Assignment 1
Computational Solid Mechanics

Arnab Samaddar Chaudhuri
MSc Computational Mechanics
April 8, 2016

Introduction:

In the present paper we discuss about the various stress surfaces and stress vs. strain plots generated by different tensile loading or compressive loading. We also discuss the effects of strain rates, viscosity and other parameters on the stress strain graph.

Part I (Rate Independent)

Here we implement in and study the supplied MATLAB code the integration algorithms (rate independent and plane strain case) for:
a. The continuum isotropic damage "non-symmetric tension-compression damage" model.
b. The "tension-only" damage model.

Inviscid Model (Tension only damage model)

Case 1:

$\Delta\sigma_1^{(1)} = \alpha \;\; ; \;\; \Delta\sigma_2^{(1)} = 0 \;\; (Unaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta \;\; ; \;\; \Delta\sigma_2^{(2)} = 0 \;\; (Unaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma \;\; ; \;\; \Delta\sigma_2^{(3)} = 0 \;\; (Unaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=250 and γ=400 and we consider yield stress as 100 N/m^2 with linear hardening modulus 0.1, getting the following plot:

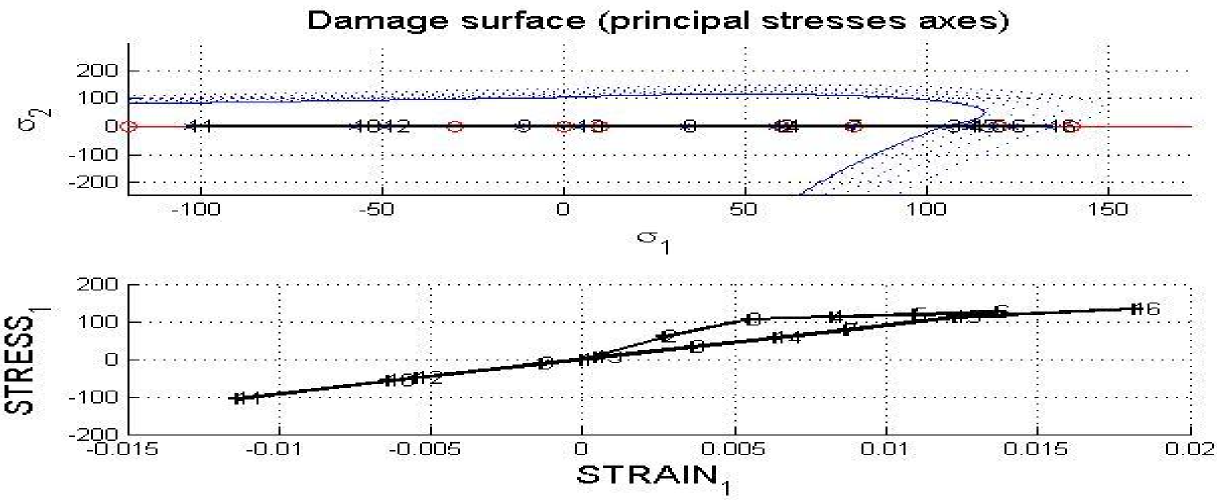


Figure 1.

From the above Figure (1), we notice that, at first tensile loading is applied and the material behaves elastically when within the yield stress of 100 N//m^2. But as soon as the loading exceeds the yield stress, the material starts deforming and it experiences hardening and the elastic domain increases. Next during compressive loading the load path remains within the yield stress and there is no deformation at this stage. And then during tensile loading the load path does not exceed the new yield stress and no deformation takes place.

Case 2:

$\Delta\sigma_1^{(1)} = \alpha \;\; ; \;\; \Delta\sigma_2^{(1)} = 0 \;\; (Unaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta \;\; ; \;\; \Delta\sigma_2^{(2)} = -\beta \;\; (Biaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma \;\; ; \;\; \Delta\sigma_2^{(3)} = \gamma \;\; (Biaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=250 and γ=400 and we consider yield stress as 100 N/m^2 with linear hardening modulus 0.1, getting the following plot:

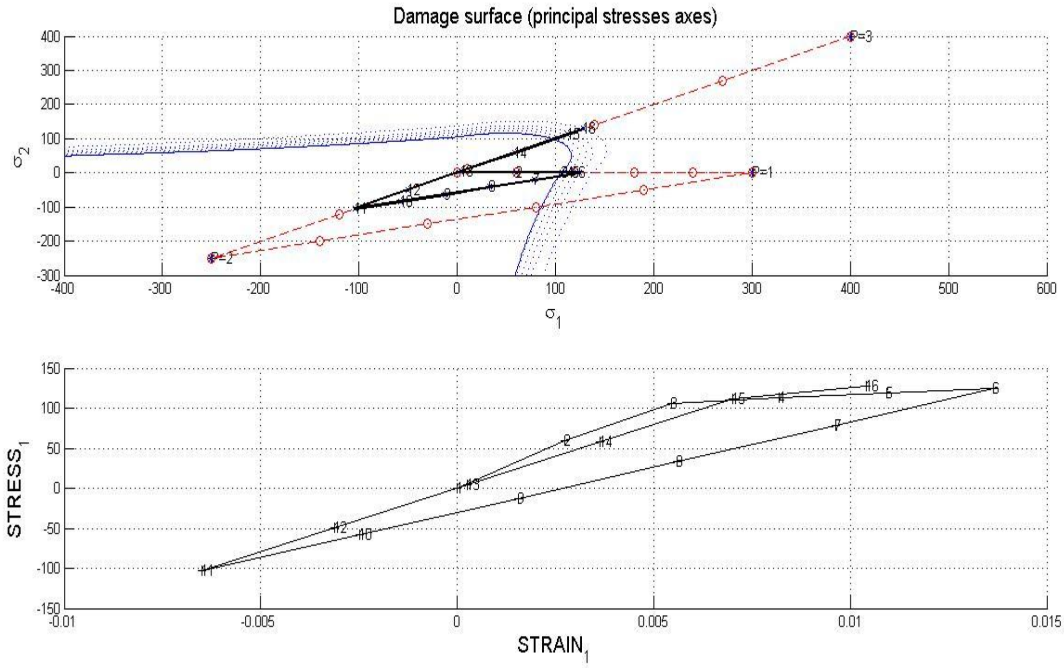


Figure 2.

