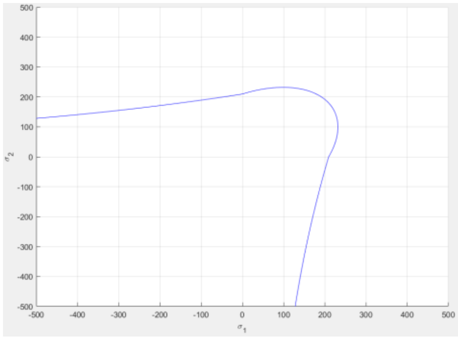

Figure 2. Only-tension damage model.

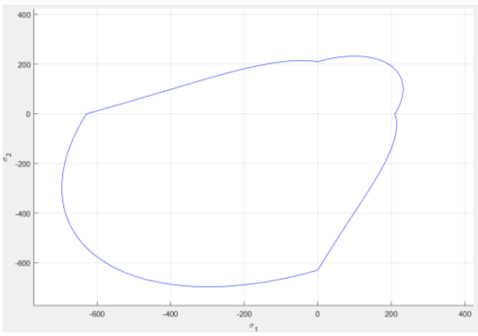

Figure 3. Non-symmetric damage model.

1.2.1. Linear and exponential Hardening/Softening.

By means of *Fig. 4* and *Fig. 5*, it is meant to be shown how the linear and the exponential hardening law behave in a simple case-scenario on purpose of the upcoming results. As it is described, hardening modulus of H = 2 is taken to describe a hardening behaviour and H = -2 to depict a softening behaviour.

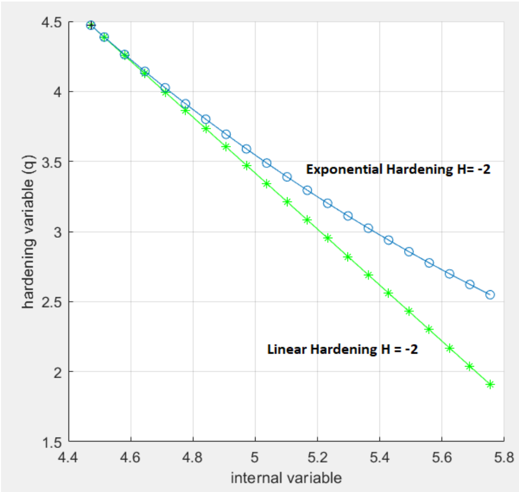

Figure 4. Linear and exponential Hardening Law H = -2.

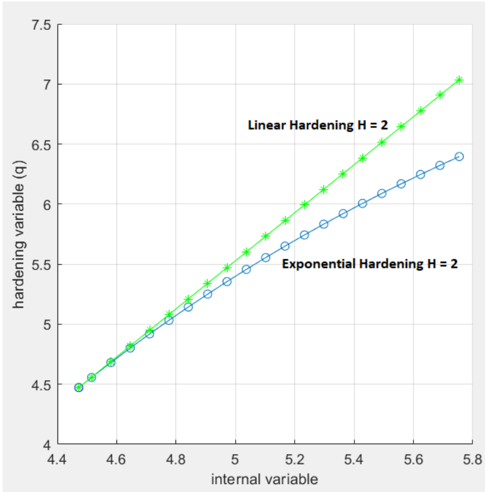

Figure 5. Linear and exponential Hardening Law H = 2.

1.3. Methodology and results. Loading paths.

On purpose of assessing the correctness of the implementation of the method and considering different material parameters and also several different load paths defining the symmetric, only-tension and non-symmetric damage models in the strain space, they will be tested under 3 cases-scenarios.

- Purely uniaxial loading/unloading.
- Uniaxial and biaxial loading/unloading.
- Purely biaxial loading/unloading.

Also note that these loading paths are consciously chosen so as to remark the properties of each one of the implemented methods.

1.3.1. Purely uniaxial loading/unloading.

1.3.1.1. Symmetric model.

It is considered then, in the next table, the following case-studio for all the aforementioned damage models.

Uniaxial tensile loading		**Uniaxial tensile unloading/compressive loading**		**Uniaxial compressive unloading/tensile loading**		**Material parameters**			
$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	**E**	σ_{yield}	**v**	**H**
300	0	-700	0	400	0	2000	200	0.3	-0.1

Table 1. Loading/Unloading parameters to assess correctness on implementation of purely uniaxial loading/unloading.

If symmetric model is assessed first, it is observed in *Fig. 6* how the model behaves when it is subjected to uniaxial stress loading and unloading and under the assumption of damage with exponential softening. It is also seen the model showing when the material behave the same both when in compression and in tension.

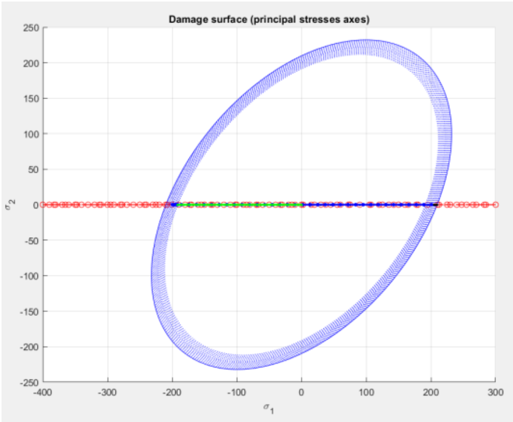

Figure 6. Symmetric model damage surface.

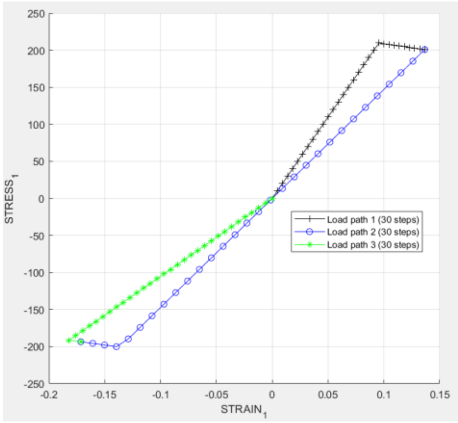

Figure 7. Stress-strain load paths. Symmetric model.

