Universitat Polytechnica de Catalunya
MSc Computational Mechanics
Spring 2018

Computational Solid Mechanics

Assignment 1

Due 23/03/2018

Alexander Keiser

Contents

1 ASSIGNMENT PART 1A

In the first part of this assignment, we were tasked with implementing in the supplied MATLAB code the integration algorithms (rate independent and plane strain case) for the continuum isotropic damage "non-symmetric tension-compression damage" model and the "tension-only' damage model. This was done successfully using the lecture slides and videos as reference. These modified codes with commentary can be found both in the Appendix I-IV.

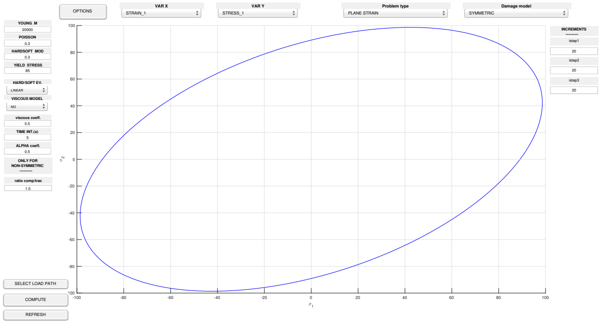

VAR X
STRAIN_1
VAR Y
STRESS_1
Problem type
PLANE STRAIN
Damage model
SYMMETRIC
YOUNG_M
20000
POISSON
0.3
HARDSOFT_MOD
0.3
YIELD_STRESS
85
HARD/SOFT EV.
LINEAR
VISCOUS MODEL
NO
viscous coeff.
0.5
TIME INT.(s)
5
ALPHA coeff.
0.5
ONLY FOR NON-SYMMETRIC
ratio comp/trac
1.5
INCREMENTS
istep1
20
istep2
20
istep3
20

Figure 1a: Symmetric Tension Compression Damage Model (STC) Already Implemented

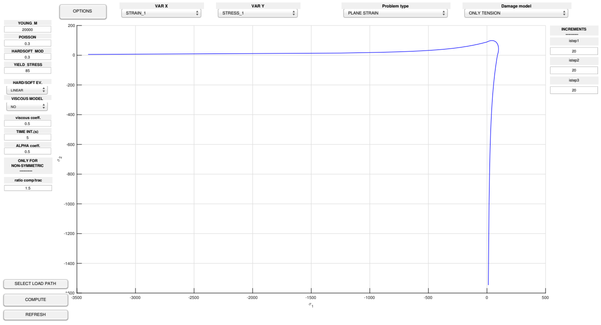

VAR X
STRAIN_1
VAR Y
STRESS_1
Problem type
PLANE STRAIN
Damage model
ONLY TENSION
YOUNG_M
20000
POISSON
0.3
HARDSOFT_MOD
0.3
YIELD_STRESS
85
HARD/SOFT EV.
LINEAR
VISCOUS MODEL
NO
viscous coeff.
0.5
TIME INT.(s)
5
ALPHA coeff.
0.5
ONLY FOR NON-SYMMETRIC
ratio comp/trac
1.5
INCREMENTS
istep1
20
istep2
20
istep3
20

Figure 1b: Implemented Tension-Only (TO) Damage Model

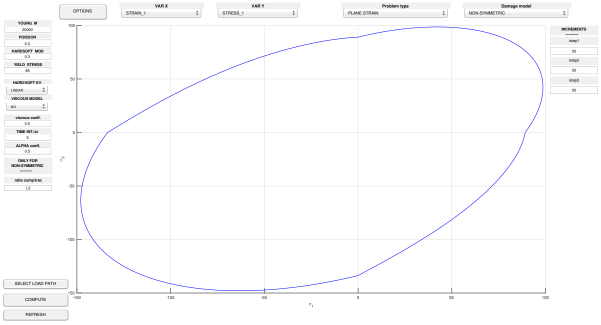

VAR X
STRAIN_1
VAR Y
STRESS_1
Problem type
PLANE STRAIN
Damage model
NON-SYMMETRIC
YOUNG_M
20000
POISSON
0.3
HARDSOFT_MOD
0.3
YIELD_STRESS
85
HARD/SOFT EV.
LINEAR
VISCOUS MODEL
NO
viscous coeff.
0.5
TIME INT.(s)
5
ALPHA coeff.
0.5
ONLY FOR NON-SYMMETRIC
ratio comp/trac
1.5
INCREMENTS
istep1
20
istep2
20
istep3
20

Figure 1c: Implemented Non-Symmetric Tension-Compression (NSTC) Damage Model

The results obtained coincide with the general shapes and theory given in the literature and the lectures. The Tension-Only model behaves asymptotically along the negative x and y axis directions, and the Non-Symmetric Tension-Compression model exhibits the desired behavior, the compression region of the graph in quadrant 3 is larger than the tension region in quadrant 1. We can assume that the implementations are correct and will move on to the next part of the assignment.

2 ASSIGNMENT PART 1B

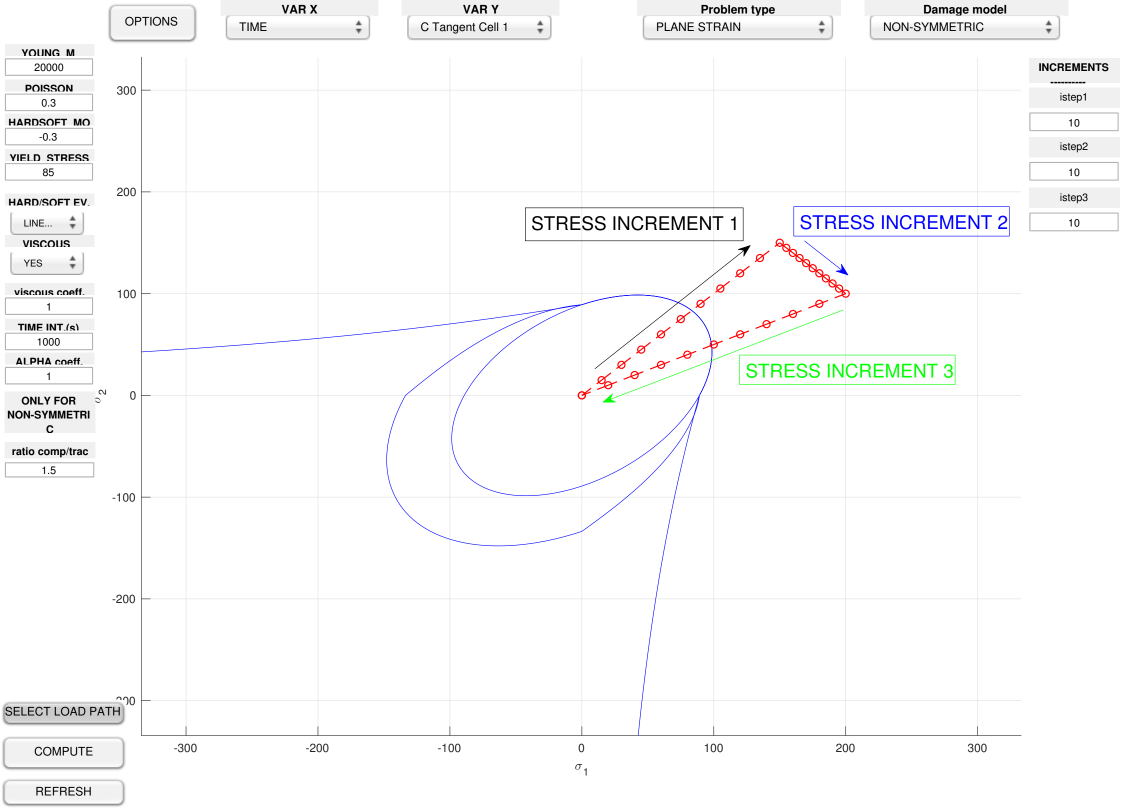
In the second part of the assignment, we were tasked with implementing the linear and exponential hardening/softening (H<0 and H>0) cases for all three of these models. This was also done successfully using the lecture slides and videos as reference. Similarly, these codes can be found with commentary in Appendix V. Additionally, in the following figure below we will find a sample loading case with with all three models overlapping.

Figure 1d: Sample Loading Case

Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 150 \ \Delta\bar{\sigma}_2^{(1)} = 150$

Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 50 \ \Delta\bar{\sigma}_2^{(2)} = -50$

Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = -200 \ \Delta\bar{\sigma}_2^{(3)} = -100$



VAR X: TIME

VAR Y: C Tangent Cell 1

Problem type: PLANE STRAIN

Damage model: NON-SYMMETRIC

YOUNG_M: 20000

POISSON: 0.3

HARDSOFT_MO: -0.3

YIELD_STRESS: 85

HARD/SOFT EV.: LINE...

VISCOUS: YES

viscous coeff.: 1

TIME INT.(s): 1000

ALPHA coeff.: 1

ONLY FOR NON-SYMMETRIC

ratio comp/trac: 1.5

INCREMENTS

istep1: 10

istep2: 10

istep3: 10

In the above figure, we have implemented a sample loading path on the overlapping of all three models to give the operations of the program context. The program imposes 3 consecutive stress increments and allows the user to plot a number of relevant results. Here we have implemented first a bi-axial tensile-loading stress of 150 in the positive direction of both principle stresses. This can be seen on the graph by the arrow labeled "STRESS INCREMENT 1". Next we implemented a "STRESS INCREMENT 2" of positive 50 in the first principle direction and -50 in the second principle direction. Finally, a third "STRESS INCREMENT 3" was imposed to bring the state of loading back to the starting point of zero in both directions. This figure can be used as reference to help visualize the loading cases for the subsequent loading paths described in this report.

3 ASSIGNMENT PART 1C

3.1 ASSIGNMENT PART 1C_CASE_1

Now that we have implemented the required codes and explained some functionality of the program, we can begin to assess the correctness of these implementations. We will begin by implementing the first prescribed case. This case has uni-axial tensile-loading as the first stress increment, uni-axial tensile-unloading/compressive-loading as the second increment, and uni-axial compressive-unloading/tensile-loading as the third increment. The values of the load paths, input parameters and relevant stress/strain curves resulting from the implementation can be seen in figure 2 below and on the next page.