From the above Figure (2), we notice that, at first tensile loading is applied and the material behaves elastically when within the yield stress of 100 N//m^2. But as soon as the loading exceeds the yield stress, the material starts deforming and it experiences hardening and the elastic domain increases. Next during biaxial compressive loading the load path remains within the yield stress and there is no deformation at this stage. And then during next biaxial tensile loading the load path exceeds the new yield stress and deformation takes place, thus increasing the damage surface.

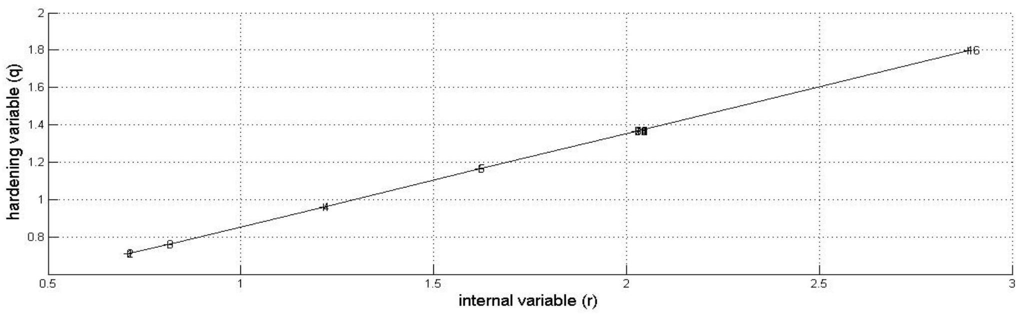


Figure 3.

From figure 3. we see the linear variation of hardening variable w.r.t. to internal variable.

Case 3:

$\Delta\sigma_1^{(1)} = \alpha$; $\Delta\sigma_2^{(1)} = \alpha$ $(Biaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta$; $\Delta\sigma_2^{(2)} = -\beta$ $(Biaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma$; $\Delta\sigma_2^{(3)} = \gamma$ $(Biaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=650 and γ=400 and we consider yield stress as 100 N/m^2 with linear hardening modulus 0.1, getting the following plot:

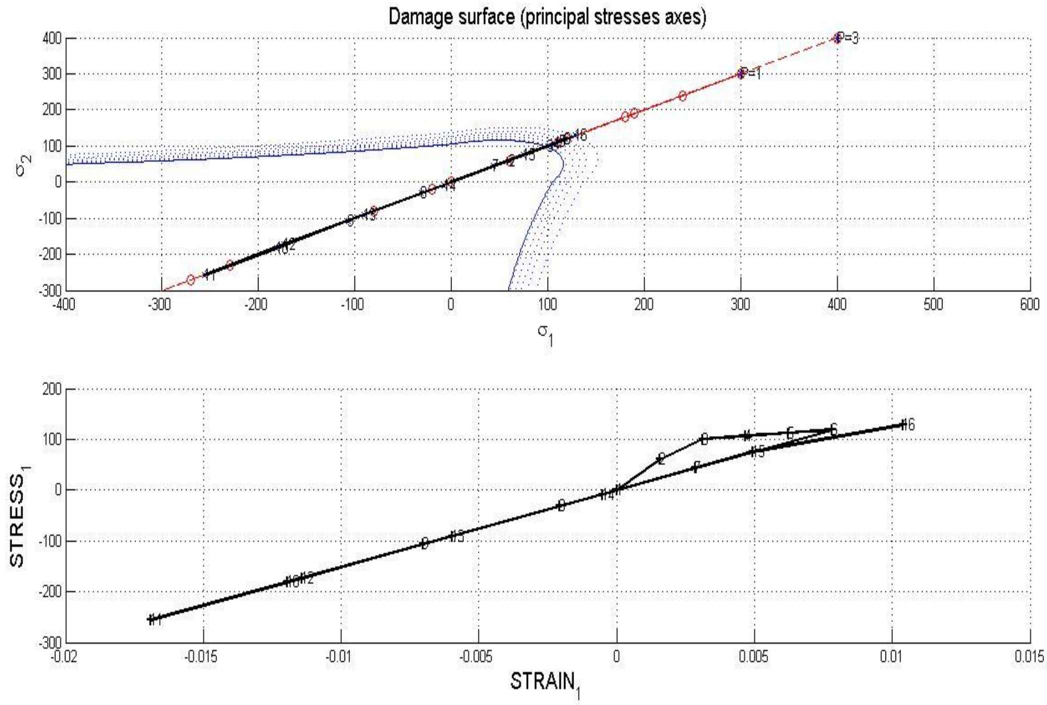


Figure 4.

From the above Figure (4), we notice that, at first tensile loading is applied and the material behaves elastically when within the yield stress of 100 N//m^2. But as soon as the loading exceeds the yield stress, the material starts deforming and it experiences hardening and the elastic domain increases. Next during biaxial compressive loading the load path exceeds the yield and but there is no deformation since it lies within the deformed surface. And then during next biaxial tensile loading the load path exceeds the new yield stress and deformation takes place, thus increasing the damage surface.

