Computational Solid Mechanics

Assignment 1: Continuum damage model

Ye Mao

Ye Mao, mao.ye@estudiant.upc.edu
Master of Numerical methods on engineering - Universitat Politècnica de Catalunya

Index

Part 1 Rate independent model

1. INTRODUCTION

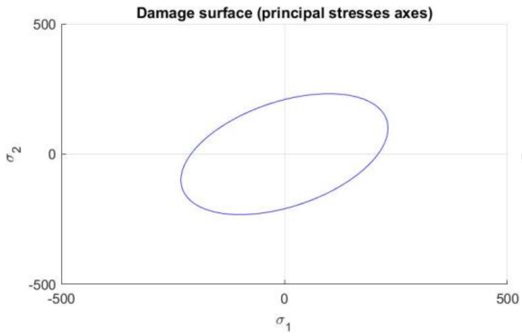


Figure 1. Symmetric damage model

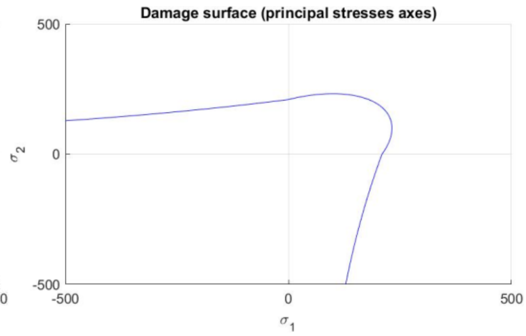


Figure 2. Only-tension damage model

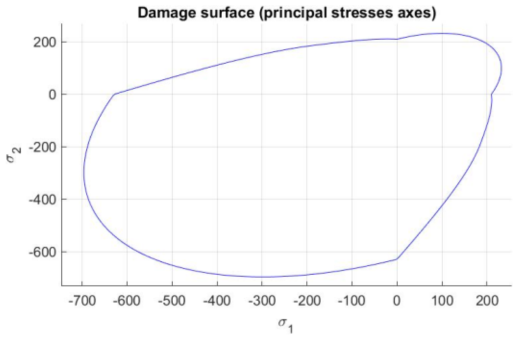


Figure 3. Non-Symmetric damage model

The above figures shown the continuum damage models by stress-strain curves in this report. Firstly, in symmetric damage model, the elastic tension area and elastic compression area are equivalent with absolute value. Second, in the only-tension damage model, the elastic tension area is smaller than the previous one and the elastic compression area is infinite. Last, in the Non-symmetric damage model, the elastic compression area is larger then elastic tension one. That mean, the parameter ‘n’ affects in it. This ‘n’ is the ratio of compression elastic limit to tension elastic limit.

In part I, modify the Matlab codes to implement the following condition:

1) Non-symmetric tension-compression damage model
2) Tension-only model
3) Linear and exponential hardening/softening ($H < 0 \; and \; H > 0$)

With the codes modification, this program can be used to get the result of Tension-only model and Non-symmetric tension-compression damage model following different load path. It contains stress-strain curves, damage variable curves and hardening/softening variable curves.

According to the result, the softening material is easier to be damaged. The multidirectional loading will improve the yield stress of the material. The Elastic

Modulus will decrease while damage happen. At the same time, the damage speed of exponential model is smaller than linear model

2. METHODOLOGY AND RESULTS

2.1 Case 1, stress-strain and damage in symmetric model

$$\cdot\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 0 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = 0 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 0 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

Case 1, the loading is only applied on x-coordinate. When the load path 1 imposed, the material is in elastic stage. We can expect that the stress-strain curve will be a straight with load path 1 which is uniaxial tensile loading. The path 2 loading is uniaxial tensile unloading and compressive loading. It is tensile unloading firstly and then go to compressive loading. Specially, we can find the stress is still inside the elastic domain. While the loading increase to 1000 in path 3, it is beyond the elastic domain and then the damage happens in this stage.

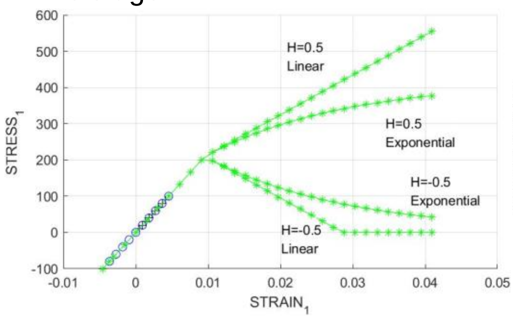


Figure 4. Case1 Stress1-strain1 curve

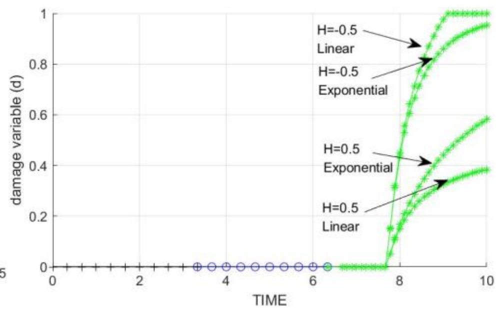


Figure 5. Case1 Damage variable respect to time

According to the above figure, 4 conditions are stated, Linear model H=0.5, Exponential model H=0.5, Linear model H=-0.5, Exponential model H=-0.5.

