Georgia Institute of Technology

ME-6203 Inelastic Deformation of Solids

Time Integration Coding Scheme Report

Author:
Pierce Heintzelman

Instructors:
Dr. Surya Kalidindi

April 12, 2021

Problem Statement

The analysis considers an isochoric deformation process [1] defined by the following strain rate parameters:

$$\dot{\epsilon}_{11} = K, \dot{\epsilon}_{22} = -K, \dot{\epsilon}_{33} = \dot{\epsilon}_{12} = 0, \dot{\epsilon}_{13} = \dot{\epsilon}_{23} = 0.5K$$

K represents a piece-wise function that takes the strain rate values as shown in Table 1. Material constants including Young's Modulus, Poisson's Ratio, rate sensitivity constant m, plastic modulus h, and initial yield stress are known. Additionally it is known that both stress and strain of the sample is zero at $t = 0$. Given the strain rate history, the goal of the analysis is to calculate stress and strain values at a new time τ where $\tau = t + \Delta t$. Both explicit and implicit time integration schemes are used to simulate the elastic-plastic deformation process. MATLAB is utilized to complete each integration scheme and create stress-strain curves in the 11, 22, and 23 directions. The explicit and implicit methods each have different strengths that will become apparent throughout the analysis.

Methods

The explicit method of integration involves directly solving for $\sigma(\tau)$ and $s(\tau)$ using known values at time t. The two equations being solved in the explicit method are [2]

$$\sigma(\tau) = (\mathbf{C}(\dot{\boldsymbol{\epsilon}} - \frac{3}{2}\dot{\epsilon}_o(\frac{\bar{\sigma}(t)}{s(t)})^{1/m} * \frac{\boldsymbol{\sigma}'(t)}{\bar{\sigma}(t)})\Delta t + \boldsymbol{\sigma}(t)$$

$$s(\tau) = (h\dot{\epsilon}_o(\frac{\bar{\sigma}(t)}{s(t)})^{1/m})\Delta t + s(t)$$

These equations are solved at each time step and used to update the stress and strain values in the desired directions. At the end of the predefined time domain, the stress and strain can be graphed to study the material response.

The implicit response, in contrast to the explicit, takes into account properties both at the current time step and the next time step. This method utilizes an initial guess that is updated at each iteration. Mathematical substitutions and an isotropic material assumption allow the original 7x7 system can be reduced to two equations with two unknowns: $\bar{\sigma}(\tau)$ and $s(\tau)$. Then, a multivariate Newton-Raphson method is applied to find the roots of the two equations below

$$f_1 = \bar{\sigma}^* - \bar{\sigma}(\tau) - 3\mu\dot{\epsilon}_o\Delta t(\frac{\bar{\sigma}(t)}{s(t)}^{1/m}) = 0$$

$$f_2 = s(\tau) - (h\dot{\epsilon}_o\Delta t(\frac{\bar{\sigma}(t)}{s(t)})^{1/m}) - s(t) = 0$$

After using the derivatives of each function to build the Jacobian, the solution for $\bar{\sigma}$ and s is updated. Until the desired tolerance of the solution is met, the iterations continue. After the roots of the equations have been identified, $\bar{\sigma}$ is plugged back in to find the solution for the components of the deviatoric stress tensor and finally the overall stress tensor.

Analysis

The explicit scheme is implemented first. This method is easy to code as it consists mainly of a for loop that solves the two equations for $\sigma(\tau)$ and $s(\tau)$ on each iteration and updates the stress and strain values in the 11, 22, and 23 directions. The ΔT parameter is defined as the total time span of 40 seconds divided by the length of the strain rate vector. Two different explicit integrations were performed while changing ΔT from .01s initially to .001. A second integration is needed because a time step of .01 is not small enough to get the explicit method to converge. This is clearly seen in the blue plotted line in Figures 1, 2, and 3. While the explicit solution at .01s tracks well with the others, it is very jagged and spiky all throughout which communicates the solution hasn't converged. Repeating this same process at with a larger strain rate vector, and consequently a smaller Δt of .001s, the solution converges much more cleanly. This solution is plotted on Figures 1, 2, and 3 in red. Using the tic/toc functions in MATLAB, the computing time for the first explicit integration is $.0168s$ while the smaller time step integration with correct convergence takes $.0889s$, over five times longer.

