UPC - BARCELONA TECH
MSc Computational Mechanics
Spring 2018

Computational Solid Mechanics

Assignment 1 Damage Models

Due 23/03/2018

Anurag Bhattacharjee

CONTENTS

Part I- Rate Independent Model

1-a) The supplied MATLAB code has been modified to include the Continuum- Isotropic damage "non-symmetric tension-compression damage" and the "tension only" damage models. The plots have been obtained for the following material properties-

Young Modulus, E= 20000 MPa
Yield Stress, σy= 200 MPa
Poisson Ratio, ν= 0.3
Hardness, H= 0.5
Ratio of compression strength to tension strength, n=3

The following plots were obtained using the above properties-

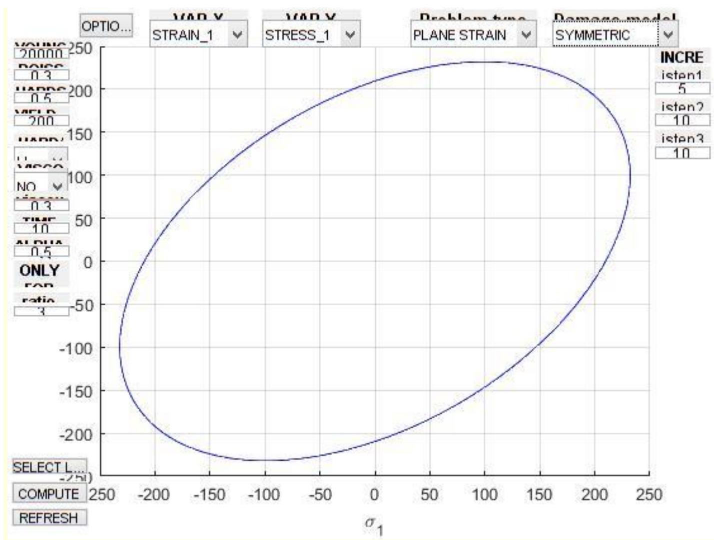

Fig-1a: Symmetric model

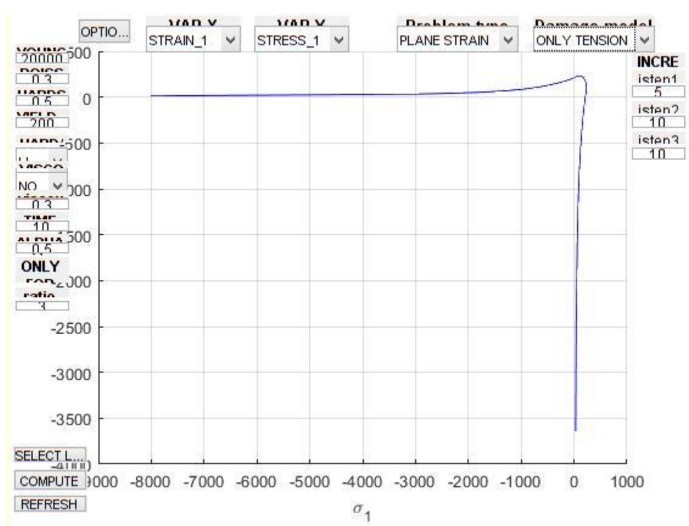

Fig-1b: Tension only model

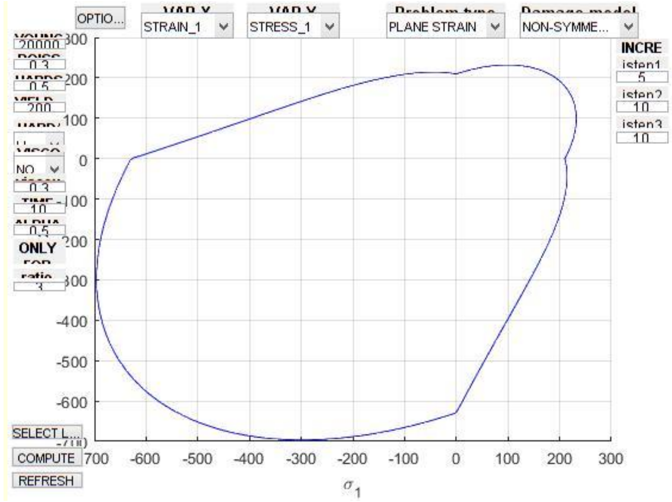

Fig-1c: Non-symmetric Tension-Compression Damage model

1-b) Linear and exponential models have been implemented for different cases considering hardening/softening (H>0 and H<0). The codes have been modified accordingly and the plots have been studied in the next section for various models.

1-c) The following cases have been considered for the analysis of different models-