While H=-0.5, the material is easier to be damaged than the condition which H=0.5. The Linear model and Exponential model will happen damage in the same point, then Linear model's damage go faster than the Exponential one.

While H=0.5, both of Linear and Exponential model have lower damage level than the condition which H=-0.5. The Linear one show low speed than Exponential one to damage.

Overall, the hardening model stronger the material's ability to against the damage.

2.2 Case 1, hardening variable q in symmetric model

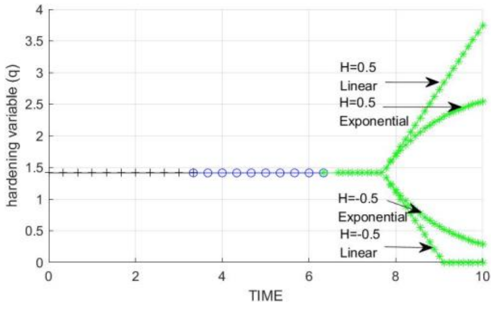


Figure 6. Case1 Hardening variable respect to time

The above figure shows the changing of hardening variable q respect to time. When the load within the initial damage surface, q keep constant. It would not be changed until the load extends outside of initial damage surface. In the same time, from that point, the Linear model's q goes faster both in increasing and decreasing process than Exponential one.

2.3 Case 2, stress-strain and damage in symmetric model

$$\cdot\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 0 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = -200 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 1000 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

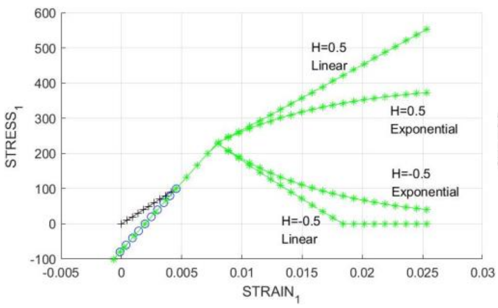


Figure 7. Case2 Stress1-strain1 curve

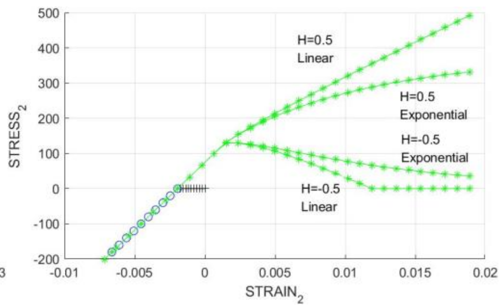


Figure 8. Case2 Stress2-strain2 curve

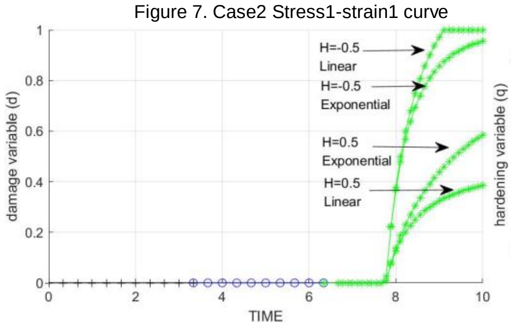


Figure 9. Case2 Damage variable respect to time

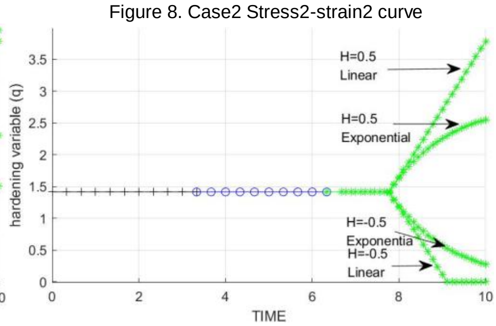


Figure 10. Case2 Hardening variable respect to time

Case 2, the loading is not only applied on x-coordinate but also on y-coordinate in path 2 and 3. It becomes to be multidirectional loading problem. We take into account the norm stress, since the damage will happen when the norm stress satisfies the damage principle. Yield stress increasing in this case comparing with Case 1.

We can find that in the damage variable figure and hardening variable figure, the damage happens point and q increasing/decreasing point appears later than case 1 but very close to it.

2.4 Case 3, stress-strain, damage and variable q in symmetric model

$$\cdot\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 100 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = -200 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 1000 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