Finally, the implicit scheme is coded and implemented. This code is significantly more complex than the explicit scheme as it starts with an initial guess to help solve equations f_1 and f_2, and then used a multivariate Newton-Raphson method to continually improve the accuracy of the solution. The code requires nested for loops, and it recalculates the derivatives of f_1 and f_2 each iteration to develop the Jacobian matrix. However, the advantages of this method become clear very quickly upon deciding the length of the strain rate vector and the value of Δt. I ultimately decided that a Δt of .1 seconds was plenty sufficient for convergence. The implicit scheme deformation solution for the three desired directions is plotted in black in Figures 1, 2, and 3. The implicit solution is visually identical to the explicit solution graphed in red, and it has a Δt value 100 times larger. This exemplifies the ease of convergence that implicit methods offer above explicit methods. The computational time required for the implicit solution is $.0599s$, which is significantly less than the converged explicit method.

Figures 4, 5, and 6 show stress over time for the 3 desired components of the strain tensor. Similar jaggedness is observed in Figure 4 for the explicit scheme when Δt is .01s. The jagged peaks disappear in Figure 5 when the time step is decreased to .001s. Figure 5 and Figure 6, with the implicit scheme, are visually identical with the stress reponses over time.

Conclusion

Overall, the solution of this isochoric elastic-plastic deformation process has exemplified the strengths and weaknesses of explicit and implicit numerical integration schemes. Explicit schemes, while more simplistic mathematically and easier to code, require time steps orders of magnitude smaller than implicit schemes to converge properly. Due to this significantly smaller time step, explicit solutions take longer to compute. Implicit schemes are more mathematically rigorous because they include information about the process both at the current time and the next time step. This makes them slightly harder to implement, but the extra effort quickly pays off with the decrease in time step for convergence and a decrease in computation time.

Appendix

Tables

Strain Rate K	Time (s)
.002 s^-1	$0 \leq t < 10$
-.001 s^-1	$10 \leq t < 30$
.001 s^-1	$30 \leq t < 40$

