MATH128B
NUMERICAL ANALYSIS

Spring 2002

TERM PROJECT

Submitted
by
Afsin SARITAS
14599793

NUMERICAL EXPLORATIONS ON PLASTIC MATERIAL MODELS

1. INTRODUCTION

Before starting to talk within the technical jargon of “Solid Mechanics”, it is better to briefly explain some physical entities that will be used in this work. Then there will be a brief introduction to Plasticity.

What is stress?
The dictionary meaning of stress taken from Oxford English
“A force acting on or within a body or structure and tending to deform it; now usu. the intensity of this, the force per unit area.”

Mathematical definition of stress is:

$$\sigma = \lim_{\Delta A \to 0} \frac{\Delta F}{\Delta A} \tag{1}$$

What is strain?
Strain is a dimensionless ratio for measuring the deformation with respect to an original configuration.

$$\varepsilon = \frac{L - L_0}{L_0} = \frac{\Delta L}{L_0} \tag{2}$$

There is a more general mathematical definition:

$$\varepsilon = \int_{L_0}^{L} \frac{1}{L} dL = \ln\left(\frac{L}{L_0}\right) = \ln\left(1 + \frac{\Delta L}{L_0}\right) \tag{3}$$

What is Plasticity?
Any material body deforms when subjected to external forces. The deformation is called “elastic” if it is reversible and time independent, that is, if the deformation vanishes instantaneously as soon as forces are removed. A reversible but time-dependent deformation is known as “viscoelastic”; in this case the deformation increases with time after application of load and it decreases slowly after the load is removed. The deformation is called “plastic” if it is irreversible or permanent.

The adjective “plastic” comes from the classical Greek verb *πλασσειν* (Plassing), meaning “to shape”; it thus describes materials, such as ductile metals, clay, which have the property that bodies made from them can have their shape easily changed by the application of appropriately directed forces, and retain their new shape upon removal of such forces.

What does theory of plasticity deal with?

The theory of plasticity deals with the stress-strain and load-deflection relationships for a plastically deforming “ductile” material or structure. The establishments of these relationships follows from

- ❑ the experimental observations
- ❑ the mathematical representation.

The stress states that are normally achieved in any experiment are simple and uniform, but the ultimate goal of any plasticity theory is a general mathematical formulation that can predict the plastic deformation of materials under complex loading and boundary conditions.

History of Theory of Plasticity

The history of plasticity theory dates back to 1864 when Tresca published his yield criterion based on his experimental results on punching and extrusion. Since then, tremendous progress has been made by many researchers, such as Saint-Venant, Levy, Von Mises, Hecky, Prandtl, and Taylor, who have established the cornerstones for the theory. Now developments in plasticity theory is an active field of mechanics.

The theories of plasticity can be established into two categories: One group is known as mathematical theories of plasticity, and the other is physical theories of plasticity. Mathematical theories are formulated to represent the experimental observations as general mathematical formulations. The physical theories, on the other hand, attempt to quantify plastic deformation at the microscopic level and explain why and how the plastic deformation occurs.

Introduction of the Numerical Analysis into Plasticity: "Computational Plasticity"

Plasticity in the field of mechanics has evolved due to the need to accurately capture the real behavior of materials. Certain assumptions had to be made for the mathematical modeling of these materials. In the early phases, only the "continuum" forms of the models were used to solve the problems. The term continuum can be attributed for no numerical approximation.

With the introduction of numerical analysis in the field of mechanics, a huge window for scientists and engineers has opened up.

In numerical analysis, the algorithmic model is an approximation to the continuum model in the sense that when time step taken to advance the solution tends to zero, we should recover the continuum. So any other "finite" sized time step Δt has an effect to all of the mathematical model. This can simply be explained through the inclusion of "Δ" or differential operators instead of time derivatives of the mathematical models. This operator will create an extra dependency of our models with respect to step sizes.

$$\begin{gathered} e.g. \\ \dot{y}(t) = f(y,t) \rightarrow Continuum \\ y(t_{n+1}) = y(t_n) + \Delta t \cdot f(y(t_{n+1}), t_{n+1}) \rightarrow Numerical \end{gathered} \tag{4}$$

Remark: References 1& 2 has been used to compile this introduction chapter.

2. MOTIVATION: One-Dimensional Plasticity

To motivate the mathematical structure of classical rate-independent plasticity for three dimensional plasticity, we examine the mechanical response of the one dimensional frictional device.

This device has a unit length(and unit area) and consists of a spring, with elastic constant E, and a Coulomb friction element, with constant $\sigma_y > 0$. We let σ be the applied stress(force) and ε total strain(change in length) in the device.

Local Governing Equations

a) The total strain ε splits into a part ε^e on the spring with constant E, referred to as the elastic part, and a strain ε^p on the friction device referred to as the plastic part.

$$\varepsilon = \varepsilon^e + \varepsilon^p \tag{5}$$

b) By equilibrium considerations, the stress on the spring with constant E is σ, and we have the elastic relationship

$$\sigma = E\varepsilon^e = E\left(\varepsilon - \varepsilon^p\right) \tag{6}$$

Irreversible Frictional Response

Assume that ε,ε^p are functions of time in an interval $[0,T] \subset \mathbb{R}$. In particular, we let

$$\varepsilon^p : [0,T] \to \mathbb{R}$$

and

$$\dot{\varepsilon}^p = \frac{\partial}{\partial t}\varepsilon^p$$

Change in the configuration of the frictional device is possible only if $\dot{\varepsilon}^p \neq 0$, to characterize this change:

We make the physical assumptions:

1) The stress σ in the frictional device can not be greater in absolute value than $\sigma_y > 0$. This means that the admissible stresses are constrained to lie in the closed interval $[-\sigma_y, \sigma_y] \subset \mathbb{R}$. We introduce

$$E_\sigma = \left\{\sigma \subset \mathbb{R} \mid f(\sigma) := \left|\sigma\right| - \sigma_y \leq 0\right\}$$

to designate the set of admissible stresses. σ_y is denoted as flow stress of the friction device. The function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(\sigma) := \left|\sigma\right| - \sigma_y$$

is referred to as yield function.

2) If the absolute value σ of the applied stress is less than the flow stress σ_y, no change in ε^p takes place, i.e., $\dot{\varepsilon}^p = 0$. This condition implies

$$\dot{\varepsilon}^p = 0 \text{ if } f(\sigma) := |\sigma| - \sigma_y < 0$$

(Fill in other equations)

and the instantaneous response of the device if elastic with spring constant E. This motivates the denomination of elastic range given to the open set

$$\text{int}(E_\sigma) = \{\sigma \in \mathbb{R} \mid f(\sigma) := |\sigma| - \sigma_y < 0\}$$

3) A change in ε^p can take place only if

$$f(\sigma) = |\sigma| - \sigma_y = 0$$

If the latter condition is met, the frictional device experiences slip in the direction of the applied stress σ , with constant slip rate. Let $\gamma \geq 0$ be the absolute value of the slip rate. Then the preceding physical assumption takes the form:

$$\dot{\varepsilon}^p = \gamma \geq 0 \ \text{ if } \sigma = \sigma_y > 0$$
$$\dot{\varepsilon}^p = -\gamma \leq 0 \ \text{ if } \sigma = -\sigma_y < 0$$

Whether $\gamma > 0$ or $\gamma = 0$ depends on further conditions involving the applied strain rate $\dot{\varepsilon}$,which are discussed below and are referred as loading/unloading conditions. For now (Eq.7) can be recast into the following single equation.

$$\dot{\varepsilon}^p = \gamma \cdot sign(\sigma) \ \text{ iff } f(\sigma) := |\sigma| - \sigma_y = 0$$

where $\gamma \geq 0$.

The boundary ∂E_σ of the convex set E_σ defined by

$$\partial E_\sigma = \{\sigma \in \mathbb{R} \mid f(\sigma) = |\sigma| - \sigma_y = 0\}$$

is called the yield surface.

In the present one dimensional model, the yield surface reduces to two points.

To complete the description of the model at hand, it remains only to determine the slip rate $\gamma \geq 0$. This involves the following essential conditions that embody the notion of irreversibility.

Loading/Unloading Conditions:
The evolution of $\varepsilon^p : [0,T] \rightarrow \mathbb{R}$ can be completely described for any admissible stress state $\sigma \in E_\sigma$ with the single equation

$$\dot{\varepsilon}^p = \gamma \cdot sign(\sigma)$$

provided that γ and σ are restricted by certain unilateral constraints.

First, we note that σ must be admissible, i.e., $\sigma \in \partial E_\sigma$ by assumption 1, and γ must be nonnegative by assumption 3.

$$\gamma \geq 0$$
$$and$$
$$f(\sigma) \leq 0$$

Second, by assumption 2, $\gamma = 0$ if $f(\sigma) < 0$. On the other hand, by assumption 3, $\dot{\varepsilon}^p \neq 0$,and, therefore, $\gamma > 0$ only if $f(\sigma) = 0$.These observations imply the conditions

$$f(\sigma) < 0 \Rightarrow \gamma = 0$$
$$\gamma > 0 \Rightarrow f(\sigma) = 0$$

It follows that we require
$\gamma f(\sigma) = 0$

Conditions described in (Eq,….) express the physical requirement that the stress must be admissible and that the plastic flow, in the sense of nonzero frictional strain $\dot{\varepsilon}^p \neq 0$, can take place only on the yield surface ∂E_σ. These conditions are known as Kuhn-Tucker conditions.

The last condition will enable us to determine the actual value of $\gamma \geq 0$ at any given time and is referred to as the consistency requirement. We need further observation to come with the formulation for this.

Let $\left\{\varepsilon(t), \varepsilon^p(t)\right\}$ be given at time $t \in \left[0, T\right]$, so that $\sigma(t)$ is also known at time t by;

$$\sigma = E\varepsilon^e = E(\varepsilon - \varepsilon^p)$$

Assume that we prescribe the total strain rate $\dot{\varepsilon}(t)$ at time t. Further, consider the case where

$$\sigma(t) \in \partial E_\sigma \Leftrightarrow \hat{f}(t) := f(\sigma(t)) = 0$$

at time t.

Then, it is easily shown that $\dot{\hat{f}} \leq 0$, since should $\dot{\hat{f}}$ be positive it would imply that $\hat{f}(t + \Delta t) > 0$ for some $\Delta t > 0$,which violates the admissibility condition $f \leq 0$.

Further, we specify that

$\gamma > 0$ only if $\dot{\hat{f}}(t) = 0$, and

set $\gamma = 0$ if $\dot{\hat{f}}(t) = 0$, and

$\gamma = 0$ if $\dot{\hat{f}}(t) < 0$

that is , dropping the hat to simplify notation, we set:

$\gamma > 0 \Rightarrow \dot{f} = 0$

$\dot{f} < 0 \Rightarrow \gamma = 0$

Therefore, we have the additional condition

$\gamma \dot{f}(\sigma) = 0$

This equation is referred to as the persistency (consistency) condition, and corresponds to the physical requirement that for $\dot{\varepsilon}^p$ to be nonzero (i.e. $\gamma > 0$) the stress point $\sigma \in \partial E_\sigma$ must "persist" on ∂E_σ so that $\dot{f}\left[\sigma(t)\right] = 0$

Frictional Slip

For the physical condition at hand, we can write

$$f = f(\sigma) \Rightarrow \dot{f} = \frac{\partial f}{\partial \sigma}\dot{\sigma} = \frac{\partial f}{\partial \sigma}E(\dot{\varepsilon} - \dot{\varepsilon}^p) = \frac{\partial f}{\partial \sigma}E\dot{\varepsilon} - \gamma\frac{\partial f}{\partial \sigma}Esign(\sigma)$$

$$f := |\sigma| - \sigma_y \Rightarrow \frac{\partial f}{\partial \sigma} = sign(\sigma)$$

$$\gamma\dot{f} = 0 \Rightarrow \gamma E(\dot{\varepsilon}sign(\sigma) - \gamma) = 0$$

$$\gamma_1 = 0, \gamma_2 = \dot{\varepsilon}sign(\sigma)$$

Now, let's observe these two solutions:

For $\gamma = \gamma_1 = 0$

$\dot{\varepsilon}^p = 0 \Rightarrow$ no additional plastic strain occurs, unloading

For $\gamma = \gamma_2 = \dot{\varepsilon}sign(\sigma)$

$\dot{\varepsilon}^p = \gamma sign(\sigma) \Rightarrow$ Plastic Loading occurs

A summary of the model is given below.

i) Elastic stress-strain relationship

$\sigma = E(\varepsilon - \varepsilon^p)$

ii) Flow rule

$\dot{\varepsilon}^p = \gamma sign(\sigma)$

iii) Yield condition

$f(\sigma) := |\sigma| - \sigma_y \le 0$

iv) Kuhn-Tucker complementary condition

$\gamma \ge 0, f(\sigma) \le =, \gamma f(\sigma) = 0$

v) Consistency condition

$\gamma\dot{f}(\sigma) = 0 \text{ (if } f(\sigma) = 0)$

This model is called perfect plasticity(rate independent).

Remark: The flow rule is related to yield condition through the potential relationship.

$$\dot{\varepsilon}^p = \gamma\frac{\partial f}{\partial \sigma}$$

where $\frac{\partial f}{\partial \sigma} = sign(\sigma)$ for our simple 1-d model.