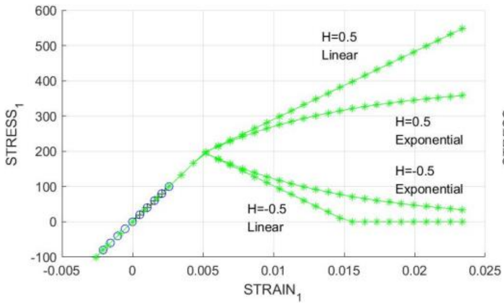


Figure 11. Case3 Stress1-strain1 curve

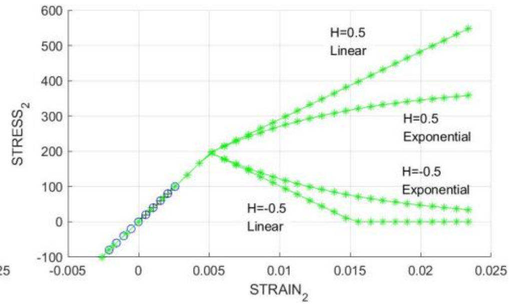


Figure 12. Case3 Stress2-strain2 curve

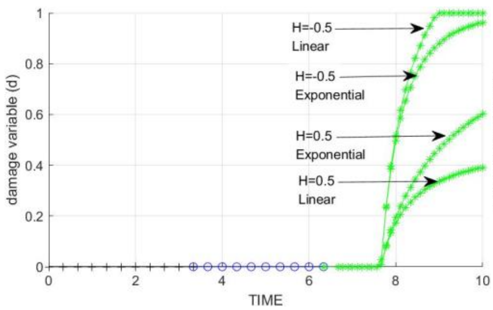


Figure 13. Case3 Damage variable respect to time

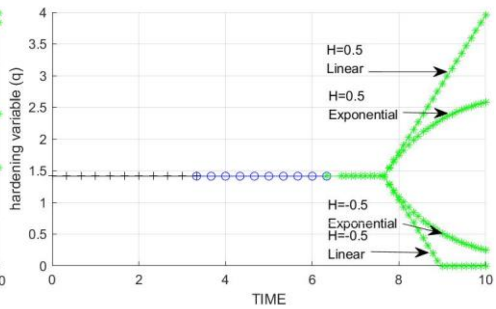


Figure 14. Case3 Hardening variable respect to time

Case 3, the loadings are all applied on x-coordinate and on y-coordinate in all path 1, 2&3. Since path 1 is not limited to x-coordinate than the Case2, so the curves of load path 1 and 2 will be the same straight line.

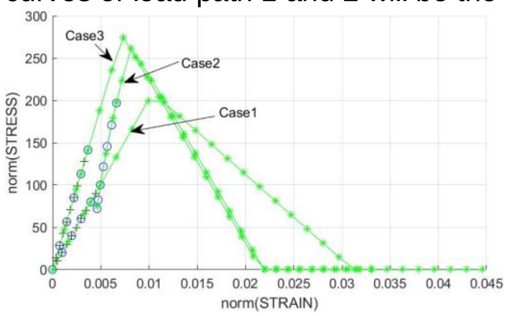


Figure 15. Comparison of nom(strain)-nom(stress) curves for different case of Symmetric model Hardening/Softening number=-0.5

From above figure, we can find the yield stress sequence is Case3>Case2>Case1.This phenomenon is caused by the reality that the yield criterion will change for different loading combination. In this case, multidirectional loading has increased the yield stress and make the material stronger. Case 3 looks like Case 2. The yield stress of case 3 is the highest one. That mean the material in case 3 is the most difficult to be damaged. In the same time, the slope of Case3 and Case2 are bigger than the slope of Case1, which means that the Elastic Modulus has increased and material becomes harder under the combined loading condition.

2.5 Case 1, stress-strain, damage and variable q in tension-only model

$$\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 0 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = 0 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 0 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

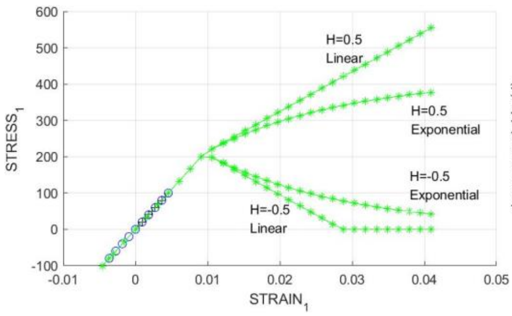


Figure 16. Case1 Stress1-strain1 curve

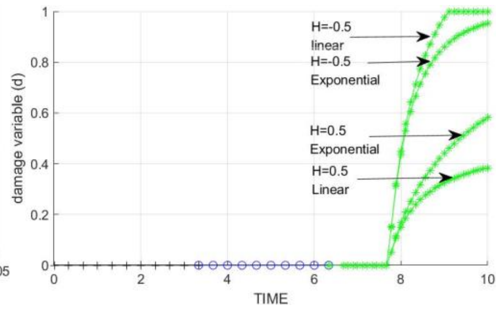


Figure 17. Case1 Damage variable respect to time

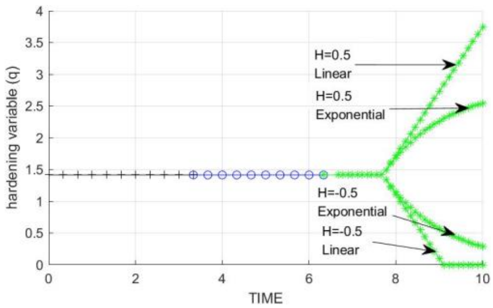


Figure 18. Case1 Hardening variable respect to time

2.6 Case 2, stress-strain, damage and hardening variable q in tension-only model

$$\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 0 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = -200 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 1000 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

