COMPUTATIONAL SOLID MECHANICS

NUMERICAL INTEGRATION OF CONSTITUTIVE DAMAGE MODELS

ASSIGNMENT 1

Alba Ruiz de Alegría

07/04/2017

Part I – Rate independent models

In this first part have been implemented the "non-symmetric" and "only-tension" damage models with linear and exponential hardening types. The codes corresponding to these implementations are on the Annex A. To test the validity of the implemented models the following cases are going to be tested.

Case 1		Case 2		Case 3	
$\Delta\overline{\sigma}_1^{(1)}=500$	$\Delta\overline{\sigma}_2^{(1)}=0$	$\Delta\overline{\sigma}_1^{(1)}=500$	$\Delta\overline{\sigma}_2^{(1)}=0$	$\Delta\overline{\sigma}_1^{(1)}=500$	$\Delta\overline{\sigma}_2^{(1)}=500$
$\Delta\overline{\sigma}_1^{(2)}=-700$	$\Delta\overline{\sigma}_2^{(2)}=0$	$\Delta\overline{\sigma}_1^{(2)}=-700$	$\Delta\overline{\sigma}_2^{(2)}=-700$	$\Delta\overline{\sigma}_1^{(2)}=-700$	$\Delta\overline{\sigma}_2^{(2)}=-700$
$\Delta\overline{\sigma}_1^{(3)}=300$	$\Delta\overline{\sigma}_2^{(3)}=0$	$\Delta\overline{\sigma}_1^{(3)}=300$	$\Delta\overline{\sigma}_2^{(3)}=300$	$\Delta\overline{\sigma}_1^{(3)}=300$	$\Delta\overline{\sigma}_2^{(3)}=300$

Case 4	
$\Delta\overline{\sigma}_1^{(1)}=-500$	$\Delta\overline{\sigma}_2^{(1)}=-500$
$\Delta\overline{\sigma}_1^{(2)}=-100$	$\Delta\overline{\sigma}_2^{(2)}=-100$
$\Delta\overline{\sigma}_1^{(3)}=600$	$\Delta\overline{\sigma}_2^{(3)}=600$

For all the cases the following data will be common:

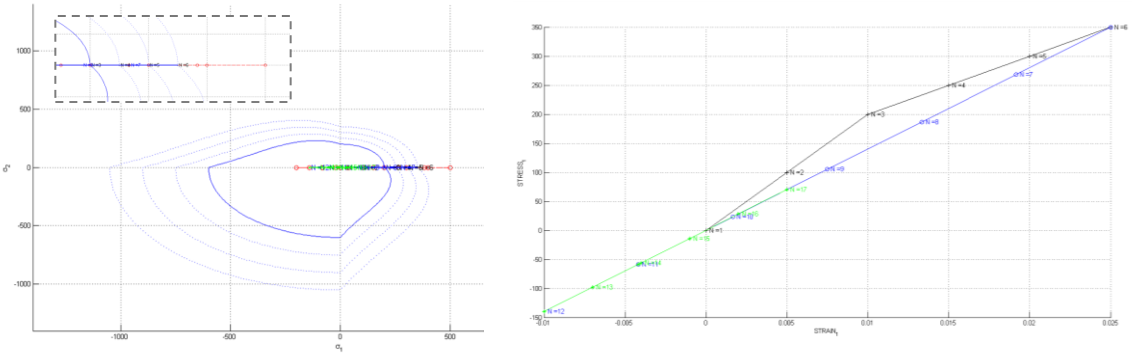
Poisson	0	Young modulus	20000	Yield stress	200

In all analyses performed with the non-symmetric model the rate between traction and compression has keep constant with a value of 3.

Case 1

In the Figure 1 the linear hardening and linear softening for a non-symmetric model is shown. When hardening is happening it can be seen that the elastic region suffers an expansion when the applied stresses are out of this region. However, in the softening case the elastic region goes in contraction. This expansion-contraction behaviour in both cases depends on the Hardening-Softening modulo (*H*). For the hardening case this parameter is set on 0.5 and in softening the parameter takes a value of -0.5. This change of the elastic region has the effect of create a positive (hardening) or negative (softening) slope on the stress-strain curve so is easy to see that the unloading has to follow a different path with a different slope due to the damage suffered by the material. As the differences between the slopes before and after the damage is bigger in softening than in hardening it can be conclude that the materials suffers more with softening behaviour.

a)



b)