A simple extension of the model(Linear Isotropic Hardening Plasticity)

We will simply summarize the equations related to this model:

i) Elastic stress-strain relationship

$\sigma = E(\varepsilon - \varepsilon^{p})$

ii) Flow rule

$\dot{\varepsilon}^{p} = \gamma sign(\sigma)$

$\dot{\alpha} = \gamma$

iii) Yield condition

$f(\sigma,\alpha) := |\sigma| - (\sigma_{y} + K\alpha) \leq 0$

α : strain like internal variable

iv) Kuhn-Tucker complementary condition

$\gamma \geq 0, f(\sigma,\alpha) \leq=, \gamma f(\sigma,\alpha) = 0$

v) Consistency condition

$\gamma \dot{f}(\sigma,\alpha) = 0$ (if $f(\sigma,\alpha) = 0$)

3. GENERAL PLASTICITY MODELS & INTEGRATION OF THESE MODELS

Numerical solution of nonlinear BVP in solid mechanics is based on an iterative solution of a discretized version of the momentum balance equations. Typically, the following steps are involved:

Step 1) The discretized momentum equations generate incremental motions which, in turn, are used to calculate the incremental strain history by kinematic relationships.

Step 2) For a given incremental strain history, new values of the state variables {σ,ε^p,q} are obtained by integrating the local constitutive equations with given initial conditions.

$\boldsymbol{\sigma}$: Stress Tensor

$\boldsymbol{\varepsilon}^p$: Plastic Strain Tensor

q : Internal parameters

Step 3) The (discrete) momentum balance equations is tested for the computed stresses and, if violated, the iteration is continued by returning to Step 1.

Computational Architecture: Steps 1&3 are carried out at a global level by finite-element/finite-difference procedures. Step 2 is regarded as the central problem of computational plasticity. From a computational standpoint, Step 2 is strain-driven in the sense that the state variables are computed for a given deformation history.

- ➢ REMARK: In this work only Local Problem (Step 2) is dealt with.

Basic Algorithmic Setup. Strain Driven Problem

Let $[0,T] \subset \mathbb{R}$ be the time interval of interest. At time $t \in [0,T]$, we assume that the total and plastic strain fields and the internal variables are known, that is

$$\left\{\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_n^p, q_n\right\}$$

are given data at t_n and

$$\boldsymbol{\varepsilon}_n^e = \boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_n^p \text{ , elastic strain tensor}$$

$$\boldsymbol{\sigma}_n = \nabla W(\boldsymbol{\varepsilon}_n^e) \text{ ,where W is stored energy function.}$$

Let Δu be the incremental displacement field, which is assumed to be given. The dimension of the problem is three.

So the basic problem is to update the fields to t_{n+1} in a manner consistent with the elastoplastic constitutive equations.

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}} &= \nabla^s(\Delta\dot{\mathbf{u}}) \\ \dot{\boldsymbol{\varepsilon}}^p &= \gamma \mathbf{r}(\boldsymbol{\sigma}, q) \\ \dot{q} &= -\gamma \mathbf{h}(\boldsymbol{\sigma}, q) \end{aligned} \tag{7}$$

∇^s : symmetric gradient operator subject to constraints(known as Kuhn-Tucker conditions)

$$f(\boldsymbol{\sigma}, q) \leq 0$$

$$\gamma \geq 0$$

$$\gamma f(\boldsymbol{\sigma}, q) = 0$$

where $f(\boldsymbol{\sigma}, q)$: yield function that defines the admissible state of stresses

and the initial conditions are

$$\left\{\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^p, q\right\}|_{t=t_n} = \left\{\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_n^p, q_n\right\}$$

CASE: Associative Flow Rule

The principle of maximum plastic dissipation leads to simplifications on the development of state variables.

a. associativity of flow rule in stress space according to the relationship
b.

$$\dot{\boldsymbol{\varepsilon}}^p = \gamma \frac{\partial f(\boldsymbol{\sigma}, q)}{\partial \boldsymbol{\sigma}}$$

$$\text{where } \boldsymbol{\sigma} = \nabla W(\boldsymbol{\varepsilon}^e)$$

c. associativity of the hardening law in stress space in the sense that

$$\dot{\alpha} = \gamma \frac{\partial f(\boldsymbol{\sigma}, q)}{\partial q}$$

where $q = -\nabla H(\alpha)$, H is a potential function

Therefore we can use associativity to state the problem.

$$\dot{q} = -\nabla^2 H(\alpha)\dot{\alpha}$$

$$\text{where } \dot{\alpha} = \gamma \frac{\partial f(\boldsymbol{\sigma}, q)}{\partial q}$$

$$D := \nabla^2 H(\alpha)$$

With these notations, the associative version of Plasticity equations become

$$\dot{\boldsymbol{\varepsilon}} = \nabla^s(\Delta \dot{\mathbf{u}})$$

$$\dot{\boldsymbol{\varepsilon}}^p = \gamma \frac{\partial f(\boldsymbol{\sigma}, q)}{\partial \boldsymbol{\sigma}}$$

$$\dot{\alpha} = \gamma \frac{\partial f(\boldsymbol{\sigma}, q)}{\partial q}$$

Before we proceed with the integration of these algorithms, let's first write down a simple but yet very effective method for the solution of Initial Value Problem for ODEs.

The Generalized Midpoint Rule:
Let $f:\mathbb{R}\to\mathbb{R}$ be a smooth function, and consider the IVP

$$\begin{gathered}\dot{x}(t)=f(x(t))\\ x(0)=x_n\\ \text{in } [t_n,t_{n+1}]\\ \Downarrow\\ x_{n+1}=x_n+\Delta t\cdot f(x_{n+\theta})\\ x_{n+\theta}=\theta x_{n+1}+(1-\theta)x_n\\ \theta\in[0,1]\end{gathered}$$

Here $x_{n+1}\cong x(t_{n+1})$ denotes the algorithmic approximation to the exact value $x(t_{n+1})$ at time $t_{n+1}=t_n+\Delta t.$

$$\begin{gathered}\theta=0\Rightarrow \text{ Forward (explicit) Euler}\\ \theta=1/2\Rightarrow \text{Midpoint rule}\\ \theta{=}1\Rightarrow \text{Backward (implicit) Euler}\end{gathered}$$

Second order accuracy is obtained only for $\theta=1/2$ whereas unconditional (linearized) stability requires $\theta\geq 1/2.$ For a more through explanation of accuracy and stability of these integration algorithms, the reader can skip to Chapter 5).

Let's Start INTEGRATING then:

The integration of the constitutive law with an implicit rule results in a nonlinear problem. If the backward Euler $(\theta=1)$ rule is used, the equations can be written:

$$\begin{aligned}\boldsymbol{\sigma}_{n+1}&=\nabla W(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}^p_{n+1})=\nabla W\left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}^p_n-\Delta\gamma_{n+1}\frac{\partial f}{\partial\boldsymbol{\sigma}}\right)\\ q_{n+1}&=-\nabla H(\alpha_{n+1})=-\nabla H\left(\alpha_n+\Delta\gamma_{n+1}\frac{\partial f}{\partial q}\right)\\ f_{n+1}&=f(\boldsymbol{\sigma}_{n+1},q_{n+1})=0\end{aligned}\tag{8}$$

For very simple plasticity models, these equations can be effectively solved and coded into the computer in closed form.

- One can find the consistency parameter in closed form and insert into above equations. The most popular of this kind of a solution is Radial Return Mapping Algorithm (Ref .4)):

$$\begin{gathered}f(\boldsymbol{\sigma},\alpha)=\|\mathbf{s}:\mathbf{s}\|+\sqrt{\frac{2}{3}}\cdot q(\alpha)\\ \mathbf{s}=\mathbf{P}^{\mathbf{dev}}:\boldsymbol{\sigma}\\ q(\alpha)=\sigma_y+H_i\alpha\end{gathered}$$

Radial Return Mapping Algorithm:

1) Compute trial elastic stress

$$\mathbf{e}_{\mathrm{n+1}} = \boldsymbol{\varepsilon}_{n+1} - \frac{1}{3}(tr\left[\boldsymbol{\varepsilon}_{n+1}\right])\mathbf{I}$$

$$\mathbf{s}_{n+1}^{trial} = 2G(\mathbf{e}_{n+1} - \mathbf{e}_{n}^{p})$$

2) Check yield condition

$$f_{n+1}^{trial} = \left\|\mathbf{s}_{n+1}^{trial}\right\| - \sqrt{\frac{2}{3}}K(\alpha_n)$$

IF$f_{n+1}^{trial} \leq 0$ THEN:

SET $(\bullet)_{n+1} = (\bullet)_{n+1}^{trial}$ *& EXIT*

ENDIF

3) Compute n_{n+1} and find $\Delta\gamma$

$$\mathbf{n}_{n+1} = \mathbf{n}_{n+1}^{trial} = \frac{\mathbf{s}_{n+1}^{trial}}{\left\|\mathbf{s}_{n+1}^{trial}\right\|}$$

$$\Delta\gamma = \frac{f_{n+1}^{trial}}{2\left(G + \frac{1}{3}H_i\right)}$$

$$\alpha_{n+1} = \alpha_n + \sqrt{\frac{2}{3}}\Delta\gamma$$

4) Update stresses and plastic strains

$$\mathbf{e}_{n+1}^{p} = \mathbf{e}_{n}^{p} + \Delta\gamma\mathbf{n}_{n+1}$$

$$\boldsymbol{\sigma}_{n+1} = K \cdot tr\left[\boldsymbol{\varepsilon}_{n+1}\right]\mathbf{I} + \mathbf{s}_{n+1}^{trial} - 2G\Delta\gamma\mathbf{n}_{n+1}$$

5) Compute consistent elastoplastic tangent moduli

$$\mathbf{C}^{\mathrm{ep}} = K \cdot \left(\mathbf{I} \otimes \mathbf{I}\right) + 2G \cdot \theta_{n+1}\left[\mathbf{II} - \frac{1}{3}\mathbf{I} \otimes \mathbf{I}\right] - 2G \cdot \bar{\theta}_{n+1}\mathbf{n}_{n+1} \otimes \mathbf{n}_{n+1}$$

$$\theta_{n+1} = 1 - \frac{2G\Delta\gamma}{\left\|\mathbf{s}_{n+1}^{trial}\right\|}$$

$$\bar{\theta}_{n+1} = \frac{1}{1 + \frac{H_i}{3G}} - (1 - \theta_{n+1})$$

- In this work, we are going to follow to construct a numerical bases for the solution of any type of Plasticity Model (as long as it is smooth, uniquely differentiable at any point on the yield surface).

Due to the non-linearity of our material relations, we need to iterate on our initial assumptions till an acceptable defined tolerance on the residual is reached. The Newton-Raphson method is maybe one the fastest converging numerical scheme for iterating. It converges quadratically as long as the initial solution is inside the "zone of attraction". We will use this method in this work.

There are modified versions of Newton-Raphson Scheme that gives us a trade-off between speed, accuracy, storage and amount of calculations that may be performed for iterations. Modified Newton-Raphson methods can be accelerated by “line search” techniques and this technique may even help to increase the radius of attraction for normal Newton-Raphson procedure for general problems (Ref. 3).

Due to convex nature of the yield surface, our problem at hand is theoretically proven to converge. One can simply appreciate this due to Dissipative nature of the mathematical formulation. The material model does not create energy, and this helps us to converge. One may try to use above techniques depending on their own considerations. In this work, we will work with full Newton-Raphson type of iterations.
Here is the plastic correction part of the General Closest Point Projection Method:

- For the plastic corrector step(i.e. $f_{n+1}^{trial} > 0$),define residuals

$$\mathbf{r}_{\sigma} = \boldsymbol{\sigma}_{n+1} + \Delta\gamma \cdot \mathbf{C}\frac{\partial f}{\partial \boldsymbol{\sigma}} - \boldsymbol{\sigma}^{\textbf{trial}}$$

$$r_{\alpha} = \alpha_{n+1} - \Delta\gamma \cdot \frac{\partial f}{\partial q} - \alpha_{n}$$

$$r_{\Delta\gamma} = f_{n+1} = f(\boldsymbol{\sigma}_{n+1}, q_{n+1})$$

$$q_{n+1} = -\nabla H(\alpha_{n+1})$$

- We want

$$\mathbf{r}(\mathbf{x}) := \begin{Bmatrix} r_{\sigma} \\ r_{\alpha} \\ r_{\Delta\gamma} \end{Bmatrix} \quad \mathit{for} \quad \mathbf{x} := \begin{Bmatrix} \boldsymbol{\sigma}_{n+1} \\ \alpha_{n+1} \\ \Delta\gamma \end{Bmatrix}$$

Consider Newton-Raphson Scheme to satisfy r(x)=0 around $x^{(k)}$

$$\mathbf{r}(\mathbf{x}) = \mathbf{r}(\mathbf{x}^{(k)}) + \frac{\partial \mathbf{r}}{\partial \mathbf{x}} \cdot (\mathbf{x} - \mathbf{x}^{(k)}) + O(h^2) = 0$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + d\mathbf{x} \rightarrow d\mathbf{x} = -\mathbf{J}^{-1}\mathbf{r}(\mathbf{x}^{(k)})$$

$$\mathbf{J} = \begin{pmatrix} 1 + \Delta\gamma\mathbf{C}\dfrac{\partial^2 f}{\partial \boldsymbol{\sigma}^2} & -\Delta\gamma H \cdot \mathbf{C}\dfrac{\partial^2 f}{\partial q \partial \boldsymbol{\sigma}} & \mathbf{C}\dfrac{\partial f}{\partial \boldsymbol{\sigma}} \\ -\Delta\gamma\dfrac{\partial^2 f}{\partial \boldsymbol{\sigma} \partial q} & 1 + \Delta\gamma H\dfrac{\partial^2 f}{\partial q^2} & -\dfrac{\partial f}{\partial q} \\ \dfrac{\partial f}{\partial \boldsymbol{\sigma}} & -H\dfrac{\partial f}{\partial q} & 0 \end{pmatrix}$$

- Iterate till $\left\| \mathbf{r}(\mathbf{x}^{(k)}) \right\| < TOL$
- Calculate consistent elastoplastic tangent

$$\mathbf{C}^{ep} = \mathbf{P}^{\mathbf{T}}\mathbf{J}^{\mathbf{-1}}\mathbf{PC}$$

where $\mathbf{P}^{T} = \left[\mathbf{I}_{\dim(\boldsymbol{\sigma})}, \mathbf{0}_{\dim(q)+1} \right]$

As stated earlier partial derivatives of the yield surface with respect to stresses might become very cumbersome to get in closed form. Algebraic manipulators such as Maple or Mathematica can be a very effective tool for obtaining analytical derivatives. However even these software programs may not achieve our purpose (Ref.7).