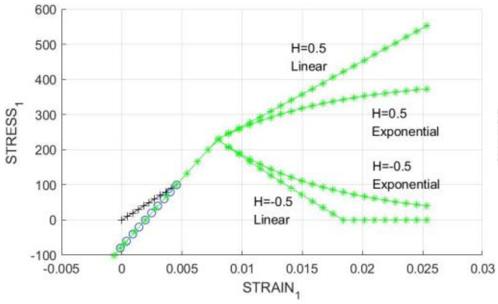


Figure 19. Case2 Stress1-strain1 curve

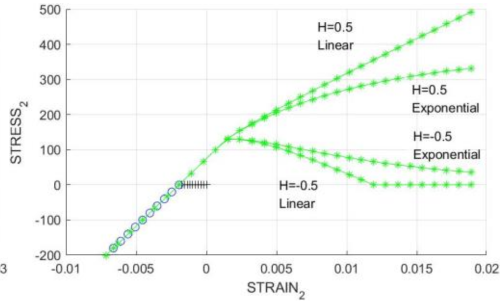


Figure 20. Case2 Stress2-strain2 curve

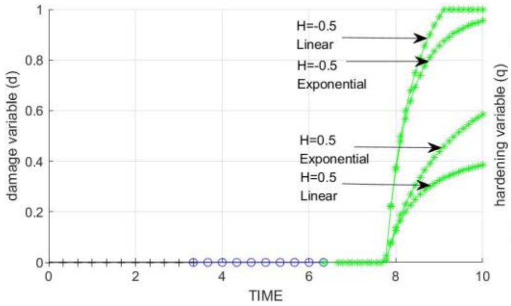


Figure 21. Case2 Damage variable respect to time

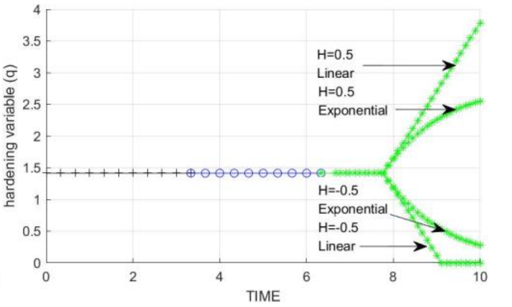


Figure 22. Case2 Hardening variable respect to time

2.7 Case 3, stress-strain, damage and hardening variable q in tension-only model

$$\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 100 \\ \Delta\sigma_1^{(2)} = -200;\ \Delta\sigma_2^{(2)} = -200 \\ \Delta\sigma_1^{(3)} = 1000;\ \Delta\sigma_2^{(3)} = 1000 \end{cases}$$

$$E = 20000, \sigma_{yield} = 200, v = 0.3, t = 10\ \alpha = 1$$

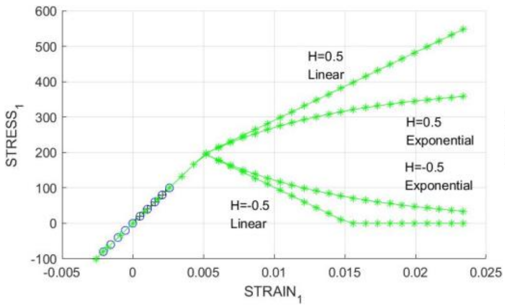


Figure 23. Case3 Stress1-strain1 curve

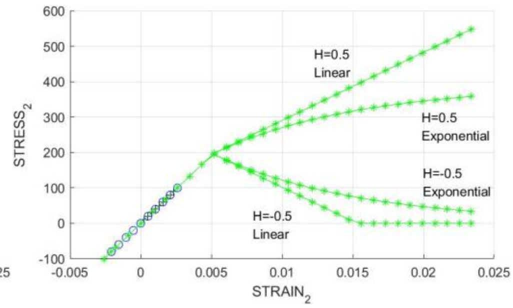


Figure 24. Case3 Stress2-strain2 curve

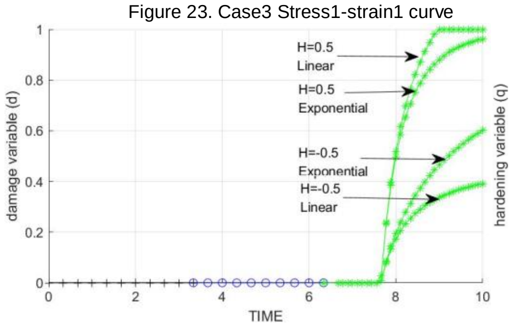


Figure 25. Case3 Damage variable respect to time

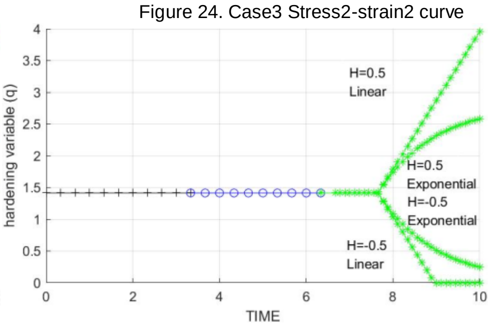


Figure 26. Case3 Hardening variable respect to time

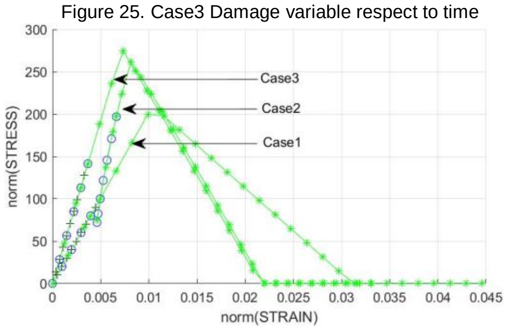


Figure 27. Case3 Hardening variable respect to time

Comparing with the case of tension-only model, we can conclude that multidirectional loading will increase the yield stress and Elastic Modulus.