Figure 2: STRESS I / STRAIN I curves of case 1 for Tension-Only
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 600$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = -700$ $\Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 900$ $\Delta\bar{\sigma}_2^{(3)} = 0$

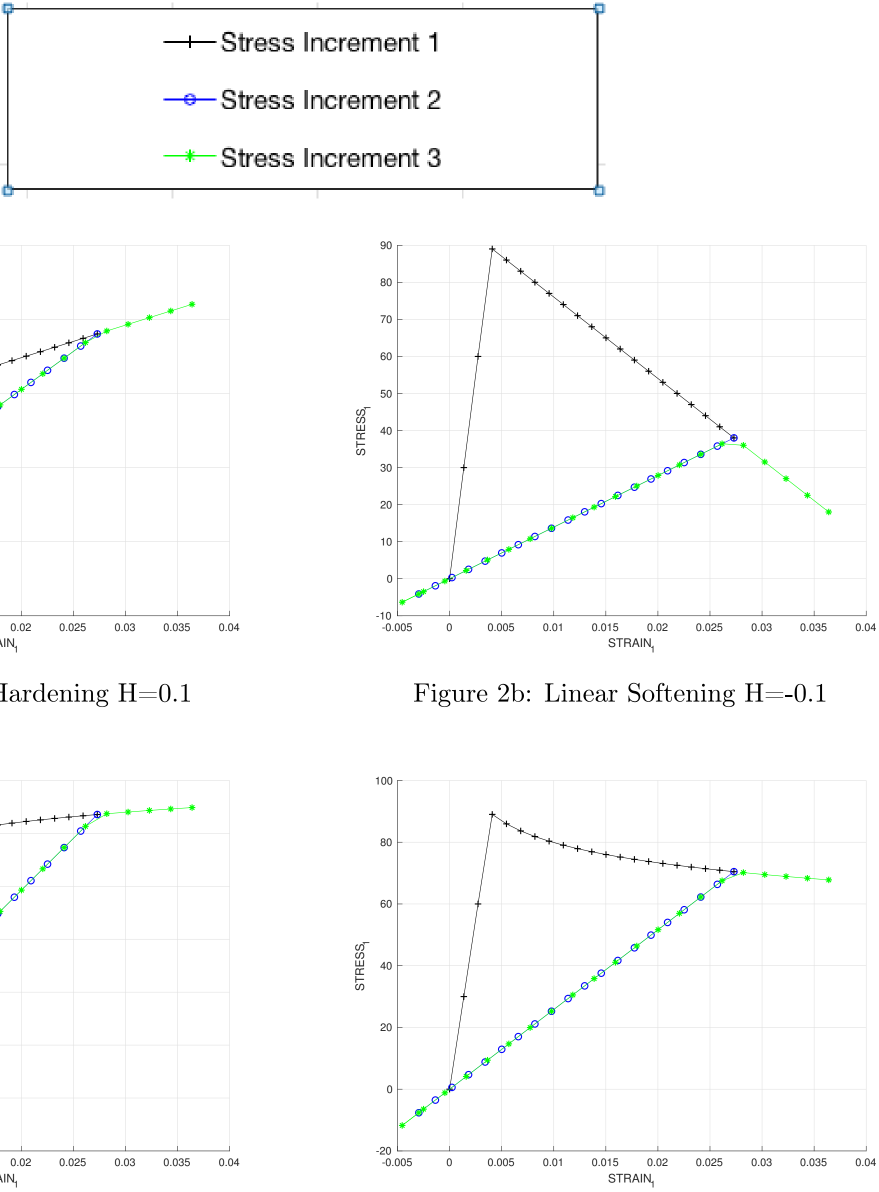


Figure 2a: Linear Hardening H=0.1

Figure 2b: Linear Softening H=-0.1

Figure 2c: Exponential Hardening H=0.1

Figure 2d: Exponential Softening H=-0.1

Above and on the previous page we can see the results of the implemented Linear Hardening, Linear Softening, Exponential Hardening, and Exponential Softening plots corresponding to the problem statement in Tension-Only case 1. All have been loaded with the same increments, these can be seen near the top of the figure. It is extremely important to note that all three of the implemented methods produced very similar results, so each of these graphs is representative of it's respective hardening/softening case for the Symmetric Tension Compression, Tension Only, and Non-Symmetric Tension Compression models. This is expected as the damage surfaces intersecting the loading paths are very similar within the context of the problem and will not give drastically different results.

We will now break down what is happening in the plots. Across the board, each plot behaves the same in the elastic loading region. The plots move linearly upwards to their yield points. Once the yield points are reached, the stress of the hardening cases continues to increase with increasing strain while the stress of the softening cases begin to decrease with increasing strain. The Linear Hardening stress increases linearly as expected and the Exponential Hardening stress increases in a curved fashion as one would expect from the theory. A similar trend is present for both cases of Linear and Exponential softening. Once the initial uni-axial stress increment is reached, all four plots begin uni-axial unloading and linear compression before resuming tensile loading. We can notice that the softened materials exhibit a lower yield stress upon reloading and the hardened materials exhibit a higher yield stress upon reloading. This is consistent with the theory and allows us to claim that the implemented models are behaving correctly for this case.

3.2 ASSIGNMENT PART 1C_CASE_2

We will now implement case two of the assignment. This case consists of uni-axial tensile loading, followed by bi-axial tensile-unloading/compressive-loading, finished with compressive-unloading/tensile-loading. The values of the load increments, the input parameters, and the relevant stress/strain curves resulting from the implementation can also be seen in figure 3 below, while some remarks and conclusions will be drawn on the next page.

Figure 3: STRESS I / STRAIN I curves of case 2 for Symmetric
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 50$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = -100$ $\Delta\bar{\sigma}_2^{(2)} = -100$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 325$ $\Delta\bar{\sigma}_2^{(3)} = 325$

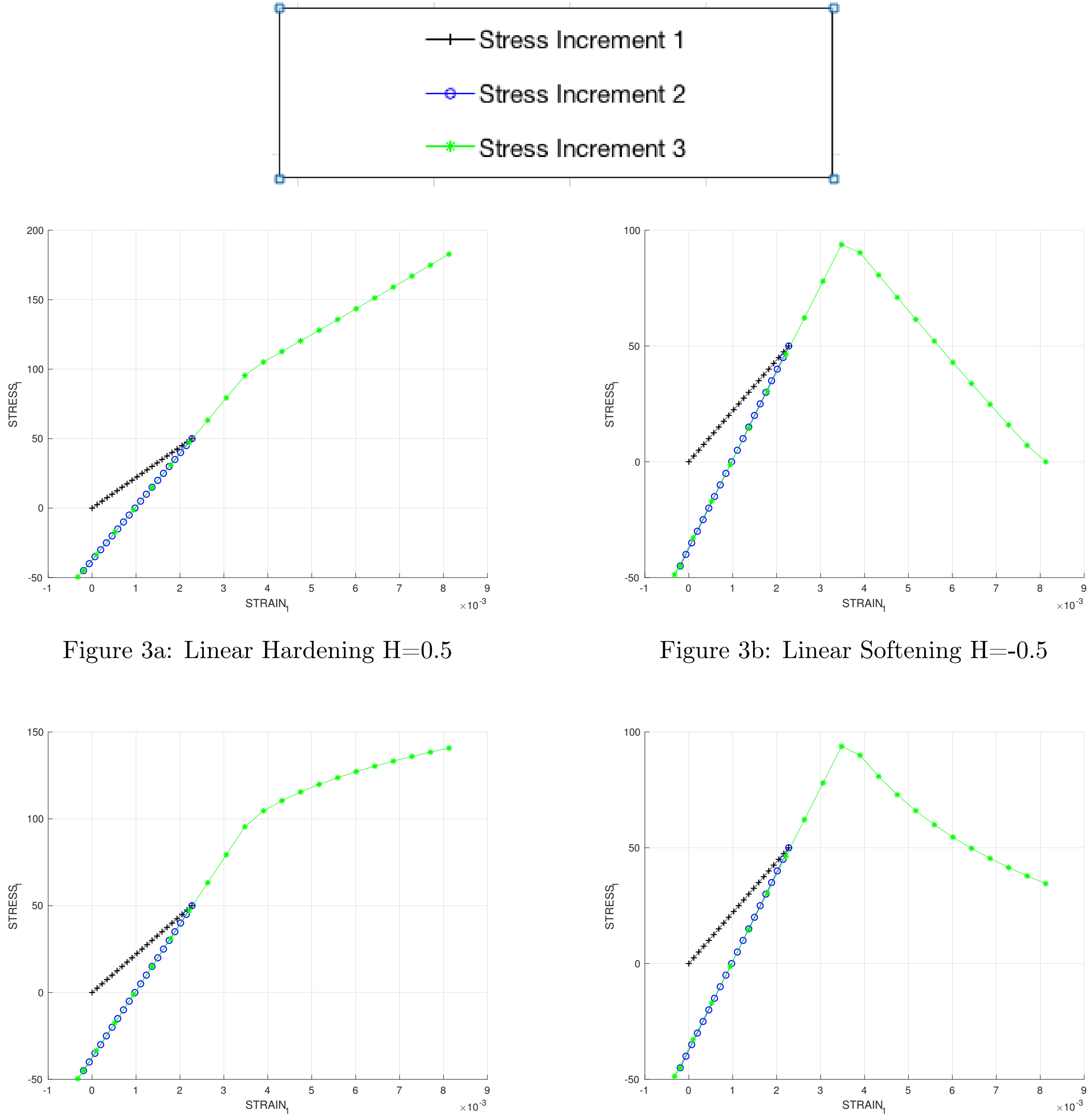


Figure 3a: Linear Hardening H=0.5

Figure 3b: Linear Softening H=-0.5

Figure 3c: Exponential Hardening H=0.5

Figure 3d: Exponential Softening H=-0.5

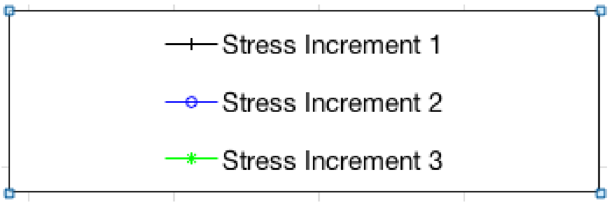
On the previous page we can see the results of the implemented Linear Hardening, Linear Softening, Exponential Hardening, and Exponential Softening plots corresponding to the problem statement in symmetric case 2. All have been loaded with the same increments, these can be seen near the top of the figure along with the input parameters. Once again, just as in the previous case, it is extremely important to note that all three of the implemented methods produced very similar results, so each of these graphs is representative of it's respective hardening/softening case for the Symmetric Tension-Compression, Tension-Only, and Non-Symmetric Tension-Compression models. This is expected for the same reason it was expected for the previous case, the damage surfaces that intersect the loading paths are very similar within the context of the problem and will not give drastically different results.