Consistent Tangent calculation expressed above is obtained by linearizing the equations to obtain the relation between $d\boldsymbol{\sigma}_{n+1} = \mathbf{C}^{ep} d\boldsymbol{\varepsilon}_{n+1}$. A simple way of doing it is used in this work (Ref.11).

To achieve quadratic convergence at the local and global levels, we need to obtain partial derivatives of yield surface correct enough. For this purpose, numerical differentiation is used to explore accuracy of the results for simple material models. For more complex models, one should look at Ref.7.

Usually first partial derivatives of the yield surface are relatively much easier to obtain analytically than second partial derivatives, so to decrease the number of numerical differentiation, we will use these first derivatives to calculate second partial derivatives.

Numerical Differentiation

Let's review some of the widely used numerical differentiation formulas, following with an unconventional one (Ref. 5).

- Forward-difference

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} f''(\xi) \quad ,\xi \in [x_0, x_0+h]$$

- Central-difference

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} - \frac{h^2}{6} f'''(\xi) \quad ,\xi \in [x_0-h, x_0+h]$$

- Complex-difference(Ref.10)

$$\mathrm{f}'(\mathrm{x}_0) = \frac{\mathrm{Im}(f(x_0+ih))}{h} + h^2 f'''(x_0)/3! + \cdots$$

A particularly important subject in the study of numerical differentiation is the effect of round-off errors. Looking at central-difference scheme more closely:

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} - \frac{h^2}{6} f'''(\xi)$$

Suppose that in evaluating $f(x_0+h)$ *and* $f(x_0-h)$ we encounter round-off errors $e(x_0+h)$ *and* $e(x_0-h)$. Then our computed values $\tilde{f}(x_0+h)$ *and* $\tilde{f}(x_0-h)$ are related to the true values by

$$f(x_0+h) = \tilde{f}(x_0+h) + e(x_0+h)$$

$$f(x_0-h) = \tilde{f}(x_0-h) + e(x_0-h)$$

The total error in the approximation,

$$f'(x_0) - \frac{f(x_0+h) - f(x_0-h)}{2h} = \frac{e(x_0+h) - e(x_0-h)}{2h} - \frac{h^2}{6} f'''(\xi)$$

will have a part due to round-off error and a part due to truncation error. If we assume that the round-off errors $e(x_0 \pm h)$ are bounded by some number $\varepsilon > 0$ and that the third derivative of f is bounded by a number M>0, then

$$\left| f'(x_0) - \frac{\tilde{f}(x_0+h) - \tilde{f}(x_0-h)}{2h} \right| \le \frac{\varepsilon}{h} + \frac{h^2}{6} M$$

To reduce truncation error, $h^2 M / 6$, we must reduce h. But as h is reduced, the round-off error ε / h grows. There are certain techniques to find an optimum h and we will briefly go over them.

Error Analysis

There is an optimal step size h^{opt} that minimizes the summation of both round-off and truncation error. The optimal step size can be written as:

$$h^{opt} = h_r^{opt} \max\{|x|, typ_x\}$$

where h_r^{opt} is the optimal step size and typ_x is a typical calue of x used to avoid choosing a null(or extremely small) h^{opt} for null (or extremely small) x. Numerical experimentation shows that typ_x can be chosen in a rather arbitrary manner, because it has a very small influence on the results(select typ_x=1). The main idea is that h_r^{opt} is independent of x. This means that a constant value of h_r^{opt} can be used all over the domain, for every load step, and all the stress components.

In general, h_r^{opt} is to be computed from the minimization of round-off and truncation errors and this is hard to calculate. Instead of that, here are some simplified values from literature:

$$1ND - O(h) : h_r^{opt} = \sqrt{\eta}$$

$$1ND - O(h^2) : h_r^{opt} = \sqrt[3]{\eta}$$

and η is the accuracy in the evaluation of f.

4. NUMERICAL SIMULATIONS

Computer Codes to compare Numerical Solution Methods

I will present the computer codes for J2 Plasticity Model written in two different formats, i.e. Radial Return Map(RRM) and General Closest Point Projection Method(GCPPM).

RRM is actually the name given for the algorithm that integrates J2 Plasticity in closed form with Backward Euler Method.

GCPPM is a main frame algorithm that will integrate any smooth plasticity model. When this method is asked to integrate J2 Plasticity, this should simply give the same result as RRM.

Radial Return Map for this material model:

```
function varargout = RRM(action,Mat_no,MatData,State,varargin)
% RRM 3-D J-2 PLASTICITY MATERIAL (full stress and strain components)
% Von Mises (J2) plasticity
% 3-D plasticity with Linear Isotropic and Kinematic Hardening
% function contributed by Afsin Saritas

global IOW;       % output file number
global HEAD_PR;   % header print indicator

switch action
case 'chec'
  % not implemented yet
  varargout = {MatData};
otherwise
  % extract material properties
  E      = MatData.E; % elastic modulus
  nu     = MatData.nu; % Poisson's Ratio
  Hi     = MatData.Hi; % Linear Isotropic Hardening
  Hk     = MatData.Hk; % Kinematic Hardening Constant
  sigmay = MatData.sigmay;
  % calculate the variables needed
  G = E/2/(1+nu);
  K = E/3/(1-2*nu);
  % define some useful matrices used in this function
  I2  = [1 1 1 0 0 0]';   % 2nd order identity
  m   = [1 1 1 0 0 0 0 0 0]';
  I4  = eye(6);           % 4th order identity
  Id  = I4 - I2*I2'./3;   % deviatoric projection
  I0  = eye(6); I0(4:6,4:6) = 0.5*eye(3);
  P   = zeros(6,9);           %##################################################
  P(1:3,1:3)=eye(3);          % P is used for the transformation between        %
  j=4;                        % the nine and six component forms of strain      %
  for i=4:2:8                 % and stresses.                                   %
    P(j,i:i+1)=[0.5 0.5];     % Six component expressions will permit final     %
    j=j+1;                    % expressions for strain and equilibrium to be    %
  end                         % written in terms of B for a general FEM Program %
  P = P';                     %##################################################
end
% material actions
% ======================================================================
switch action
  % implement this later
% ======================================================================
case 'init'
  % response history variables
  sig = zeros(6,1);
  State.sig = sig;
  Ct  = K.*(I2*I2') + (2*G).*(I0-I2*I2'./3);
  State.Ct = Ct;
  State.Pres.sig   = sig;
  State.Pres.Ct    = Ct;   % Elastic tangent moduli
  State.Pres.alpha = 0;    % equivalent plastic strain for isotropic hardening
  State.Pres.epsP  = zeros(6,1);    % plastic strains
  State.Pres.q     = zeros(9,1);    % initiliaze backstress in 9 comp (only affects deviatoric stress)
  varargout = {State};
% ======================================================================
case{'stif','forc'}
  % Retrieve history variables from Past
  ePn    = P*State.Past.epsP ; % plastic strain from last converged (no volumetric plastic strain)
  qn     = State.Past.q;     % back stress from last converged( already in 9 component form)
  alphan = State.Past.alpha; % strain hardening from last converged

  % update Strain Tensor. Compute Trial Elastic Stresses
  en1    = P*(Id*State.eps(:,1));  % total deviatoric strain at n+1
  epsV   = I2'*State.eps(:,1);     % volumetric strain at n+1
  sn1Trial   = 2*G * ( en1 - ePn);   % trial deviatoric stress
  ksin1Trial = sn1Trial - qn;      % subtract deviatoric backstress

  % check yield condition
  norm_ksin1Trial = sqrt(ksin1Trial'*ksin1Trial);
  Yn = sigmay + Hi*alphan;
  fn1Trial = norm_ksin1Trial - sqrt(2/3)*Yn;

  if fn1Trial<=0
    % elastic step
    sn1  = sn1Trial;
    ePn1 = ePn;
    alphan1 = alphan;
    qn1 = qn;
    % algorithmic elastoplastic tangent
    Ct  = K.*(I2*I2') + (2*G).*(I0-I2*I2'./3);
  else
    % plastic step
    % 3) compute nn1 and dgamma
    Gp = G + (Hi+Hk)/3;
    dgamma = fn1Trial/(2*Gp);  % dgamma is found in closed form
    nn1    = ksin1Trial ./norm_ksin1Trial;  % normal to the yield surface
    % update stress,strain
    alphan1 = alphan + sqrt(2/3)*dgamma;
    ePn1    = ePn + dgamma*nn1;
    ksin1   = ksin1Trial - (2*G+2/3*Hk)*dgamma.*nn1;
    qn1     = qn + 2/3*Hk*dgamma.*nn1;
    sn1     = ksin1+qn1;
    % 4) calculate algorithmic elastoplastic tangent
    theta = 2*G*dgamma/norm_ksin1Trial;
    nn1   = P'*nn1;  % project normal to 6 component
    Ct   = K .*(I2*I2') + (2*G).*((1-theta).*(I0-I2*I2'./3) - (G/Gp-theta).*(nn1*nn1'));
  end

  % update total stress and total plastic strain values and project to 6 component form
  sign1 = P'*(sn1 + (K*epsV).*m);
  epsPn1= I0 \ (P'*ePn1);

  % update values in
  State.sig = sign1;
  State.Pres.sig   = sign1;
  State.Pres.Ct    = Ct;
  State.Pres.alpha = alphan1;
  State.Pres.epsP  = epsPn1;
  State.Pres.q     = qn1;

  if action == 'stif'
    State.Ct = Ct;
  end
  varargout = {State};
% ======================================================================
otherwise
end
```