Inviscid Model (Non Symmetric tension compression damage model):

Case 1:

$\Delta\sigma_1^{(1)} = \alpha$; $\Delta\sigma_2^{(1)} = 0$ $(Unaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta$; $\Delta\sigma_2^{(2)} = 0$ $(Unaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma$; $\Delta\sigma_2^{(3)} = 0$ $(Unaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=250 and γ=400 and we consider yield stress as 100 N/m^2 with linear hardening modulus 0.1, getting the following plot:

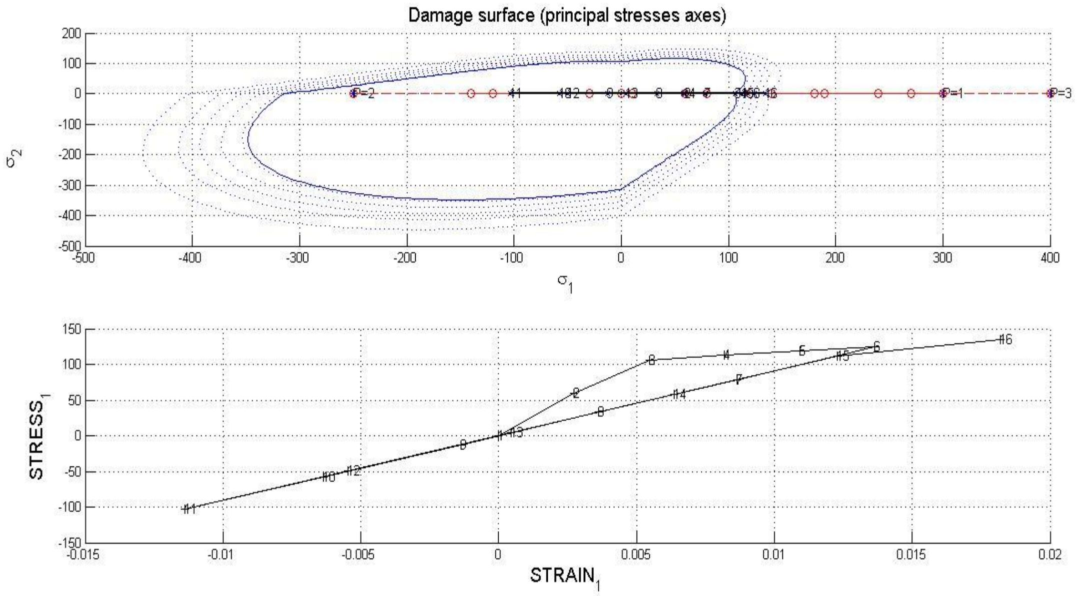


Figure 5.

From the above Figure (5), we notice that, at first tensile loading is applied and the material behaves elastically when within the yield stress of 100 N/m^2. But as soon as the loading exceeds the yield stress, the material starts deforming and it experiences hardening and the elastic domain increases. Next during uniaxial compressive loading the load path remains within the yield stress and there is no deformation at this stage. And then during uniaxial tensile loading the load path exceeds the new yield stress and deformation takes place, thus increasing the damage surface.

Case 2:

$\Delta\sigma_1^{(1)} = \alpha \quad ; \quad \Delta\sigma_2^{(1)} = 0 \;\; (Unaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta \quad ; \quad \Delta\sigma_2^{(2)} = -\beta \;\; (Biaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma \quad ; \quad \Delta\sigma_2^{(3)} = \gamma \;\; (Biaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=250 and γ=400 and we consider yield stress as 100 N/m^2 with <u>exponential hardening</u> modulus 0.1, getting the following plot:

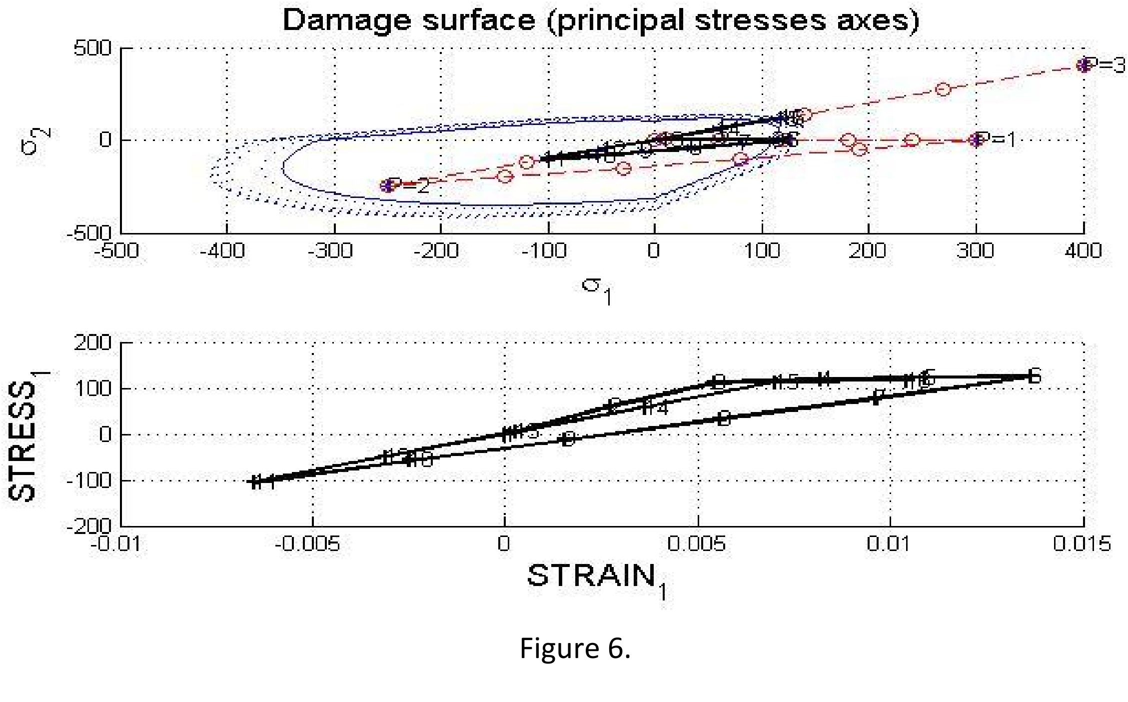


Figure 6.