Case-1

$\Delta\sigma^1_{(1)} = 100$ MPa ; $\Delta\sigma^2_{(1)} = 0$

$\Delta\sigma^1_{(2)} = -500$ MPa ; $\Delta\sigma^2_{(2)} = 0$

$\Delta\sigma^1_{(3)} = 900$ MPa ; $\Delta\sigma^2_{(3)} = 0$

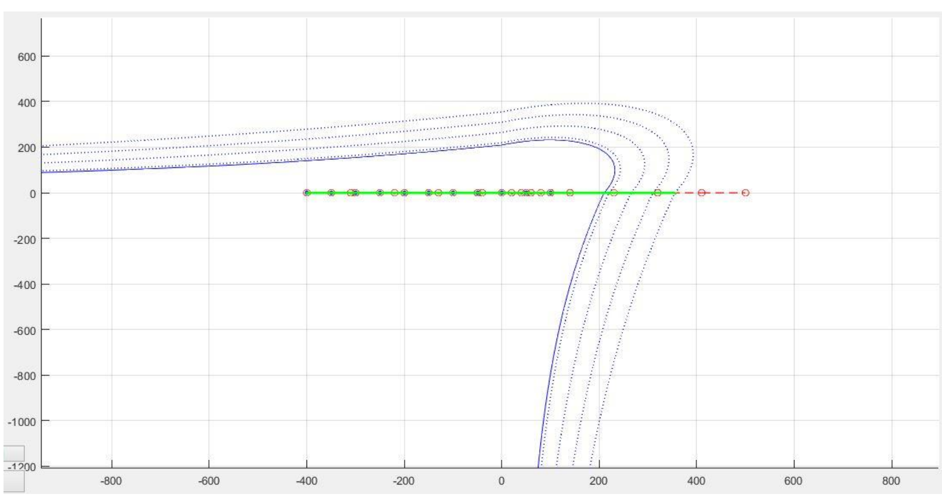

Fig-2a: σ_1 vs σ_2 for Tension only model (H=0.5)

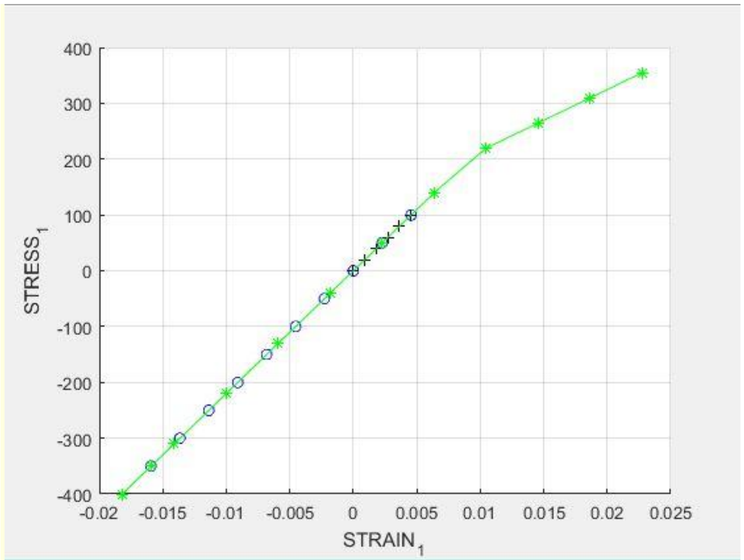

Fig- 2b: Linear Hardening for Tension only model

Black-crossed line shows uniaxial tensile loading, blue-circled line represents uniaxial tensile unloading/ compressive loading and the green-starred line represents uniaxial compressive unloading/tensile loading.

As can be seen from the plot, the loading stresses do not go beyond the yield stress, so the loading falls in the elastic regime which is reflected in the plot as all the points lie in a straight line with slope E. The tension only model does not take into account the yield stresses in the compression region so tensile unloading/ compressive loading is always elastic in this region (third quadrant of Fig- 2a), which also means that the material can only fail in tension.

We also note that during the next stage of uniaxial loading, as the stresses cross the yield stress the material undergoes hardening (H=0.5) and the stresses continue to increase with increasing strain. In case of linear hardening the curve increases in a linear fashion.

Next we take the case of linear softening for non-symmetric tension-compression damage model for the same values as the earlier model.

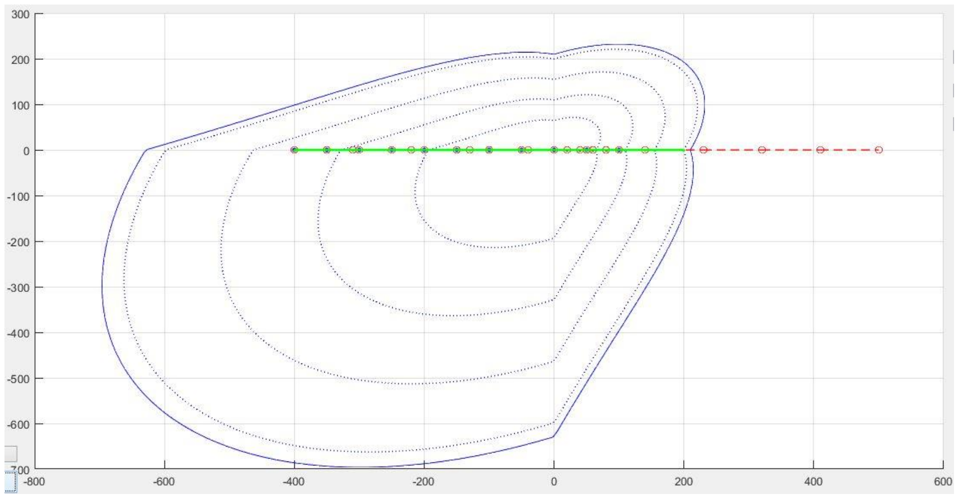

Fig-2c: σ_1 vs σ_2 for Non-symmetric tension-compression damage model (H=-0.5)

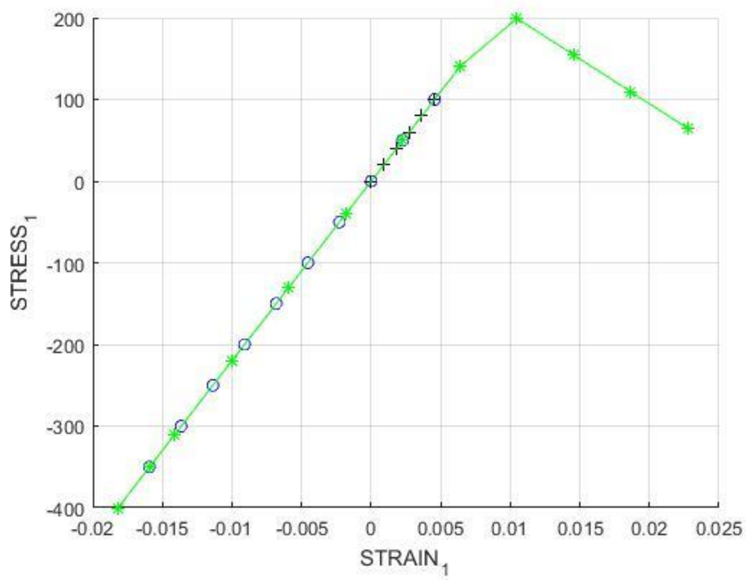

Fig- 2d: Linear Softening for Non-symmetric tension-compression model

As in the previous case, the plot is consistent for the first stage of elastic loading and then tensile unloading/ compressive loading. Next during the subsequent loading we observe that as the stresses cross the material yield stress, with increasing strain the stresses decrease which is the result of softening (H<0) and this behavior is as expected.