GCPPM code:

```
function varargout = GCPPM(action,Mat_no,MatData,State,varargin)
% General Closest Point Projection Algorithm
% uses linear hardening material (Yield function and its derivatives can be modified)
% Yield Surface :  f(sig,alpha) = PHI(sig) + q(alpha)
% this function can be modified to include Non-associative hardening very easily


global IOW;       % output file number
global HEAD_PR;   % header print indicator
TOL1 = 1e-12;
TOL2 = 1e-12;

switch action
case 'chec'
   % not implemented yet
   varargout = {MatData};
otherwise
   % extract material properties
   E      = MatData.E; % elastic modulus
   nu     = MatData.nu; % Poisson's Ratio
   % material parameters
   G = E/2/(1+nu);
   K = E/3/(1-2*nu);
   % define some useful matrices used in this function
   I2  = [1 1 1 0 0 0]';   % 2nd order identity
   m   = [1 1 1 0 0 0 0 0 0]';
   I4  = eye(6);           % 4th order identity
   Id  = I4 - I2*I2'./3;   % deviatoric projection
   I0  = eye(6); I0(4:6,4:6) = 0.5*eye(3);
   P   = zeros(6,9);           %##################################################
   P(1:3,1:3)=eye(3);          % P is used for the transformation between        %
   j=4;                        % the nine and six component forms of strain      %
   for i=4:2:8                 % and stresses.                                   %
      P(j,i:i+1)=[0.5 0.5];    % Six component expressions will permit final     %
      j=j+1;                   % expressions for strain and equilibrium to be    %
   end                         % written in terms of B for a general FEM Program %
   P = P';                     %##################################################

   Pvol = m*m'./3;             % volumetric projection in 9-d
   Pdev = eye(9)-Pvol;         % deviatoric projection in 9-d
   % calculate C
   C  = 3*K.*Pvol + (2*G).*Pdev;  % 9-d form
end

% material actions
% ========================================================================
switch action
case 'data'
   % implement this later

% ========================================================================
case 'init'
  % response history variables
  sig = zeros(6,1);
  Ct   = P'*C*P;
  State.sig = sig;
  State.Ct = Ct;  % project to 6-d
  State.Pres.sig   = sig;
  State.Pres.Ct    = Ct;              % Elastic tangent moduli
  State.Pres.alpha = 0;               % equivalent plastic strain for isotropic hardening
  State.Pres.epsP  = zeros(6,1);    % plastic strains
  varargout = {State};

% ========================================================================
case{'stif','forc'}

% retrieve history values
epsP_n  = P*State.Past.epsP;
alpha_n = State.Past.alpha;
% exract current strain
eps_n1  = P*State.eps;   % project to 9-d

% 1) Initiliaze General Closest Point Projection Iterations
epsP_n1  = epsP_n;     % project to 9-d
alpha_n1 = alpha_n;
```

```
% 2) Calculate values at t_n1
sig_n1Trial = C*(eps_n1 - epsP_n1);
sig_n1      = sig_n1Trial;
qPOT        = GradH(MatData,alpha_n1);
q_n1        = -qPOT.DH;
YieldFun    = Yield(sig_n1,q_n1,Pdev);
f_n1Trial   = YieldFun.f;
if f_n1Trial <=0
   %elastic step, exit with current values
   Ct     = P'*C*P;   % project to 6-d
else
   % plastic step, Start General Closest Projection Algorithm
   k = 0;
   Dgamma_n1 = 0;

   % calculate derivatives
   dfdsig   = YieldFun.dfdsig;
   d2fdsig2 = YieldFun.d2fdsig2;
   dfdq     = YieldFun.dfdq;
   H        = qPOT.D2H;   % Current Plastic Modulus

   r_sig    = zeros(9,1);
   r_alpha  = 0;
   r_Dgamma = f_n1Trial;
   R = [r_sig;r_alpha;r_Dgamma];

   while sqrt(R'*R)>TOL2
      %calculate components of the Jacobian
      J = [  (eye(9)+Dgamma_n1*C*d2fdsig2)   zeros(9,1)       C*dfdsig;
                   zeros(1,9)                  1             -dfdq ;
                     dfdsig'               -H*dfdq'            0    ];

      % find update values
      dr        = -J\R;
      dsig      = dr(1:9,1);
      dalpha    = dr(10,1);
      dDgamma   = dr(11,1);
      sig_n1    = sig_n1 + dsig;
      alpha_n1  = alpha_n1 + dalpha;
      Dgamma_n1 = Dgamma_n1 + dDgamma;

      % calculate new derivatives
      qPOT     = GradH(MatData,alpha_n1);
      q_n1     = -qPOT.DH;
      H        = qPOT.D2H;
      YieldFun = Yield(sig_n1,q_n1,Pdev);
      f_n1     = YieldFun.f;
      dfdsig   = YieldFun.dfdsig;
      d2fdsig2 = YieldFun.d2fdsig2;
      dfdq     = YieldFun.dfdq;

      % calculate residuals
      r_sig     = sig_n1 + Dgamma_n1*C*dfdsig - sig_n1Trial;
      r_alpha   = alpha_n1 - Dgamma_n1*dfdq - alpha_n;
      r_Dgamma  = f_n1;
      R = [r_sig;r_alpha;r_Dgamma];
   end
   epsP_n1 = epsP_n + Dgamma_n1*dfdsig;
   % calculate Elastoplastic tangent with Perez-Foguet Way
   Pr = [   eye(9)    zeros(9,2)]';  % Projection Matrix onto Stress Space
   Ct = (Pr'*(J\Pr))*C;
   Ct = P'*Ct*P;   %project to 6-d
   % calculate elastoplastic tangent with Simo-Hughes way
   AM = inv(C)+Dgamma_n1*d2fdsig2;
   AM = inv(AM);
   N_n1 = dfdsig'*AM*dfdsig;
   N_n1 = AM*dfdsig./N_n1;
   Ct_SH = AM - N_n1*N_n1';
end

sig_n1 = P'*sig_n1;         % project back to 6-d
epsP_n1 = I0\(P'*epsP_n1);  % project back to 6-d
% store output values
```

```
State.sig = sig_n1;
State.Ct  = Ct;
State.Pres.sig = sig_n1;
State.Pres.alpha = alpha_n1;
State.Pres.epsP = epsP_n1;
State.Pres.Ct = Ct;

varargout = {State};
end

return

%============================================================================
%============================================================================
function YF = Yield(sig,q,Pdev)
sqrt23 = sqrt(2/3);
s = Pdev*sig; % project to 9-d
norms = sqrt(s'*s);
f = norms + sqrt23*q;
if norms ==0
   norms = 1;
end
dfdsig = s./norms;
d2fdsig2 = (Pdev - dfdsig*dfdsig')./norms;
dfdq = sqrt23;
YF.f = f;
YF.dfdsig = dfdsig;
YF.d2fdsig2 = d2fdsig2;
YF.dfdq = dfdq;
YF.d2fdq2 = 0;
YF.d2fdsigdq = 0;

% derivatives with numerical differentiation
h = 1e-5;
k=length(sig);
imagine = sqrt(-1);
for i=1:k
   ei = zeros(9,1);
   ei(i) = 1;     % disturb i-th component
   YF1 = YieldSimple(sig+h*ei,q,Pdev);
   YF2 = YieldSimple(sig-h*ei,q,Pdev);
   Ndfdsig(i,1) = (YF1.f-YF2.f)/2/h;
   Nd2fdsig2(:,i) = (YF1.dfdsig-YF2.dfdsig)/2/h;
end

for i=1:k
   ei = zeros(9,1);
   ei(i) = 1;     % disturb i-th component
   YF1 = YieldSimple(sig+h*imagine*ei,q,Pdev);
   Nidfdsig(i,1) = imag(YF1.f)/h;
   Nid2fdsig2(:,i) = imag(YF1.dfdsig)/h;
end

YF1 = YieldSimple(sig,q+h,Pdev);
YF2 = YieldSimple(sig,q-h,Pdev);
Ndfdq = (YF1.f - YF2.f)/2/h
%+++++++ SUB ROUTINE for the SUB ROUTINE
function YF = YieldSimple(sig,q,Pdev)
sqrt23 = sqrt(2/3);
s = Pdev*sig; % project to 9-d
norms = sqrt(s'*s);
f = norms + sqrt23*q;
if norms ==0
   norms = 1;
end
dfdsig = s./norms;
YF.f = f;
YF.dfdsig = dfdsig;
return
%++++++++++++++++++++++++++++++++++++++++++
return
%==============================================================
%==============================================================
%-------------------------------
function qPOT = GradH(MatData,alpha)
sigY = MatData.sigY;
Hi= MatData.Hi;
qPOT.DH = sigY + Hi*alpha;
qPOT.D2H = Hi;
return
%--------------------------------
```

Saturation Hardening model (just change GradH to new one in GCPPM)

```
%-------------------------------
function qPOT = GradH(MatData,alpha)
sigY = MatData.sigY;
sigY_inf = MatData.sigY_inf;
delta = MatData.delta;
qPOT.DH = sigY+(sigY_inf-sigY)*(1-exp(-delta*alpha));
qPOT.D2H = (sigY_inf-sigY)*delta*exp(-delta*alpha);
return
%--------------------------------
```

4.2 Numerical Examples

Compare accuracy of GCPPM with RRM

Here is some portion of the input script that defines the material parameters. We are going to run a Isotropic Linear Hardening Case in RRM and GCPPM.

```
MatData.E = 10;
MatData.nu = 0.2;
MatData.sigmay = 20;
MatData.Hi = 2;
MatData.Hk = 0;

% define time and strain
tmax = 3;
dt = 0.05;
t = 0:dt:tmax;
k = length(t);
% create strain history to be applied
for i=1:k
   %epsilon(:,i) = (12*t(i)).*[1 1 1 0 0 0]' + (6*sin(t(i))).*[0 1 3/2 4 0 0]'; % engineering strains
   epsilon(:,i) = (6*sin(t(i))).*[0 0 0 1 0 0]'; % engineering strains
end
```

Results

```
At time t=  0.15 sec (RRM)
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
3.735953311840E+000  0.000000000000E+000  4.166666666667E+000

At time t=  0.15 sec (GCPPM)
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
3.735953311840E+000  0.000000000000E+000  4.166666666667E+000

At time t=  0.70 sec (RRM)
```

```
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
1.217575528685E+001  5.445133886728E-001  5.747126436782E-001

At time t=  0.70 sec (GCPPM)
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
1.217575528685E+001  5.445133886728E-001  5.747126436782E-001


At time t=  1.45 sec (RRM)
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
1.337746323099E+001  1.585222996225E+000  5.747126436782E-001

At time t=  1.45 sec (GCPPM)
---------------------------------------------------------
    sigma(4)              alpha         C1212
---------------------------------------------------------
1.337746323099E+001  1.585222996225E+000  5.747126436782E-001
```

So two methods give the same answer. However, there are a couple points that has been observed in the numerical runs.

The way consistent elastoplastic tangent calculated in GCPPM is slightly influenced by:

Ct = (Pr'*(J\Pr))*C; (Better way to do)

When instead of this
Ct = (Pr'*inv(J)*Pr)*C; (creates round off)

However these small errors do not create a problem, because the elastoplastic tangent disturbed with errors is still algorithmicly more consistent than any other tangent.

Numerical Assesment of the Local Convergence of GCPPM

TOL=1e-12
Performance of Newton Scheme in decreasing the residual for GCPPM

```
At time t=  0.15 sec
----------------------------------
    NormR
----------------------------------
0.000000000000E+000

At time t=  0.70 sec
----------------------------------
    NormR
----------------------------------
1.379964191411E+000
1.526079710569E-002
1.815880342093E-006
2.842176364046E-014

At time t=  1.45 sec
----------------------------------
    NormR
----------------------------------
2.567950570683E-001
```

```
2.406126075023E-004
2.087894301894E-010
3.552713678801E-015

At time t=  2.20 sec
----------------------------------
    NormR
----------------------------------
0.000000000000E+000

At time t=  2.70 sec
----------------------------------
    NormR
----------------------------------
0.000000000000E+000

At time t=  3.00 sec
----------------------------------
    NormR
----------------------------------
0.000000000000E+000
```

Graphs from Test Run on GCPPM are attached at the end of the report..

- Time step does not have so significant effect on stress response of the material in terms of stability of the solution, however the consistent tangent changes significantly. Cep(1,1,1,1) stays almost constant for very small time steps, but for pretty large time steps, dt=1, Cep deviates. However this does not mean that this Cep is wrong. It just shows us that this is the tangent that is consistent with the time step.
- We can clearly see the smooth hardening behavior of the material in stress-strain curves(Figure-4)
- GCPPM bases its algorithm on Implicit Euler. We can see the strong stable behavior of the response for varying time step changes.

Results from Numerical Differentiation

Looking at Figure-5, we get very good and expected results from Central Difference method. It has an optimum step size 10^{-5} which is comparable with

$$h_r^{opt} = \sqrt[3]{\eta} = \sqrt[3]{eps} = 6 \cdot 10^{-6}$$

Complex differentiation didn't give satisfactory results. However in Ref.6, this method is shown to behave as good as central difference and even better.

5. ACCURACY & STABILITY OF INTEGRATION ALGORITHMS FOR ELASTOPLASTIC CONSTITUTIVE RELATIONS (Ref. 14)

An acceptable integration algorithm for plasticity models has to satisfy three basic requirements a) consistency or first order accuracy; b) numerical stability; specific for this kind of problems c) incremental plastic consistency. A non-required but nevertheless desirable feature to add to the above list is: d) second order accuracy.

Conditions (a) and (b) are needed to obtain convergence of the numerical solution as the time step becomes very small. Condition (c) is the algorithmic counterpart of the plastic consistency condition and requires that states of stress computed from the algorithm be contained within the elastic domain.

Here are the equations that we are trying to solve:

$$\begin{aligned} \varepsilon_{ij} &= \varepsilon_{ij}^{e} + \varepsilon_{ij}^{p} \\ \sigma_{ij} &= C_{ijkl} : \varepsilon_{kl}^{e} \\ \dot{\varepsilon}_{ij}^{p} &= \gamma r_{ij}(\boldsymbol{\sigma},\mathbf{q}) \\ \dot{q}_i &= \gamma h_i(\boldsymbol{\sigma},\mathbf{q}) \end{aligned} \tag{9}$$

$$\begin{aligned} f(\boldsymbol{\sigma},\mathbf{q}) &\le 0 \\ \gamma &\ge 0 \\ f\gamma &= 0 \end{aligned} \tag{10}$$

For

$$\begin{aligned} f < 0 &\Rightarrow \gamma = 0(Elastic) \\ \gamma > 0 &\Rightarrow f = 0 \,\&\, \dot{f} = 0 \end{aligned}$$

$$\begin{aligned} \dot{f} &= \frac{\partial f}{\partial \sigma_{ij}}\dot{\sigma}_{ij} + \frac{\partial f}{\partial q_i}\dot{q}_i = \eta_{ij}\dot{\sigma}_{ij} + \xi_i\dot{q}_i = 0 \\ \eta_{ij}(\boldsymbol{\sigma},\mathbf{q}) &= \frac{\partial f}{\partial \sigma_{ij}} \\ \xi_i(\boldsymbol{\sigma},\mathbf{q}) &= \frac{\partial f}{\partial q_i} \end{aligned} \tag{11}$$

For a particular form of plasticity (associated plasticity), we have:

$$r_{ij} = \eta_{ij} \tag{12}$$

<u>*Generalized Trapezoidal Rule:*</u>

A class of algorithms for integration of Eq.9-11 can be written as:

$$\begin{aligned} \boldsymbol{\sigma}_{n+1} &= \mathbf{C} : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{p}) \\ \boldsymbol{\varepsilon}_{n+1}^{p} &= \boldsymbol{\varepsilon}_{n}^{p} + \Delta\gamma\left[(1-\theta)\mathbf{r}_n + \theta\mathbf{r}_{n+1}\right] \\ \mathbf{q}_{n+1} &= \mathbf{q}_n + \Delta\gamma\left[(1-\theta)\mathbf{h}_n + \theta\mathbf{h}_{n+1}\right] \\ f_{n+1} &= 0 \end{aligned} \tag{13}$$

The symbol $(:)$ signifies doubly contracted tensor product, e.g. $(C:\varepsilon)_{ij} = C_{ijkl}\varepsilon_{kl}$, $\Delta\gamma$ is incremental plastic consistency parameter, and θ ranges from 0 to 1.