2.8 According to the appendix codes, the Non-symmetric model has the similar behavior as Tension-only model. In this report, due to the space of this report, the result will not be shown again.

Part 2 Rate dependent model

1. INTRODUCTION

This part mainly discussed several coefficient's effects on the result of stress-strain curve. Compare different α to discuss its effect on the C11 component of tangent and algorithmic constitutive operators. We should take care of choosing the appropriate value of α. The value of α should be suggested to larger than 0.5, since the result will be expansive on time and show oscillation while the value of α less than 0.5.

This part is symmetric tension-compression model but under uniaxial stress. Linear hardening/softening parameter should be used. To show the viscosity parameter behaver clearly, we can set following condition to compare:

Loading path:

$$\begin{cases} \Delta\sigma_1^{(1)} = 100;\ \Delta\sigma_2^{(1)} = 0 \\ \Delta\sigma_1^{(2)} = 200;\ \Delta\sigma_2^{(2)} = 0 \\ \Delta\sigma_1^{(3)} = 300;\ \Delta\sigma_2^{(3)} = 0 \end{cases}$$

2 METHODOLOGY AND RESULTS

2.1 Effect of viscosity coefficient η

Case1: $E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0, t = 10\ \alpha = 1, \eta = 0,1,2,10$

Case2: $E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0.1, t = 10\ \alpha = 1, \eta = 0,1,2,10$

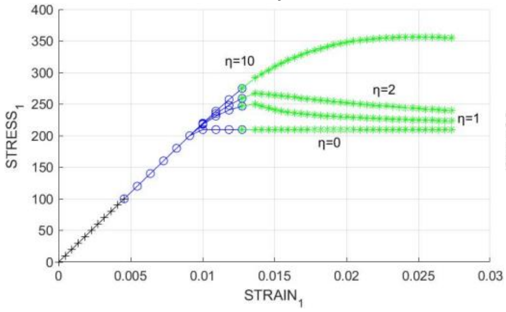


Figure 28. Case1 Stress-strain curve η=0,1,2,10

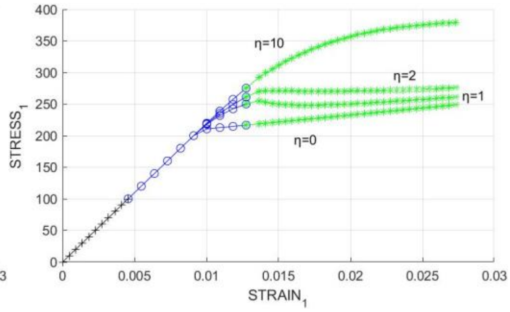


Figure 29. Case2 Stress-strain curve η=0,1,2,10

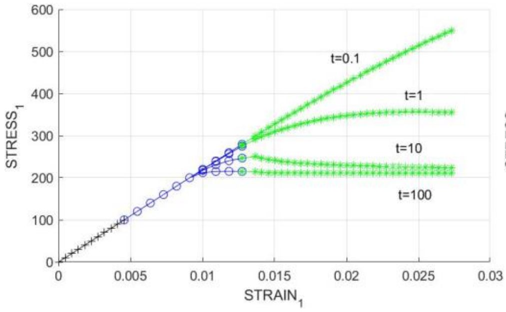
In Case 1, it can be found that while the viscous parameter increases, the yield stress will increase. At the same time, the yield stress will increase with the hardening/softening parameter increase form Case 2.

2.2 Effect of strain rate coefficient $\frac{d\varepsilon}{dt}$

To check the effect of strain rate for this method. We take different time values parameters since the strain rate is depend on time.

Case1: $E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0, \eta = 1, \alpha = 1, t = 0.1,1,10,100$

Case2: $E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0.1, \eta = 1, \alpha = 1, t = 0.1,1,10,100$



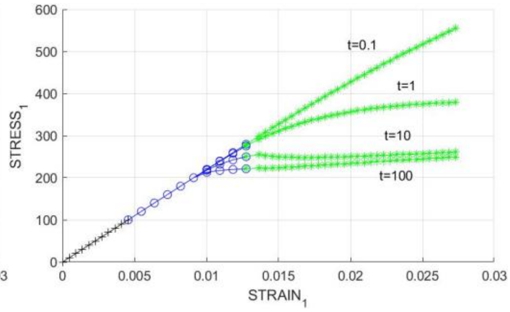


Figure 30. Case1 Stress-strain curve t=0.1,1,10,100

Figure 31. Case2 Stress-strain curve t=0.1,1,10,100

From above figures, rate coefficient gets the similar behaver as viscosity coefficient. While the strains rate is very low, t is very big, this system can be considered as quasi-static. That means load is applied very slowly and the material's deformation will become very slowly. In this case, the inertia force might be neglected. That means the material will be in equilibrium in whole time. To the opposite, while the strain rate is high, t is small, the material can not omit the energy applied consequently. In this case, a process of damage will appear.