Figure 7 comments on how the processes of loading and unloading are carried out. If enlighten first the load path 1 (black line), it is understood how the applied elastic loading generates a process of damage loading when it reaches and overcomes the yield stress. Following, within the load path 2 (blue line) there is an elastic unloading preceding a compressive loading. And finally, the material shows a process of unloading (green line) until it reaches the first initial state. Showing purely elastic behaviour.

If referring to *Figure 9,* but also *Fig. 7,* it might be appreciated how the process on load paths 2 and 3 are reduced due to a possible material degradation.

Generally, damage evolution in time gives information about loading/unloading stages of the material, whereas, particularly, its horizontal lines on *Fig.9* depict a constant behaviour of the damage in the material.

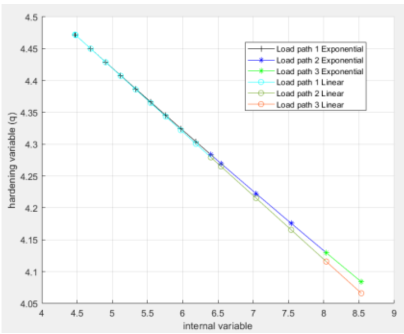

Figure 8. Hardening Law, Exponential vs Linear.

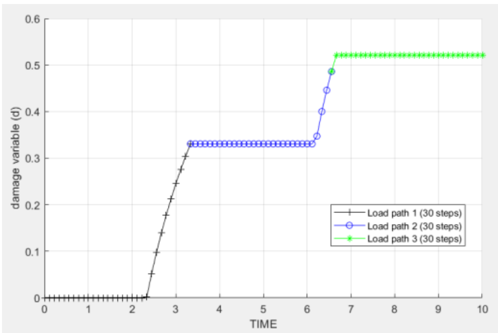

Figure 9. Evolution of damage variable in time. Symmetric m.

Figure 8 may be of help to understand that little change in the behaviour of the material is occurring due to the applied assumption of the exponential softening.

1.3.1.2. Only-tension model

Next it is shown how an only-tension damage model behaves when it is applied a loading path such as the one described above.

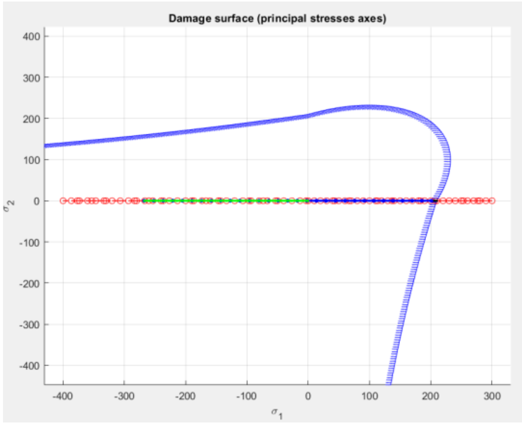

Figure 10. Only-tension model damage surface.

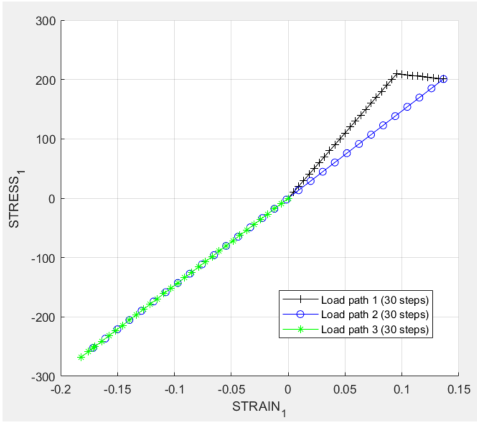

Figure 11. Stress-strain load paths. Only-tension model.

In *Figure 11,* it is described the different applied elastic loading paths. For the first one there is a uniaxial tensile loading until it reaches the yield stress. And, whilst along the second path it is found a tensile unloading/compressive loading, the last path relates to a compressive unloading.

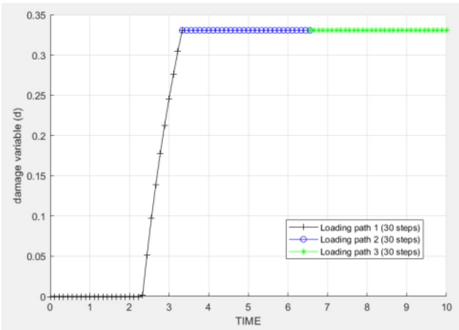

Figure 12. Evolution of damage variable in time. Only-t. model.

Throughout the *Fig 12,* one can observe not just the applied exponential hardening commented before but also the material degradation during the first stage of the loading cycle. As it is shown, this model only accounts for tensile forces, this is, the model does not take into account failure by compression and that is mainly the reason why in the second and third loading paths there is no degradation of the material obtained.

1.3.1.3. Non-symmetric model

This damage model, as illustrated in *Fig. 13,* differs from only-tension in the sense that the latter one have an infinite elastic region.

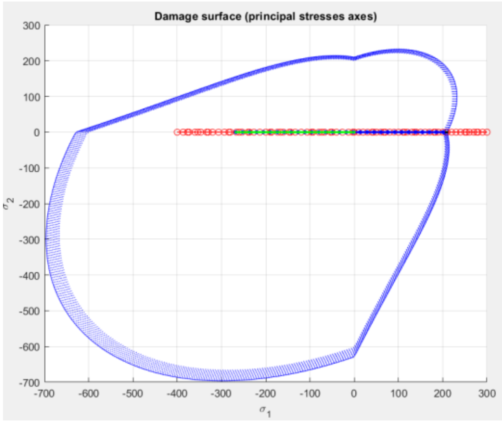

Figure 13. Non-symmetric model damage surface.

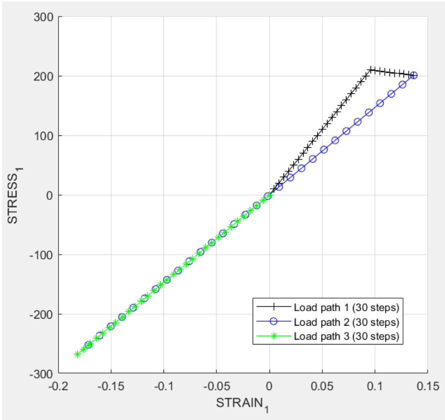

Figure 14. Stress-strain load paths. Non-symm. model

The behaviour of the material for the chosen path behaves similarly to the only-tension damage model. It is, certainly, a damage model in which one knows that the testing material will work fine under compression forces but might go into failure when applying tensile forces.