From the above figure 6. we can see that the material initially behaves like in the previous case. But after the final biaxial tensile loading the load path does not exceed the new yield stress and no deformation takes place at the final step. Now the hardening variable vs internal variable is plotted below:

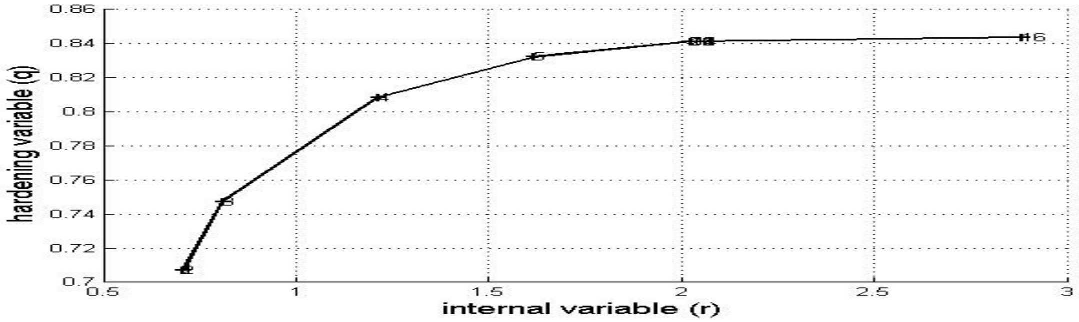


Figure 7

From the above plot we see that q’(r) is always greater than zero. The hardening variable exponentially approaches the q_infinity.

Case 3:

$\Delta\sigma_1^{(1)} = \alpha$; $\Delta\sigma_2^{(1)} = \alpha$ $(Biaxial\ Tensile\ Loading)$

$\Delta\sigma_1^{(2)} = -\beta$; $\Delta\sigma_2^{(2)} = -\beta$ $(Biaxial\ Tensile\ Unloading/Compressive\ Loading)$

$\Delta\sigma_1^{(3)} = \gamma$; $\Delta\sigma_2^{(3)} = \gamma$ $(Biaxial\ Compressive\ Unloading/Tensile\ Loading)$

Here we compute for α=300, β=650 and γ=400 and we consider yield stress as 100 N/m^2 with <u>linear hardening</u> modulus 0.5, getting the following plot:

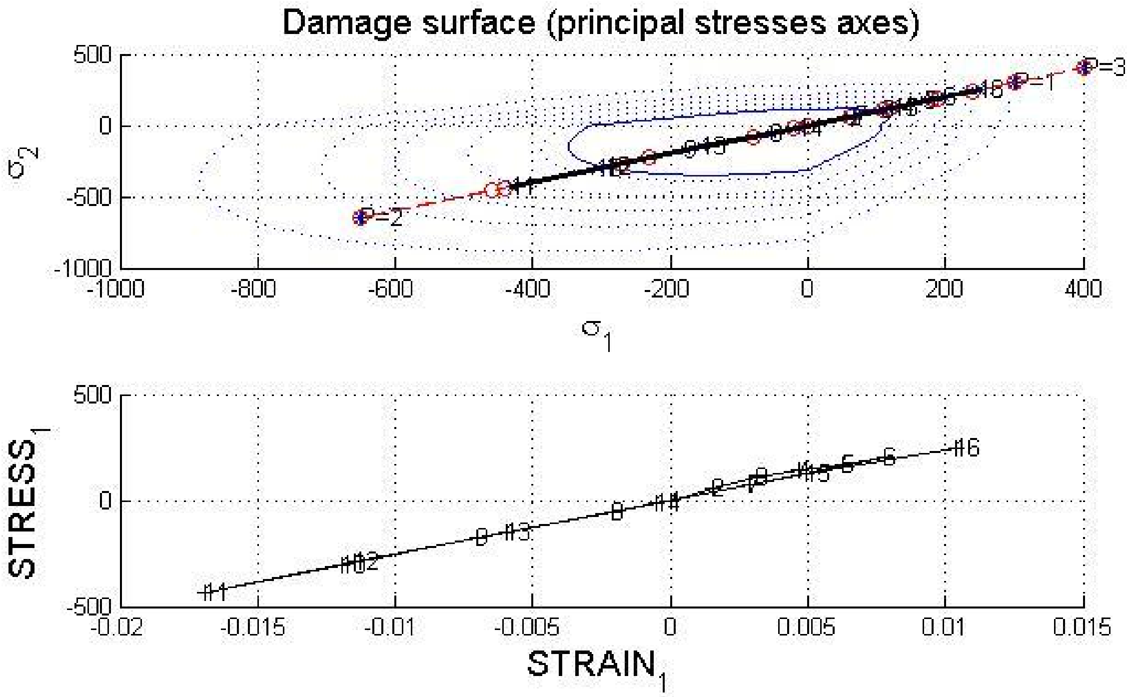


Figure 8

From the above Figure (8), we notice that, at first tensile loading is applied and the material behaves elastically when within the yield stress of 100 N//m^2. But as soon as the loading exceeds the yield stress, the material starts deforming and it experiences hardening and the elastic domain increases. Next during biaxial compressive loading the load path exceeds the yield stress and deformation takes place at this stage. And then during biaxial tensile loading the load path exceeds the new yield stress and deformation takes place, thus increasing the damage surface again. Now the internal variable vs time is plotted below:

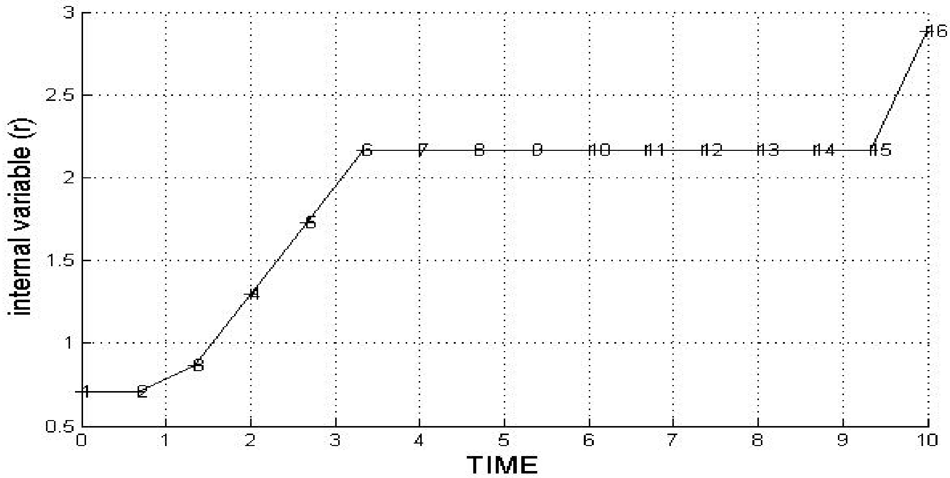


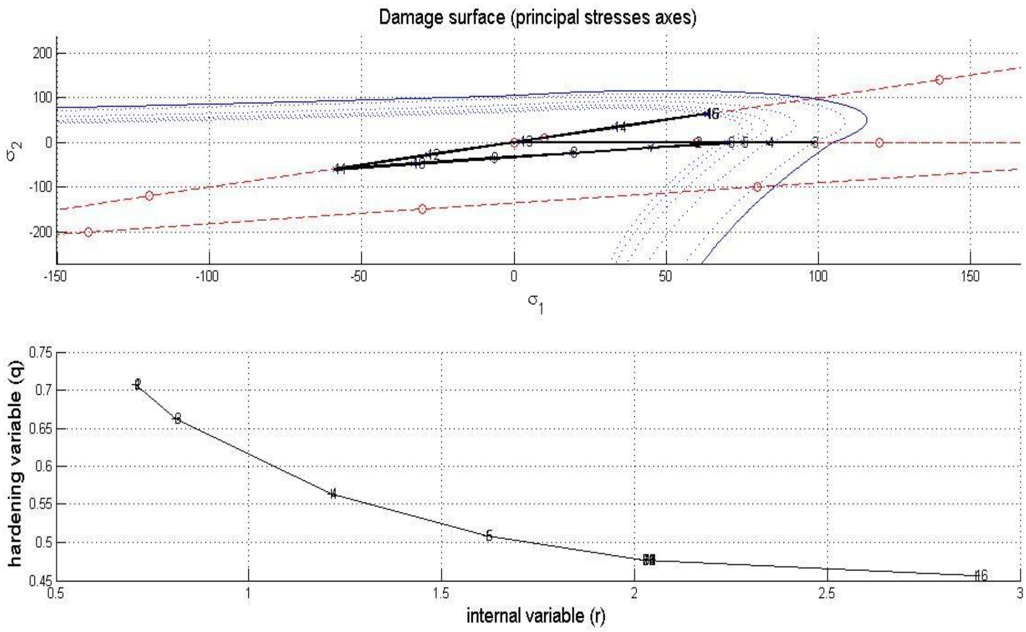
Figure 9.

From the figure 9. we see that $\dot{r}$ is always greater than equal to zero.

A brief discussion on exponential softening is given below for tension only damage model:

$$\Delta\sigma_1^{(1)} = \alpha \;\; ; \;\; \Delta\sigma_2^{(1)} = 0 \;\; (Unaxial\ Tensile\ Loading)$$
$$\Delta\sigma_1^{(2)} = -\beta \;\; ; \;\; \Delta\sigma_2^{(2)} = -\beta \;\; (Biaxial\ Tensile\ Unloading/Compressive\ Loading)$$
$$\Delta\sigma_1^{(3)} = \gamma \;\; ; \;\; \Delta\sigma_2^{(3)} = \gamma \;\; (Biaxial\ Compressive\ Unloading/Tensile\ Loading)$$

Here we compute for α=300, β=250 and γ=400 and we consider yield stress as 100 N/m^2 with exponential softening modulus 0.5, getting the following plot:



From the above plot we see that q'(r) is less than zero. The hardening variable exponentially approaches the q_infinity.

By studying all the above cases, the correctness of the implementation has been concluded.

Part II (Rate Dependent)

Here we study and implement in the supplied MATLAB code the integration algorithms (plane strain case) for the continuum isotropic visco-damage "symmetric tension compression" model."