Table 1: K Parameter Values

Figures

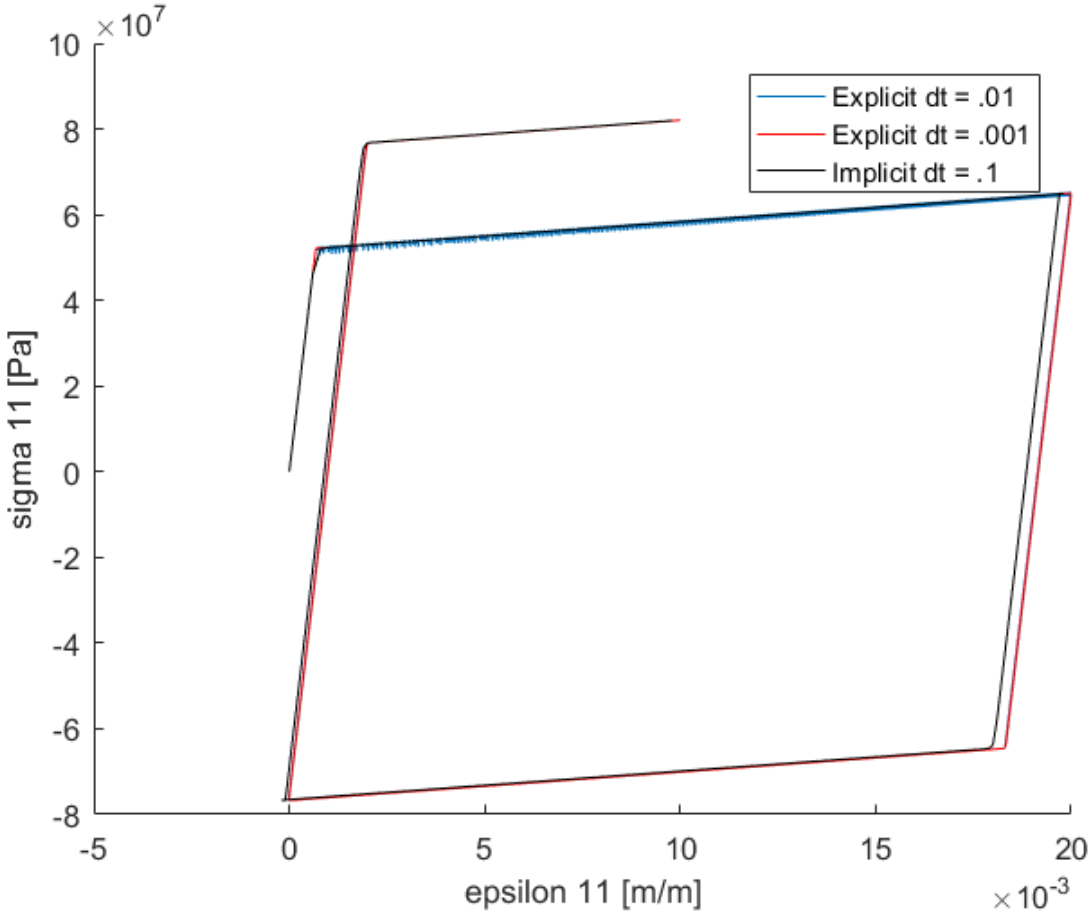

Figure 1: Response in 11 Direction

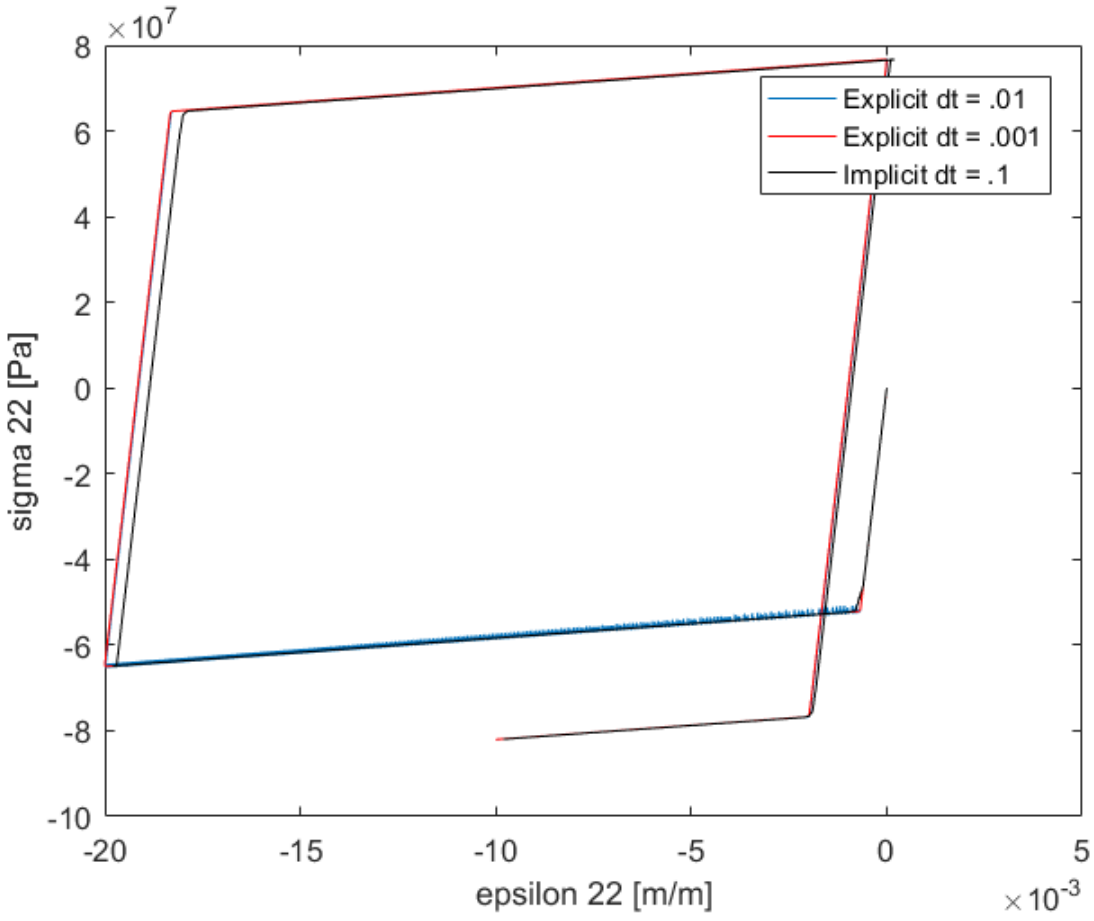

Figure 2: Response in 22 Direction

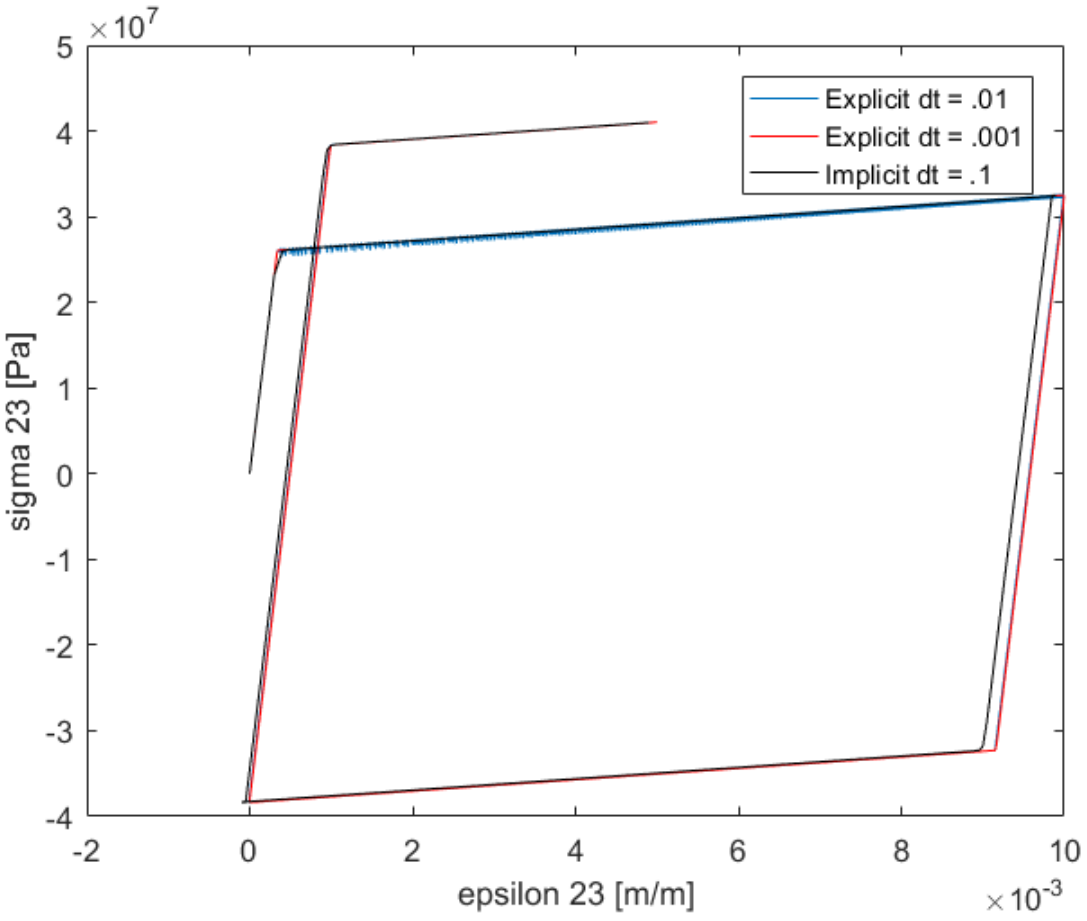