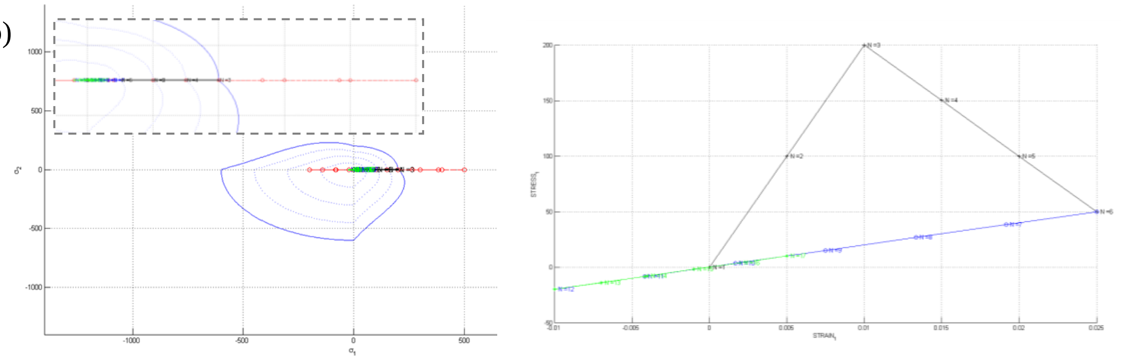


Figure 1: *Non-Symmetric model case 1. a) Linear hardening b) Linear softening*

Case 2

This second case is going to be used to analyse the difference between linear and exponential hardenings. Up to the yield surface the response is identical in the both cases. Once this limit is exceeded the linear model continuous linearly with a slope that depends on the value of H. However, in the exponential case the curve continuous exponentially. In that way the unloading slopes are different.

When these linear-exponential models are compared for a positive hardening can be seen that in the exponential case the material suffers less than with linear behaviour (Figure 2). The same happens when softening is occurring (Figure 3).

In neither case does the material break but it would be interesting to mention that in the case of total damage no matter how big are the applied stress, the strains are going to remain always in zero.

a)

b)

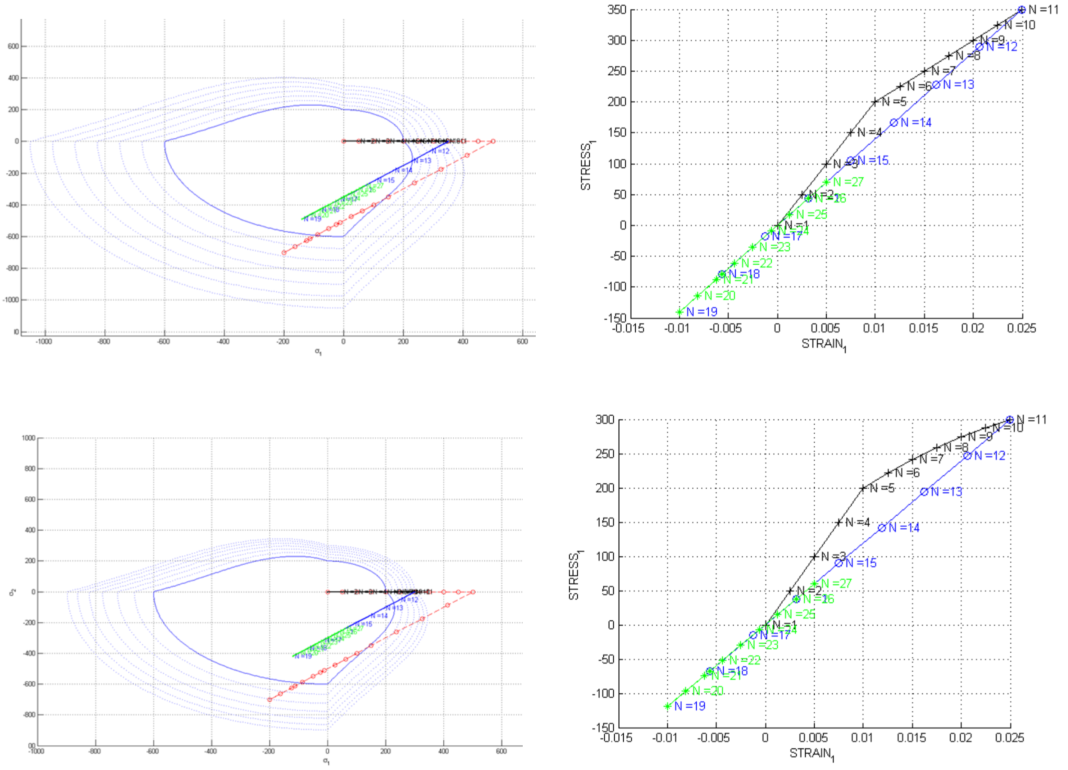


Figure 2: *Non-Symmetric model case 2. A) Linear hardening B) Exponential hardening*

a)

b)