Case-2

$\Delta\sigma^1_{(1)}$ = 100 MPa ; $\Delta\sigma^2_{(1)}$ = 0

$\Delta\sigma^1_{(2)}$ = −300 MPa ; $\Delta\sigma^2_{(2)}$ = -300 MPa

$\Delta\sigma^1_{(3)}$ = 600 MPa ; $\Delta\sigma^2_{(3)}$ = 600 MPa

Linear hardening and softening have been analyzed for this case for various models.

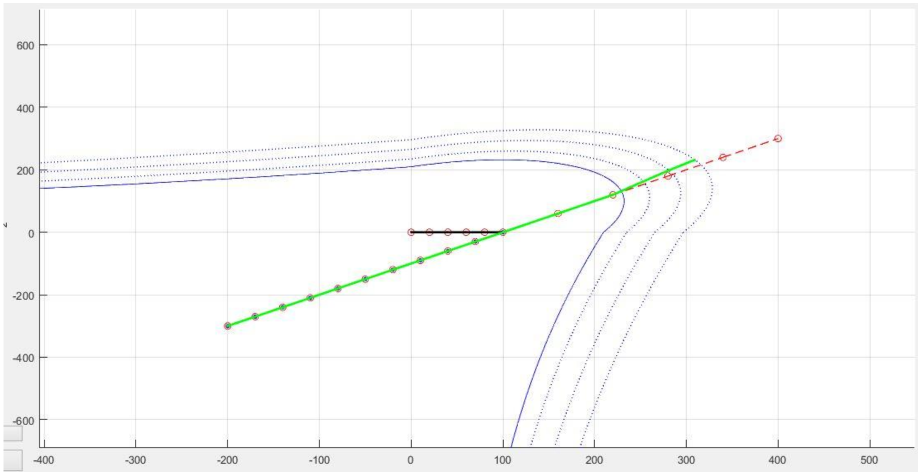

Fig-3a: σ_1 vs σ_2 for Tension only model (H=0.5)

Just as in the previous case we observe in the above plot that the initial uniaxial tensile loading gives an elastic response as the stresses do not cross the yield stress. Also in the case of biaxial tensile unloading/ compressive loading, the response is still elastic irrespective of the value of the loads as the tension only model do not account for the yield stresses in the compressive regime. Next during biaxial loading the domain expands as the stresses cross the yield stress boundary.

This is evident from figure 3b, that as the stresses cross the yield boundary during the last loading step, the stresses increase with the strain causing a hardening effect on the material and the internal variable (q) evolves accordingly.

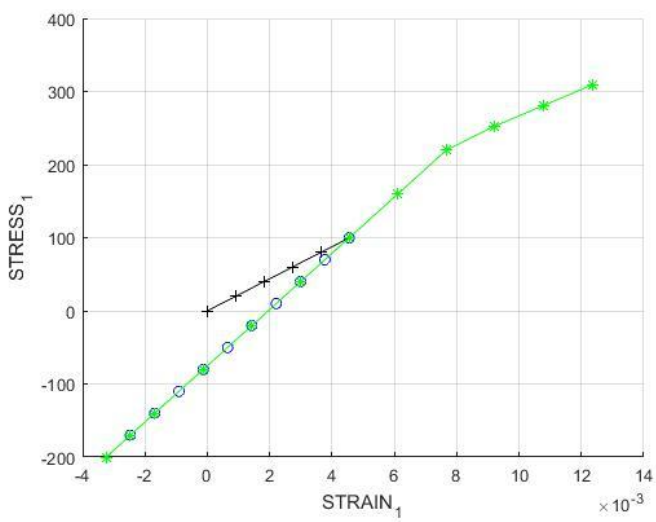

Fig- 3b: Linear Hardening for Tension only model

Now we consider the case of non-symmetric tension-compression model to represent the case of linear softening.

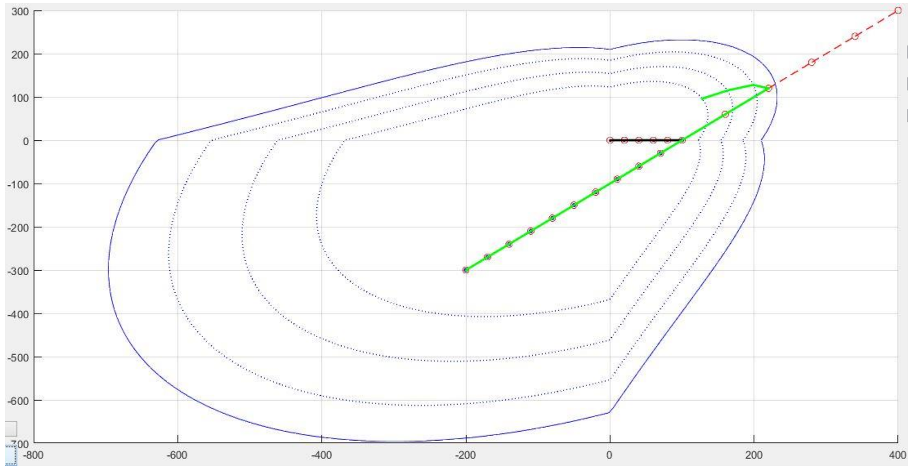

Fig-3c: σ_1 vs σ_2 for Non-symmetric tension-compression damage model (H=-0.5)

For this case also we observe that the initial loading and unloading fall within the elastic limits and hence, we get purely elastic response. However as the stresses exceed the yield limit, the stresses begin to drop with the evolution of internal variable as the strain increases, as can be seen from the following stress strain curve highlighting the linear softening for H=-0.5.