Figure 3: Response in 23 Direction

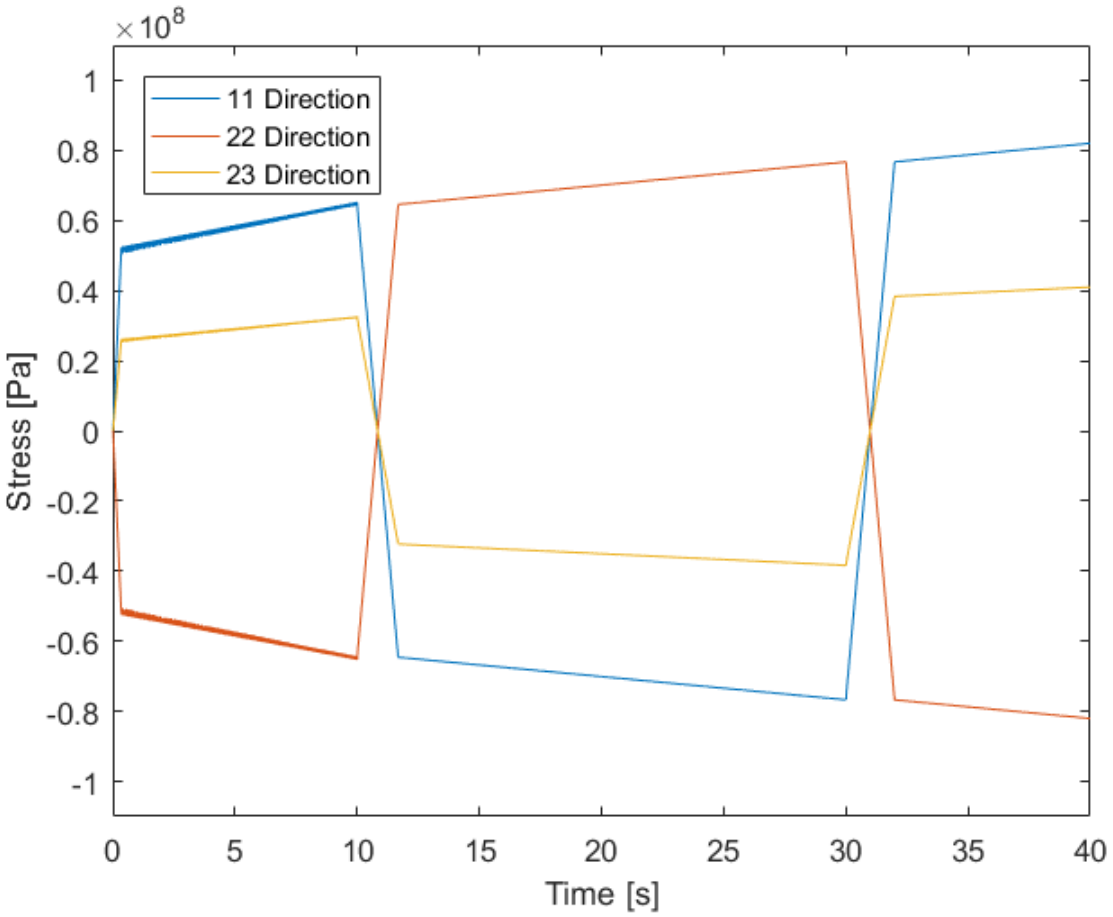

Figure 4: Stress vs. Time for Explicit Scheme with dt=.01

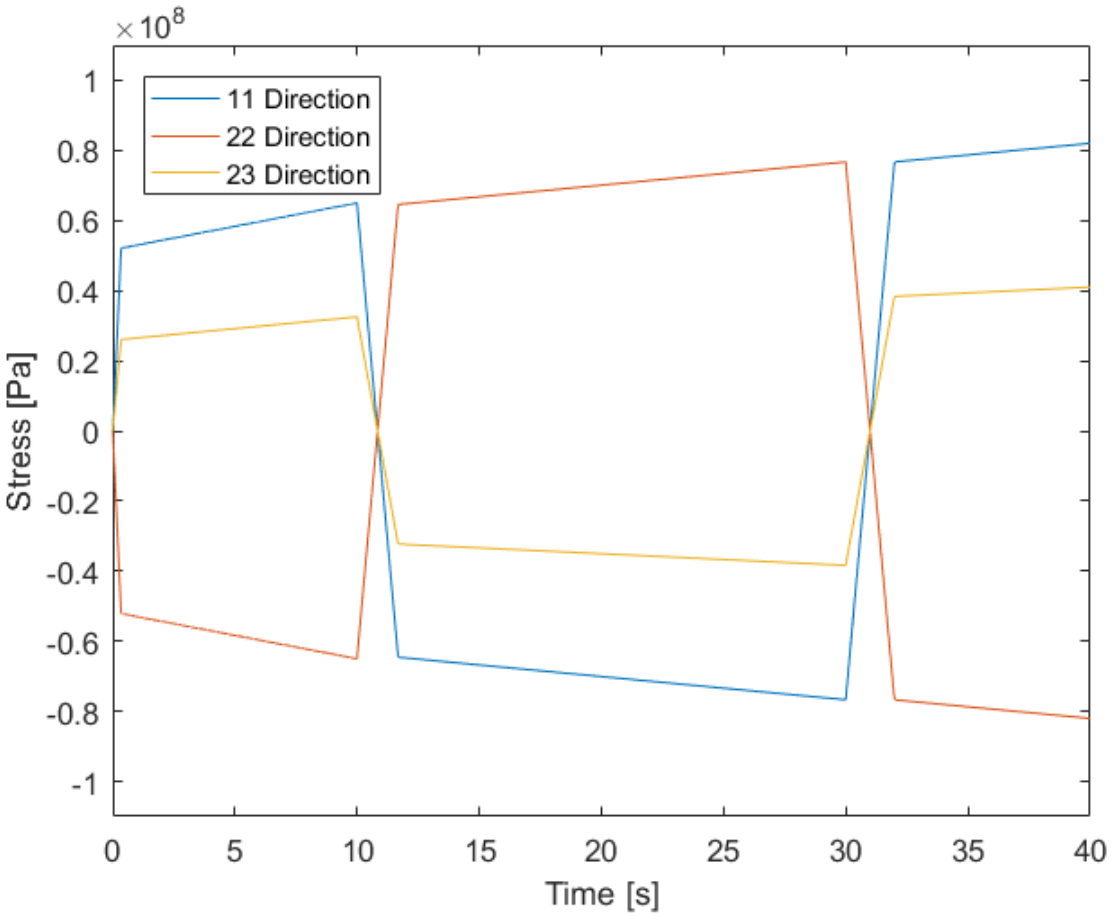