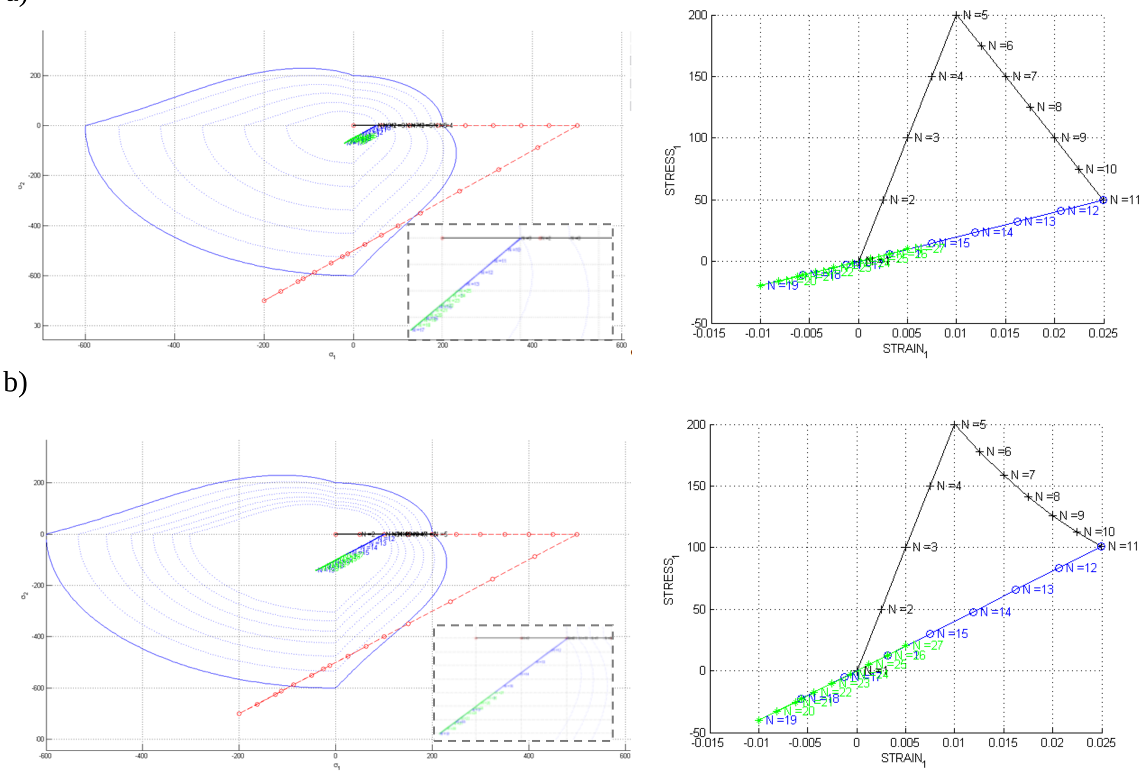


Figure 3: *Non-Symmetric model case 2. A) Linear softening B) Exponential softening*

Case 3

With this case the non-symmetric model and only-tension model for a positive load path and linear hardening are going to be represented. As it is shown in the figures below (Figure 4) the behaviour in the non-symmetric case is equal to the one obtained with the only-tension model. This occurs because the applied load path cross the yield surface from a region that is equal for both models. Nevertheless for negative stresses the response must be different, so to prove this the case 4 has been proposed.