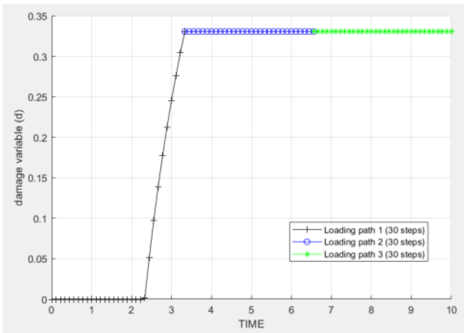

Figure 15. Evolution of damage variable in time. N-S model.

1.3.2. Uniaxial and biaxial loading/unloading.

To carry out this loading case, the considered parameters are listed below. Here it is also considered exponential hardening.

Uniaxial tensile loading		Biaxial tensile unloading/compressive loading		Biaxial compressive unloading/tensile loading		Material parameters			
$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	E	σ_{yield}	v	H
300	0	-300	-300	100	100	2000	200	0.3	-0.1

Table 2. Loading/Unloading parameters to assess correctness on implementation of uniaxial/biaxial loading/unloading.

1.3.2.1. Symmetric model.

For the load path of uniaxial and biaxial loading/unloading, it is considered a first uniaxial tensile loading so as it crosses the damage surface. (*see Fig. 16)*

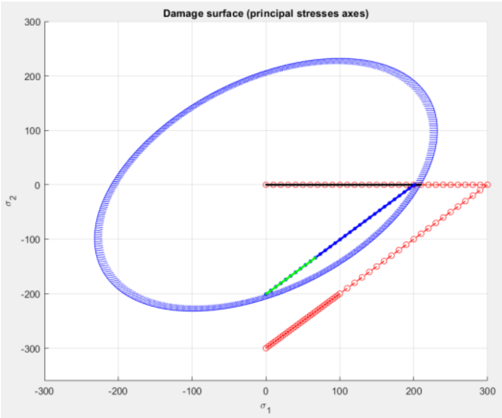

Figure 16. Symmetric model damage surface. Symmetric model.

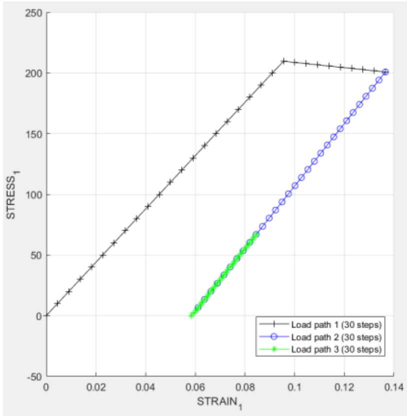

Figure 17. Stress-strain load paths. Symmetric model.

Then, when it is applied a biaxial tensile unloading/compressive loading (load path 2) it is remarked that this current state remains inside the damage surface. This behaviour may also be observed in terms of the stress-strain curve where it is first produced an elastic loading preceding a process of damage loading. This is then followed by an elastic tensile unloading/compressive loading and at last, the compressive unloading/tensile reloading.

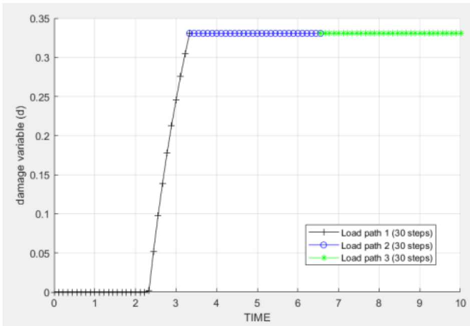

Figure 18. Evolution of damage variable in time. Symm. model.

1.3.2.2. Only-tension model

From the only-tension model it might also be observed that the model shows damage when the first load path applied tries to go out of the damage surface. This is when overcomes the yield stress.

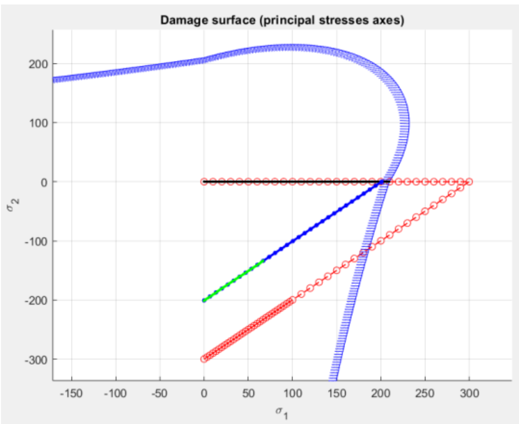

Figure 19. Only-tension model damage surface. Only-tension model.

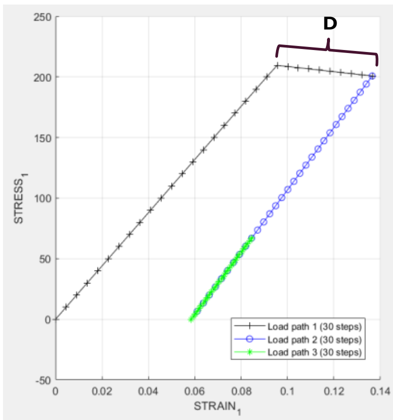

Figure 20. Stress-strain load paths. Only-tension model.

As it is clearly seen, *D* regions in *Fig. 20* and *Fig. 21,* are coincident since they represent the process of damage.

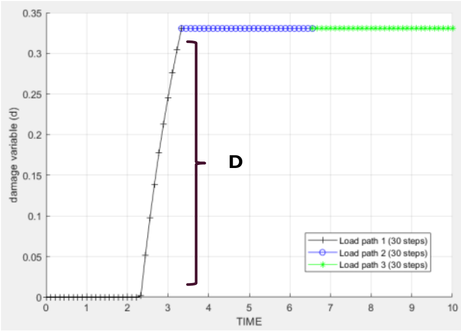

Figure 21. Evolution of damage variable in time. O-t model.