Figure 5: Stress vs. Time for Explicit Scheme with dt=.001

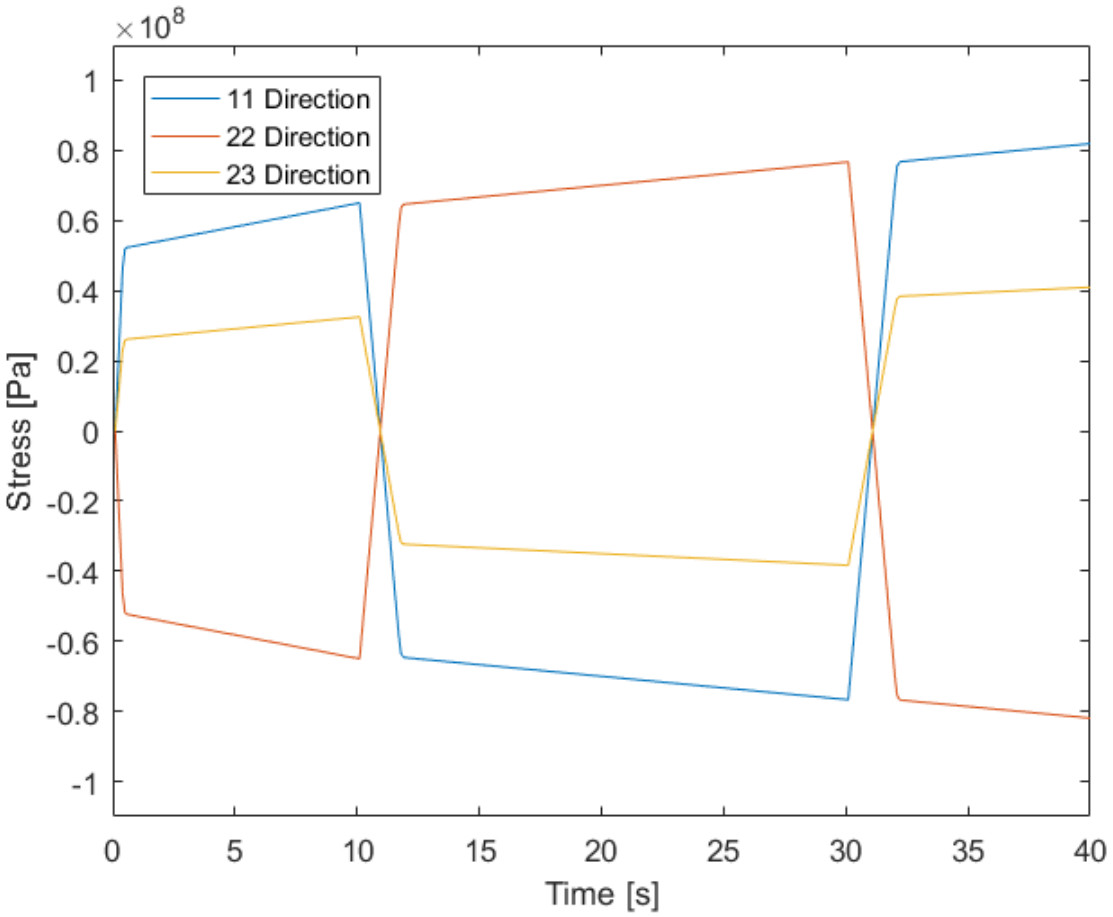

Figure 6: Stress vs. Time for Implicit Scheme with dt=.1

References

[1] Kalidindi, Surya. "Time Integration Scheme Assignment" ME 6203, 30 Mar. 2021, Georgia Institute of Technology. Assignment.

[2] Kalidindi, Surya. "Integration Scheme for Isoptropic Rate Dependent Plasticity." ME 6203, 30 Mar. 2021, Georgia Institute of Technology. Lecture.