We will now break down what is happening in the plots. Across the board, each plot behaves the same in the elastic loading region. The plots begin by moving linearly upwards to their yield points. Before they can be reached, bi-axial tensile-unloading/compressive-loading takes place followed by bi-axial compressive-unloading/tensile-loading to take the graphs to their yield points and damage surfaces. Several comments can now be made. First, it can now be noted the effect of the magnitude of the hardening modulus on the slope of the respective hardening/softening regions. We can confirm from the theory that the higher the magnitude of this modulus, the steeper in slope the regions of hardening and softening will be. This is another good indicator that our models are behaving correctly. Second, a key difference in loading path from this case and the previous case is the application of bi-axial tensile-unloading/compressive-loading from within the elastic region after the uni-axial elastic tension but before reapplication of bi-axial compressive-unloading/tensile-loading. We can notice that the addition of a negative second principle stress value during the second phase of the prescribed increments changes the slope of the elastic region. This is to be expected and the third phase follows this new higher slope until reaching the plot's yield points at which the models undergo their respective forms of hardening/softening. From this point on the Linear Hardening graph increases linearly and the Exponential Hardening graph increases in a curved fashion. A similar trend is present for both cases of Linear and Exponential Softening. These plots are behaving exactly as expected and are consistent with the theory and allows us to further argue the claim that the implemented models are correct.

3.3 ASSIGNMENT PART 1C_CASE_3

We will now move on to the third and final case of part 1 of the assignment. This case consists of bi-axial tensile-loading, followed by bi-axial tensile-unloading/compressive-loading, finished with bi-axial compressive-unloading/tensile-loading. Once again, the load increments, along with the input parameters and relevant stress/strain curves resulting from the implementation can be seen in figure 4 on the below and on next page, with some drawn remarks and conclusions.

Figure 4: STRESS I / STRAIN I curves of case 3 for Non-Symmetric
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 150$ $\Delta\bar{\sigma}_2^{(1)} = 150$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = -200$ $\Delta\bar{\sigma}_2^{(2)} = -200$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 300$ $\Delta\bar{\sigma}_2^{(3)} = 300$



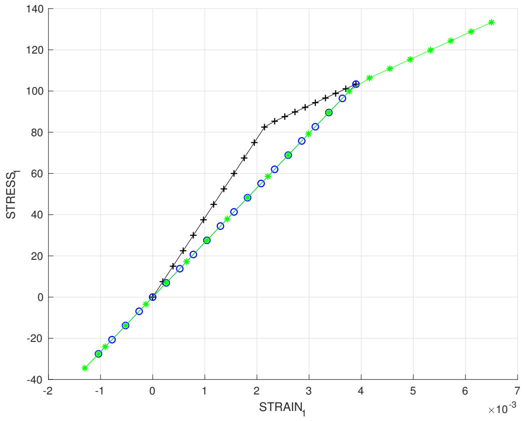


Figure 4a: Linear Hardening H=0.3

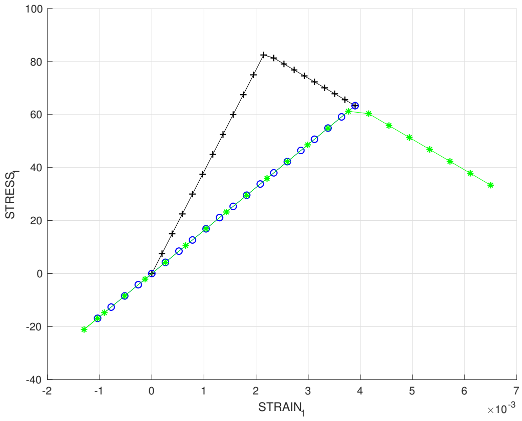


Figure 4b: Linear Softening H=-0.3

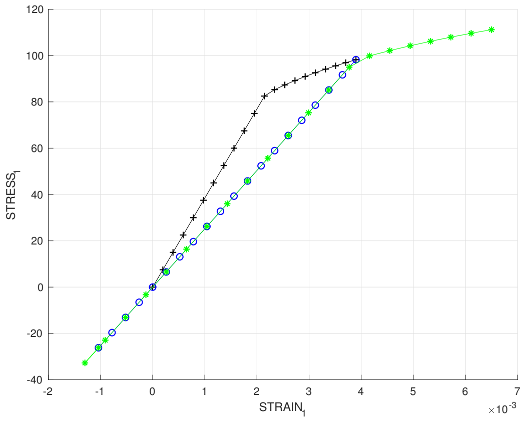


Figure 4c: Exponential Hardening H=0.3

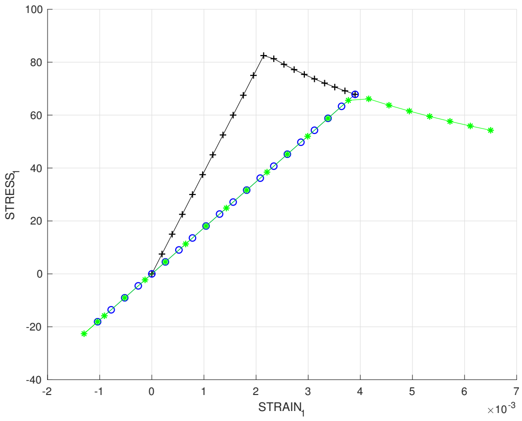


Figure 4d: Exponential Softening H=-0.3

Above we can see the implemented methods undergoing Linear Hardening, Linear Softening, Exponential Hardening, and Exponential Softening corresponding to the problem statement of Non-Symmetric case 3. All have been loaded with the same increments, these can be seen near the top of the figure. Once again, as is consistent with the other two previous cases, it is extremely important to note that all three of the implemented methods produced very similar results, so each of these graphs is representative of it's respective hardening/softening case for the Symmetric Tension-Compression, Tension-Only, and Non-Symmetric Tension-Compression models. This is expected for the same reasons as it was for the previous two cases. The damage surfaces intersecting the loading paths are very similar within the context of the problem and will not give drastically different results.

We will now break down what is happening in the plots. As is consistent with previous cases, each plot behaves the same in the elastic loading region. The plots move linearly upwards to their yield points. Once the yield points are reached, the stress of the hardening cases continues to increase with increasing strain while the stress of the softening cases begin to decrease with increasing strain. The Linear Hardening graph increases linearly and the Exponential Hardening graph increases in a curved fashion as seen in the theory. A similar trend is present for both cases of Linear and Exponential softening. Once the initial bi-axial stress increment is reached, all four plots begin bi-axial linear-unloading/compressive-loading before taking on bi-axial compressive-unloading/tensile-loading back up to their new hardened/softened yield stresses. We can notice that the softened materials exhibit a lower yield stress upon reloading and the hardened materials exhibit a higher yield stress upon reloading. This is consistent with the theory and results of the previous uni-axial case of the same nature, and further strengthens our claim that the implemented models are correct.

4 ASSIGNMENT PART 2D

For this part of the assignment we will be implementing in the supplied MATLAB code the integration algorithm (plane strain case) for the continuum isotropic visco-damage Symmetric Tension Compression model. To do this, I first added the eps.n, viscpr, and delta_t variables as input to the rmap_dano1.m after defining eps_n within the loop in damage_main.m and imported the eta and ALPHA variables using Eprop(7) and Eprop(8) respectivley. Next we define our rtrial_Nminus1 and rtrial_NminusALPHA variables in the damage surface before implementation in case of viscid parameters. This code implementation can be found in APPENDIX VI, and some sample results of this implementation comparing viscous effects vs. inviscid under identical conditions can be found below and on the next page.

Figure 5: STRESS I / STRAIN I curves of inviscid and viscous plots
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 150 \; \Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = -100 \; \Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 200 \; \Delta\bar{\sigma}_2^{(3)} = 0$