In Eq.13, $\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_n^p, \boldsymbol{\sigma}_n, \mathbf{q}_n$ are the known strains, plastic strains, stress and plastic variables at time t_n, whereas $\boldsymbol{\varepsilon}_{n+1}^p, \boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}$ are the corresponding unknown variables at time t_{n+1}. The updated strains ε_{n+1} are assumed given and one writes

$$\begin{aligned} \mathbf{r}_n &= \mathbf{r}(\boldsymbol{\sigma}_n, \mathbf{q}_n); \mathbf{r}_{n+1} = \mathbf{r}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \\ \mathbf{h}_n &= \mathbf{h}(\boldsymbol{\sigma}_n, \mathbf{q}_n); \mathbf{h}_{n+1} = \mathbf{h}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \end{aligned} \tag{14}$$

Generalized Midpoint Rule:

$$\begin{aligned} \boldsymbol{\sigma}_{n+1} &= \mathbf{C} : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^p) \\ \boldsymbol{\varepsilon}_{n+1}^p &= \boldsymbol{\varepsilon}_n^p + \Delta\gamma \mathbf{r}_{n+\theta} \\ \mathbf{q}_{n+1} &= \mathbf{q}_n + \Delta\gamma \mathbf{h}_{n+\theta} \\ f_{n+1} &= 0 \end{aligned} \tag{15}$$

where

$$\begin{aligned} \mathbf{r}_{n+\theta} &= \mathbf{r}\left((1-\theta)\boldsymbol{\sigma}_n + \theta\boldsymbol{\sigma}_{n+1}, (1-\theta)\mathbf{q}_n + \theta\mathbf{q}_{n+1}\right) \\ \mathbf{h}_{n+\theta} &= \mathbf{h}\left((1-\theta)\boldsymbol{\sigma}_n + \theta\boldsymbol{\sigma}_{n+1}, (1-\theta)\mathbf{q}_n + \theta\mathbf{q}_{n+1}\right) \end{aligned} \tag{16}$$

Accuracy Analysis

As mentioned in Chapter 3, we are going to deal with Step 2 of the solution of the Boundary Value Problem that is tried to be solved within the context of Finite Element Method Analysis. As a result, the updated strains $\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}(t_{n+1}) = \boldsymbol{\varepsilon}(t_n + h)$ may be viewed as known functions of the step size h. The remaining update variables $\boldsymbol{\sigma}_{n+1}, \boldsymbol{\varepsilon}_{n+1}^p, \mathbf{q}_{n+1}$, as well as the incremental plastic parameter $\Delta\gamma$, also become functions of h implicitly defined through relations (Eq.13-16). It is clear from these relations that as $h \to 0$ and, hence, $\boldsymbol{\varepsilon}_{n+1} \to \boldsymbol{\varepsilon}_n$ the limiting values

$$\boldsymbol{\sigma}_{n+1} \to \boldsymbol{\sigma}_n; \boldsymbol{\varepsilon}_{n+1}^p \to \boldsymbol{\varepsilon}_n^p; \mathbf{q}_{n+1} \to \mathbf{q}_n; \Delta\gamma \to 0 \tag{17}$$

are obtained. Furthermore, by implicit function theorem $\boldsymbol{\sigma}_{n+1}, \boldsymbol{\varepsilon}_{n+1}^p, \mathbf{q}_{n+1}$ *and* $\Delta\gamma$ are differentiable functions of h provided the functions r,h and f are sufficiently smooth.

First-order accuracy or consistency of the algorithms Eq 13-16 with constitutive relations Eq.9 necessitates that the numerically integrated variables $\boldsymbol{\sigma}_{n+1}, \boldsymbol{\varepsilon}_{n+1}^p, \mathbf{q}_{n+1}$ agree with their exact values $\boldsymbol{\sigma}(t_{n+1}), \boldsymbol{\varepsilon}^p(t_{n+1}), \mathbf{q}(t_{n+1})$ to within second-order terms in h. An alternative statement requires that

$$\begin{aligned}
&\frac{d}{dh}(\boldsymbol{\sigma}_{n+1})_{h=0} = \dot{\boldsymbol{\sigma}}_n = \mathbf{C} : (\dot{\boldsymbol{\varepsilon}}_n - \dot{\boldsymbol{\varepsilon}}_n^p) \\
&\frac{d}{dh}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} = \dot{\boldsymbol{\varepsilon}}_n^p = \gamma_n \mathbf{r}_n \\
&\frac{d}{dh}(\mathbf{q}_{n+1})_{h=0} = \dot{\mathbf{q}}_n = \gamma_n \mathbf{h}_n \\
&\frac{d}{dh}(\Delta\gamma)_{h=0} = \gamma_n
\end{aligned} \tag{18}$$

where plastic consistency parameter γ_n is determined with the aid of the plastic consistency condition

$$\dot{f}_n = \boldsymbol{\eta}_n : \dot{\boldsymbol{\sigma}}_n + \boldsymbol{\xi}_n \cdot \dot{\mathbf{q}}_n \tag{19}$$

Let's investigate whether or not Eq.18-19 are satisfied by Generalized Trapezoidal rule and Generalized Midpoint Rule:

Generalized Trapezoidal Rule:

$$\begin{aligned}
&\frac{d}{dh}\boldsymbol{\sigma}_{n+1} = \mathbf{C} : \left(\frac{d}{dh}\boldsymbol{\varepsilon}_{n+1} - \frac{d}{dh}\boldsymbol{\varepsilon}_{n+1}^p \right) \\
&\frac{d}{dh}\boldsymbol{\varepsilon}_{n+1}^p = \frac{d(\Delta\gamma)}{dh}\left[(1-\theta)\mathbf{r}_n + \theta\mathbf{r}_{n+1}\right] + \theta \cdot \Delta\gamma \left[\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}}\right)_{n+1} : \frac{d}{dh}\boldsymbol{\sigma}_{n+1} + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}}\right)_{n+1} : \frac{d}{dh}\mathbf{q}_{n+1} \right] \\
&\frac{d}{dh}\mathbf{q}_{n+1} = \frac{d(\Delta\gamma)}{dh}\left[(1-\theta)\mathbf{h}_n + \theta\mathbf{h}_{n+1}\right] + \theta \cdot \Delta\gamma \left[\left(\frac{\partial \mathbf{h}}{\partial \boldsymbol{\sigma}}\right)_{n+1} : \frac{d}{dh}\boldsymbol{\sigma}_{n+1} + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{q}}\right)_{n+1} : \frac{d}{dh}\mathbf{q}_{n+1} \right] \\
&0 = \frac{d}{dh}f_{n+1} = \boldsymbol{\eta}_{n+1} : \frac{d}{dh}\boldsymbol{\sigma}_{n+1} + \boldsymbol{\xi}_{n+1} \cdot \frac{d}{dh}\mathbf{q}_{n+1}
\end{aligned} \tag{20}$$

Taking the limit $h \to 0$ in these expressions and recalling Eq.17:

$$\begin{aligned}
&\frac{d}{dh}(\boldsymbol{\sigma}_{n+1})_{h=0} = \mathbf{C} : \left(\dot{\boldsymbol{\varepsilon}}_n - \frac{d}{dh}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} \right) \\
&\frac{d}{dh}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} = \left(\frac{d(\Delta\gamma)}{dh} \right)_{h=0} \mathbf{r}_n \\
&\frac{d}{dh}(\mathbf{q}_{n+1})_{h=0} = \left(\frac{d(\Delta\gamma)}{dh} \right)_{h=0} \mathbf{h}_n \\
&0 = \frac{d}{dh}(f_{n+1})_{h=0} = \boldsymbol{\eta}_n : \frac{d}{dh}(\boldsymbol{\sigma}_{n+1})_{h=0} + \boldsymbol{\xi}_n \cdot \frac{d}{dh}(\mathbf{q}_{n+1})_{h=0}
\end{aligned} \tag{21}$$

Comparing Eq.21 with Eq.18 it becomes apparent that $\left(\frac{d(\Delta\gamma)}{dh} \right)_{h=0} = \gamma_n$ and, consequently the remaining consistency conditions in Eq.18 are also satisfied.

Taking the accuracy analysis a step further, second order accuracy of algorithm Eq.13-14 requires that the numerically integrated variables agree with their exact values to within third-order terms in h. In other words, in addition to Eq.18 it is now required

$$\begin{aligned}
&\frac{d^2}{dh^2}(\boldsymbol{\sigma}_{n+1})_{h=0} = \ddot{\boldsymbol{\sigma}}_n = \mathbf{C} : (\ddot{\boldsymbol{\varepsilon}}_n - \ddot{\boldsymbol{\varepsilon}}_n^p) \\
&\frac{d^2}{dh^2}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} = \ddot{\boldsymbol{\varepsilon}}_n^p = \dot{\gamma}_n \mathbf{r}_n + \gamma_n \left[\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}} \right)_n : \dot{\boldsymbol{\sigma}}_n + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}} \right)_n : \dot{\mathbf{q}}_n \right] \\
&\frac{d^2}{dh^2}(\mathbf{q}_{n+1})_{h=0} = \ddot{\mathbf{q}}_n = \dot{\gamma}_n \mathbf{h}_n + \gamma_n \left[\left(\frac{\partial \mathbf{h}}{\partial \boldsymbol{\sigma}} \right)_n : \dot{\boldsymbol{\sigma}}_n + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{q}} \right)_n : \dot{\mathbf{q}}_n \right] \\
&\frac{d^2}{dh^2}(\Delta\gamma)_{h=0} = \dot{\gamma}_n
\end{aligned} \tag{22}$$

where $\dot{\gamma}_n$ is to be determined from imposing a second order oscillatory satisfaction of the plastic consistency condition f=0, i.e.

$$0 = \ddot{f}_n = \dot{\boldsymbol{\eta}}_n : \dot{\boldsymbol{\sigma}}_n + \boldsymbol{\eta}_n : \ddot{\boldsymbol{\sigma}}_n + \dot{\boldsymbol{\xi}}_n \cdot \dot{\mathbf{q}}_n + \boldsymbol{\xi}_n \cdot \ddot{\mathbf{q}}_n \tag{23}$$

To check whether this condition is satisfied, differentiating Eq.20 with respect to h and taking the limit $h \to 0$ it is obtained

$$\begin{aligned}
&\frac{d^2}{dh^2}(\boldsymbol{\sigma}_{n+1})_{h=0} = \mathbf{C} : \left(\ddot{\boldsymbol{\varepsilon}}_n - \frac{d^2}{dh^2}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} \right) \\
&\frac{d^2}{dh^2}(\boldsymbol{\varepsilon}_{n+1}^p)_{h=0} = \left(\frac{d^2(\Delta\gamma)}{dh^2} \right)_{h=0} \mathbf{r}_n + 2\theta \cdot \gamma \left[\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}} \right)_n : \dot{\boldsymbol{\sigma}}_n + \left(\frac{\partial \mathbf{r}}{\partial \mathbf{q}} \right)_n : \dot{\mathbf{q}}_n \right] \\
&\frac{d^2}{dh^2}(\mathbf{q}_{n+1})_{h=0} = \left(\frac{d^2(\Delta\gamma)}{dh^2} \right)_{h=0} \mathbf{h}_n + 2\theta \cdot \Delta\gamma \left[\left(\frac{\partial \mathbf{h}}{\partial \boldsymbol{\sigma}} \right)_n : \dot{\boldsymbol{\sigma}}_n + \left(\frac{\partial \mathbf{h}}{\partial \mathbf{q}} \right)_n : \dot{\mathbf{q}}_n \right] \\
&0 = \frac{d^2}{dh^2}(f_{n+1})_{h=0} = \dot{\boldsymbol{\eta}}_n : \dot{\boldsymbol{\sigma}}_n + \boldsymbol{\eta}_n : \frac{d^2}{dh^2}(\boldsymbol{\sigma}_{n+1})_{h=0} + \dot{\boldsymbol{\xi}}_n \cdot \dot{\mathbf{q}}_n + \boldsymbol{\xi}_n \cdot \frac{d^2}{dh^2}(\mathbf{q}_{n+1})_{h=0}
\end{aligned} \tag{24}$$

Comparison of Eq.24 with Eq.22 and Eq.23 leads to the conclusion that $\left(\frac{d^2}{dh^2}(\Delta\gamma) \right)_{h=0} = \dot{\gamma}_n$ and that second-order accuracy is achieved provided one chooses $2\theta = 1, i.e. \theta = 1/2$

A similar analysis leads to the same conclusions for the generalized midpoint rule, i.e. it is consistent for all θ and second-order accurate for $\theta = 1/2$.

From these results it is concluded that the choice of $\theta = 1/2$ leads to optimal accuracy for small strain increments. By contrast, the stability analysis carried out below indicate that larger values of θ may prove advantageous in terms accuracy and even necessary for stability in the realm of large time steps.

Numerical Stability

The concept of numerical stability plays a central role in approximation theory for initial value problems. Its relevance stems largely from the fact that consistency and stability are necessary and sufficient conditions for convergence as time step size is allowed to tend to zero. In the context of linear analysis, the numerical stability of a time stepping algorithm can be characterized with the aid of suitably defined energy norms. By contrast, when the initial value problem under consideration is nonlinear it is not always clear from the structure of the governing equations how to characterize numerical stability.