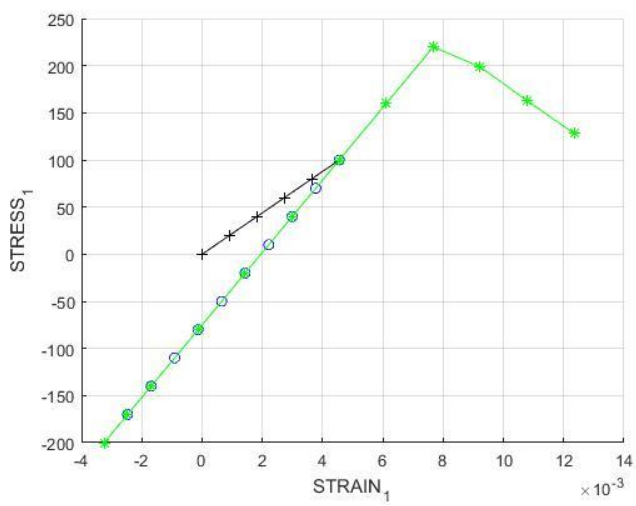

Fig- 3d: Linear Softening for Non-symmetric tension-compression model

Case-3

$\Delta\sigma^1{}_{(1)}$ = 100 MPa ; $\Delta\sigma^2{}_{(1)}$ = 100 MPa

$\Delta\sigma^1{}_{(2)}$ = −300 MPa ; $\Delta\sigma^2{}_{(2)}$ = -300 MPa

$\Delta\sigma^1{}_{(3)}$ = 600 MPa ; $\Delta\sigma^2{}_{(3)}$ = 600 MPa

Exponential hardening and softening have been analyzed for this case for various models.

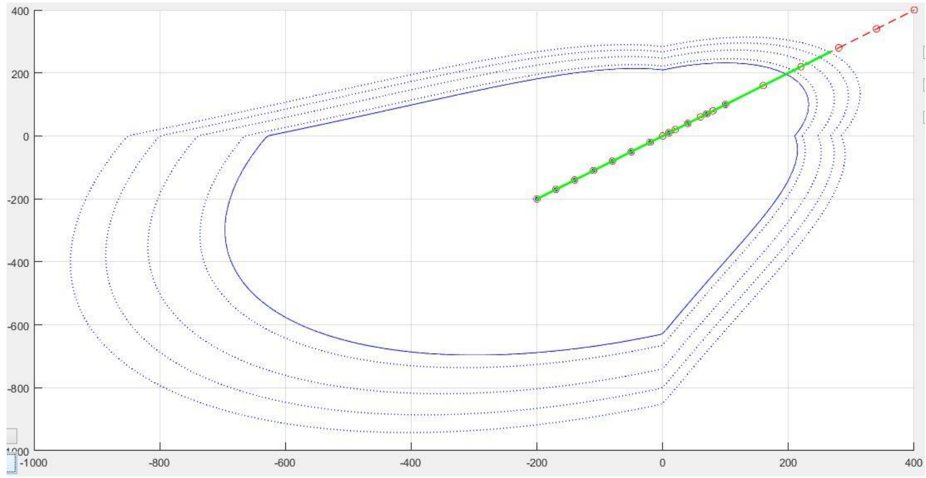

Fig-4a: σ_1 vs σ_2 for Non-symmetric tension-compression damage model (H=0.5)

Here we analyze exponential hardening for non-symmetric tension-compression model. We can see from the plot that the biaxial tensile loading falls within the elastic region and therefore exhibits elastic response, straight line with slope E. The biaxial tensile unloading/ compressive loading also does not exceed the elastic threshold and therefore shows elastic behavior. However, during the subsequent stage of biaxial compressive unloading/ tensile loading, hardening effect is observed as the stresses begin to grow with the increase in strains. This physical significance of this behavior can be interpreted as the material trying to increase it's yield stress limit to try and account for this hardening effect.

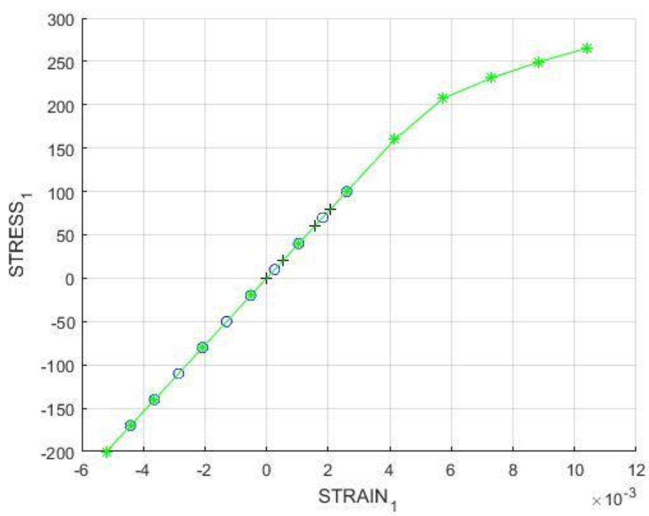

Fig- 4b: Exponential Softening for Non-symmetric tension-compression model

Now we observe the effect of exponential softening for Tension only model. It behaves in a similar fashion for the initial biaxial tensile loading and biaxial tensile unloading/compressive loading. However during the stage of biaxial compressive unloading/tensile loading we find that the stresses decrease with the increasing strain which results due to the softening property of the material as can be seen from the figure 4d. However the stresses outside the yield limit follows an exponential profile because of considering exponential nature of softening.