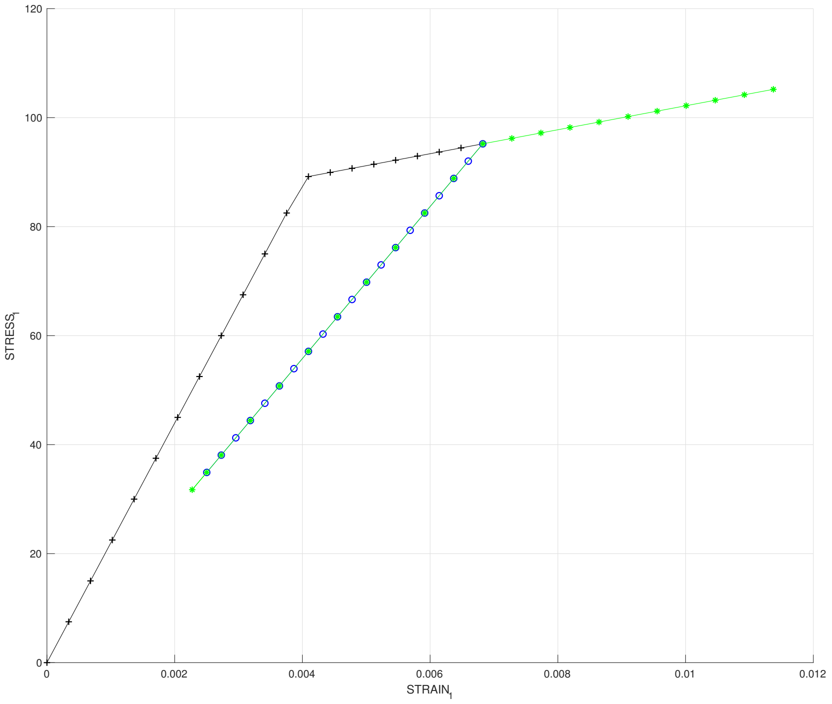


Figure 5a: STRESS I / STRAIN I curve of inviscid Linear Hardening H=0.1

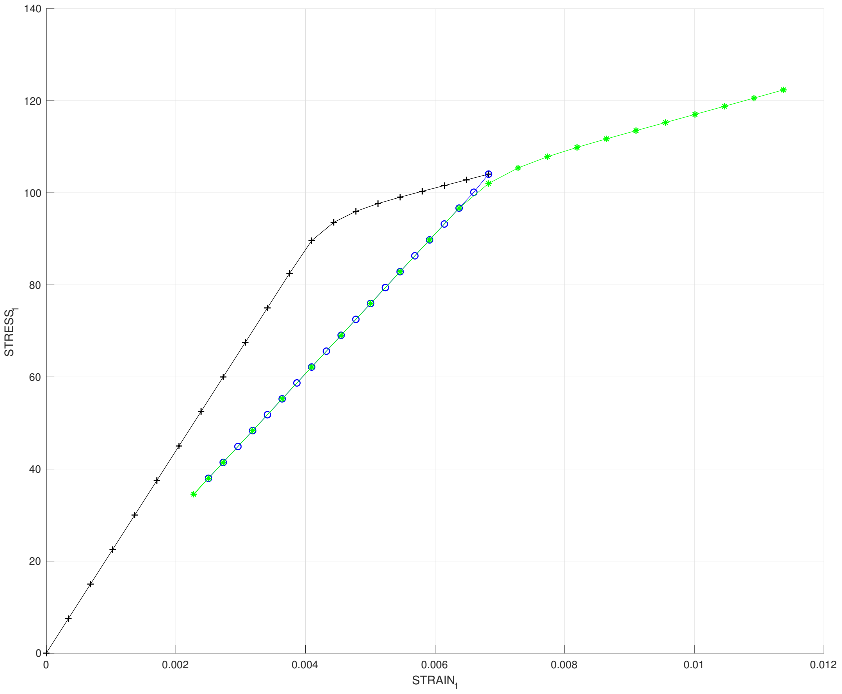


Figure 5b: STRESS I / STRAIN I curve of viscous Linear Hardening H=0.1 $\eta = 0.1$

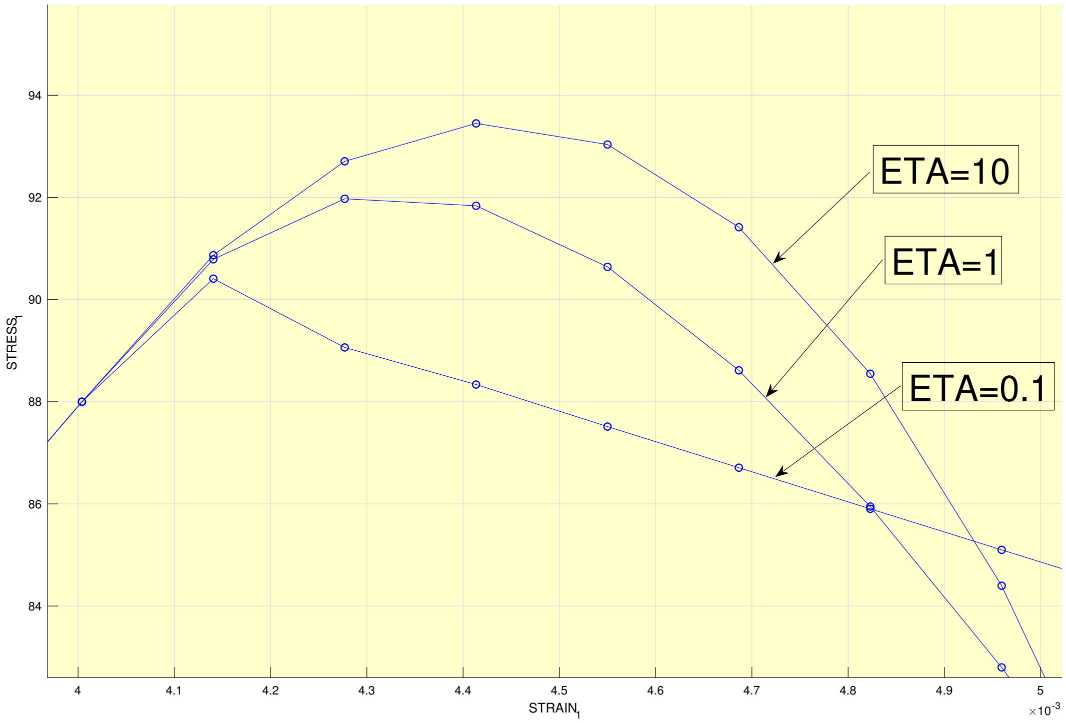
In the above plots, we can see a clear difference between the inviscid linear hardening plot in figure 5a, and the viscous linear hardening plot in figure 5b. We can notice on the viscous plot that at both the initial yield point and hardened yield point, the plot exhibits a smoother and higher stressed transition to the linear hardening region. This result can be explained by the addition of the viscous parameter seen in the implemented codes in Appendix VI. These results confirm what is expected from the theory and literature.

5 ASSIGNMENT PART 2E

5.1 VARIATION OF STRESS WITH VISCOUS PARAMETER

Now that we have implemented the visco-damage Symmetric Tension Compression model, we can assess the correctness of the implementation. This will be done by varying three different variables individually while holding everything else constant. The first variable that we will vary is the viscosity parameter eta (η). Below we can see some zoomed in plots of varying η parameters.

Figure 6: STRESS I / STRAIN I peaks from variation of viscous parameter eta (η)
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 100$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 100$ $\Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 100$ $\Delta\bar{\sigma}_2^{(3)} = 0$

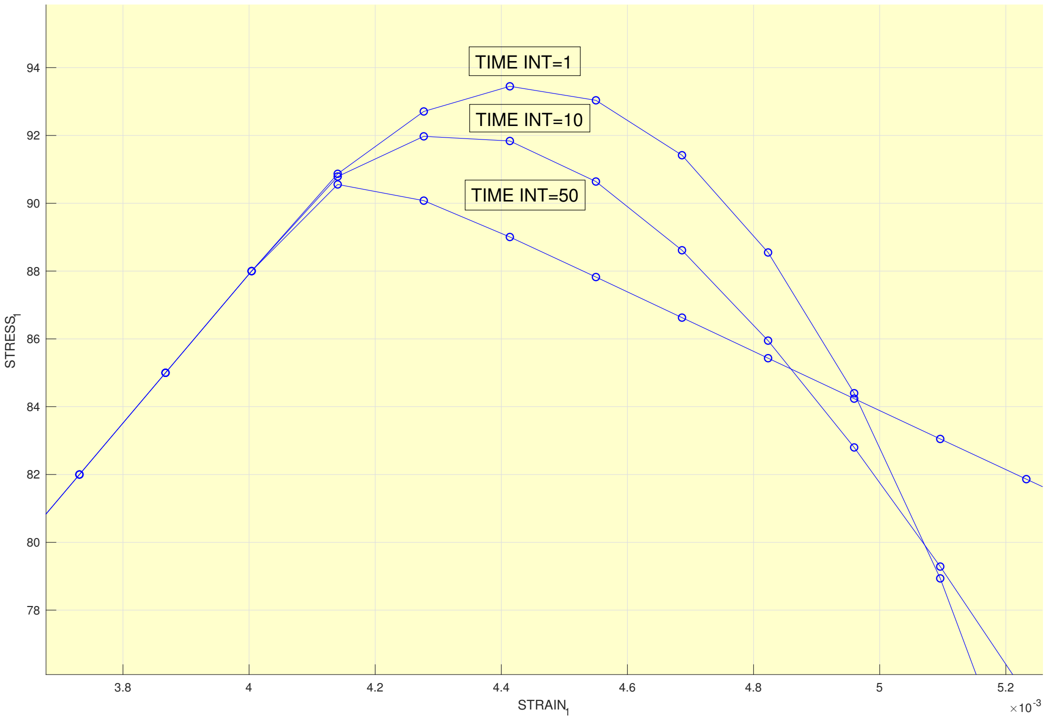


Above in Figure 6 we can see three STRESS I / STRAIN I peaks all of different magnitude corresponding to the variation of the viscous parameter eta (η). We can notice in the figure that as we increase the value of the viscous parameter, the amplitude of the peak increases accordingly. This corresponds to the theory stating that there is a proportional relationship between stress and the negative inverse of the viscous parameter (η) multiplied by some other terms. So in theory, as we increase the value of eta, we decrease the amount of stress subtracted by the corresponding combination of terms, therefore effectively increasing the stress. The figure confirms this predicted phenomenon and we can begin to formulate the argument that the implemented visco-damage Symmetric Tension Compression model is behaving correctly.

5.2 VARIATION OF STRESS WITH STRAIN RATE

Now that we have proven the variation of the viscous parameter with stress, we will now attempt to prove the variation of strain rate with the variation of stress to further our claim that the implemented visco-damage Symmetric Tension Compression model is behaving correctly.

Figure 7: STRESS I / STRAIN I peaks from variation of the variable TIME_INT
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 100 \ \Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 100 \ \Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 100 \ \Delta\bar{\sigma}_2^{(3)} = 0$