Large-scale and small-scale stability: The purpose of the stability analysis that follows is to determine under what conditions finite perturbations in the initial stresses are attenuated by the algorithm, i.e.

$$d\left(\boldsymbol{\sigma}_{n+1}^{(2)},\boldsymbol{\sigma}_{n+1}^{(1)}\right) \le d\left(\boldsymbol{\sigma}_{n}^{(2)},\boldsymbol{\sigma}_{n}^{(1)}\right) \tag{25}$$

where $d(\cdot,\cdot)$ is some suitable distance to be defined on the yield surface and $\boldsymbol{\sigma}_{n+1}^{(1)}$ and $\boldsymbol{\sigma}_{n+1}^{(2)}$ are two sets of updated stresses corresponding to arbitrary initial values $\boldsymbol{\sigma}_{n}^{(1)}$ and $\boldsymbol{\sigma}_{n}^{(2)}$ respectively, all of which are assumed to lie on the yield surface. Stability in the sense of Eq.25 will be therefore referred to as 'large-scale' stability. It has been shown in Reference 15 that for nonlinear IVP defined on Banach manifolds consistency and large-scale stability with respect to a complete metric are sufficient for convergence.

The stability analysis is significantly simplified, however, once it is recognized that attention can be confined to infinitesimal perturbations in the initial conditions of the type $\boldsymbol{\sigma}_n \to \boldsymbol{\sigma}_n + d\boldsymbol{\sigma}_n$. This rests on the fact that attenuation by the algorithm of the infinitesimal perturbations,

$$\left\| d\boldsymbol{\sigma}_{n+1} \right\| \le \left\| d\boldsymbol{\sigma}_{n} \right\| \tag{26}$$

with respect to some suitable norm $\|\cdot\|$, of 'small-scale' stability, implies large-scale stability in the sense in Eq.25. Let this norm be defined as the energy norm:

$$\|\boldsymbol{\sigma}\|^2 = \sigma_{ij} D_{ijkl} \sigma_{kl} \tag{27}$$

where **D**=**C**$^{-1}$, and let the distance on the yield surface be defined as

$$d(\boldsymbol{\sigma}^{(1)},\boldsymbol{\sigma}^{(2)}) = \inf_{\lambda} \int_{\lambda} \|\boldsymbol{\sigma}'(s)\| ds \tag{28}$$

where infimum is taken over all stress paths λ on the yield surface joining $\boldsymbol{\sigma}^{(1)}$ and $\boldsymbol{\sigma}^{(2)}$.

Consider next any two initial states of stress $\boldsymbol{\sigma}_{n}^{(1)}$ and $\boldsymbol{\sigma}_{n}^{(2)}$ for which the infimum in Eq.28 is attained. If λ_n is such curve, then by definition

$$d(\boldsymbol{\sigma}_{n}^{(1)},\boldsymbol{\sigma}_{n}^{(2)}) = \int_{\lambda_n} \|\boldsymbol{\sigma}'(s)\| ds \tag{29}$$

Let λ_{n+1} be the transform of λ_n by the algorithm. λ_{n+1} lies on the yield surface and joins $\boldsymbol{\sigma}_{n+1}^{(1)}$ and $\boldsymbol{\sigma}_{n+1}^{(2)}$. Then by definition it follows that

$$d(\boldsymbol{\sigma}_{n+1}^{(1)},\boldsymbol{\sigma}_{n+1}^{(2)}) \le \int_{\lambda_{n+1}} \|\boldsymbol{\sigma}'(s)\| ds \tag{30}$$

However under the assumption of small scale stability of the algorithm one has $\|\boldsymbol{\sigma}'(s_{n+1})\|ds = \|d\boldsymbol{\sigma}(s_{n+1})\| \le \|d\boldsymbol{\sigma}(s_n)\| = \|\boldsymbol{\sigma}'(s_n)\|ds$ for every pair of corresponding points s_n and s_{n+1} and λ_n and λ_{n+1} respectively and hence

$$\int_{\lambda_{n+1}} \|\boldsymbol{\sigma}'(s)\|ds \le \int_{\lambda_n} \|\boldsymbol{\sigma}'(s)\|ds \tag{31}$$

Combining Eq.29,30 and 31 it is concluded that

$$d(\boldsymbol{\sigma}_{n+1}^{(1)}, \boldsymbol{\sigma}_{n+1}^{(2)}) \le d(\boldsymbol{\sigma}_n^{(1)}, \boldsymbol{\sigma}_n^{(2)}) \tag{32}$$

which proves large scale stability.

Conclusion: small-scale stability in the energy norm is equivalent to large-scale stability.

This conclusion has a practical importance since it shows that stability analysis may be confined to the assessment of small-scale stability. Now let's investigate this with our algorithms.

Generalized Trapezoidal Rule

To carry out a small-scale stability analysis of the generalized trapezoidal rule it is necessary to determine first how it propagates infinitesimal perturbations in the initial conditions. First differentiate Eq.13

$$\begin{aligned}
&d\boldsymbol{\sigma}_{n+1} = -\mathbf{C} : d\boldsymbol{\varepsilon}_{n+1}^p ; d\boldsymbol{\sigma}_n = -\mathbf{C} : d\boldsymbol{\varepsilon}_n^p \\
&d\boldsymbol{\varepsilon}_{n+1}^p - d\boldsymbol{\varepsilon}_n^p = d(\Delta\gamma)\left[(1-\theta)\mathbf{r}_n + \theta\mathbf{r}_{n+1}\right] + \Delta\gamma\left[(1-\theta)\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}}\right)_n : d\boldsymbol{\sigma}_n + \theta\left(\frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}}\right)_{n+1} : d\boldsymbol{\sigma}_{n+1}\right] \\
&0 = df_{n+1} = \boldsymbol{\eta}_{n+1} : d\boldsymbol{\sigma}_{n+1}
\end{aligned} \tag{33}$$

Introducing the notation

$$\mathbf{B} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}}; \quad \overline{\mathbf{r}}_{n+\theta} = (1-\theta)\mathbf{r}_n + \theta\mathbf{r}_{n+1} \tag{34}$$

Equations 33 can be simplified to read

$$\begin{aligned}
&(\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1}) : d\boldsymbol{\sigma}_{n+1} = d\Delta\gamma\overline{\mathbf{r}}_{n+\theta} + (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n) : d\boldsymbol{\sigma}_n \\
&0 = \boldsymbol{\eta}_{n+1} : d\boldsymbol{\sigma}_{n+1}
\end{aligned} \tag{35}$$

Substituting 35a into 35b, the value of $d\Delta\gamma$ may be solved for, leading to

$$d\Delta\gamma = \frac{\boldsymbol{\eta}_{n+1} : (\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n) : d\boldsymbol{\sigma}_n}{\boldsymbol{\eta}_{n+1} : (\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : \overline{\mathbf{r}}_{n+\theta}} \tag{36}$$

Solving for $d\boldsymbol{\sigma}_{n+1}$ and making use of Eq.36,

$$d\boldsymbol{\sigma}_{n+1} = \mathbf{P} : (\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n) : d\boldsymbol{\sigma}_n \tag{37}$$

where

$$\mathbf{P} = \mathbf{I} - \frac{\hat{\mathbf{r}}_{n+\theta} \otimes \boldsymbol{\eta}_{n+1}}{\hat{\mathbf{r}}_{n+\theta} : \boldsymbol{\eta}_{n+1}} \qquad (38)$$
$$\hat{\mathbf{r}}_{n+\theta} = (\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : \overline{\mathbf{r}}_{n+\theta}$$

To derive small scale stability from Eq.37, we can define the energy norm of a matrix in the usual fashion:

$$\|\mathbf{A}\| \equiv \sup_{\sigma} \frac{\|\mathbf{A} : \boldsymbol{\sigma}\|}{\|\boldsymbol{\sigma}\|} \qquad (39)$$

Taking the energy norm of Eq.37 and using the inequalities $\|A:\sigma\| \le \|A\| \cdot \|\sigma\|$ and $\|A_1 : A_2\| \le \|A_1\| \cdot \|A_2\|$, finally it is obtained:

$$\|d\boldsymbol{\sigma}_{n+1}\| \le \|\mathbf{P}\| \cdot \left\|(\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n)\right\| \cdot \|d\boldsymbol{\sigma}_n\| \qquad (40)$$

Now we have to estimate the components of Eq.40. Concerning the norm of P it suffices to note that P defines an oblique projection along the direction $\hat{\mathbf{r}}_{n+\theta}$ onto the hyperplane orthogonal to $\boldsymbol{\eta}_{n+1}$, *i.e.* $\mathbf{P} : \hat{\mathbf{r}}_{n+\theta} = 0$ and $\mathbf{P} : \boldsymbol{\sigma} = \boldsymbol{\sigma}$ for every σ orthogonal to $\boldsymbol{\eta}_{n+1}$. From these properties and the definition in Eq.39, it follows that $\|\mathbf{P}\| = 1$.

Assuming that the flow direction r derives from a convex potential or loading function which is a common assumption made in many plasticity models, we get $\mathbf{B} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\sigma}}$ symmetric and positive definite everywhere on the yield surface.

Under these assumptions, we get:

$$\left\|(\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n)\right\| = \left|\frac{\boldsymbol{\delta}_{\max} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n) : \boldsymbol{\delta}_{\max}}{\boldsymbol{\delta}_{\max} : (\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : \boldsymbol{\delta}_{\max}}\right| \qquad (41)$$

where $\delta_{\max}$ is the eigenvector corresponding to the maximum eigenvalue of the eigenproblem

$$\left[(\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n) - \mu(\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1}\right] : \boldsymbol{\delta} = 0 \qquad (42)$$

which may be normalized to satisfy

$$\|\boldsymbol{\delta}_{\max}\|^2 = \boldsymbol{\delta}_{\max} : \mathbf{D} : \boldsymbol{\delta}_{\max} = 1 \qquad (43)$$

Denoting

$$\beta_n = \boldsymbol{\delta}_{\max} : \mathbf{B}_n : \boldsymbol{\delta}_{\max}; \quad \beta_{n+1} = \boldsymbol{\delta}_{\max} : \mathbf{B}_{n+1} : \boldsymbol{\delta}_{\max} \qquad (44)$$

and making use of the Eq.43, 41 reduces to

$$\left\|(\mathbf{D} + \theta\Delta\gamma\mathbf{B}_{n+1})^{-1} : (\mathbf{D} - (1-\theta)\Delta\gamma\mathbf{B}_n)\right\| = \left|\frac{1 - (1-\theta)\Delta\gamma\beta_n}{1 + \theta\Delta\gamma\beta_{n+1}}\right| \qquad (45)$$

which substituted into Eq.40,

$$\left\|d\boldsymbol{\sigma}_{n+1}\right\| \le \left|\frac{1-(1-\theta)\Delta\gamma\beta_n}{1+\theta\Delta\gamma\beta_{n+1}}\right|\left\|d\boldsymbol{\sigma}_n\right\| \tag{46}$$

β_n *and* β_{n+1} are positive scalars due to the symmetry and positive definiteness of B. We can say that

$$\left|\frac{1-(1-\theta)\Delta\gamma\beta_n}{1+\theta\Delta\gamma\beta_{n+1}}\right| \le \left|\frac{1-\theta}{\theta}\right|\frac{\beta_n}{\beta_{n+1}}, 0 \le \theta \le 1 \tag{47}$$

To find an upper bound for this, we can define the quantity

$$s = \sup\frac{\boldsymbol{\delta} : \mathbf{B}_1 : \boldsymbol{\delta}}{\boldsymbol{\delta} : \mathbf{B}_2 : \boldsymbol{\delta}} \tag{48}$$

where B_1 and B_2 are evaluated at arbitrary points σ_1 and σ_2. For reasons that are clarified below, s will be referred to as the distortion index of the loading surface. From Eq.48,

$$\frac{\beta_n}{\beta_{n+1}} \le s \tag{49}$$

which in combination with 46 and 47 yields

$$\left\|d\boldsymbol{\sigma}_{n+1}\right\| \le \left|\frac{1-\theta}{\theta}\right| s\left\|d\boldsymbol{\sigma}_n\right\| = c\left\|d\boldsymbol{\sigma}_n\right\|, 0 \le \theta \le 1 \tag{50}$$

where one writes

$$c \equiv \left|\frac{1-\theta}{\theta}\right| s \tag{51}$$

Unconditional stability requires $c \le 1$ which in view of Eq.51 gives us,

$$\theta \ge \frac{s}{1-s} \equiv \theta_{\min} \tag{52}$$

As discussed below, it follows from definition in Eq.48 that the distortion index s ranges from 1 for loading functions of constant curvature such as J2 Plasticity to infinity for loading surfaces with corners. As a consequence, the minimum value of $\theta_{\min}$ of θ to achieve unconditional stability ranges from ½ for s=1 to 1 for $s = \infty$

For $\theta < \theta_{\min}$ stability is conditional. A little reflection soon leads to the condition that

$$\left|\frac{1-(1-\theta)\Delta\gamma\beta_n}{1+\theta\Delta\gamma\beta_{n+1}}\right| \le 1 \tag{53}$$

if and only if $\Delta\gamma$ is confined to be

$$\Delta\gamma \le \frac{2/\beta_{\max}}{s-(1+s)\theta} = \Delta\gamma_{crit}, \theta < \theta_{\min} \tag{54}$$

where $\beta_{\max}$ is the maximum eigenvalue among all tensors B over the yield surface. For $\theta = \theta_{\min}$, the critical value $\Delta\gamma_{crit}$ of the incremental plastic parameter becomes infinity and unconditional stability is recovered.