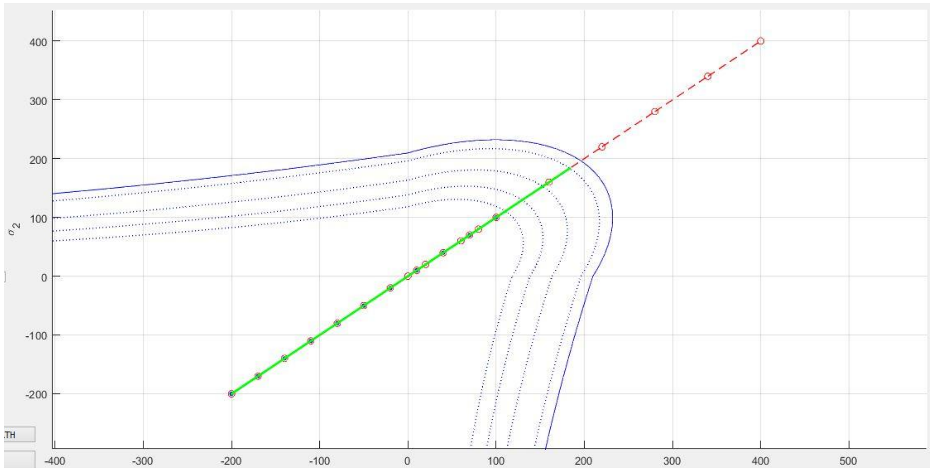

Fig-4c: σ_1 vs σ_2 for Tension only model (H=-0.5)

The exponential nature of the softening is even more evident from the stress-strain plot given below.

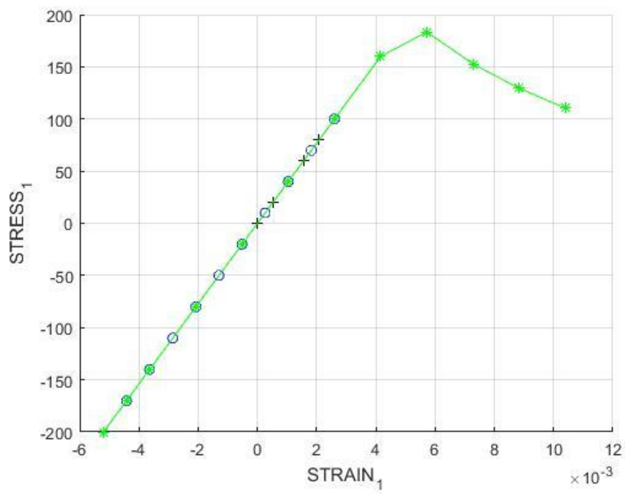

Fig- 4d: Exponential softening for Tension only model

Part II- Rate Dependent Model

In this section we consider time as an independent variable which means that the stress tensor can still change even when the strain tensor remains constant because of its dependence on time.

In the subsequent sections we have implemented the MATLAB code to account for Rate-dependent model for plane strain case of isotropic visco-damage "symmetric tension-compression" model.

2-a) Here we check the correctness of the implementation by considering dependency on various parameters. We consider an uniaxial loading/unloading and the following parameters for the subsequent simulations.

$\Delta\sigma^1{}_{(1)}$ = 100 MPa ; $\Delta\sigma^2{}_{(1)}$ = 0
$\Delta\sigma^1{}_{(2)}$ = −200 MPa ; $\Delta\sigma^2{}_{(2)}$ = 0
$\Delta\sigma^1{}_{(3)}$ = 600 MPa ; $\Delta\sigma^2{}_{(3)}$ = 0
Young Modulus, E= 20000 MPa
Yield Stress, σy= 200 MPa
Poisson Ratio, ν= 0.3
Hardness, H= 0.5
Ratio of compression strength to tension strength, n=3

Case-1: Different viscosity parameters η

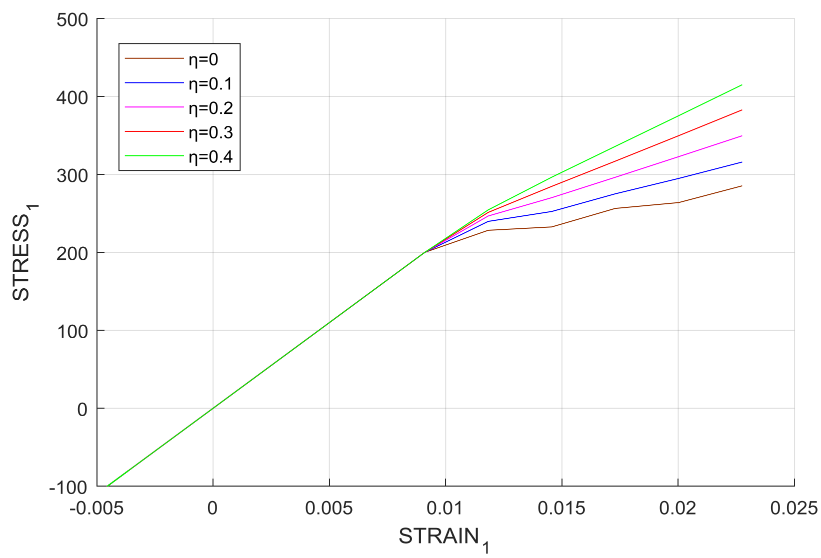

Fig-5a: Dependency of stress-strain curve on the variation of η

For uniaxial loading /unloading cycle, the above plot is obtained. It is evident from the stress-strain curve that the stresses outside the elastic region increase with the increase in viscosity which is consistent with the damper model which requires a high value of force for higher values of viscosity. And in accordance with the theory we find the elastic region remains unaffected by the effects of viscosity.