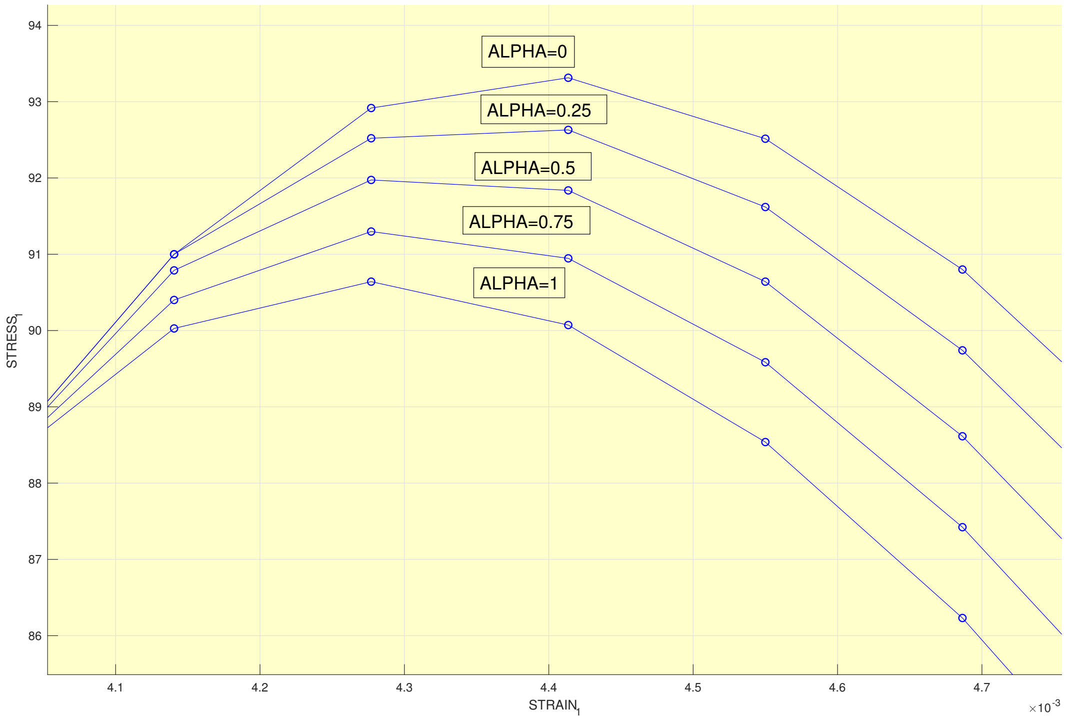


In the above Figure 7, we can see that the there is a relationship between the total time variable TIME_INT and the stress. It can be observed that as we decrease the value of TIME_INT, the amplitude of the stress/strain peaks increase. From the theory, we can correlate an inverse relationship between strain rate and the total time interval. We know that as total time increases, the strain rate decreases. From the theory we can also say that as the strain rate decreases, the maximum stress peaks should decrease since the deformation process is taking much longer when compared to if the process was happening in a short period of time. So, combining these statements allows us to say that as the value of the variable for total time TIME_INT increases, we should see lower stress peaks on the graph. And as we can see in figure 6 above, our model conforms to these postulations by decreasing stress peaks with decreasing strain rate caused my an increase in total time interval. This is an excellent indicator that the implemented model is correct.

5.3 VARIATION OF STRESS WITH ALPHA

We will now take a look at how variations in the parameter ALPHA change the stress/strain relationship. We know that the parameter ALPHA represents the numerical method used to obtain the solution. Below in Figure 8 we can see the effects on the STRESS I / STRAIN I curve of the variation of numerical methods due to changing the value of ALPHA.

Figure 8: STRESS I / STRAIN I peaks from variation of ALPHA parameter
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 100$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 100$ $\Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 100$ $\Delta\bar{\sigma}_2^{(3)} = 0$

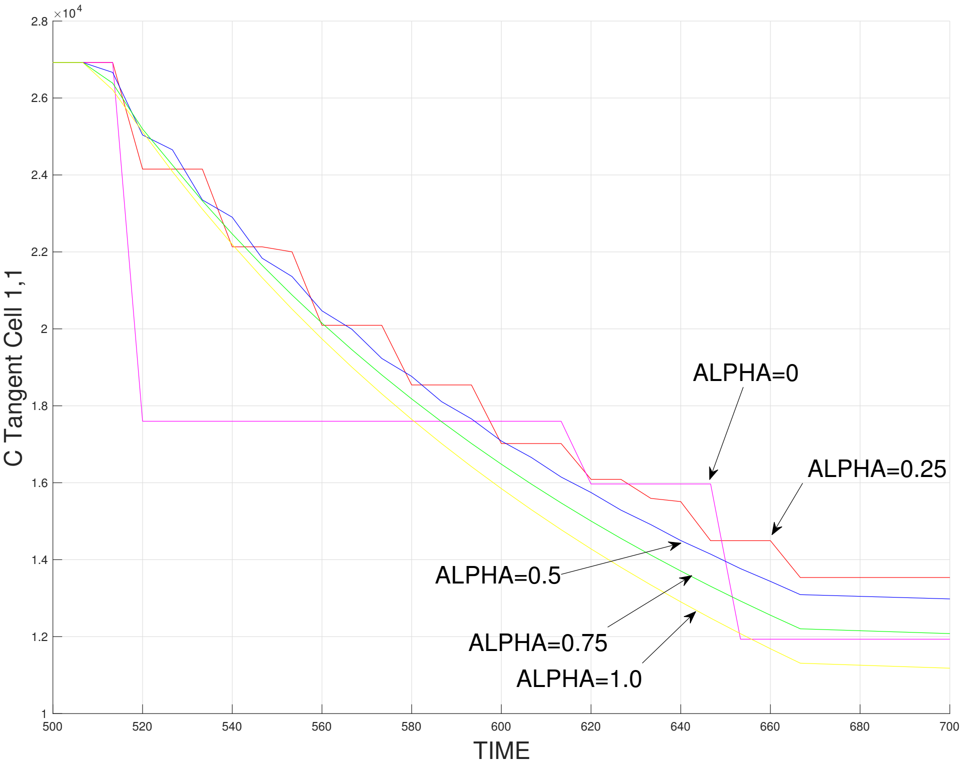


As we can notice from the figure above, if we increase the value of ALPHA, the maximum stress of the plot decreases. In the case when ALPHA equals 1, the peak is the lowest of the group. On the other hand, when ALPHA=0, the peak is the highest. It is safe to assume that the exact solution will fall somewhere between these upper and lower bounds created by ALPHA=0 and ALPHA=1. The differences between the results of changing the parameter ALPHA stem from the different solution method that each ALPHA value represents. All of these plots have some errors as none of these solutions are perfect. But it is safe to assume that the Crank-Nicholson method corresponding to ALPHA=0.5 is the most accurate of the group due to its unconditional stability and capacity to handle second order equations with an error of third order magnitude.

5.4 VARIATION OF C TANGENT OPERATOR WITH ALPHA

We will now take a look at the effects of varying ALPHA values on the time evolution of the $C_{1,1}$ tangent operator component. The results are displayed below in figure 9.

Figure 9: C Tangent operator against time with varying ALPHA values
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 100$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 100$ $\Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 100$ $\Delta\bar{\sigma}_2^{(3)} = 0$

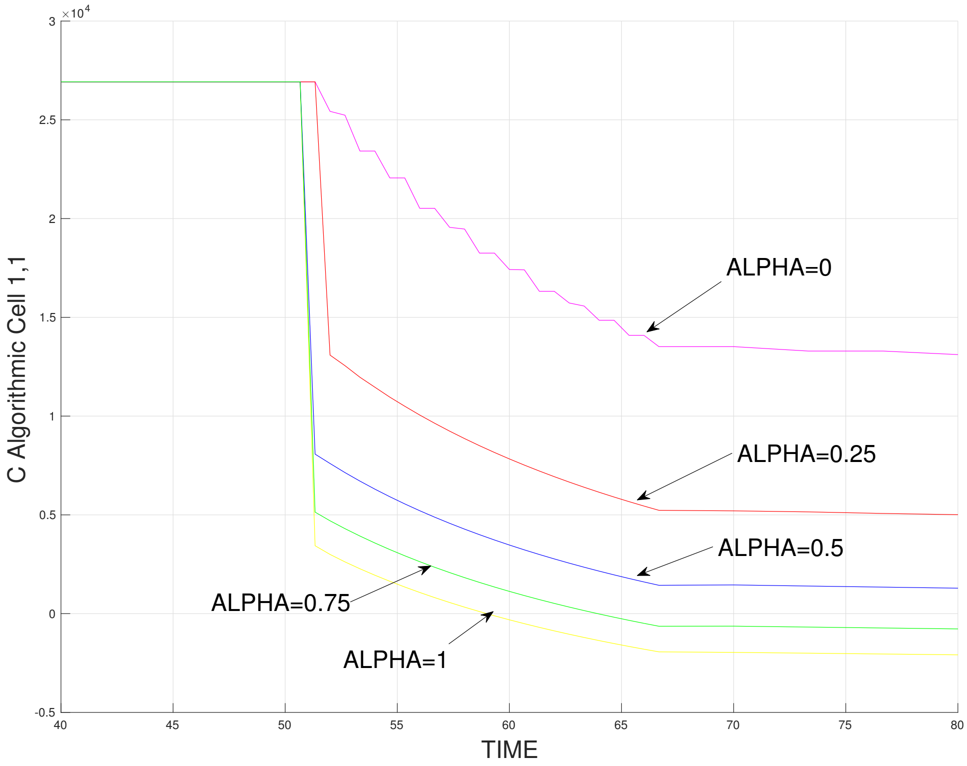


Above in figure 9, we can see 5 different plots corresponding to different values of ALPHA. A total time of 1000 was used to highlight some instabilities that may surface. The first thing we can notice is that we have a high degree of instability with time for ALPHA values of 0 and 0.25. This is consistent with the theory of θ-methods stating that values of $\theta < 0.5$ are explicit methods and will produce instability because they are only conditionally stable. It is also worth noting that there is a general downward trend in all plots. This is consistent with the theory because as damage increases with time, the C tangent operator is related to C by the factor (1-d). So as d increases with time, we expect the C tangent operator to decrease. Figure 9 confirms this trend.

5.5 VARIATION OF C ALGORITHMIC OPERATOR WITH ALPHA

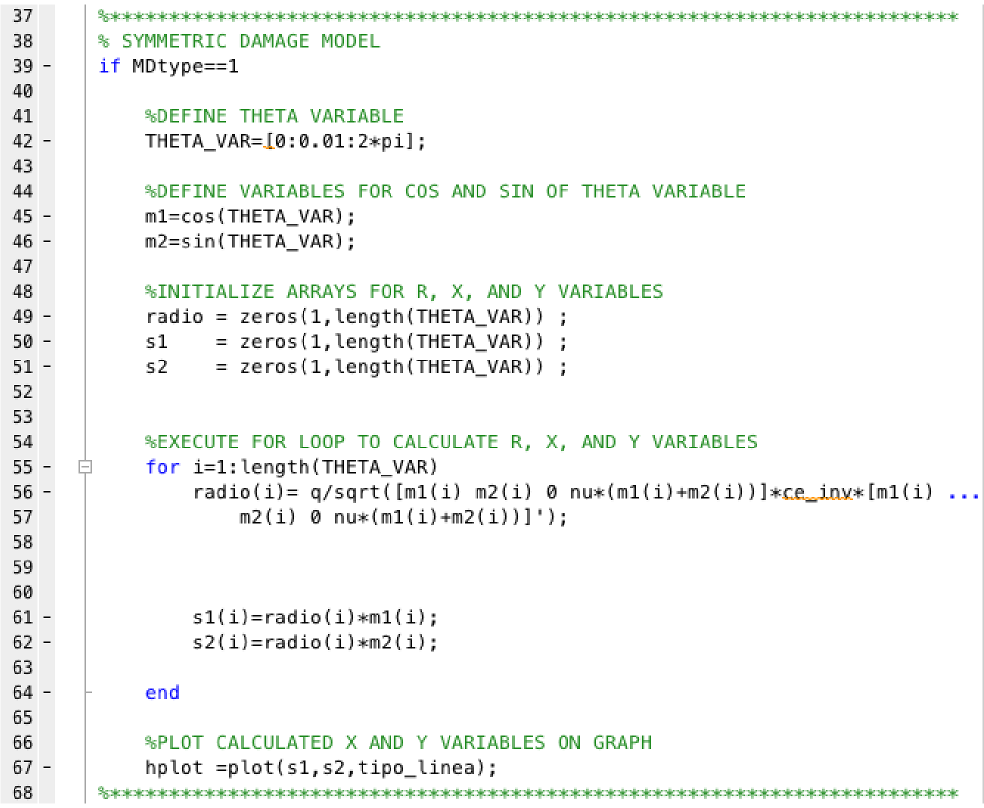
We will now show the variation of the implemented C algorithmic operator with respect to time for varying ALPHA values. These plots can be found in the figure below.