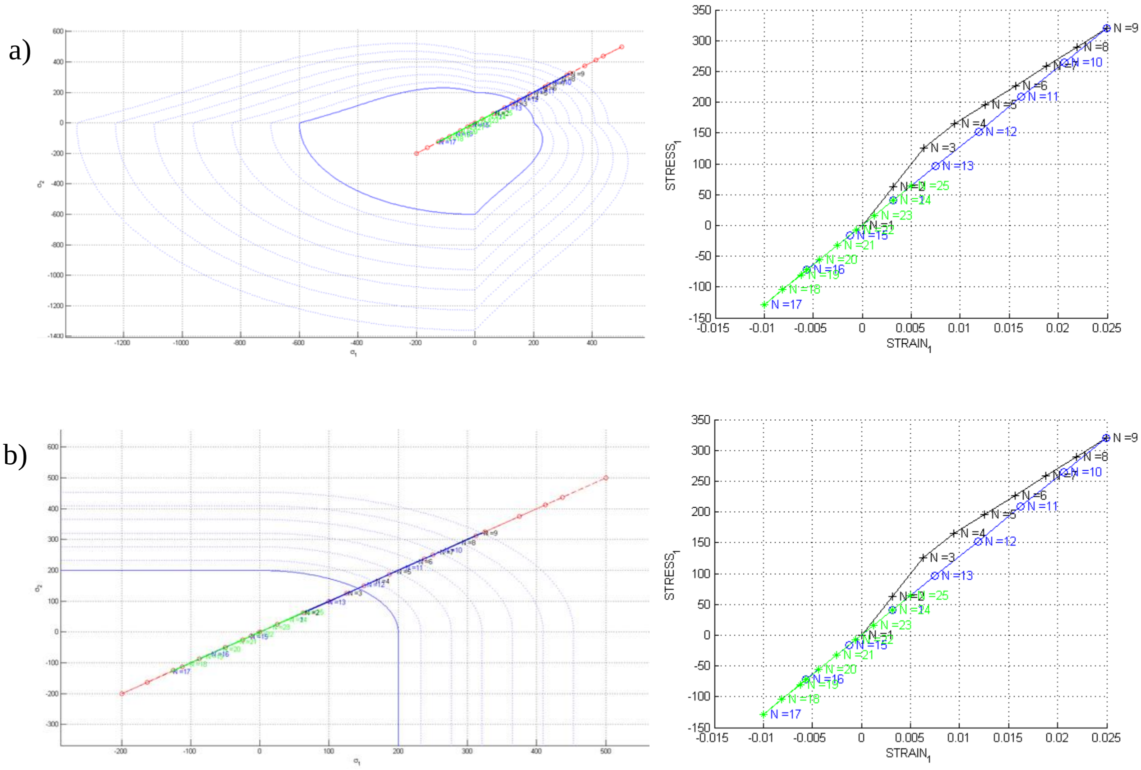


Figure 4: *Case 3. a) Non-Symmetric model with linear hardening b) Only-tension model with linear hardening*

Case 4

As can be seen in the Figure 5 in this case the behaviour is totally different. In the only-tension model, as it does not take into account any type of damage under compression, the stresses remains in the elastic region. Nonetheless, when the non-symmetric model is being used the material is damaged when the stresses exceed the yield surface and the behaviour of the material change.