1.3.2.3. Non-symmetric model

Differences between these three models are not found for the case of uniaxial and biaxial loading/unloading. But as a remark, it can be affirmed that non-symmetric and only-tension damage models behave in a more conservative manner than the symmetric model when a biaxial compressive is applied. *Figure 23,* shows a partially elastic behaviour with 2 differentiated regions. Residual strain (corresponding to first bracketed area) and elastic recovery (related to the second one). This behaviour is also seen in figures presented above.

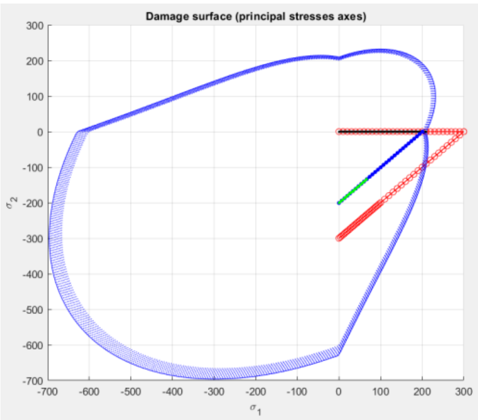

Figure 22. Non-symmetric model damage surface.

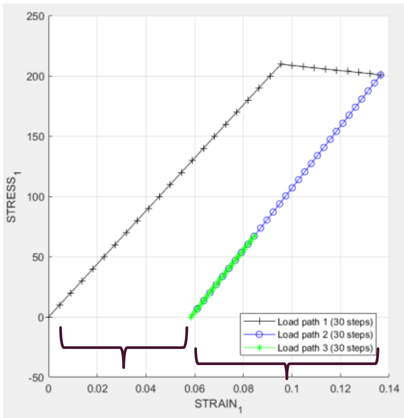

Figure 23. Stress-strain load paths. N-S model.

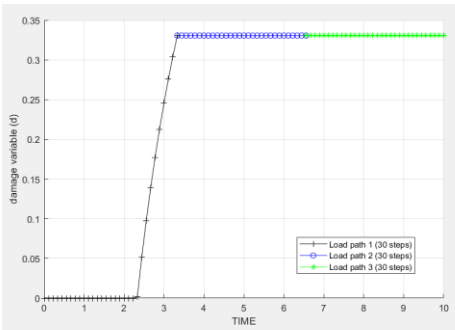

Figure 24. Evolution of damage variable in time. N-S model.

1.3.3. Purely biaxial loading/unloading.

And the case of biaxial loading/unloading the following parameters are used.

Biaxial tensile loading		Biaxial tensile unloading/compressive loading		Biaxial compressive unloading/tensile loading		Material parameters			
$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	**E**	σ_{yield}	**v**	**H**
300	300	-700	-700	400	400	2000	200	0.3	-0.1

Table 3. Loading/Unloading parameters to assess correctness on implementation of purely biaxial loading/unloading.

1.3.3.1. Symmetric model

In the context of the symmetric model, it is observable how the model behaves when facing a purely biaxial loading/unloading. From *Fig. 25,* one sees that both tensile and compressive loadings cross the damage surface yielding to a process of material degradation.

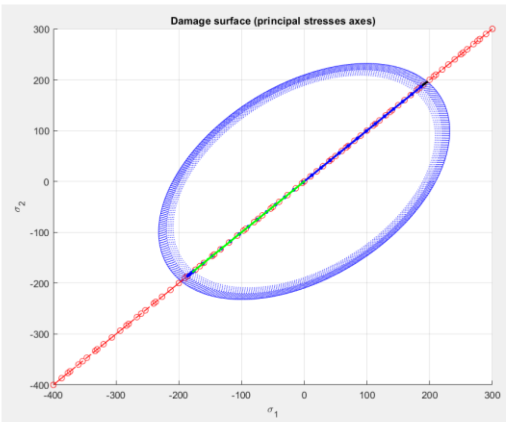

Figure 25. Symmetric model damage surface.

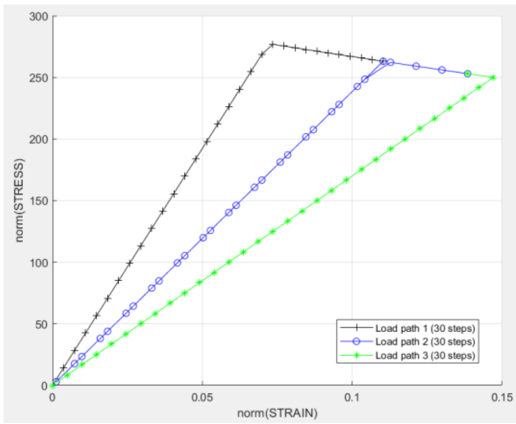

Figure 26. Stress-strain load paths. Symmetric model.

So as to have a better insight of what is happening in *Fig. 26,* it is explained the loading/unloading/loading cycle path-by-path. Within the load path 1, the elastic loading overcomes the stress limit of the material leading to a subsequent process of damage in such material. This is then followed by the load path 2 which, at its turn, applies an elastic tensile unloading/compressive loading that, since it also exceeds yield stress, and gives, consequently, a second process of damage. Finally, an elastic compressive unloading is applied.

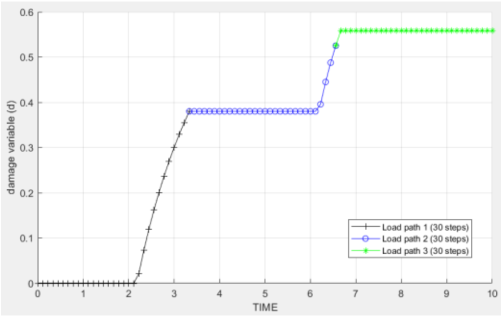

Figure 27. Evolution of damage variable in time. Symm. model.

Through *Fig.27* it is depicted the two levels of damage commented before. First one corresponds to a tensile loading while the second one it properly does by means of a compressive loading.