2.3 Effect of α

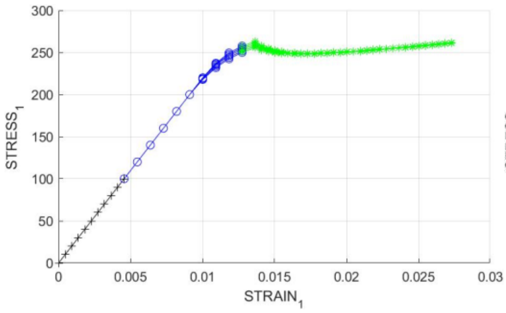
To check the effect of α, we consider the range [0,1]. We should expect that as long as the α is changed, it will be affect as one stability method.

Case:1

$$E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0.1, \eta = 1, t = 10,$$
$$\alpha = 0, 0.25, 0.5, 0.75, 1$$

Case:2

$$E = 20000, \sigma_{yield} = 200, v = 0.3, H = 0.1, \eta = 1, t = 1000,$$
$$\alpha = 0, 0.25, 0.5, 0.75, 1$$



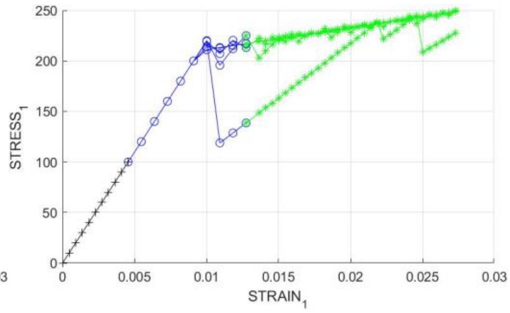


Figure 32. Case1 Stress-strain curve t=10, α=0,0.25,0.5,0.75,1

Figure 33. Case1 Stress-strain curve t=1000, α=0,0.25,0.5,0.75,1

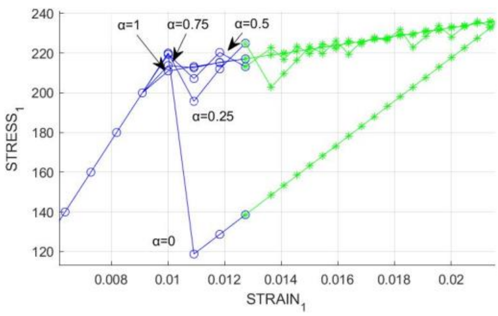


Figure 34. Zoom of Figure 33

From the above figures we can find while the strain rate is low, whether α is large or small, the system keep stability. But while the strain rate is high, equal to quasi-static, α will play a role in stability. If $\alpha \leq 0.5$, result will be oscillation.

On the other hand, it would be stable while $\alpha \geq 0.5$. Actually, $\alpha = 0.5$ make the method owe second order accurate.

2.4 Effect of α on the evolution of C_{tg11} and C_{ALg11}

From Figure 35 and Figure 37 we can also find that while $\alpha \leq 0.5$, the instability is shown again. These values accuracy is preserved. So we can get the stable range $0.5 \leq \alpha \leq 1$ conditionally. By the other hand, for Lax Theorem, the convergent value is $0 \leq \alpha \leq 1$

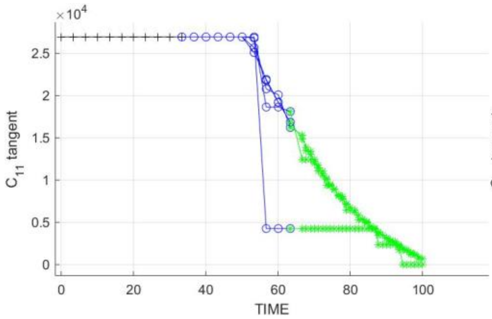


Figure 35. C_{tg11} respect to time, α=0,0.25,0.5,0.75,1

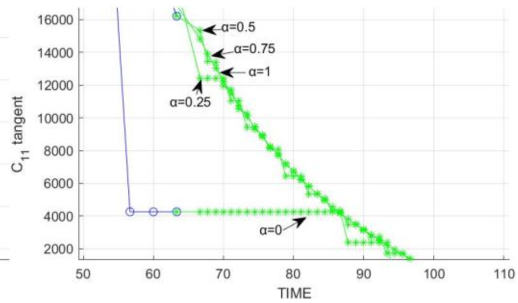


Figure 36. Zoom of Figure 35

From the above figure, it is understood where the material is undamaged, it is the continuous line; and where is the process of damage starts, it is where discontinuities start to show up.