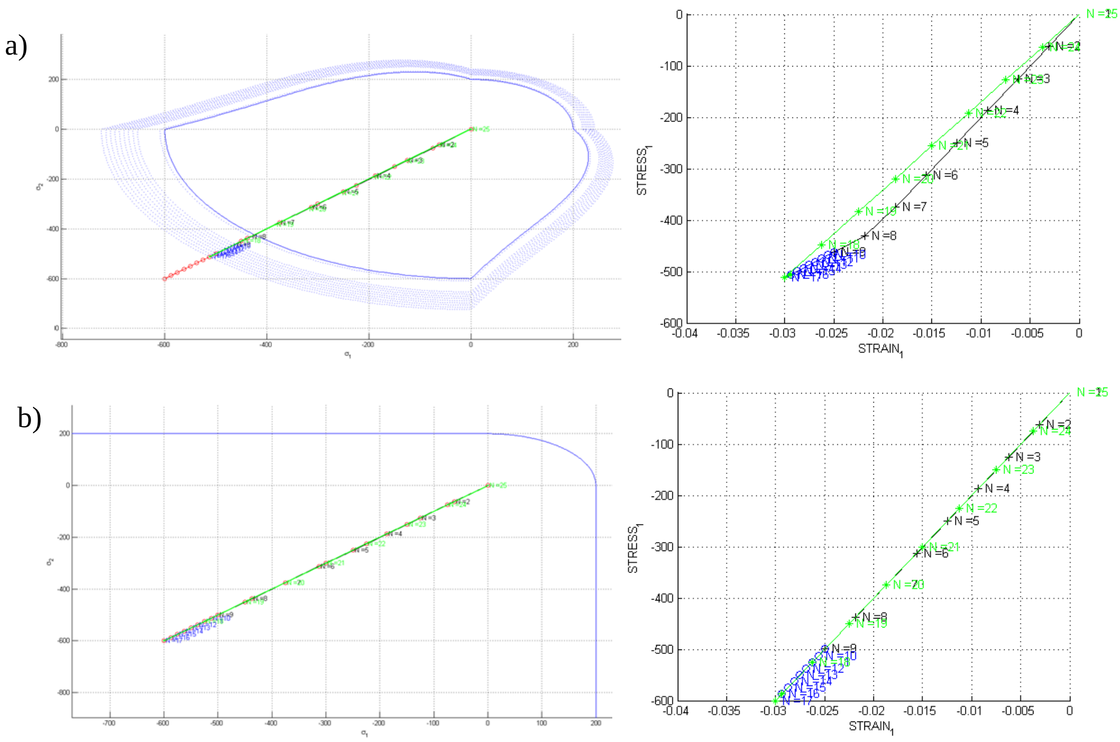


Figure 5: *Case 4. a) Non-Symmetric model with linear hardening b) Only-tension model with linear hardening*

Part II – Rate dependent models

The second part of the assignment contains several results obtained after the implementation of the viscous model. The codes corresponding to these implementations are on the Annex B.

Theoretically, when in the viscous model the viscosity parameter (η) and the α values are set on 0 and 1 respectively the inviscid model has to be restored. This has been the basis of the validation of the code. Below can be seen the results for the non-viscous model (Figure 6 a), the viscous model with the parameters η and α equal to 0 and 1 (Figure 6 b) and the viscous model with η and α values in 0.5 (Figure 6 c). In every case a symmetric model with linear hardening has been use with the following stress path.

$\Delta\overline{\sigma}_1^{(1)} = 300$	$\Delta\overline{\sigma}_2^{(1)} = 0$
$\Delta\overline{\sigma}_1^{(2)} = 200$	$\Delta\overline{\sigma}_2^{(2)} = 0$
$\Delta\overline{\sigma}_1^{(3)} = 200$	$\Delta\overline{\sigma}_2^{(3)} = 0$

As can be seen the inviscid model is obtained so the code is validate. In the Figure 6 c a saw effect appears that is a typical behaviour of the brittle materials. When they reach to the damage surface links inside the material breaks and which provide some resistance to the material, this is why this backward jump is happening in the stresses.

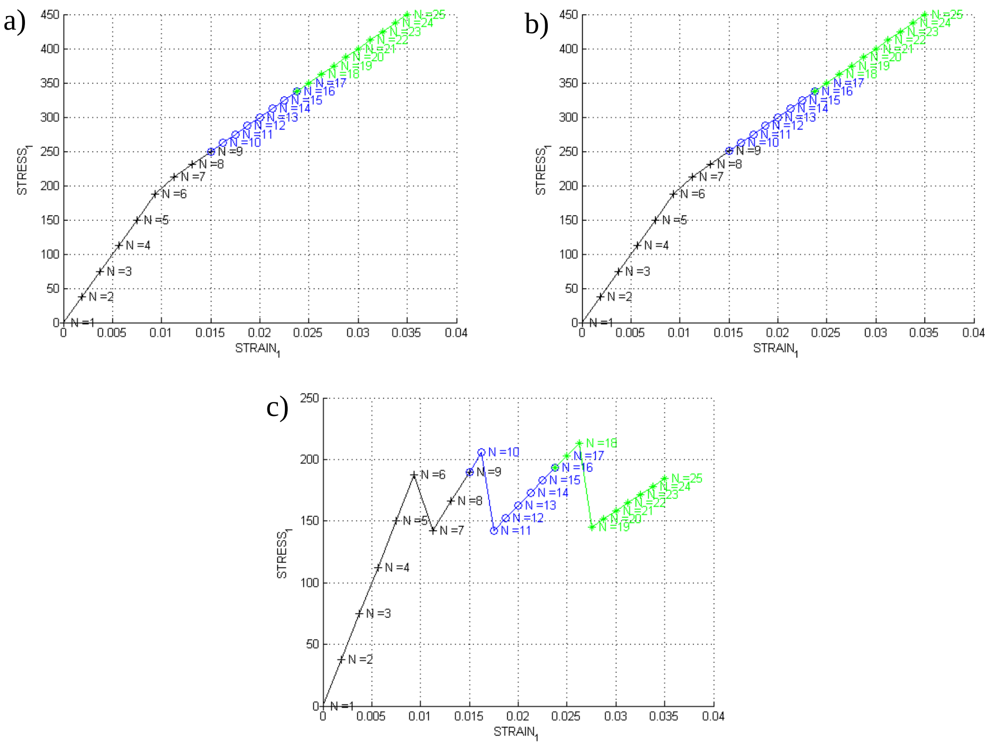


Figure 6: *Validation of the viscous model. a) Inviscid model b) Viscous model with η=0 and α=1 c) Viscous model with η=0.5 and α=0.5*