Matlab Code

```
% ME 6203 Time Integration Scheme
% Pierce Heintzelman
clear
clc
clf
% Start with explicit scheme

% Known Constants
E = 100e9; %Pa
nu = 0.3;

% Shear Modulus
mu=E/2/(1+nu);
k = E/3/(1-2*nu);
lam = k-2/3*mu;

% Tensor - Reduced Form 6x6
C=[[lam+2*mu,lam,lam,0,0,0];[lam,lam+2*mu,lam,0,0,0];...
    [lam,lam,lam+2*mu,0,0,0];[0,0,0,2*mu,0,0];...
    [0,0,0,0,2*mu,0];[0,0,0,0,0,2*mu]];


% plasticity parameters
edot = 0.001;
m = 0.01;
h = 1000e6; %Pa
so = 100e6; %Initial Yield Stress in PA

% Initial strain based on problem statemnt
eo = [.002*ones(1000,1)', -.001*ones(1000,1)', -.001*ones(1000,1)',...
    .001*ones(1000,1)'];

% Initialize stress
sigma0 = [0,0,0,0,0,0]';

ttime = 40;

dtime = ttime/length(eo);
t = dtime:dtime:40;
%Initialize arrays
sig11 = zeros(length(eo),1);
sig22 = zeros(length(eo),1);
sig23 = zeros(length(eo),1);
```

```
e11 = zeros(length(eo),1);
e22 = zeros(length(eo),1);
e23 = zeros(length(eo),1);

tic
for i = 1:length(eo)
    if i == 1
        sigma_t = sigma0;
        s_t = so;
    end
    sig11(i) = sigma_t(1);
    sig22(i) = sigma_t(2);
    sig23(i) = sigma_t(6);


    erate = [eo(i),-1*eo(i), 0, 0, 0.5*eo(i), 0.5*eo(i)]';

    % Keeping track of strain
    if i>=2
        e11(i) = e11(i-1) + erate(1)*dtime;
        e22(i) = e22(i-1) + erate(2)*dtime;
        e23(i) = e23(i-1) + erate(6)*dtime;
    end

    mises_t = sqrt(0.5*((sigma_t(1)-sigma_t(2))^2 +...
    (sigma_t(2) - sigma_t(3))^2 ...
        + (sigma_t(1) - sigma_t(3))^2+3*(sigma_t(4)^2+sigma_t(5)^2+...
    sigma_t(6)^2)));
    peeqdot = edot * (mises_t/s_t)^(1/m);
    press = (sigma_t(1) + sigma_t(2) + sigma_t(3))/3;
    dev_t = sigma_t - press*[1,1,1,0,0,0]';
    if mises_t==0
        epdot = [0,0,0,0,0,0]';
    else
        epdot = 3/2*peeqdot*dev_t/mises_t;
    end
    % update stress and strength values

    sigma_tau = C * (erate-epdot)*dtime + sigma_t;
    sigma_t = sigma_tau;
    s_tau = h*peeqdot*dtime + s_t;
    s_t = s_tau;

end
toc
```

```
hold on

figure(1)
plot(e11, sig11)

figure(2)
plot(e22, sig22)

figure(3)
plot(e23, sig23)

figure(4)
plot(t, sig11, t, sig22, t, sig23)
xlabel('Time [s]')
ylabel('Stress [Pa]')
legend('11 Direction','22 Direction', '23 Direction', 'Location', 'northwest'
ylim([-1.1e8 1.1e8])

%DT = .001
% Initial strain based on problem statemnt
eo = [.002*ones(10000,1)', -.001*ones(10000,1)', -.001*ones(10000,1)',...
    .001*ones(10000,1)'];

% Initialize stress
sigma0 = [0,0,0,0,0,0]';

ttime = 40;

dtime = ttime/length(eo);
t = dtime:dtime:40;
%Initialize arrays
sig11 = zeros(length(eo),1);
sig22 = zeros(length(eo),1);
sig23 = zeros(length(eo),1);
e11 = zeros(length(eo),1);
e22 = zeros(length(eo),1);
e23 = zeros(length(eo),1);

tic
for i = 1:length(eo)
    if i == 1
        sigma_t = sigma0;
        s_t = so;
    end
    sig11(i) = sigma_t(1);
```

```
    sig22(i) = sigma_t(2);
    sig23(i) = sigma_t(6);


    erate = [eo(i),-1*eo(i), 0, 0, 0.5*eo(i), 0.5*eo(i)]';

    % Keeping track of strain
    if i>=2
        e11(i) = e11(i-1) + erate(1)*dtime;
        e22(i) = e22(i-1) + erate(2)*dtime;
        e23(i) = e23(i-1) + erate(6)*dtime;
    end

    mises_t = sqrt(0.5*((sigma_t(1)-sigma_t(2))^2 +...
    (sigma_t(2) - sigma_t(3))^2 ...
        + (sigma_t(1) - sigma_t(3))^2+3*(sigma_t(4)^2+...
    sigma_t(5)^2+sigma_t(6)^2)));
    peeqdot = edot * (mises_t/s_t)^(1/m);
    press = (sigma_t(1) + sigma_t(2) + sigma_t(3))/3;
    dev_t = sigma_t - press*[1,1,1,0,0,0]';
    if mises_t==0
        epdot = [0,0,0,0,0,0]';
    else
        epdot = 3/2*peeqdot*dev_t/mises_t;
    end
    % update stress and strength values