Figure 10: C Algorithmic operator against time with varying ALPHA values
E= 20000 YIELD_STRESS=85 POSSION=0.3
Stress Increment 1: $\Delta\bar{\sigma}_1^{(1)} = 100$ $\Delta\bar{\sigma}_2^{(1)} = 0$
Stress Increment 2: $\Delta\bar{\sigma}_1^{(2)} = 100$ $\Delta\bar{\sigma}_2^{(2)} = 0$
Stress Increment 3: $\Delta\bar{\sigma}_1^{(3)} = 100$ $\Delta\bar{\sigma}_2^{(3)} = 0$



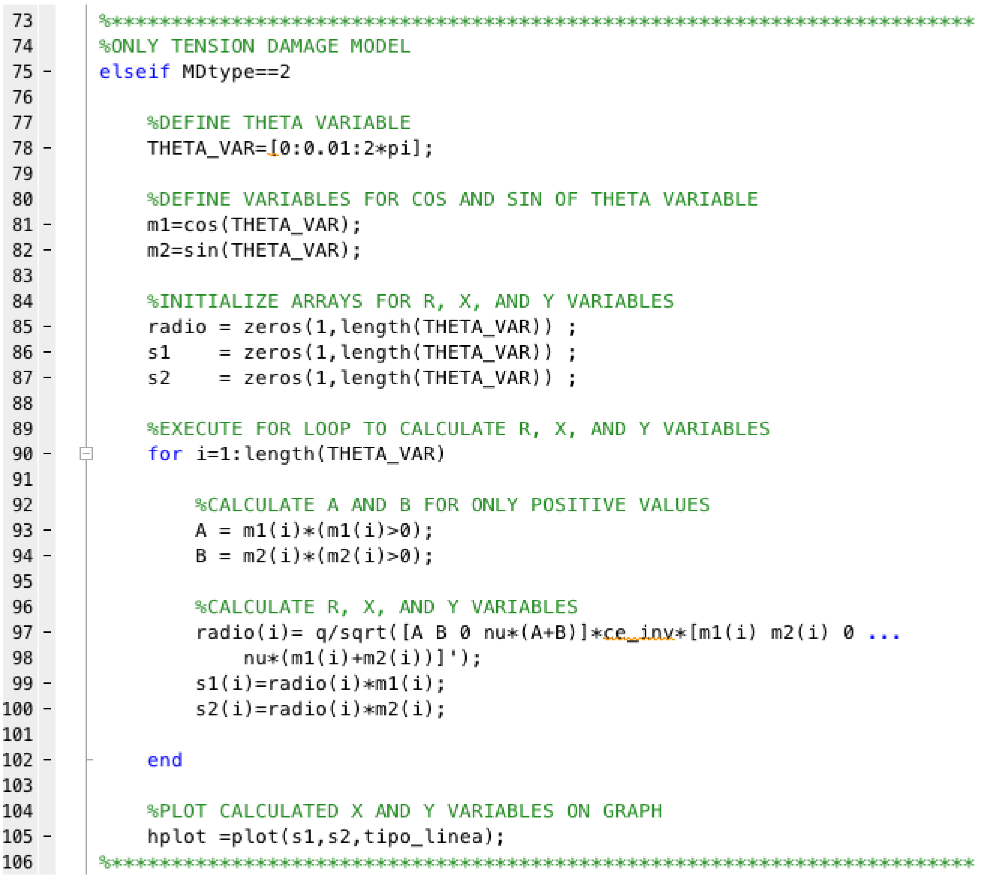
Above in Figure 10, similarly to the previous section we can see 5 plots of the C algorithmic operator versus time corresponding to differing ALPHA values, however this time with a smaller total time of 100s. We will first note the general downward trend of all the plots. Similarly to the C tangent operator, the C algorithmic operator is also related to C by the factor (1-d), so as d increases with time, we expect the C algorithmic operator to also decrease. We will not expect them to be the same though due to the addition of another term during the formulation of the C algorithmic operator. However, the exception to this statement is the plot of ALPHA=0. In this case, ALPHA=0 causes the entire second term to vanish leaving us with the exact results of the C tangent operator. These plots are consistent with the theory and bolster the correctness of the implementation. The code for both the C algorithmic and C tangent implementations can be found in Appendix XII.

6 APPENDIX



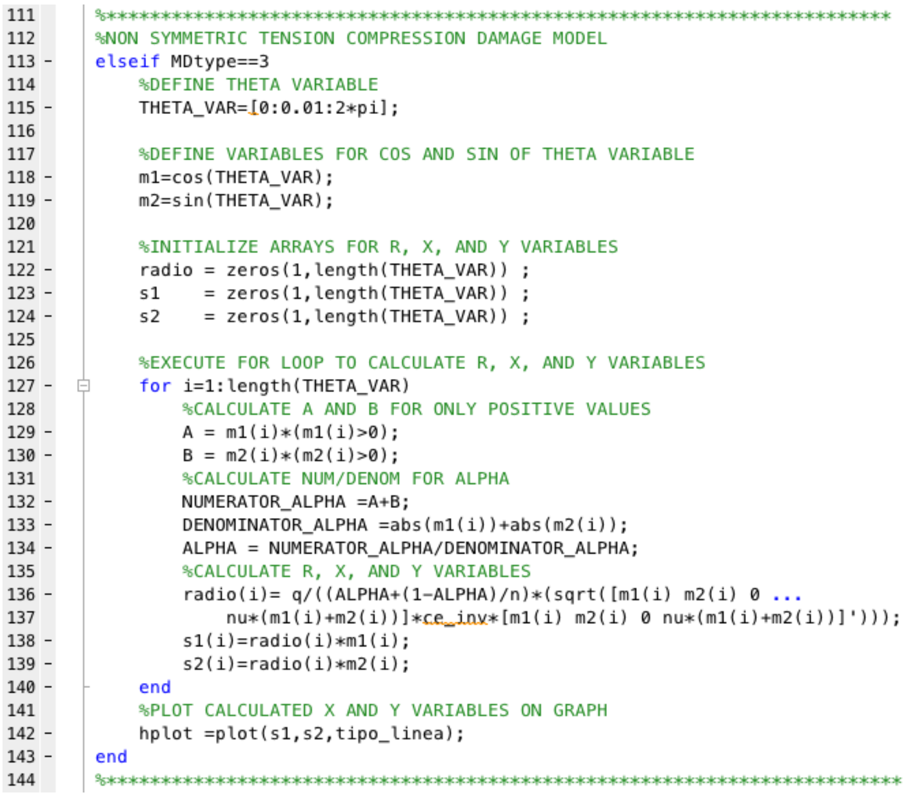
```
%************************************************************************
% SYMMETRIC DAMAGE MODEL
if MDtype==1

    %DEFINE THETA VARIABLE
    THETA_VAR=[0:0.01:2*pi];

    %DEFINE VARIABLES FOR COS AND SIN OF THETA VARIABLE
    m1=cos(THETA_VAR);
    m2=sin(THETA_VAR);

    %INITIALIZE ARRAYS FOR R, X, AND Y VARIABLES
    radio = zeros(1,length(THETA_VAR)) ;
    s1    = zeros(1,length(THETA_VAR)) ;
    s2    = zeros(1,length(THETA_VAR)) ;


    %EXECUTE FOR LOOP TO CALCULATE R, X, AND Y VARIABLES
    for i=1:length(THETA_VAR)
        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

    %PLOT CALCULATED X AND Y VARIABLES ON GRAPH
    hplot =plot(s1,s2,tipo_linea);
%************************************************************************
```

APPENDIX I: Revision of Symmetric Damage Model (dibujar_criterio_dano1.m)