Remarks:

(1) Under the assumption of a loading function ψ acting as a potential for the plastic flow direction $\mathbf{r} = \frac{\partial\psi}{\partial\boldsymbol{\sigma}}$, one has $\mathbf{B} = \frac{\partial^2\psi}{\partial\boldsymbol{\sigma}^2}$, i.e. tensor **B** coincides with the Hessian of ψ. Thus **B** determines how the normal **r** to the loading surfaces $\psi = const$ varies over neighboring points, i.e. it is a measure of the curvature of the loading surface.

(2) Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be evaluated at two arbitrary points $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ on the yield surface. Then the scalar

$$s(\boldsymbol{\sigma_1},\boldsymbol{\sigma_2}) = \sup_{\delta}\frac{\boldsymbol{\delta}:\mathbf{B_1}:\boldsymbol{\delta}}{\boldsymbol{\delta}:\mathbf{B_2}:\boldsymbol{\delta}} \tag{55}$$

where supremum is taken over all incremental stresses δ, is a measure of the curvature of the loading surface at point $\boldsymbol{\sigma_1}$ relative to that at point $\boldsymbol{\sigma_2}$. From this perspective,

$$s = \sup_{\sigma_1,\sigma_2} s(\boldsymbol{\sigma_1},\boldsymbol{\sigma_2}) \tag{56}$$

where supremum is taken over all pairs of stress point σ_1 and σ_2 on the yield surface, gives an idea of the extent of variation of the curvature of the loading surface over points of the yield surface. This provides the reason for calling s the distortion index of the loading surface.

(3) The shape of the loading surface strongly influences the stability properties of the generalized trapezoidal rule.

Generalized Midpoint Rule

En entirely similar analysis leads to the following estimate for generalized midpoint rule:

$$\|d\boldsymbol{\sigma}_{n+1}\| \le \left|\frac{1-(1-\theta)\Delta\gamma\beta_{n+\theta}}{1+\theta\Delta\gamma\beta_{n+\theta}}\right| \|d\boldsymbol{\sigma}_n\| \tag{57}$$

where now $\beta_{n+\theta}$ is the maximum eigenvalue of the tensor $B_{n+\theta}$. Making use of Eq.47 and 57 can be written

$$\|d\boldsymbol{\sigma}_{n+1}\| \le \left|\frac{1-\theta}{\theta}\right| \|d\boldsymbol{\sigma}_n\| = c\|d\boldsymbol{\sigma}_n\| \tag{58}$$

$$c \equiv \left|\frac{1-\theta}{\theta}\right| \tag{59}$$

Unconditional stability requires $c \le 1$ which in view of 59 implies

$$\theta \ge \theta_{\min} = \frac{1}{2} \tag{60}$$

In conclusion, generalized midpoint rule is unconditionally stable for $\theta \geq 1/2$, regardless of the choice of the loading potential. This is completely in sharp contrast with the generalized trapezoidal rule for which stability has been shown to be strongly dependent on the shape of the loading surface.

For $\theta < 1/2$ the stability of generalized midpoint rule is only conditional. Making use of Eq.53 and 54,

$$\Delta\gamma \leq \frac{2/\beta_{\max}}{1-2\theta} = \Delta\gamma_{crit}, \theta < 1/2 \tag{61}$$

which is unlike 54 is independent of the choice of loading potential. For $\theta = 1/2$, we recover unconditional stability again.

6. CONCLUSION & DISCUSSION

It has been studied that numerical analysis has a very big potential application in calculation of response of Plastic Materials. Among integration schemes, Implicit Euler method is the most widely and successfully used one. For the integration with this method, we needed to calculate partial derivatives of yield function with respect to stress like variables. GCPPM has been shown to be very easily extensible to other type of yield functions and hardening types. In conjunction with GCPPM, numerical differentiation resulted in very satisfactory results.

GCPPM is shown to give quadratic convergence at the local level. The elastoplastic tangent resulted from this method is compared with RRM and they matched, and this suggests that GCPPM will provide us quadratic convergence in Global, as well.

The next level of application that the author suggests is Multi-surface plasticity with GCPPM and using numerical differentiation. However, as shown in Ref.7, smoothening out the yield surface speeds up singular points on the yield surface and thus reduces the cost.

In Chapter-5, two families of algorithms have been presented. It is observed that the stability properties of the generalized trapezoidal rule are very sensitive to the degree of distortion of the loading surface. In particular, in the presence of corners, stability of the trapezoidal rule requires $\theta = 1$,which corresponds to the closest point procedure. By contrast, the generalized midpoint rule is unconditionally stable for $\theta \geq 1/2$, regardless of the choice of loading surface. This remarkable fact would appear to point to the generalized midpoint rule as preferable to the generalized trapezoidal rule, apart from simple cases such as the J2 model for which both integration rules coincide (since J2 model has constant curvature).

REFERENCE

1. A.S. Khan, S. Huang, “Continuum Theory of Plasticity”, John Wiley & Sons, New York,1995
2. J. Lubliner, “Plasticity Theory”, Macmillan Publishing Company, New York, 1990
3. O.C. Zienkiewicz, R.L. Taylor, “The Finite Element Method – Volume 2”, Butterworth-Heinemann,2000
4. J.C. Simo, T.J.R. Hughes, “Computational Inelasticity”, Springer, New York, 1998
5. R.L. Burden, J.D. Faires, “Numerical Analysis”, Brooks/Cole Publishing Company, 1998
6. A. Perez-Foguet, A. Rodriguez-Ferran, A. Huerta, “Numerical Differentiation for local and global tangent operators in computational plasticity”, Computer Methods in Applied Mechanics and Engineering, 189(2000),277-296
7. A. Perez-Foguet, A. Rodriguez-Ferran, A. Huerta, “Numerical Differentiation for non-trivial consistent tangent matrices: an application to the MRS-Lade Model”, International Journal for Numerical Methods in Engineering, 48(2000),159-184
8. F.Armero, A.Perez-Foguet, “On the formulation of closest-point projection algorithms in elastoplasticity – Part 1: The variational Structure”, International Journal for Numerical Methods in Engineering, 53(2002),297-329
9. F.Armero, A.Perez-Foguet, “On the formulation of closest-point projection algorithms in elastoplasticity – Part 2: Globally Convergent Schemes”, International Journal for Numerical Methods in Engineering, 53(2002),331-374
10. W.Squire, G.Trapp, “Using Complex Variables to estimate derivatives of real functions”, SIAM Review, 40,1,110-112
11. M. Ortiz, J.B. Martin, “Symmetry Preserving Return Mapping Algorithms and Incremental Extremal Paths: A Unification of Concepts”, International Journal for Numerical Methods in Engineering, 28 (1989), 1839-1853
12. P. Papadopoulos, R.L. Taylor, “On the application of multi-step integration methods to infititesimal elastoplasticity”, International Journal for Numerical Methods in Engineering,37(1994),3169-3184
13. S.W. Sloan, “Substepping schemes for the numerical integration of elastoplastic stress-strain relations”, International Journal for Numerical Methods in Engineering,24(1987),893-911
14. M. Ortiz, E.P. Popov, “Accuracy and Stability of Integration Algorithms for elastoplastic constitutive relations”, International Journal for Numerical Methods in Engineering,, 21(1985),1561-1576
15. J.Marsden, ”On Product formulas for nonlinear semi-groups”, J. Functional Anal, 13,51-72 (1973)

FIGURE- 1
Effect of Time discretization on Stress component σ_{12}

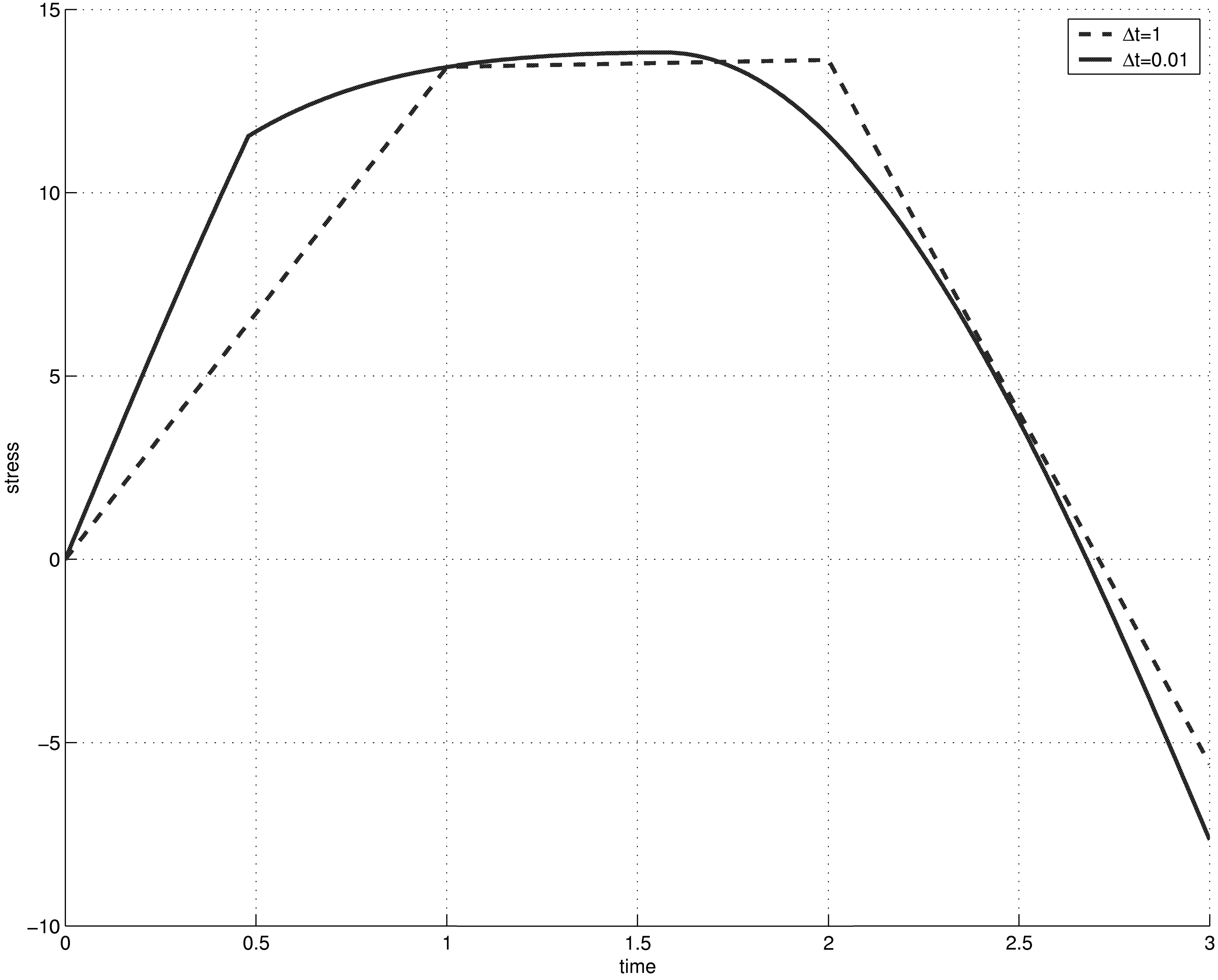
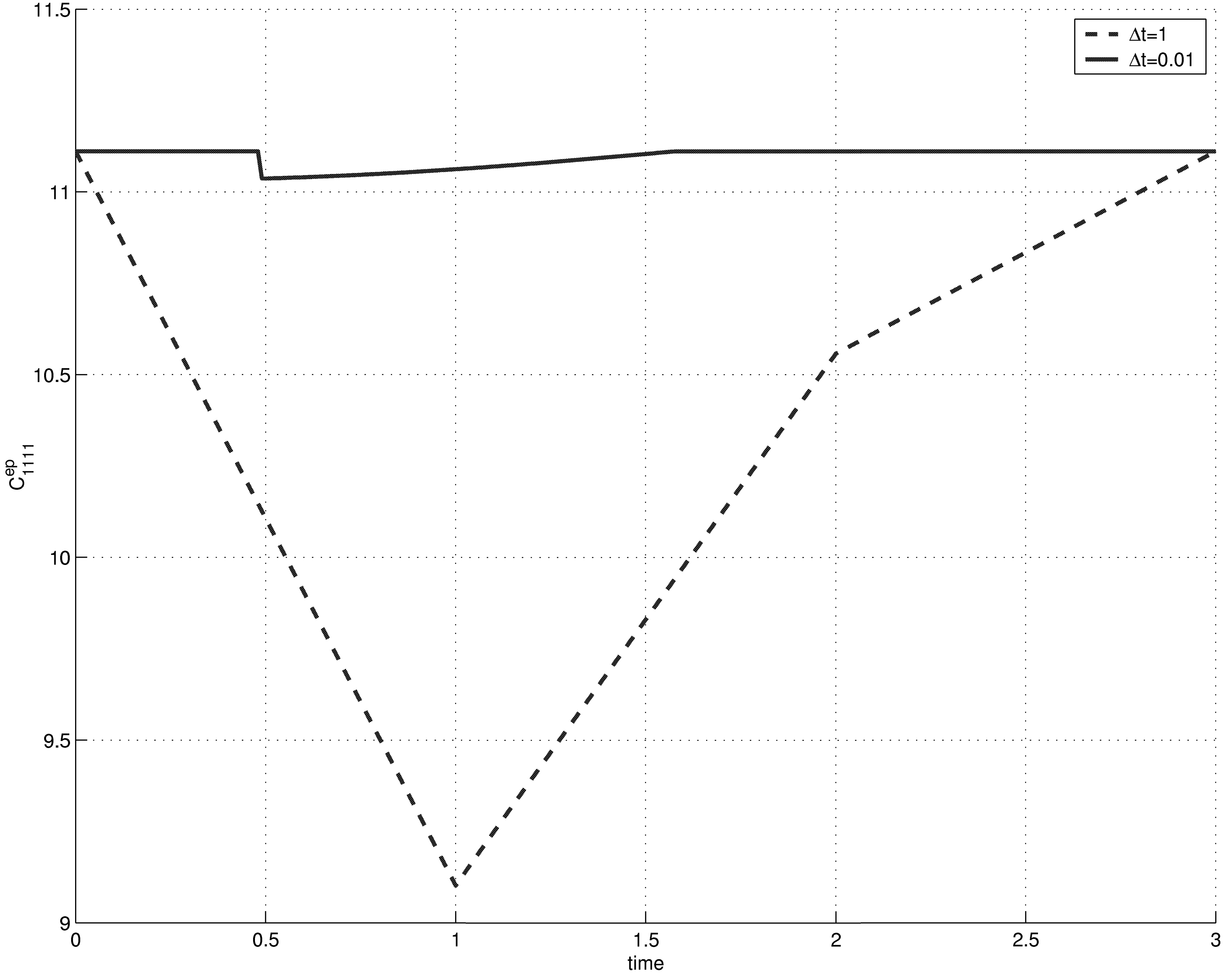
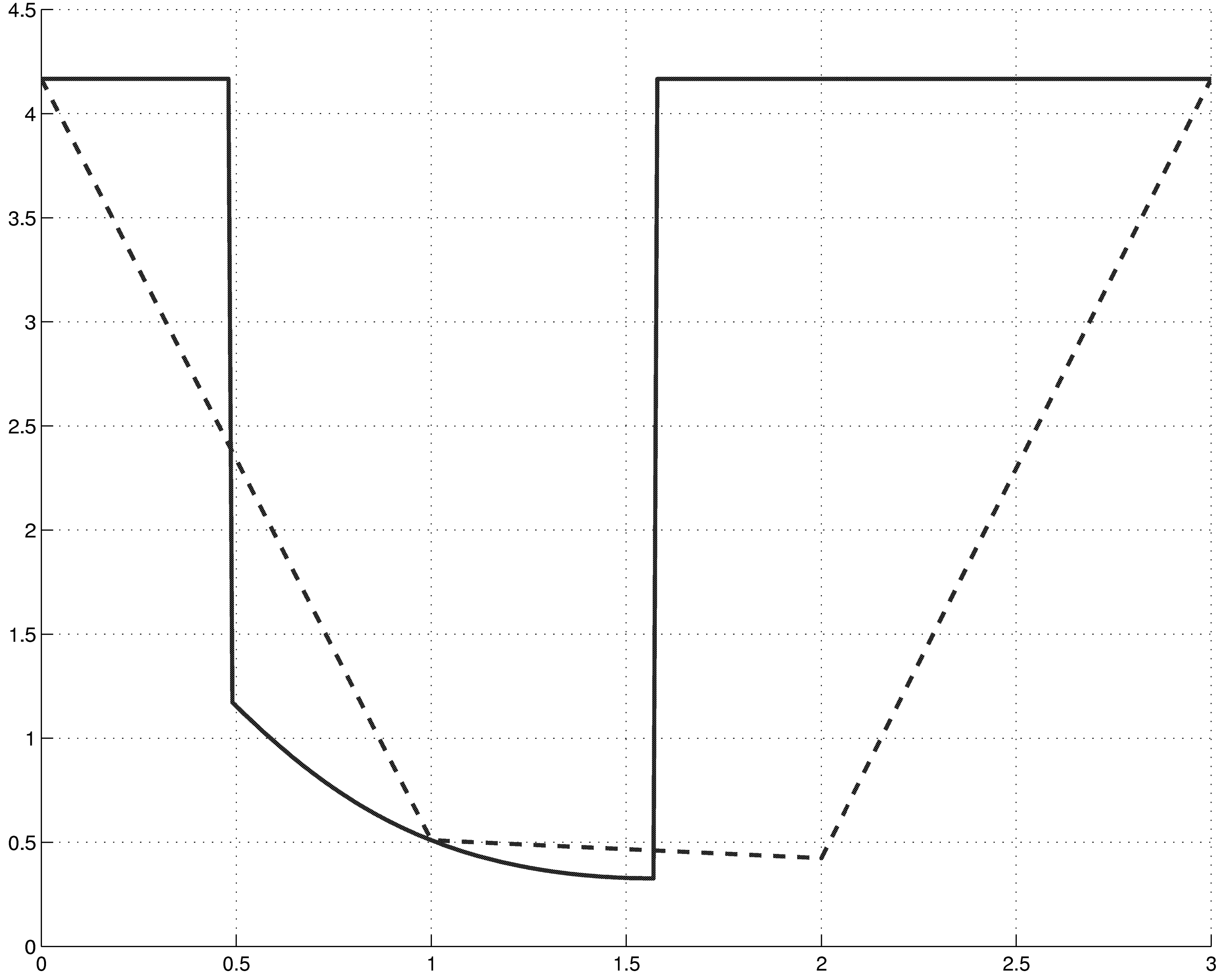