Case-2: Different strain rate values

For this case we chose to vary the strain rate by assigning different values of total time to carry out the simulation and then analyzing the effect on the resulting stress with the variation of time. It is clear from the figure obtained in 5b, that there is no variation in stress within the elastic limit as was the observation in the earlier case of variation of viscosity. However we observe that once the stresses cross the yield stress for the material, it starts deforming non-elastically and the stresses show variation with strain rates. Higher values of strain rates yield higher values of stresses.

This could probably be attributed to the fact that a slow strain rate or in other words allowing the strains to slowly increase over time produces a softening effect on the material and as a result the stresses begin to decrease.

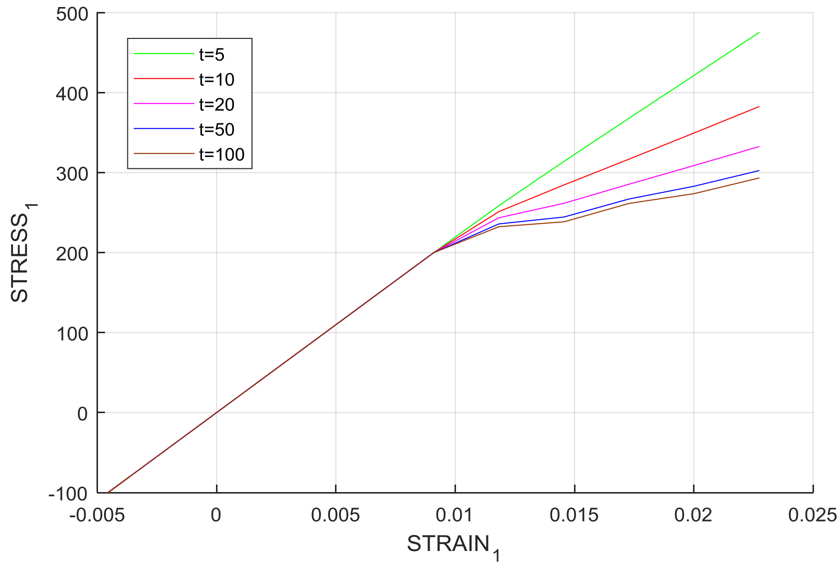

Fig-5b: Dependency of stress-strain curve on the variation of total time

Case-3: Different α values

The α time integration method gives different numerical methods for different values of α. The following figure-5c illustrates how the stress-strain relationship varies for different numerical methods or for different values of α.

It is evident from the plot that stresses obtained are different for different values of α. α=0 α=0.25, α=0.5, α=0.75 and α=1. These deviations result from the different discretization schemes employed by these methods. Although explicit schemes have low computational costs but they are not unconditionally stable which can give rise to unreliable solutions. While all the other methods here are first order accurate, Crank-Nicholson (α=0.5) has second order accuracy and is unconditionally stable. So this method for α=0.5 is expected to give most reliable results in most situations compared to the other methods.

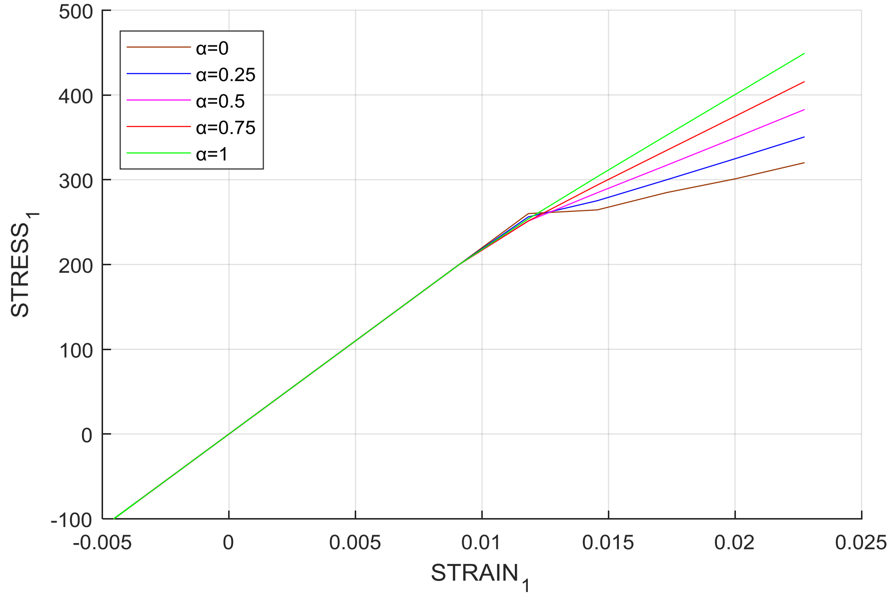

Fig-5c: Dependency of stress-strain curve on the variation of total time

2-b) Effects of variation of α on the evolution of C_{11} component of the tangent and algorithmic constitutive operators

In this section we analyze how the C_{11} components vary with the variation of α. The following values were considered in this analysis-

$\Delta\sigma^1_{(1)}$ = 150 MPa ; $\Delta\sigma^2_{(1)}$ = 0
$\Delta\sigma^1_{(2)}$ = 150 MPa ; $\Delta\sigma^2_{(2)}$ = 0
$\Delta\sigma^1_{(3)}$ = 150 MPa ; $\Delta\sigma^2_{(3)}$ = 0
Young Modulus, E= 20000 MPa
Yield Stress, σy= 200 MPa
Poisson Ratio, ν= 0.8
Hardness, H=- 0.5

Figure 6a shows the variation of C_{11} tangent component with alpha for the time of simulation. We can see that it varies with different methods. The plot also highlights the presence of instabilities for α<0.5 which is to be expected as these methods are explicit and are conditionally stable leading to fluctuations in the result which is not the case for the implicit methods. We also find that the values of C_{11} tangent component is gradually falling with time which can be attributed to the fact that C_{tan}=(1-d)C. So with the increase in damage, which increases with time, the value of Tangent operator falls.