Now some parameters are going to be modified to know more about their effect. First of all the different values of the viscosity parameter are going to be tested with α equal to 0.5 and a time interval of 10 seconds. The values that has been set are 0 (Figure 7 a), 0.25 (Figure 7 b), 0.5 (Figure 7 c), 0.75 (Figure 7 d) and 1(Figure 7 e). As can be seen the value of η change the jump in the stresses change. With small η values the jumps are higher than with small values. So with the same stress path as the smaller is this parameter less is the damage.

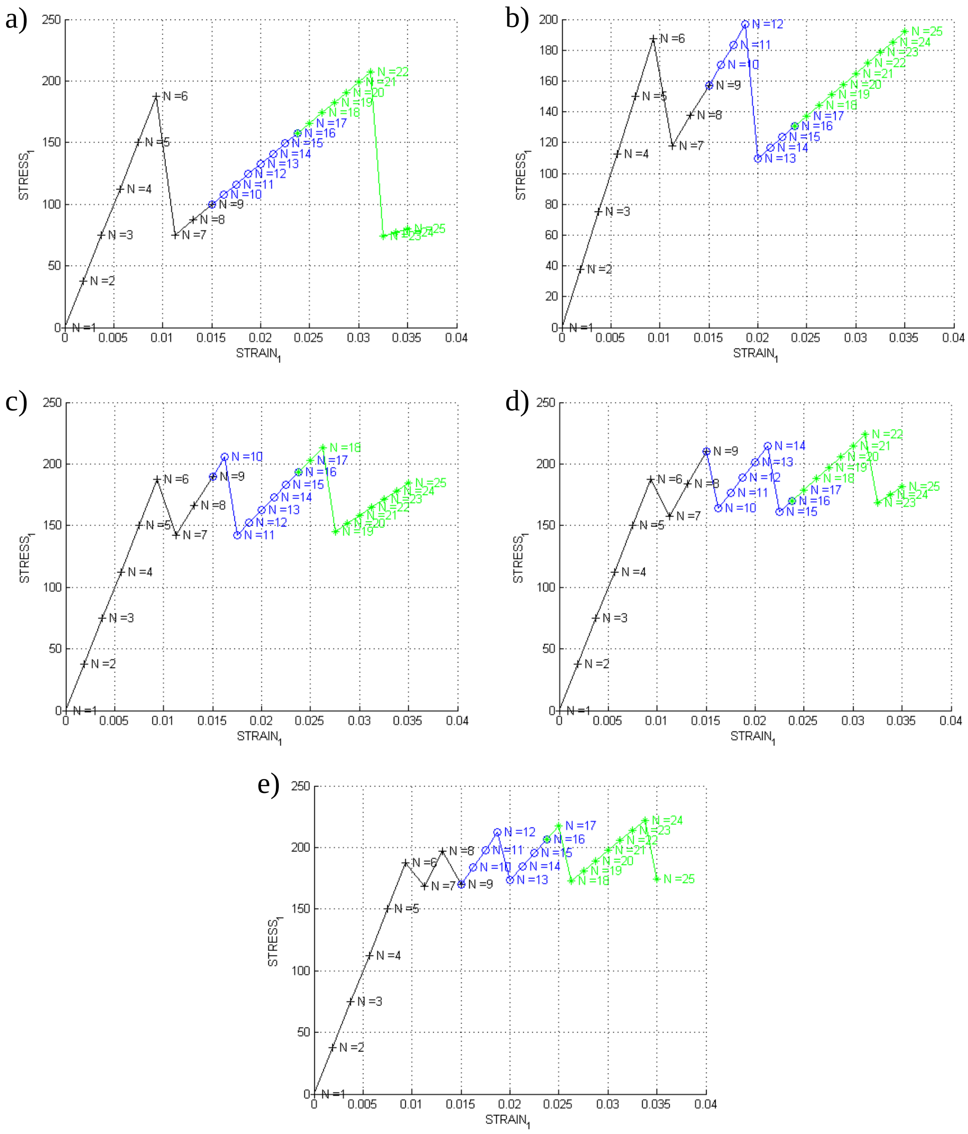


Figure 7: *Viscous model with α=0.5 time interval of 10 s and a) η=0 b) η=0.25 c) η=0.5 d) η=0.75 e) η=1*

When η and α have the constant value of 0.5 and the strain rate is changed the same stress path has to be applied in different time interval so as longer is the interval lower is going to be the material damage so the jumps are going to be higher (see Figure 8).