Variable Viscosity Parameter η:

Here we compute for Young's Modulus = 20000, Poisson ratio = 0.3, Linear Hardness Parameter = 0.2, Yield Stress = 100, Sigma1 = [300,0], Sigma2 = [600,0], Sigma3 = [900,0], No of time increments = 5

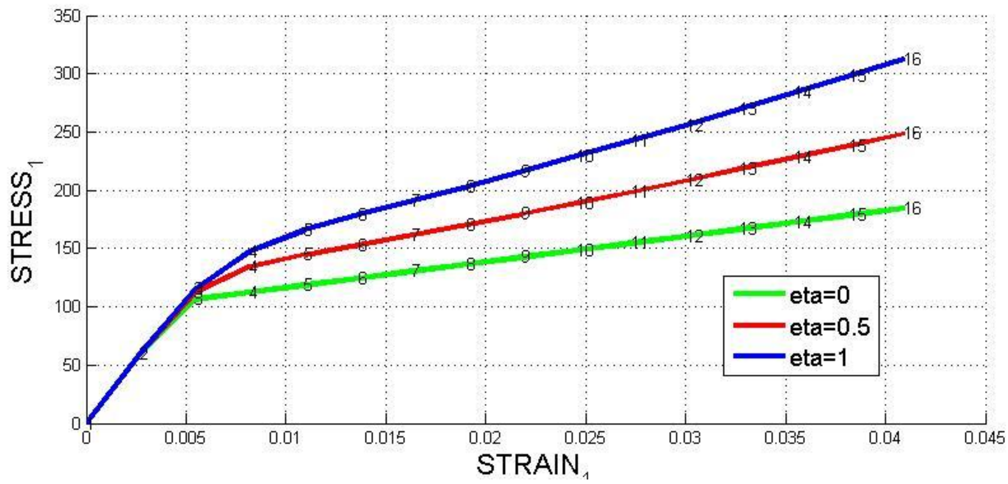


Figure 11

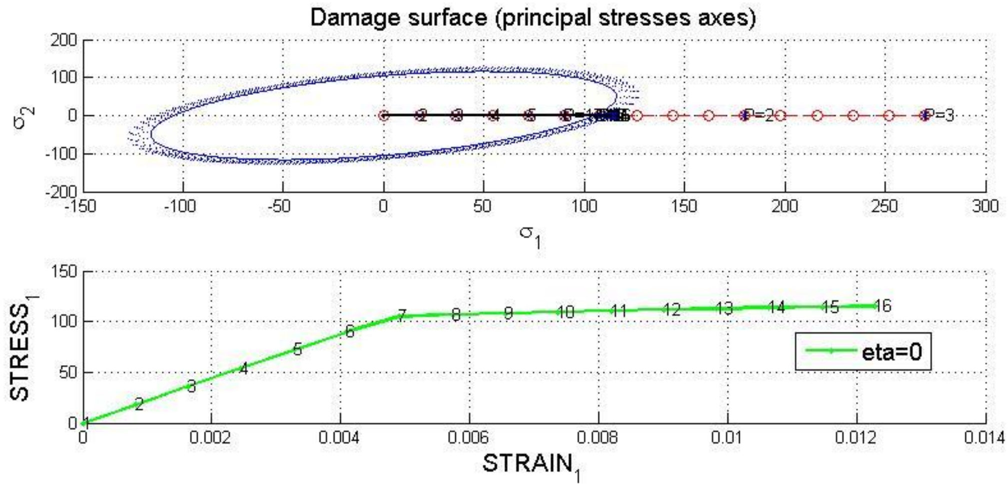


Figure 12

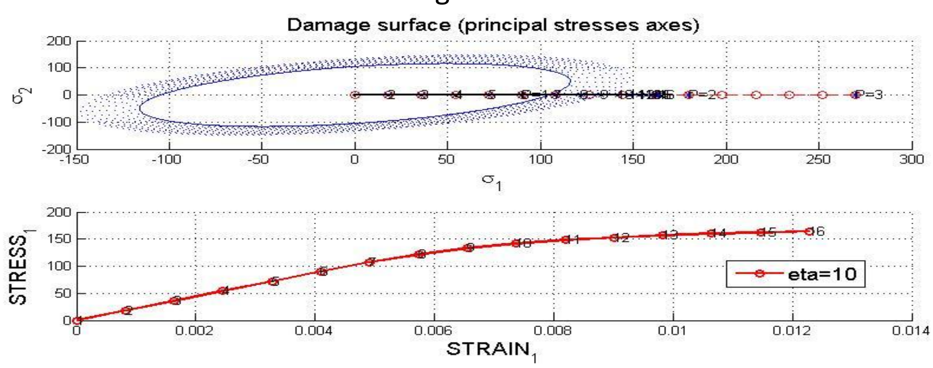


Figure 13

From Figure 12 & 13 we can deduce that as the viscosity increases, the stresses can increase at much faster than the damage surface. Conversely if the viscosity is zero, then the stresses cannot cross over the damage surface.

Variable Viscosity Parameter α:

Here we compute for Young's Modulus = 20000, Poisson ratio = 0.3, Linear Hardness Parameter = 0.2, Yield Stress = 100, Sigma1 = [300,0], Sigma2 = [600,0], Sigma3 = [900,0], No of time increments = 5, Viscosity= 1