Similarly, the C_{11} component of algorithmic operator also follows a gradual fall in its values with the increase in time just as the case with Tangent operator. This can be clearly seen from figure 6b. We also find huge oscillations in the result which is to be expected for α<0.5 as the methods are explicit and conditionally stable. The solutions are however stable for methods with α>0.5.

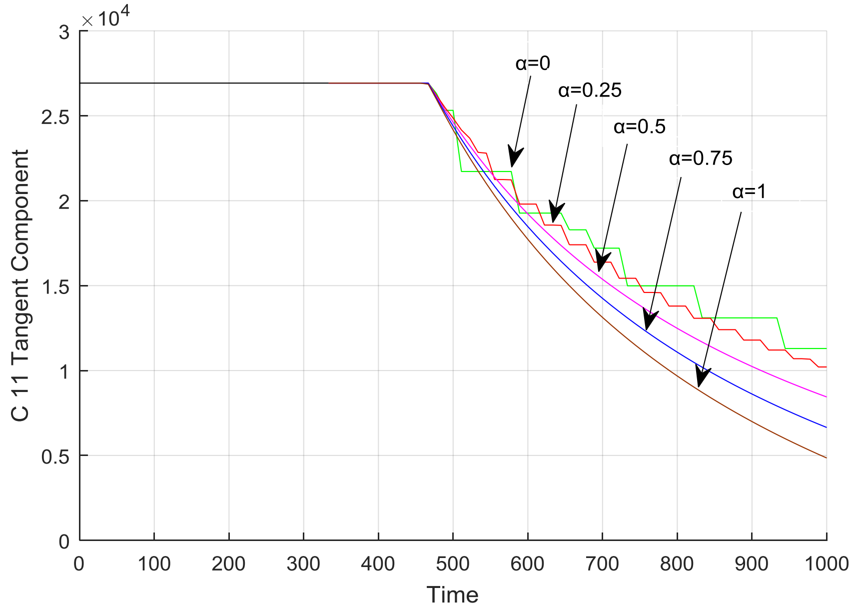

Fig-6a: Variation of C11 tangent Component with α

One interesting aspect to note is that both the curves for tangent operator and algorithmic operator coincide for α=0 which is in accordance with the theory. This can also be viewed as a verification for the correctness of the code implemented.

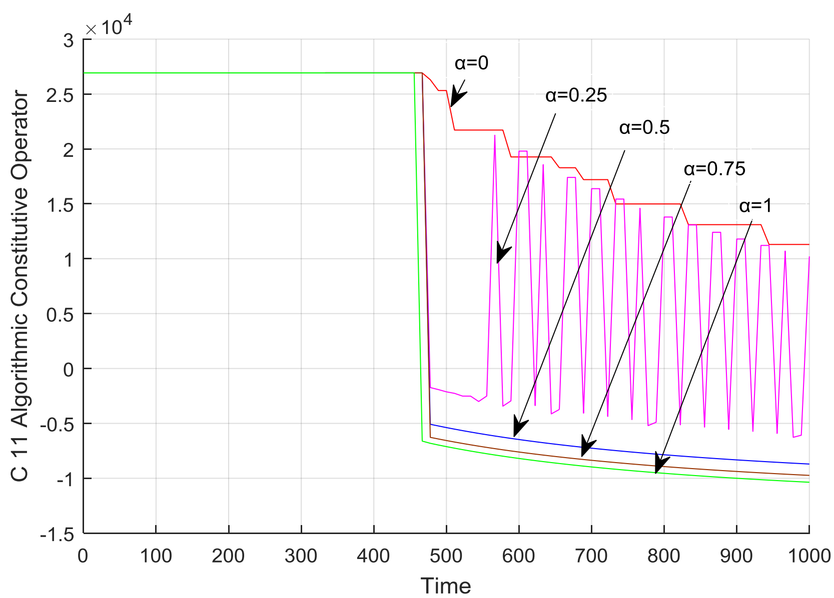

Fig-6b: Variation of C11 Algorithmic Constitutive Operator with α

Part III- Appendix

APPENDIX I: Changes in codes for tension only damage model(dibujar_criterio_dano1.m)

```
elseif MDtype==2 %Tension only damage model

    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
         %Defining the macaulay variables for Sigma(+)
    m1_plus=m1(i)*(m1(i)>0);
    m2_plus=m2(i)*(m2(i)>0);

    %Defining Tau(Epsalon) for Tension only model
    Tau_eps_plus= sqrt([m1_plus m2_plus 0 nu*(m1_plus+m2_plus)]*ce_inv*[m1(i) m2(i) 0 ...
            nu*(m1(i)+m2(i))]');
        radio(i)= q/Tau_eps_plus;
        % Sigma(1)= r(thera)*cos(theta)
        % Sigma(2)= r(theta)*sin(theta)

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

    end
    hplot =plot(s1,s2,tipo linea);
```