1.3.3.2. Only-tension model

Only-tension model only illustrates the behaviour of tensile effects on the material. Therefore, as it is observed from *Fig. 29* and *Fig. 30,* after overcoming the yield stress and generating a process of damage (load path 1), material submitted under compressive loading (load path 2) does not represent a damage surface crossing when overcoming such stress limit. This is mainly due to be, the only-tension model, representing an infinite compressive elastic region.

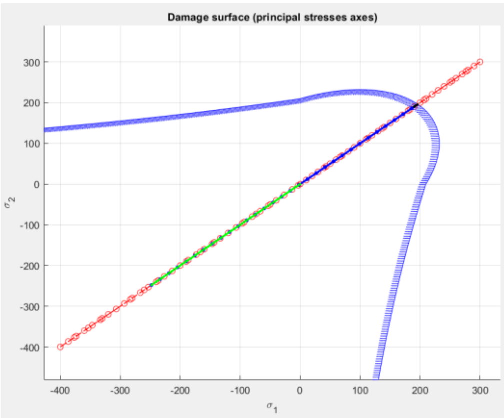

Figure 28. Only-tension model damage surface.

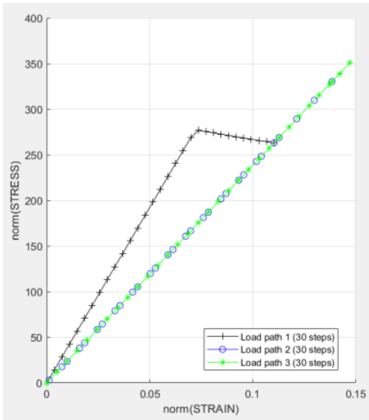

Figure 29. Stress-strain load paths. Only-tension model.

Because of the commented above, *Fig. 30,* only displays a level of damage.

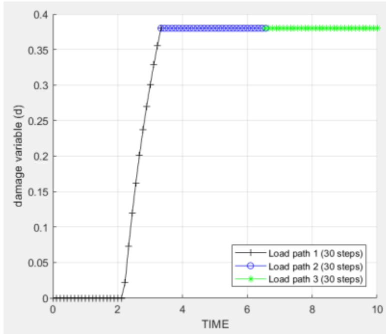

Figure 30. Evolution of damage variable in time. O-t m.

1.3.3.3. Non-symmetric model

Similar observations to those from only-tension damage model can be applied to non-symmetric model. In this case, compressive reloading does not cross the damage surface leading, this way, into a no damage process.

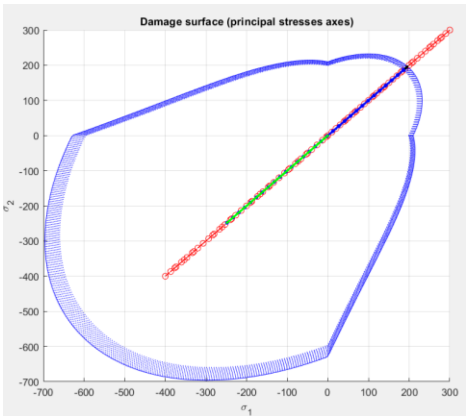

Figure 31. Non-symmetric model damage surface.

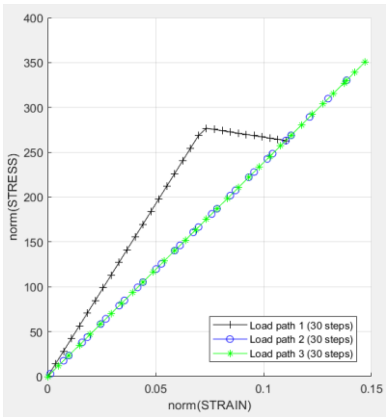

Figure 32. Stress-strain load paths. N-S model.

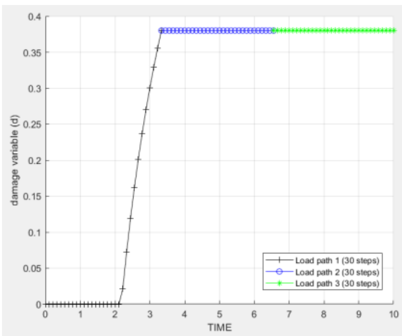

Figure 33. Evolution of damage variable in time. N-S m.

2. PART 2. RATE DEPENDENT MODELS

For reasons of clearance, the model assessed within this part is a symmetric tension-compression model but under uniaxial stress state. Linear hardening/softening parameter will be used, as well. In the table shown below there is detailed the loading path for the current case-scenario. Moreover, it is thought to be important that the loading path crosses the damage surface to better understand the variation of the material damage model when some of the tested parameters change.

Loading path (1)		**Loading path (2)**		**Loading path (3)**	
$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$	$\Delta\bar{\sigma}_1$	$\Delta\bar{\sigma}_2$
100	0	100	0	300	0

Table 4. Loading/Unloading parameters to assess correctness on implementation of symmetric tension-compression model.

2.2. Variability in the viscosity parameter.

Also, so as to clearly show how the viscosity parameter behaves, perfect damage with $H^d(r) = 0$ will be first chosen as a first sight and then linear hardening $H^d(r) > 0$.

Cases	**Material Parameters**				**Integration Parameters**		**Viscosity Parameter η**			
	E	σ_{yield}	**v**	**H**	T. int.	α	**1**	**2**	**3**	**4**
Case 1	2000	200	0.3	0	10	1	0	0.5	1	10
Case 2				0.1						

Table 5. Material, integration and viscosity parameters for perfect and linear hardening cases.

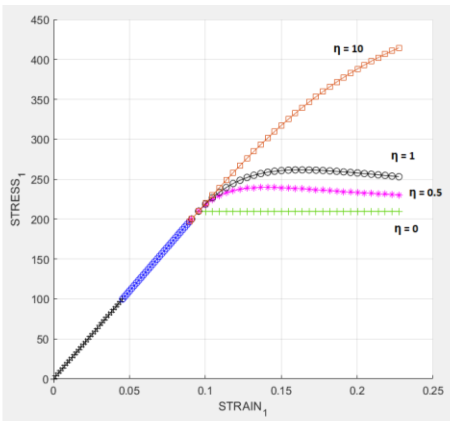

Figure 34. Case 1. Stress-strain curve for different η values.