    sigma_tau = C * (erate-epdot)*dtime + sigma_t;
    sigma_t = sigma_tau;
    s_tau = h*peeqdot*dtime + s_t;
    s_t = s_tau;

end
toc

figure(1)
hold on
plot(e11, sig11,'r')
xlabel('epsilon 11 [m/m]')
ylabel('sigma 11 [Pa]')
% title('Response in 11 Direction')
legend('Explicit dt = .01','Explicit dt = .001')

figure(2)
hold on
```

```
plot(e22, sig22,'r')
xlabel('epsilon 22 [m/m]')
ylabel('sigma 22 [Pa]')
% title('Response in 22 Direction')
legend('Explicit dt = .01','Explicit dt = .001')

figure(3)
hold on
plot(e23, sig23,'r')
xlabel('epsilon 23 [m/m]')
ylabel('sigma 23 [Pa]')
% title('Response in 23 Direction')
legend('Explicit dt = .01','Explicit dt = .001')

figure(5)
plot(t, sig11, t, sig22, t, sig23)
xlabel('Time [s]')
ylabel('Stress [Pa]')
legend('11 Direction','22 Direction', '23 Direction', 'Location', 'northwest'
ylim([-1.1e8 1.1e8])

%Implicit Scheme

%Initialize arrays

eo = [.002*ones(100,1)', -.001*ones(100,1)', -.001*ones(100,1)',...
    .001*ones(100,1)'];

sig11 = zeros(length(eo),1);
sig22 = zeros(length(eo),1);
sig23 = zeros(length(eo),1);
e11 = zeros(length(eo),1);
e22 = zeros(length(eo),1);
e23 = zeros(length(eo),1);


dtime = ttime/length(eo);
t = dtime:dtime:40;

tol = 1e-3; %set tolerance
tic
for i = 1:length(eo)
    if i==1
        sigma_t = sigma0;
        s_t = so;
```

```
end
sig11(i) = sigma_t(1);
sig22(i) = sigma_t(2);
sig23(i) = sigma_t(6);

%strain rate at current time step
erate = [eo(i),-1*eo(i), 0, 0, 0.5*eo(i), 0.5*eo(i)]';

% strain(tau) = strain(t) + strain_rate*dtime
if i>=2
    e11(i) = e11(i-1)+erate(1)*dtime;
    e22(i) = e22(i-1)+erate(2)*dtime;
    e23(i) = e23(i-1)+erate(6)*dtime;
end

%calculate trial stresses - elastic predictor
sig_tr = C*erate*dtime+sigma_t;
press = (sig_tr(1) + sig_tr(2) + sig_tr(3))/3;
dev_tr = sig_tr - press*[1,1,1,0,0,0]';

% Find mises equivalent of t and trial stress
mises_t = sqrt(0.5*((sigma_t(1)-sigma_t(2))^2 + ...
(sigma_t(2) - sigma_t(3))^2 ...
    + (sigma_t(1) - sigma_t(3))^2+3*(sigma_t(4)^2+...
sigma_t(5)^2+sigma_t(6)^2)));
mises_tr = sqrt(0.5*((sig_tr(1)-sig_tr(2))^2 +...
(sig_tr(2) - sig_tr(3))^2 ...
    + (sig_tr(1) - sig_tr(3))^2+3*(sig_tr(4)^2+...
sig_tr(5)^2+sig_tr(6)^2)));

%Newton Raphson iterations
mises_tau = mises_tr;
s_tau = s_t;
for j = 1:500
    peeqdot = edot*(mises_tau/s_tau)^(1/m);
    g1 = mises_tr-mises_tau - 3*mu*peeqdot*dtime;
    g2 = s_tau-s_t-h*peeqdot*dtime;
    g = [g1 g2]';
    y = [mises_tau s_tau]';
    g1m = -1-3*mu*edot*dtime/m*((mises_tau/s_tau)^(1/m-1))/s_tau;
    g1s = 3*mu*dtime/m*peeqdot/s_tau;
    g2m = -1*h*edot*dtime/m*((mises_tau/s_tau)^(1/m-1))/s_tau;
    g2s = 1+h*dtime/m*peeqdot/s_tau;
    J = [[g1m g1s];[g2m g2s]];
    y = y-(inv(J))*g;
```

```
        mises_tau = y(1);
        s_tau = y(2);
        if abs(g1)<tol*so && abs(g2) < tol*so
            break
        end
        if j==50
            disp('Implicit method not converged.')
        end
    end
    corrf = 1+3*mu*edot*dtime*(mises_tau/s_tau)^(1/m-1)/s_tau;
    dev_tau = dev_tr/corrf;
    sigma_tau = dev_tau+press*[1,1,1,0,0,0]';

    %stress strength update
    sigma_t = sigma_tau;
    s_t = s_tau;

end
toc

figure(1)
hold on
plot(e11, sig11,'k')
legend('Explicit dt = .01','Explicit dt = .001','Implicit dt = .1')

figure(2)
hold on
plot(e22, sig22,'k')
legend('Explicit dt = .01','Explicit dt = .001','Implicit dt = .1')

figure(3)
hold on
plot(e23, sig23,'k')
legend('Explicit dt = .01','Explicit dt = .001','Implicit dt = .1')

figure(6)
plot(t, sig11, t, sig22, t, sig23)
xlabel('Time [s]')
ylabel('Stress [Pa]')
legend('11 Direction','22 Direction', '23 Direction', 'Location', 'northwest'
ylim([-1.1e8 1.1e8])
```