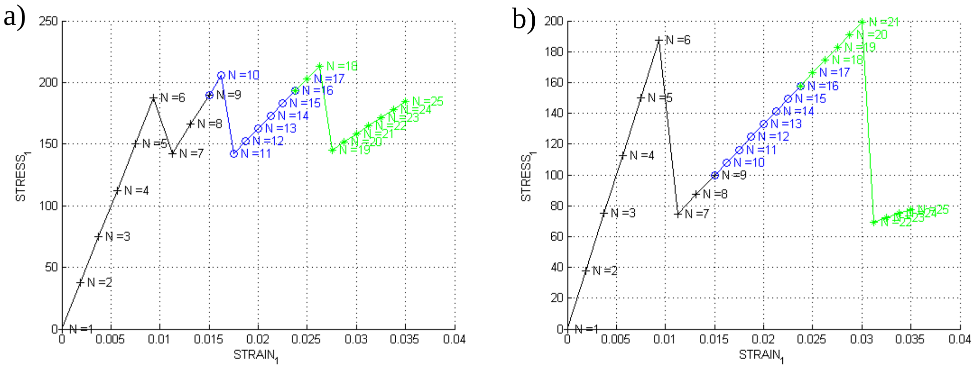


Figure 8: *Viscous model with α=0.5, η=0.5 and time interval of a) 10 s and b)10000 s*

The last parameter that is going to be modified is the α parameter that change the time iterative method. When the value of α is between 0.5 and 1 the method is unconditionally stable but for values below 0.5 the stability is going to depend on the time step. For big time steps the schemes become instable.

In the graphs below the η is of 0.5 and the time interval is of 10. A take the values that correspond to Forward Euler scheme (α=0)(Figure 9 a), Crank-Nicholson scheme (α=0.5) (Figure 9 b) and Backward Euler scheme (α=1) (Figure 9 c).

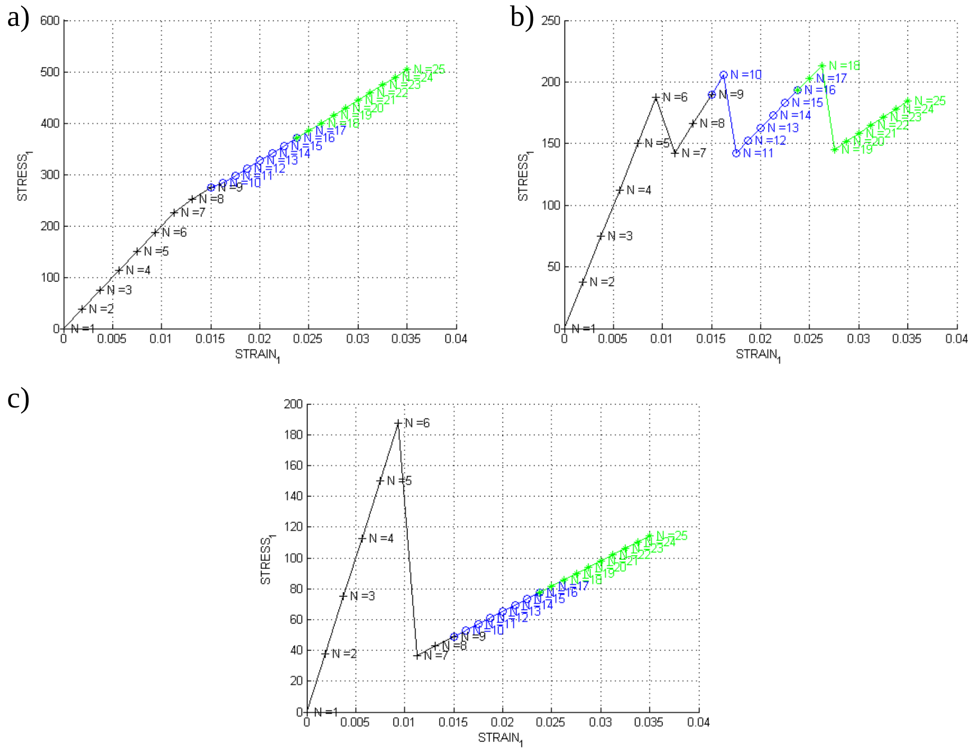


Figure 9: *Viscous model with η=0.5, time interval of 10 s and α equal to a) 0-Forward Euler scheme b)0.5-Crank-Nicholson scheme and c)1- Backward Euler scheme*