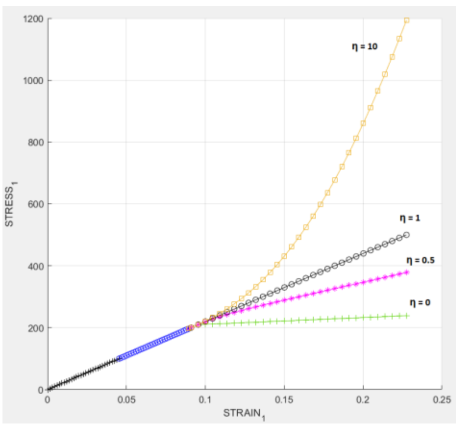

Figure 35. Case 2. Stress-strain curve for different η values.

From Case 1, it is better observed that as long as the viscous parameter increases, the initial damage threshold for yield stress (σ_{yield}) and initial strain (ε_0) also increase. On the other hand, if Case 2 is pointed out, one observes that for higher values of viscosity, the straight line that represents a material with a η = 0 becomes an exponential curve for these larger η values. This way, the material behaves as if it was either a “rubber” or an elastic tissue.

2.3. Variability in the strain rate.

In order to check how the variability in the strain rate is affecting the implementation of the method, different values for the time integration parameters will be recalled as long as the strain rate is time dependent, $\frac{d\varepsilon}{dt}$.

Cases	Material Parameters				Viscosity Parameter	Integration Parameters				
	E	σ_{yield}	v	H	η	α	1	2	3	4
Case 1	2000	200	0.3	0	1	1	0.1	5	10	100
Case 2				0.1						

Table 6. Material, integration and viscosity parameters for perfect and linear hardening cases.

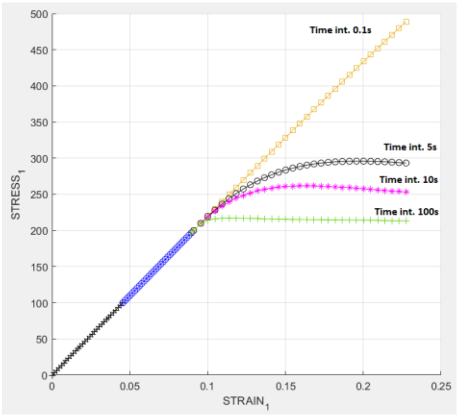

Figure 36. Case 1. Stress-strain curve for different* time integration *values.

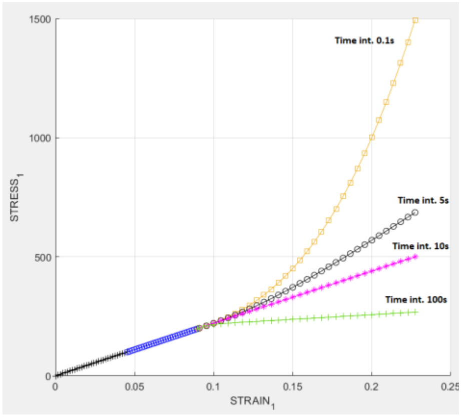

Figure 37. Case 2. Stress-strain curve for different* time integration *values.

As it is checked out from *Fig. 36* and *Fig. 37,* and in terms of stress-strain, the variability in the strain rate behaves similar as if it was by means of varying the viscous parameter. Another remarkable point is that if the strain rate is very low, the system can be considered quasi-static. Meaning that, the load is applied so slowly that the material deforms also very slowly and the inertia force exerted might be neglected and it is, the material, in equilibrium the whole time. On the contrary, if the strain rate is highly increased, that material could not dissipate the energy applied and consequently it would appear a process of damage.

2.4. Variability of alpha

To emphasize the effect of the time integration scheme on the stress-strain curve, different alpha values ranging from $0 \leq \alpha \leq 1$ are selected. Moreover, to be noted that, as long as the viscosity parameter is increased or decreased, it starts playing a role in terms of the stability of the method.

Material Parameters				Viscosity Parameter	Integration Parameters					
E	σ_{yield}	v	H	η	Time int.	α_1	α_2	α_3	α_4	α_5
2000	200	0.3	0.1	0.1	100	0	0.25	0.5	0.75	1

Table 7. Material, integration and viscosity parameters for perfect and linear hardening cases.

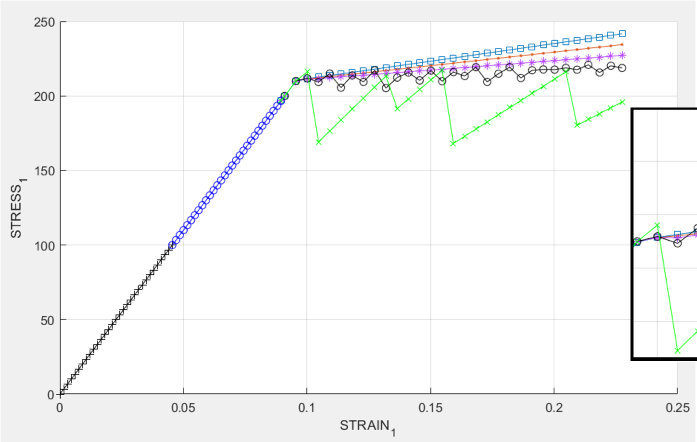

Figure 38. Stress-strain curve for different α values.

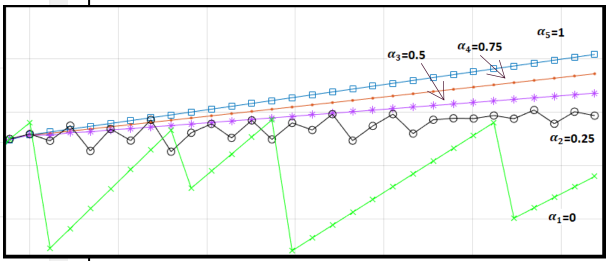

Figure 39. Zoom of Stress-strain curve for different α.

As it is seen in *Fig. 38,* the stability of the time integration method ranges from

$\frac{1}{2} \leq \alpha \leq 1$. It is perceptible that, in this range, the method is also accurate. On the other side, for values of $\alpha = 0, 0.25$ the method becomes unstable.