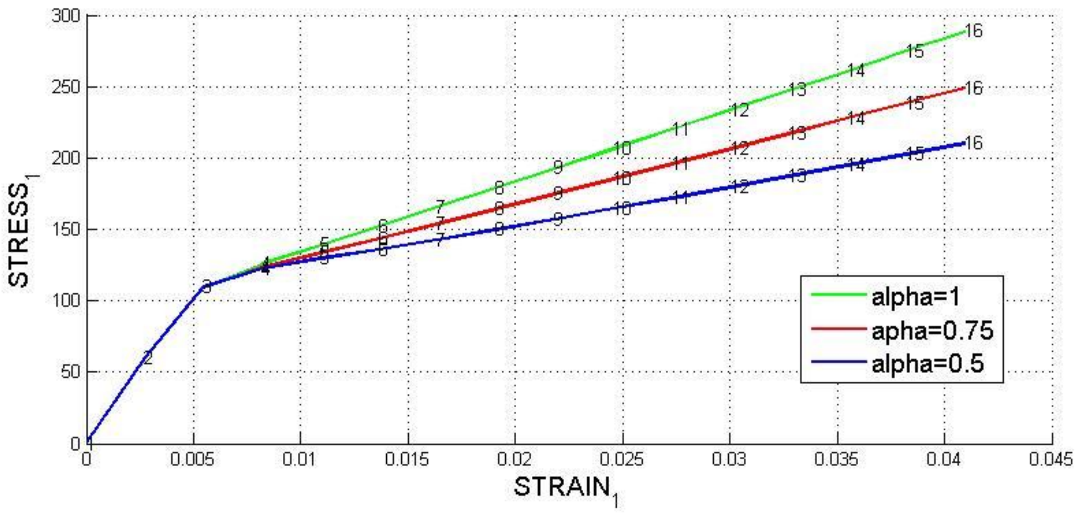


Figure 14

From the above figure we see that the solution is stable for $0.5 \leq \alpha \leq 1$. For values of α less than 0.5, the solution is unstable and there are a lot of oscillations. This is validated in theory too since when $\alpha = 0$, the solution is obtained by Forward Euler method (Explicit method thus stability issues) and for $\alpha = 1$, it is obtained by Backward Euler method (Implicit method).

Variable Strain Rate $\dot{\varepsilon}$:

Here we compute for Young's Modulus = 20000, Poisson ratio = 0.3, Linear Hardness Parameter = 0.2, Yield Stress = 100, Sigma1 = [300,0], Sigma2 = [600,0], Sigma3 = [900,0], Viscosity= 1, alpha=0