ANNEXES

Annex A

Implementation of the rate independent models

Modified codes:

- dibujar_criterio_dano1

```
elseif MDtype==2
    tetha=[(-pi/2)*0.9999:0.01:pi*0.9999];
    %*************************************************************************************
    %* RADIUS
    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
        sigma= [m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
        sigmapos=sigma.*(sigma>0);
        radio(i)= q/sqrt(sigmapos*ce_inv*sigma');

        s1(i)=radio(i)*m1(i);
        s2(i)=radio(i)*m2(i);

    end
    hplot =plot(s1,s2,tipo_linea);
elseif MDtype==3
    tetha=[0:0.01:2*pi];
    %*************************************************************************************
    %* RADIUS
    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

        sigma=[m1(i) m2(i) 0 nu*(m1(i)+m2(i))];
        sigmapos=sigma.*(sigma>0);
        tita=sum(sigmapos)/sum(abs(sigma));

        radio(i)= (q/sqrt(sigma*ce_inv*sigma'))/(tita+(1-tita)/n);

        s1(i)=radio(i)*m1(i);
        s2(i)=radio(i)*m2(i);

    end
    hplot =plot(s1,s2,tipo_linea);

end
```

- Modelos_de_dano1

```
%*******************************************************************************************
if (MDtype==1)      %* Symmetric

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

elseif (MDtype==2)  %* Only tension

    sigmab=(eps_n1*ce);
    sigmabpos=sigmab.*(sigmab>0);

    rtrial=sqrt(sigmabpos*eps_n1');

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

    sigma=(eps_n1*ce);
    sigmapos=sigma.*(sigma>0);
    sigmaabs=abs(sigma);
    tita=sum(sigmapos)/sum(sigmaabs);
    C=(tita+(1-tita)/n);
    rtrial= C*sqrt(eps_n1*ce*eps_n1');


end
%*******************************************************************************************
```

- rmap_dano1

```
%Exponential
% zero_q es q_inf por abajo
q_inf=r0+(r0-zero_q);
if H>0
    q_n1=q_n+((H*(q_inf-r0)/r0)*exp(H*(1-rtrial/r0)))*delta_r;
else
    q_n1=q_n+((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial/r0))))*delta_r;
end
```

Annex B

Implementation of the rate dependent models

Modified codes:

- rmap_dano1

  Input values have been added to the function.

```
[sigma_n1,hvar_n1,aux_var] = rmap_dano1 (eps_n1,eps_n0,hvar_n,Eprop,ce,MDtype,n,viscpr,delta_t)
```

The code has been divided by an if to separate the viscous model from the inviscid one. Below is only the implementation of the part of the viscous model.

```
eta = Eprop(7);
ALPHA_COEFF = Eprop(8);


[rtrial_n] = Modelos_de_dano1 (MDtype,ce,eps_n0,n); % TauEps_n (Viscous model)
[rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);
[rtrial_nalpha]=(1-ALPHA_COEFF)*rtrial_n+ALPHA_COEFF*rtrial ; % TauEps_n+alpha (Viscous model)
```

```
if viscpr == 1
    if (rtrial_nalpha > r_n)
        %*   Loading
        fload=1;
        delta_r=rtrial_nalpha-r_n;
        r_n1=((eta-delta_t*(1-ALPHA_COEFF))/(eta-ALPHA_COEFF*delta_t))*r_n+...
            (delta_t*rtrial_nalpha)/(eta+ALPHA_COEFF*delta_t);
        if hard_type == 0
            %  Linear
            q_n1= q_n+ H*delta_r;
        else
            %Exponential
            % zero_q es q_inf por abajo
            q_inf=r0+(r0-zero_q);
            if H>0
                q_n1=q_n+((H*(q_inf-r0)/r0)*exp(H*(1-rtrial_nalpha/r0)))*delta_r;
            else
                q_n1=q_n+((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial_nalpha/r0))))*delta_r;
            end
        end

         if(q_n1<zero_q)
            q_n1=zero_q;
        end

    else ...
```

- damage_main

  In the main loop the following line has been added before calling rmap_dano1

```
eps_n0 = strain(i-1,:) ;
```