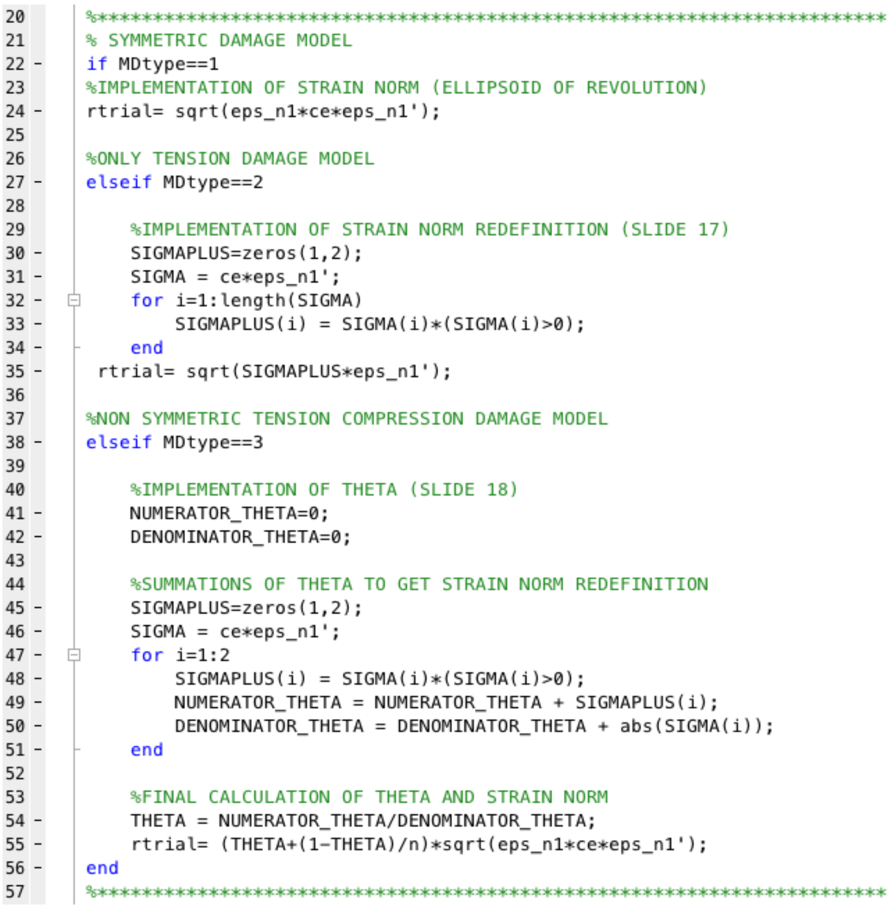
```
%************************************************************************
%ONLY TENSION DAMAGE MODEL
elseif MDtype==2

    %DEFINE THETA VARIABLE
    THETA_VAR=[0:0.01:2*pi];

    %DEFINE VARIABLES FOR COS AND SIN OF THETA VARIABLE
    m1=cos(THETA_VAR);
    m2=sin(THETA_VAR);

    %INITIALIZE ARRAYS FOR R, X, AND Y VARIABLES
    radio = zeros(1,length(THETA_VAR)) ;
    s1    = zeros(1,length(THETA_VAR)) ;
    s2    = zeros(1,length(THETA_VAR)) ;

    %EXECUTE FOR LOOP TO CALCULATE R, X, AND Y VARIABLES
    for i=1:length(THETA_VAR)

        %CALCULATE A AND B FOR ONLY POSITIVE VALUES
        A = m1(i)*(m1(i)>0);
        B = m2(i)*(m2(i)>0);

        %CALCULATE R, X, AND Y VARIABLES
        radio(i)= q/sqrt([A B 0 nu*(A+B)]*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

    %PLOT CALCULATED X AND Y VARIABLES ON GRAPH
    hplot =plot(s1,s2,tipo_linea);
%************************************************************************
```

APPENDIX II: Implementation of Tension Only Damage Model (dibujar_criterio_dano1.m)



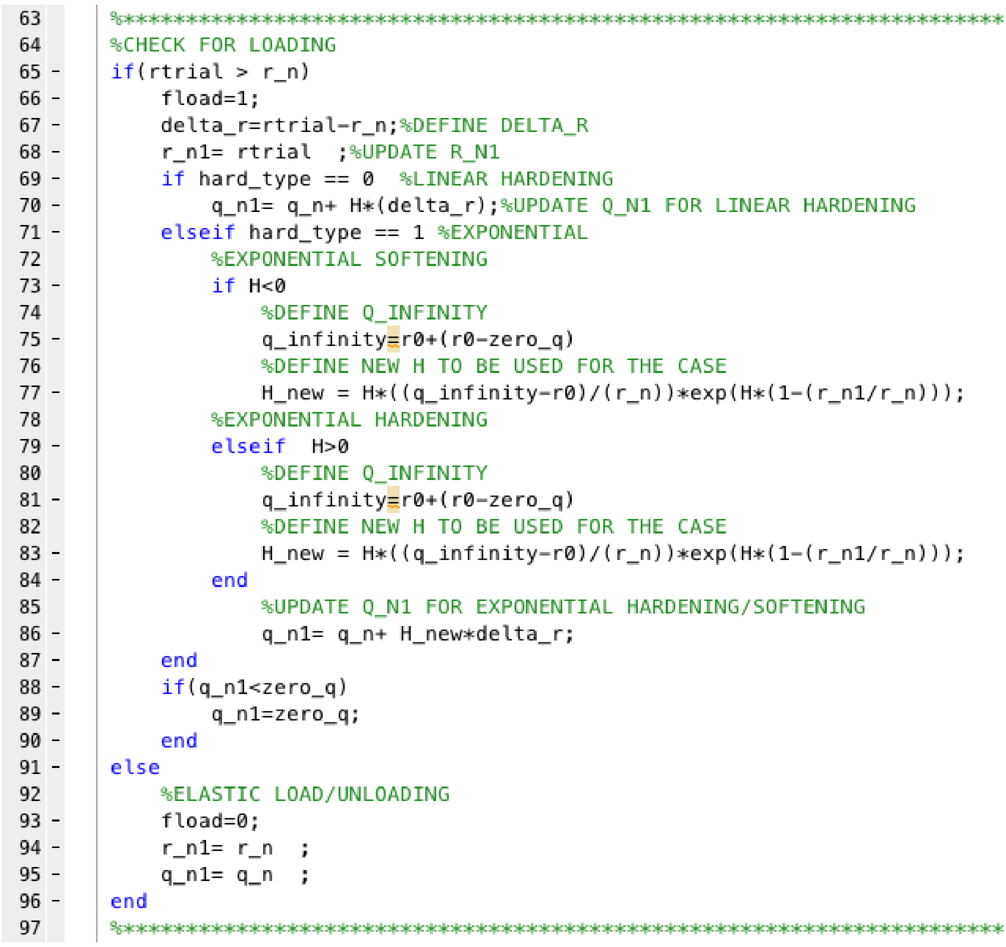
```
%***************************************************************************
%NON SYMMETRIC TENSION COMPRESSION DAMAGE MODEL
elseif MDtype==3
    %DEFINE THETA VARIABLE
    THETA_VAR=[0:0.01:2*pi];

    %DEFINE VARIABLES FOR COS AND SIN OF THETA VARIABLE
    m1=cos(THETA_VAR);
    m2=sin(THETA_VAR);

    %INITIALIZE ARRAYS FOR R, X, AND Y VARIABLES
    radio = zeros(1,length(THETA_VAR)) ;
    s1    = zeros(1,length(THETA_VAR)) ;
    s2    = zeros(1,length(THETA_VAR)) ;

    %EXECUTE FOR LOOP TO CALCULATE R, X, AND Y VARIABLES
    for i=1:length(THETA_VAR)
        %CALCULATE A AND B FOR ONLY POSITIVE VALUES
        A = m1(i)*(m1(i)>0);
        B = m2(i)*(m2(i)>0);
        %CALCULATE NUM/DENOM FOR ALPHA
        NUMERATOR_ALPHA =A+B;
        DENOMINATOR_ALPHA =abs(m1(i))+abs(m2(i));
        ALPHA = NUMERATOR_ALPHA/DENOMINATOR_ALPHA;
        %CALCULATE R, X, AND Y VARIABLES
        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
    %PLOT CALCULATED X AND Y VARIABLES ON GRAPH
    hplot =plot(s1,s2,tipo_linea);
end
%***************************************************************************
```

APPENDIX III: Non-Symmetric Tension Compression Damage Model (dibujar_criterio_dano1.m)



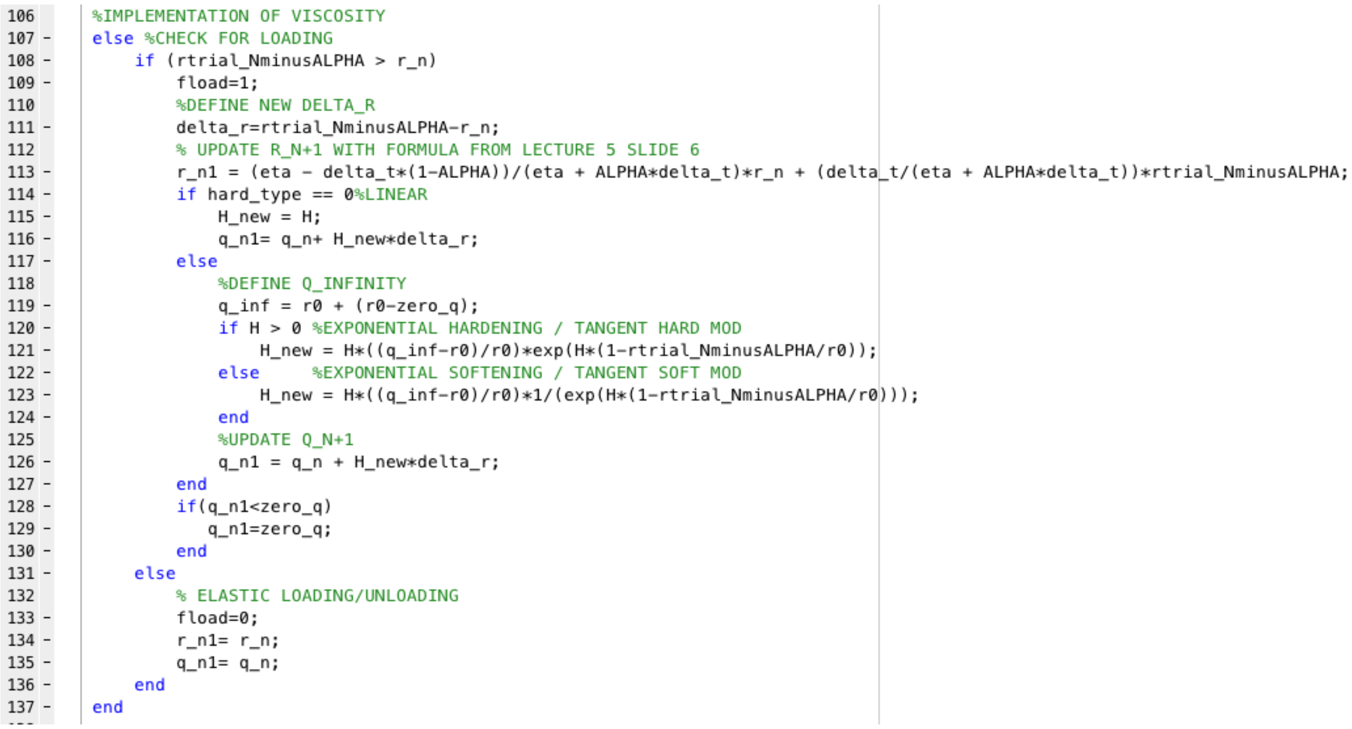
```
%***************************************************************************
% SYMMETRIC DAMAGE MODEL
if MDtype==1
%IMPLEMENTATION OF STRAIN NORM (ELLIPSOID OF REVOLUTION)
rtrial= sqrt(eps_n1*ce*eps_n1');

%ONLY TENSION DAMAGE MODEL
elseif MDtype==2

    %IMPLEMENTATION OF STRAIN NORM REDEFINITION (SLIDE 17)
    SIGMAPLUS=zeros(1,2);
    SIGMA = ce*eps_n1';
    for i=1:length(SIGMA)
        SIGMAPLUS(i) = SIGMA(i)*(SIGMA(i)>0);
    end
 rtrial= sqrt(SIGMAPLUS*eps_n1');

%NON SYMMETRIC TENSION COMPRESSION DAMAGE MODEL
elseif MDtype==3

    %IMPLEMENTATION OF THETA (SLIDE 18)
    NUMERATOR_THETA=0;
    DENOMINATOR_THETA=0;

    %SUMMATIONS OF THETA TO GET STRAIN NORM REDEFINITION
    SIGMAPLUS=zeros(1,2);
    SIGMA = ce*eps_n1';
    for i=1:2
        SIGMAPLUS(i) = SIGMA(i)*(SIGMA(i)>0);
        NUMERATOR_THETA = NUMERATOR_THETA + SIGMAPLUS(i);
        DENOMINATOR_THETA = DENOMINATOR_THETA + abs(SIGMA(i));
    end

    %FINAL CALCULATION OF THETA AND STRAIN NORM
    THETA = NUMERATOR_THETA/DENOMINATOR_THETA;
    rtrial= (THETA+(1-THETA)/n)*sqrt(eps_n1*ce*eps_n1');
end
%***************************************************************************
```