For $\alpha = \frac{1}{2}$ method is second order accurate.

2.5. Effects of alpha on the evolution of C_{tg11} and C_{Alg11}

In this section it is studied the behaviour of the algorithm and tangential constitutive matrices with the influence of the variability of alpha. C_{11} component of both matrices in particular it is presented here.

2.5.1. Evolution of C_{tg11} in time.

Throughout *Figure 40* and *Figure 42,* it is observed the lack of stability for alpha values lower than 0.5. For these values accuracy is preserved. The method is conditionally stable for the range of $\frac{1}{2} \leq \alpha \leq 1$. Although, the method is consistent and stable, it is therefore, by the *Lax Theorem, also* convergent for values of $0 \leq \alpha \leq 1$.
In the figure shown below, it is understood where the material is undamaged, this is the continuous line, and where the process of damage starts, this is where discontinuities start to show up.

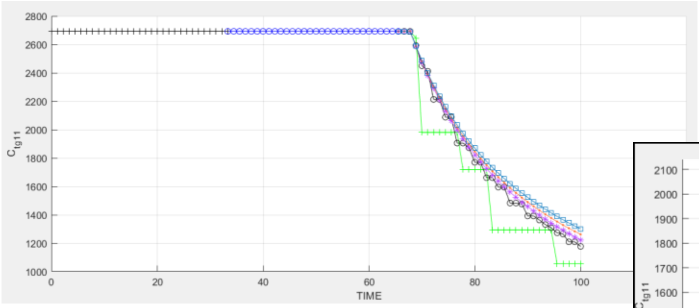

Figure 40. Evolution of C_{tg11} in time for different α values.

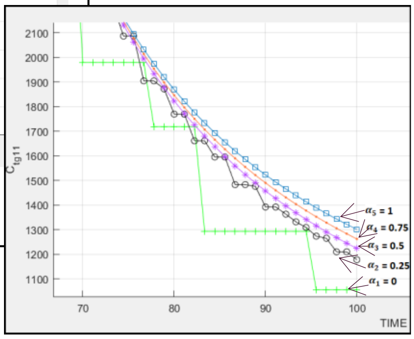

Figure 41. Zoom of Figure 40.

2.5.2. Evolution of C_{Alg11} in time.

The algorithm constitutive matrix shows a discontinuity in its behaviour. This is indeed due to the fact that the expression of the algorithm constitutive matrix is a piecewise function in which $C_{Alg} = C_{tg}$ where the elastic region is in and $C_{Alg} \neq C_{tg}$ in the damage region.

Moreover, stability for different alpha values behave similarly as it does for the constitutive tangent matrix.

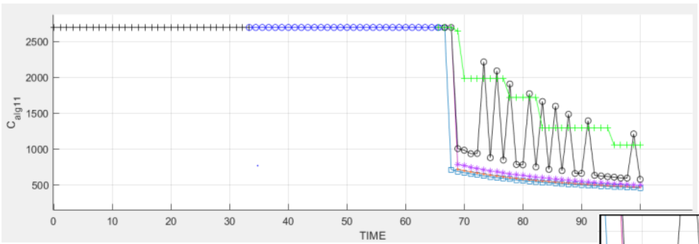

Figure 42. Evolution of C_{alg11} in time for different α values.

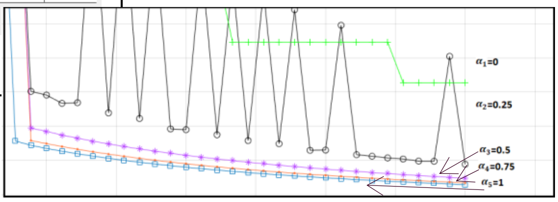

Figure 43. Zoom of Figure 42.

3. APPENDIX

Here are listed the modified routines. Parts of the routines which are not changed were removed so as to optimize space in the report. These erased parts are identified as **[…]** within the function**.**

3.2. dibujar_criterio_dano1

```
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);
%*******************************************************************************
***
[...]
elseif MDtype==2

%*******************************************************************************
***
    %* RADIUS
    tetha=[-pi/2+0.01:0.01:pi-0.01];
    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
        %Implementation of Macaulay brackets in "mx(i)*(mx(i)>0)" -> If mx(i)>0 get
1, otherwise get 0
        cos_part = m1(i)*(m1(i)>0);
        sin_part = m2(i)*(m2(i)>0);
        radio(i)= q/sqrt([cos_part sin_part 0 nu*(cos_part+sin_part)]*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);

elseif MDtype==3

%*******************************************************************************
***
    %* RADIUS
    tetha=[0:0.01:2*pi];
    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) ;
    alpha_Num=0;
    alpha_Den=0;
    for i=1:Contador
        %Implementation of Macaulay brackets in "mx(i)*(mx(i)>0)" -> If mx(i)>0 get
1, otherwise get 0
        cos_part = m1(i)*(m1(i)>0);
        sin_part = m2(i)*(m2(i)>0);
        alpha_Num =cos_part+sin_part;
        alpha_Den =abs(m1(i))+abs(m2(i));
        alpha = alpha_Num/alpha_Den;
        radio(i)= q/((alpha+(1-alpha)/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
```