APPENDIX II: Changes in codes for Non-symmetric Tension-Compression damage model(dibujar_criterio_dano1.m)

```
elseif MDtype==3 %Non-symmetric Tension-Compression damage model
     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
         %Defining the macaulay variables for Sigma(+)
    m1_plus=m1(i)*(m1(i)>0);
    m2_plus=m2(i)*(m2(i)>0);

    %Defining theta for Non-symmetric tension-compression model
    A= m1_plus+m2_plus;
    B=abs(m1(i))+abs(m2(i));
    thetaa= A/B;
    %Defining Tau(sigma) for Tension only model
    Tau_sigma= sqrt([m1(i) m2(i) 0 nu*(m1(i)+m2(i))]*ce_inv*[m1(i) m2(i) 0 ...
            nu*(m1(i)+m2(i))]');
        radio(i)= q/(thetaa+(1-thetaa)/n)*Tau_sigma;
        % Sigma(1)= r(thera)*cos(theta)
        % Sigma(2)= r(theta)*sin(theta)

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

APPENDIX III: Changes in codes for Tension only and Non-Symmetric Tension-Compression damage models (Modelos_de_dano1.m)

```

elseif MDtype==2  %Tension only model

    %Defining sigma(plus) and sigma required to find out r(trial)
    S1_plus=zeros(1,2);
    S1 = ce*eps_n1';
 %Macaulay functions for sigma(plus) matrix
    S1_plus(1) = S1(1)*(S1(1)>0);
    S1_plus(2) = S1(2)*(S1(2)>0);

 rtrial= sqrt(S1_plus*eps_n1');


elseif MDtype==3 %Non-symmetric tension-compression model

    %Introducing theta for non-symmetric tension-compression model
    A=0;
    B=0;

    %For theta formulation
    S1_plus=zeros(1,2);
    S1 = ce*eps_n1';
    for i=1:length(S1_plus)
        S1_plus(i) = S1(i)*(S1(i)>0);%macaulay function for sigma(plus)
        A = A + S1_plus(i);%Numerator for theta
        B = B + abs(S1(i));%Denominator for theta
    end
    thetaa = A/B;
    rtrial= (thetaa+(1-thetaa)/n)*sqrt(eps_n1*ce*eps_n1');
end
```

APPENDIX IV: Changes in codes for including hardening and softening for viscous models(rmap_dano1.m)

```

    %Checking whether the model is viscous or not
    if viscpr == 0 %variable defined in damage_main
    if(rtrial > r_n)
        fload=1;
        delta_r=rtrial-r_n;
        r_n1= rtrial  ;
        if hard_type == 0  %Linear Hardening
            q_n1= q_n+ H*delta_r;
        elseif hard_type == 1 %Exponential Hardening
            %Exponential Softening
            if H<0
                %q(infinity)
                q_in=r0+(r0-q0);
                H_n = H*((q_in-r0)/(r_n))*exp(H*(1-(r_n1/r_n)));
            %Exponential Hardening
            elseif  H>0 %same calculations as above
                q_in=r0+(r0-q0);
                H_n = H*((q_in-r0)/(r_n))*exp(H*(1-(r_n1/r_n)));
            end
                q_n1= q_n+ H_n*delta_r; %q(n+1) obtained from earlier nth step
        end
        if(q_n1<q0)
            q_n1=q0;
        end
    else
        %*   Elastic load/unload
        fload=0;
```

APPENDIX V: Changes in codes for including continuum isotropic visco-damage models for symmetric model(rmap_dano1.m)

```
    %For viscous regime
    else
        if (rtrialn_alpha > r_n)%Outside elastic limits
            fload=1;
            delta_r=rtrialn_alpha-r_n;
            %Defining the r for (n+1)th step
            r_n1 = (eta - delta_t*(1-ALPHA))/(eta + ALPHA*delta_t)*r_n + (delta_t/(eta + ALPHA*delta_t))*rtrialn_alpha;
            if hard_type == 0%Linear
                H_n = H;
                q_n1= q_n+ H_n*delta_r;
            else if hard_type==1 %Exponential
                q_inf = r0 + (r0-q0);
                if H > 0 %Hardening
                    H_n = H*((q_inf-r0)/r0)*exp(H*(1-rtrialn_alpha/r0));
                else     %Softening
                    H_n = H*((q_inf-r0)/r0)*1/(exp(H*(1-rtrialn_alpha/r0)));
                end
                %Defining q for (n+1)thstep
                q_n1 = q_n + H_n*delta_r;
            end
            if(q_n1<q0)
               q_n1=q0;
            end
        else
            %*  Elastic load/unload
            fload=0;
            r_n1= r_n;
            q_n1= q_n;
        end
    end
```

APPENDIX VI: Changes in codes to determine C_{11} components of Tangent and Algorithmic Constitutive Operators(rmap_dano1.m)

```matlab

%For viscous region, symmetric
if viscpr == 1
    if rtrialn_alpha > r_n

        %From the definition of Consistent algorithmic operator for loading
        Algorithmic = (1.d0-dano_n1)*ce+((ALPHA*delta_t)/(eta+ALPHA*delta_t))*...
            (1/rtrialn_alpha)*((H_n*r_n1-q_n1)/(r_n1^2))*((ce*eps_n1')'*(ce*eps_n1')); %Slide 14, lecture-15

        %Now that we have the algorithmic matrix, we identify the (1,1)
        %element
        C_alg11 = Algorithmic(1,1);

        %Computing the tangent operator
        Tangent=(1.d0-dano_n1)*ce; %(1-d)*C

        %(1,1) element for tangent matrix
        C_tan11 = Tangent(1,1);

    else rtrialn_alpha <= r_n

        %It lies in the elastic regime
        Algorithmic = (1.d0-dano_n1)*ce; %(1-d)*C- elastic/unloading

        C_alg11 = Algorithmic(1,1);
        %Tangent matrix
        Tangent=(1.d0-dano_n1)*ce; %Same as Algorithmic for this case
        C_tan11 = Tangent(1,1);
    end
end
```