APPENDIX IV: Strain Norm Calculations For All Models (Modelos_de_dano1.m)



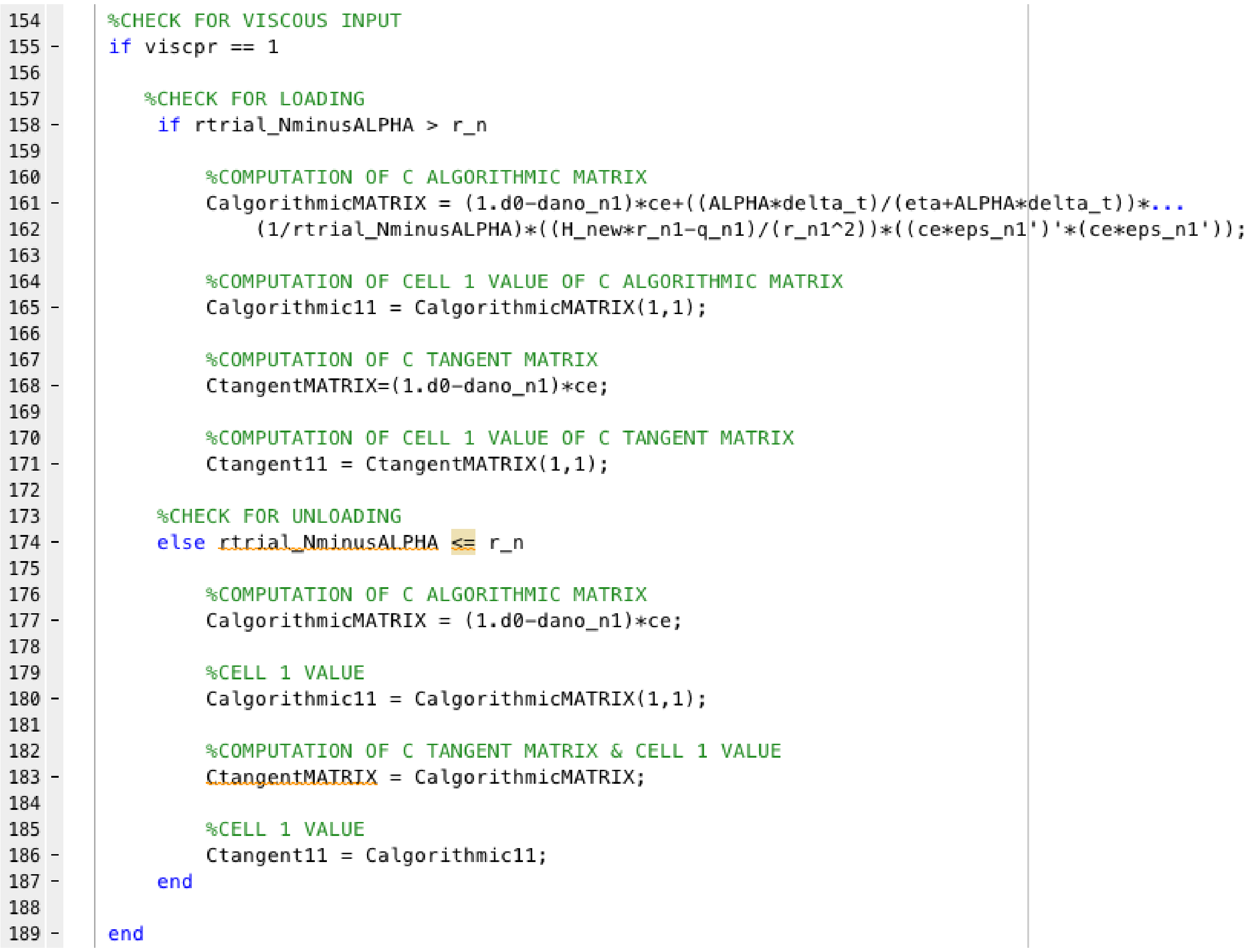
```
%**********************************************************************
%CHECK FOR LOADING
if(rtrial > r_n)
    fload=1;
    delta_r=rtrial-r_n;%DEFINE DELTA_R
    r_n1= rtrial  ;%UPDATE R_N1
    if hard_type == 0  %LINEAR HARDENING
        q_n1= q_n+ H*(delta_r);%UPDATE Q_N1 FOR LINEAR HARDENING
    elseif hard_type == 1 %EXPONENTIAL
        %EXPONENTIAL SOFTENING
        if H<0
            %DEFINE Q_INFINITY
            q_infinity=r0+(r0-zero_q)
            %DEFINE NEW H TO BE USED FOR THE CASE
            H_new = H*((q_infinity-r0)/(r_n))*exp(H*(1-(r_n1/r_n)));
        %EXPONENTIAL HARDENING
        elseif  H>0
            %DEFINE Q_INFINITY
            q_infinity=r0+(r0-zero_q)
            %DEFINE NEW H TO BE USED FOR THE CASE
            H_new = H*((q_infinity-r0)/(r_n))*exp(H*(1-(r_n1/r_n)));
        end
            %UPDATE Q_N1 FOR EXPONENTIAL HARDENING/SOFTENING
            q_n1= q_n+ H_new*delta_r;
    end
    if(q_n1<zero_q)
        q_n1=zero_q;
    end
else
    %ELASTIC LOAD/UNLOADING
    fload=0;
    r_n1= r_n  ;
    q_n1= q_n  ;
end
%**********************************************************************
```

APPENDIX V: Code for Linear/Exponential and Hard/Soft cases (rmap_dano1.m)



```
%IMPLEMENTATION OF VISCOSITY
else %CHECK FOR LOADING
    if (rtrial_NminusALPHA > r_n)
        fload=1;
        %DEFINE NEW DELTA_R
        delta_r=rtrial_NminusALPHA-r_n;
        % UPDATE R_N+1 WITH FORMULA FROM LECTURE 5 SLIDE 6
        r_n1 = (eta - delta_t*(1-ALPHA))/(eta + ALPHA*delta_t)*r_n + (delta_t/(eta + ALPHA*delta_t))*rtrial_NminusALPHA;
        if hard_type == 0%LINEAR
            H_new = H;
            q_n1= q_n+ H_new*delta_r;
        else
            %DEFINE Q_INFINITY
            q_inf = r0 + (r0-zero_q);
            if H > 0 %EXPONENTIAL HARDENING / TANGENT HARD MOD
                H_new = H*((q_inf-r0)/r0)*exp(H*(1-rtrial_NminusALPHA/r0));
            else     %EXPONENTIAL SOFTENING / TANGENT SOFT MOD
                H_new = H*((q_inf-r0)/r0)*1/(exp(H*(1-rtrial_NminusALPHA/r0)));
            end
            %UPDATE Q_N+1
            q_n1 = q_n + H_new*delta_r;
        end
        if(q_n1<zero_q)
           q_n1=zero_q;
        end
    else
        % ELASTIC LOADING/UNLOADING
        fload=0;
        r_n1= r_n;
        q_n1= q_n;
    end
end
```

APPENDIX VI: Implementation for isotropic visco-damage (rmap_dano1.m)



```
%CHECK FOR VISCOUS INPUT
if viscpr == 1

   %CHECK FOR LOADING
    if rtrial_NminusALPHA > r_n

        %COMPUTATION OF C ALGORITHMIC MATRIX
        CalgorithmicMATRIX = (1.d0-dano_n1)*ce+((ALPHA*delta_t)/(eta+ALPHA*delta_t))*...
            (1/rtrial_NminusALPHA)*((H_new*r_n1-q_n1)/(r_n1^2))*((ce*eps_n1')'*(ce*eps_n1'));

        %COMPUTATION OF CELL 1 VALUE OF C ALGORITHMIC MATRIX
        Calgorithmic11 = CalgorithmicMATRIX(1,1);

        %COMPUTATION OF C TANGENT MATRIX
        CtangentMATRIX=(1.d0-dano_n1)*ce;

        %COMPUTATION OF CELL 1 VALUE OF C TANGENT MATRIX
        Ctangent11 = CtangentMATRIX(1,1);

    %CHECK FOR UNLOADING
    else rtrial_NminusALPHA <= r_n

        %COMPUTATION OF C ALGORITHMIC MATRIX
        CalgorithmicMATRIX = (1.d0-dano_n1)*ce;

        %CELL 1 VALUE
        Calgorithmic11 = CalgorithmicMATRIX(1,1);

        %COMPUTATION OF C TANGENT MATRIX & CELL 1 VALUE
        CtangentMATRIX = CalgorithmicMATRIX;

        %CELL 1 VALUE
        Ctangent11 = Calgorithmic11;
    end

end
```

APPENDIX VII: Implementation for $C_{tangent}$ and $C_{algorithmic}$ (rmap_dano1.m)