3.3. rmap_dano1

```
function [sigma_n1,hvar_n1,aux_var] = rmap_dano1
(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_n,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) ;
viscpr = Eprop(6);
eta = Eprop(7);
alpha = Eprop(8);
[...]

%*************************************************************************************
**
%*       Damage surface
%*
[rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);
[rtrial_prev] = Modelos_de_dano1 (MDtype,ce,eps_n,n); %It is computed rtial at
previous time step.
rtrial_n_alpha = rtrial_prev*(1-alpha)+rtrial*alpha;
%*************************************************************************************
**


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

%* Check if model is viscous or inviscid
if viscpr == 0
    eta = 0;
    alpha = 1;
    if (rtrial > r_n)
        %*   Loading
        fload=1;
        delta_r = rtrial-r_n;
        r_n1 = rtrial;
        if hard_type == 0
            %  Linear Hardening Law
            H_n1 = H;
            q_n1= q_n+ H*delta_r;
        else
            % Exponential Hardening Law
            q_inf = r0 + (r0-zero_q); %First it is computed q infinity
            if H > 0
                H_n1 = H*((q_inf-r0)/r0)*exp(H*(1-rtrial_n_alpha/r0));
```

```
%calculation...
                ...of tangent hardening modulus
            else
            H_n1 = H*((q_inf-r0)/r0)*1/(exp(H*(1-rtrial_n_alpha/r0)));
%calculation...
            ...of tangent softening modulus
            end
        q_n1=q_n+H_n1*(delta_r);
        end

        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

else %viscpr == 1 --> viscous model
    if (rtrial_n_alpha > r_n)
        %*   Loading
        fload = 1;
        delta_r=rtrial_n_alpha-r_n;
        r_n1 = (eta - delta_t*(1-alpha))/(eta + alpha*delta_t)*r_n + (delta_t/(eta +
...
            alpha*delta_t))*rtrial_n_alpha;
        if hard_type == 0
            % Linear Hardening Law
            H_n1 = H;
            q_n1= q_n+ H_n1*delta_r;
        else
            % Exponential Hardening Law
            q_inf = r0 + (r0-zero_q); %First q inf is computed
            if H > 0
                H_n1 = H*((q_inf-r0)/r0)*exp(H*(1-rtrial_n_alpha/r0));
%calculation...
                ...of the tangent hardening modulus
            else
            H_n1 = H*((q_inf-r0)/r0)*1/(exp(H*(1-rtrial_n_alpha/r0)));
%calculation...
            ...of the tangent softening modulus
            end
        q_n1 = q_n + H_n1*delta_r;
        end
        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
end

% Damage variable
```

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

%*************************************************************************************
**
% calculation of the Ce_tang_n1
if viscpr == 1
    if rtrial_n_alpha > r_n
        %Algorithm Constitutive Tangent Matrix
        Ce_alg_n1 = (1.d0-dano_n1)*ce+((alpha*delta_t)/(eta+alpha*delta_t))*...
            (1/rtrial_n_alpha)*((H_n1*r_n1-
q_n1)/(r_n1^2))*((ce*eps_n1')'*(ce*eps_n1'));
        C_alg = Ce_alg_n1(1,1);
        %Constitutive Tangent Matrix Operator
        Ce_tan_n1=(1.d0-dano_n1)*ce;
        C_tan = Ce_tan_n1(1,1);
    else
        %Algorithm Constitutive Tangent Matrix
        Ce_alg_n1 = (1.d0-dano_n1)*ce;
        C_alg = Ce_alg_n1(1,1);
        %Constitutive Tangent Matrix Operator
        Ce_tan_n1 = Ce_alg_n1;
        C_tan = C_alg;
    end
end


%*************************************************************************************
**
%* Updating historic variables                                          %*
% hvar_n1(1:4)  = eps_n1p;
hvar_n1(5)= r_n1 ;
hvar_n1(6)= q_n1 ;
hvar_n1(7)= dano_n1;
%If viscous update variables
if viscpr == 1
    hvar_n1(8)= C_alg;
    hvar_n1(9)= C_tan;
end
%*************************************************************************************
**

%*************************************************************************************
**
%* Auxiliar variables
%*
aux_var(1) = fload;
aux_var(2) = q_n1/r_n1;
%*aux_var(3) = (q_n1-H*r_n1)/r_n1^3;
%*************************************************************************************
**
```

3.4. Modelos_de_dano1

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

%*********************************************************************************
***
if (MDtype==1)      %* Symmetric damage model

rtrial= sqrt(eps_n1*ce*eps_n1')                          ;

elseif (MDtype==2)  %* Only tension damage model

sigma_e = ce*eps_n1';

for i=1:length(sigma_e)
    sigma_e_pos(i) = sigma_e(i)*(sigma_e(i)>0);
end
rtrial= sqrt(sigma_e_pos*eps_n1');

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

theta_Num=0; %Preallocation
theta_Den=0;
sigma_e = ce*eps_n1'; %Taking stresses from strains.

for i=1:length(sigma_e)
    sigma_e_pos(i) = sigma_e(i)*(sigma_e(i)>0); %Taking positive part of sigma.
    theta_Num = theta_Num + sigma_e_pos(i);
    theta_Den = theta_Den + abs(sigma_e(i));
end

theta = theta_Num/theta_Den;
rtrial= (theta+(1-theta)/n)*sqrt(eps_n1*ce*eps_n1');
end

%*********************************************************************************
***
return
```

3.5. damage_main

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

% SET LABEL OF "vartoplot" variables  (it may be defined also outside this function)
% ----------------------------------
 LABELPLOT = {'hardening variable (q)','internal variable','damage variable (d)',
'C_a_l_g_1_1', 'C_t_g_1_1'};

E      = Eprop(1) ;
nu = Eprop(2) ;
viscpr = Eprop(6) ;
sigma_u = Eprop(4);
eta = Eprop(7);
alpha = Eprop(8);

[...]

% INITIALIZING  (i = 1) !!!!
% ***********i*
i = 1 ;
r0 = sigma_u/sqrt(E);
hvar_n(5) = r0; % r_n
hvar_n(6) = r0; % q_n
% hvar_n(6)/hvar_n(5) = 0; % --> damage at t=0
hvar_n(7) = 0;
hvar_n(8) = ce(1,1); %C_alg11
hvar_n(9) = ce(1,1); %C_tg11

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 ; %number of variables to plot
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)
vartoplot{i}(4) = hvar_n(8);
vartoplot{i}(5) = hvar_n(9);
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_n1 = strain(i,:) ; %eps at current time step
        eps_n= strain(i-1,:); %eps at previous time step

%*********************************************************************************
***
        %*      DAMAGE MODEL
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        [sigma_n1,hvar_n,aux_var] =
```

```
rmap_dano1(eps_n1,hvar_n,Eprop,ce,MDtype,n,eps_n,delta_t);
        % PLOTTING DAMAGE SURFACE
        if viscpr == 0
        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)
;
        else
        end

        else
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %**********************************************************************
        % 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)
        vartoplot{i}(4) = hvar_n(8) ;
        vartoplot{i}(5) = hvar_n(9) ;
    end
end
```