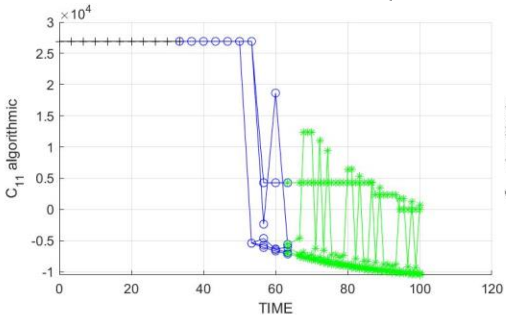


Figure 37. C_{Alg11} respect to time, α=0,0.25,0.5,0.75,1

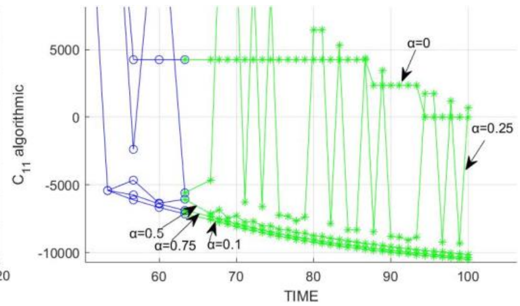


Figure 38. Zoom of Figure 37

The algorithm constitutive matrix shows a discontinuity in its behavior. Because the expression of algorithm constitutive matrix is a piecewise function while $C_{Alg11} = C_{tg11}$ in elastic region and $C_{Alg11} \neq C_{tg11}$ in the damage region. The stability of different α value behave similarly as it does for the constitutive tangent matrix.

REFERENCE

[1] Lecture slides in Computational Solid Mechanics.

APPENDIX

In this appendix, mainly show the modification of the function dbujar_criterio_dano1, rmap_dano1, modelos_de_dano1 and damage_main. Here, we use mark […] instead of the parts not modified in the original function.

1. dbujar_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);
%*****************************************************************
***********************
[…]

%*****************************************************************
***********************
% POLAR COORDINATES
[…]
%%%
%%%
elseif MDtype==2
    % Comment/delete lines below once you have implemented this
case
    % *****************************************************
    %menu({'Damage surface "ONLY-TENSION" has not been
implemented yet. '; ...
    %    'Modify files "Modelos_de_dano1" and
"dibujar_criterio_dano1"' ; ...
    %    'to include this option'},  ...
    %    'STOP');
    %error('OPTION NOT AVAILABLE')

    tetha=[0:0.1:2*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) ;
    M1    = zeros(1,Contador) ;
    M2    = zeros(1,Contador) ;


    for i=1:Contador
```

```
        M1(i)=m1(i);
        M2(i)=m2(i);
        if m1(i)<0
            M1(i)=0;
        end
        if m2(i)<0
            M2(i)=0;
        end
        radio(i)= q/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);
%%%
%%%
elseif MDtype==3
    % Comment/delete lines below once you have implemented this
case
    % ******************************************************
%     menu({'Damage surface "NON-SYMMETRIC" has not been
implemented yet. '; ...
%         'Modify files "Modelos_de_dano1" and
"dibujar_criterio_dano1"' ; ...
%         'to include this option'},  ...
%         'STOP');
%     error('OPTION NOT AVAILABLE')

    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
        sum_stress=0;
        vec_stress=[m1(i) m2(i) nu*(m1(i)+m2(i))];
        sum_stress_abs=sum(abs(vec_stress));
        for j=1:3
            if vec_stress(j)<0
                vec_stress(j)=0;
            end
            sum_stress=sum_stress+vec_stress(j);
        end

        theta_stress=sum_stress/sum_stress_abs;
        coeft=theta_stress+(1-theta_stress)/n;

        radio(i)= q/(coeft*(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
```

2. rmap_dano1

```
function [sigma_n1,hvar_n1,aux_var] = rmap_dano1
(eps_n,eps_n1,hvar_n,Eprop,ce,MDtype,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)  ;

eta=Eprop(7);%add
alpha=Eprop(8); %add
%*****************************************************************
**********************
[…]

%*****************************************************************
**********************
%*
initializing                                            %*
 r0 = sigma_u/sqrt(E);
 zero_q=1.d-6*r0;
 A=abs(H);
 q_inf=r0+sign(H)*0.99*r0;
% if(r_n<=0.d0)
%     r_n=r0;
%     q_n=r0;
% end
%*****************************************************************
**********************
[…]
%*****************************************************************
**********************
%*        Damage
surface
%*
%[rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);

[tau_eps_n]=Modelos_de_dano1
(MDtype,ce,eps_n,n);%sqrt(eps_n*ce*eps_n');
[tau_eps_n1]=Modelos_de_dano1
```

```
(MDtype,ce,eps_n1,n);%sqrt(eps_n1*ce*eps_n1');
[rtrial]=(1-alpha)*tau_eps_n+alpha*tau_eps_n1;
%*****************************************************************
**********************