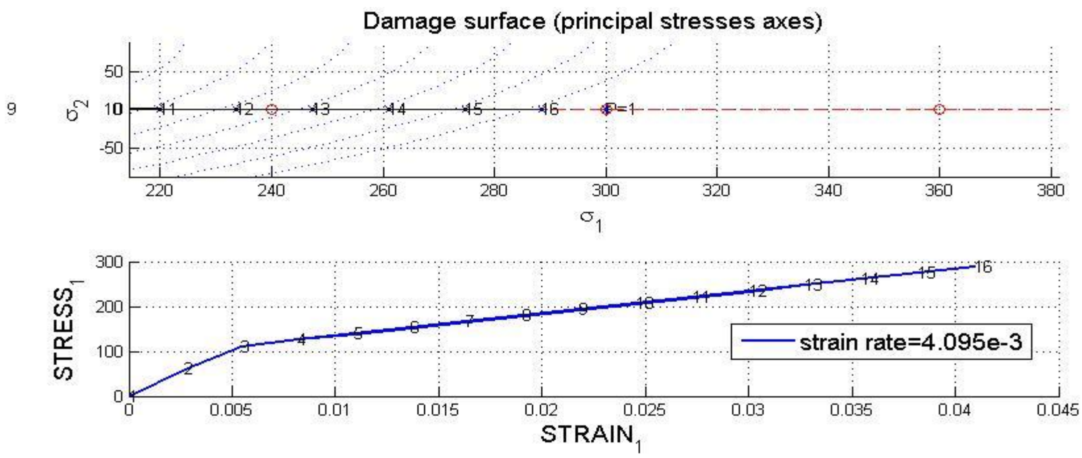


Figure 15

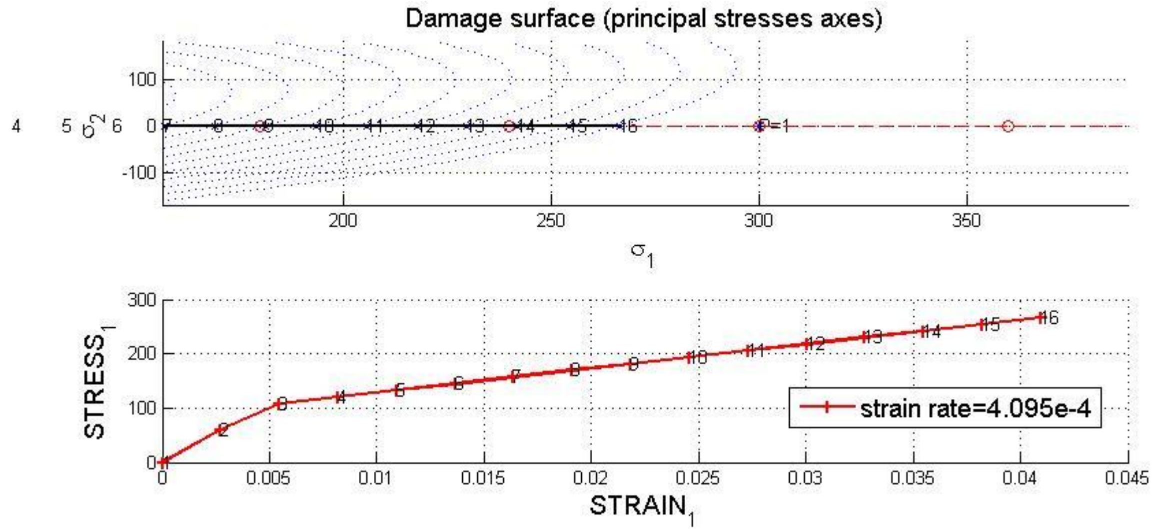


Figure 16

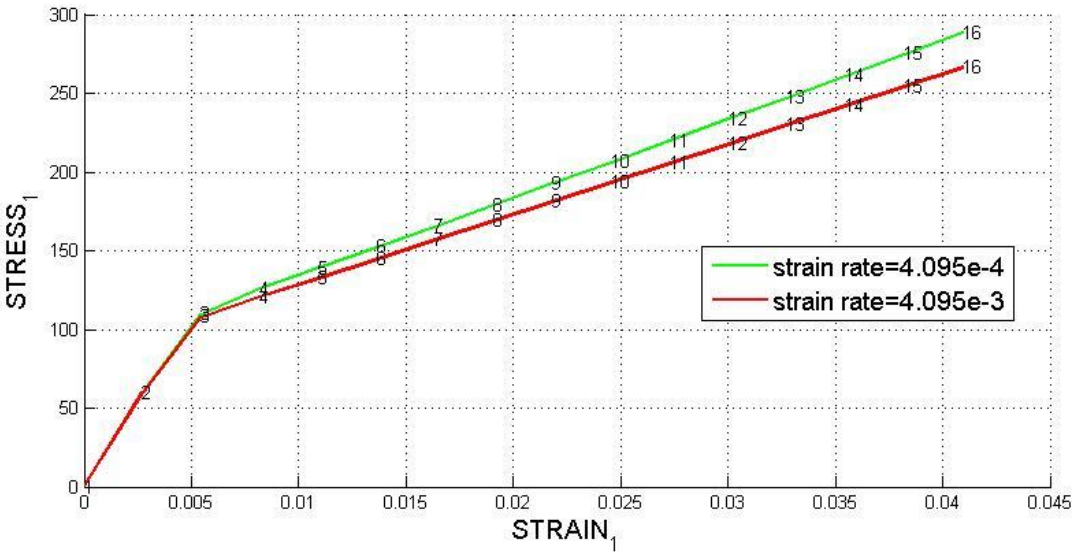


Figure 17

We change the Total Time of 100s and 1000s to vary the strain rate. At low strain rates the apparent stress does not exceed the yield stress and remains within the elastic domain, as apparent from Figure 15 & 16. From Figure 17 we see that stress vs. strain almost remains same and there is little effect of varying strain rate.

Conclusion:

By studying various models, different cases and taking into account of effects of various process parameters, we come to the conclusion that the MATLAB code has been properly implemented, giving satisfactory results which in turn can be validated by present theory of Computational Solid Mechanics theory.

ANNEXTURE

Modified Routines in Modelos_de_dano1 :

```
%*************************************************************************************
if (MDtype==1)      %* Symmetric
rtrial= sqrt(eps_n1*ce*eps_n1');

elseif (MDtype==2)  %* Only tension
stress=ce*eps_n1';
stress(stress<0)=0;   %stores only the positive part , rest are put zero
rtrial=sqrt(eps_n1*stress);  %Computation of strain norm

elseif (MDtype==3)  %*Non-symmetric
    stress=ce*eps_n1';
    stress_plus=stress;
    stress_plus(stress_plus<0)=0;  %stores only the positive part , rest are put zero
    num = sum((stress_plus));
    den = sum((abs(stress)));
    theta = num/den;
    rtrial = (theta + (1-theta)/n)* sqrt(eps_n1 * ce*eps_n1'); %Computation of strain norm

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

Modified Routines in dibujar_criterio_dano1 :

```
elseif MDtype==2            %tension only
    tetha=[0:0.01:2*pi];
    %*************************************************************************************
    %* RADIUS
    D=size(tetha);                        %*  Range
    m1=cos(tetha);                        %*
    n1=m1;
    n1(n1<0)=0;
    m2=sin(tetha);                        %*
    n2=m2;
    n2(n2<0)=0;
    Contador=D(1,2);                      %*

    radio = zeros(1,Contador) ;
    s1    = zeros(1,Contador) ;
    s2    = zeros(1,Contador) ;

    for i=1:Contador
        radio(i)= q/sqrt([n1(i) n2(i) 0 nu*(n1(i)+n2(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);
    axis([-400 600 -300 400])
```

```
    elseif MDtype==3  % Non symmetric
    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
      den = abs(m1(i))+abs(m2(i));
      n1 = m1(i);
      n2 = m2(i);
      if n1<0  n1 = 0;
      end
      if n2<0  n2 = 0;
      end
      num = n1+n2;
      radio(i)= q/(((num/den)+(1-(num/den))/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
```

<u>Modified Routines in</u> rmap_dano1:

```
if (viscrp == 0 ) % Inviscid
 if(rtrial > r_n)
    %*   Loading

    fload=1;
    delta_r=rtrial-r_n;
    r_n1= rtrial   ;
    if hard_type == 0
        %  Linear
        q_n1= q_n+ H*delta_r;
    else
              if H>0
               q_infi=r0*1.1;
          A = (H*r0)/(q_infi-r0);
          H_new= (A*(q_infi-r0)*exp(A*(1-rtrial/r0)))/r0;
          q_n1= q_n + H_new*delta_r;
          else
               q_infi=r0*0.5;
          A = (H*r0)/(q_infi-r0);
          H_new= (A*(q_infi-r0)*exp(A*(1-rtrial/r0)))/r0;
          q_n1= q_n + H_new*delta_r;
          end
    end
    if(q_n1<zero_q)
        q_n1=zero_q;
    end
```

```
else %FOR VISCOUS ('viscrp == 1')

    rtrial=(1-alpha)*r_n+alpha*rtrial;

    if(rtrial > r_n)
    %*   Loading

    fload=1;
    delta_r=rtrial-r_n;
    r_n1= (((eta-delta_t*(1-alpha))/(eta+alpha*delta_t))*r_n) + ....
        ((delta_t/(eta+alpha*delta_t))*rtrial);
    if hard_type == 0
        %  Linear
        q_n1= q_n+ H*delta_r;
    else
       %Exponential
         q_infi=r0*1.3;
         A = (H*r0)/(q_infi-r0);
         H_new= (A*(q_infi-r0)*exp(A*(1-rtrial/r0)))/r0;
         q_n1= q_n + H_new*delta_r;
    end

    if(q_n1<zero_q)
        q_n1=zero_q;
    end
```