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


if(rtrial > r_n)
    %*   Loading

    fload=1;
    %r_n1= rtrial  ;
    r_n1= ((eta-delta_t*(1-
alpha))/(eta+alpha*delta_t))*r_n+(delta_t/(eta+alpha*delta_t))*r
trial;
    delta_r=r_n1-r_n;

    if hard_type == 0
        %  Linear
        q_n1= q_n+ H*delta_r;
    else
        % Comment/delete lines below once you have implemented
this case
        %
*******************************************************
        %menu({'Hardening/Softening exponential law has not been
implemented yet. '; ...
        %'Modify file "rmap_dano1" ' ; ...
        %'to include this option'},  ...
        %'STOP');
        %error('OPTION NOT AVAILABLE')

        %Hexp=A*(zero_q-r0)/r0*exp(A*(1-r_n/r0));


        q_n1= q_inf-(q_inf-r0)*exp(A*(1-r_n1/r0));


    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
% 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')

%***************************************************************
**********************


%***************************************************************
**********************
%* Updating historic
variables                                                  %*
%  hvar_n1(1:4)  = eps_n1p;
hvar_n1(5)= r_n1 ;
hvar_n1(6)= q_n1 ;
%***************************************************************
**********************




%***************************************************************
**********************
%* Auxiliar
variables
%*
aux_var(1) = fload;
aux_var(2) = q_n1/r_n1;
aux_var(3) = (q_n1-H*r_n1)/r_n1^3;
Ce_tan=(1-dano_n1)*ce;
%Ce_tan=(1-dano_n1)*ce-
aux_var(1)*aux_var(3)*((ce*eps_n1')*(ce*eps_n1')');
Ce_alg=Ce_tan-
aux_var(1)*((alpha*delta_t)/(eta+alpha*delta_t)*aux_var(3)*r_n1/
(sqrt(sigma_n1'*(inv(ce))*sigma_n1)/(1-
dano_n1))*((ce*eps_n1')*(ce*eps_n1')'));
%Ce_alg=(1-dano_n1)*ce-
aux_var(1)*((alpha*delta_t)/(eta+alpha*delta_t)*aux_var(3)*r_n1/
(1/tau_eps_n1)*((ce*eps_n1')*(ce*eps_n1')'));
aux_var(4)=Ce_tan(1,1); %Ce_tan
aux_var(5)=Ce_alg(1,1); %Ce_alg
%***************************************************************
**********************
```

3. modelos_de_dano1

```
function [rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n)
%***************************************************************
***********************
[…]                              %*
%***************************************************************
***********************
[…]
%***************************************************************
***********************
if (MDtype==1)      %* Symmetric
rtrial= sqrt(eps_n1*ce*eps_n1')                         ;

elseif (MDtype==2)  %* Only tension

    sigma_n1 =ce*eps_n1';
```

```
    for j=1:4
            if  sigma_n1(j)<0
                sigma_n1(j)=0;
            end
    end
    cei=inv(ce);
    eps_n1m=(cei*sigma_n1)';

    rtrial= sqrt(eps_n1m*ce*eps_n1m')                       ;


elseif (MDtype==3)  %*Non-symmetric
        sigma_n1 =ce*eps_n1';
        sum_stress=0;
        vec_stress=[sigma_n1(1) sigma_n1(2) sigma_n1(4)];
        sum_stress_abs=sum(abs(vec_stress));
        for j=1:3
            if  vec_stress(j)<0
                vec_stress(j)=0;
            end
            sum_stress=sum_stress+vec_stress(j);
        end

        theta_stress=sum_stress/sum_stress_abs;
        coeft=theta_stress+(1-theta_stress)/n;

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


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

4. 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_{11} tangent','C_{11}
algorithmic'}; %%add
%LABELPLOT{3}='damage variable (d)';
E       = Eprop(1) ; nu = Eprop(2) ;
viscpr = Eprop(6) ;
sigma_u = Eprop(4);



if ntype == 1
    menu('PLANE STRESS has not been implemented yet','STOP');
    error('OPTION NOT AVAILABLE')
elseif ntype == 3
    menu('3-DIMENSIONAL PROBLEM has not been implemented
yet','STOP');
```

```
    error('OPTION NOT AVAILABLE')
else
    mstrain = 4     ;
    mhist   = 6     ;
end

if viscpr == 1
    % Comment/delete lines below once you have implemented this
case
    % ******************************************************
    %menu({'Viscous model has not been implemented yet. '; ...
    %    'Modify files "damage_main.m","rmap_dano1" ' ; ...
    %    'to include this option'},  ...
    %    'STOP');
    %error('OPTION NOT AVAILABLE')

else
    Eprop(7)=0;
    Eprop(8)=1;

end


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)
vartoplot{i}(4) = ce(1,1); % Ce_tan11
vartoplot{i}(5) = ce(1,1); % Ce_alg11
```

```
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_n=strain(i-1,:);
        eps_n1 = strain(i,:) ;
        %*******************************************************
*******************************
        %*      DAMAGE MODEL
        % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
        [sigma_n1,hvar_n,aux_var] =
rmap_dano1(eps_n,eps_n1,hvar_n,Eprop,ce,MDtype,n,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

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
        %*******************************************************
***************
        % 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) = aux_var(4); % Ce_tan11
        vartoplot{i}(5) = aux_var(5); % Ce_alg11
    end
end
```