FIGURE – 2
Effect of time dicsretization on Elastoplastic Tangent component C^{ep}_{1111}



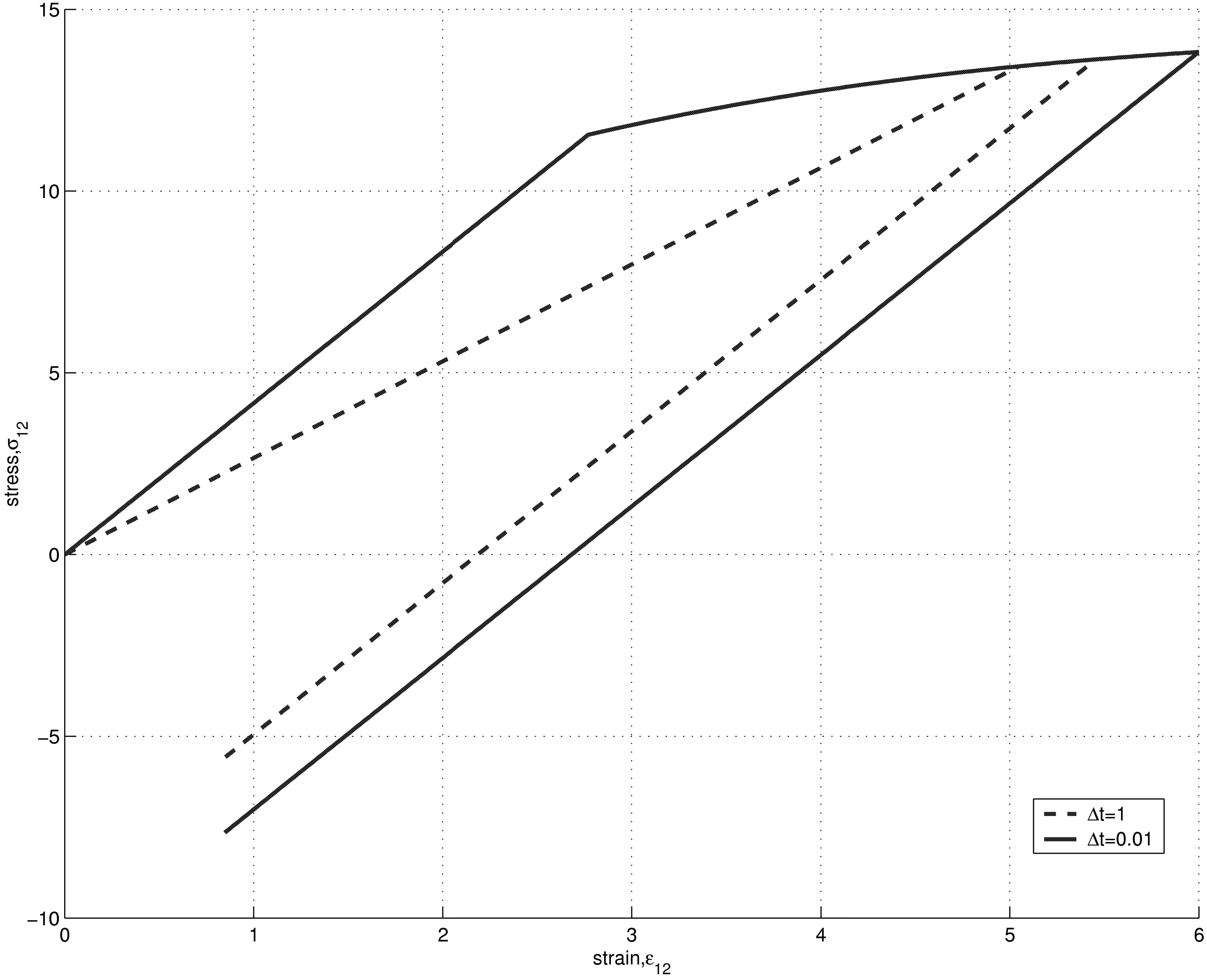
FIGURE–3

Effect of time discretization on Elastoplastic tangent component C^{ep}_{1212}

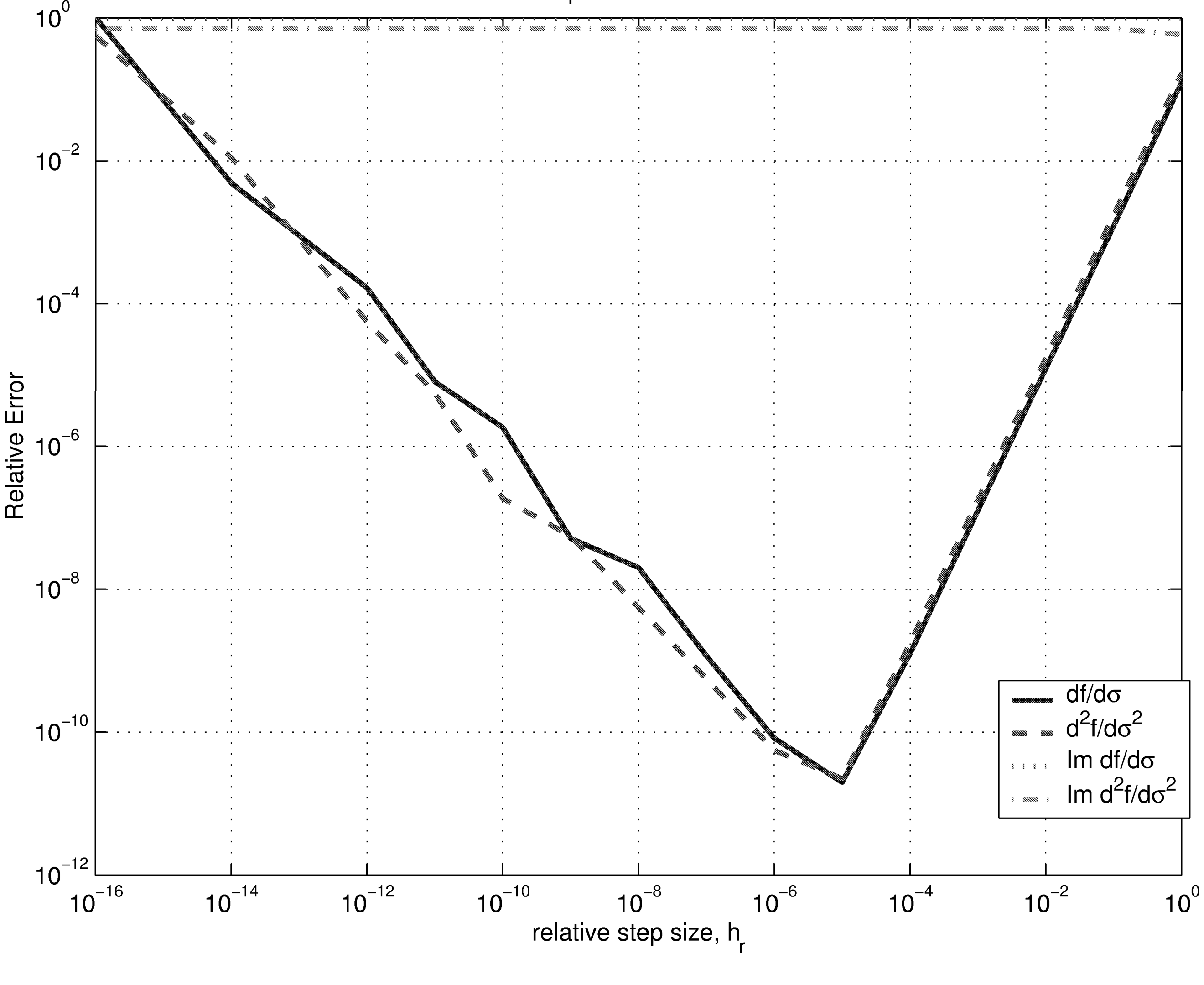


FIGURE–4

Effect of time discretization on stress component σ_{12}



FIGURE−5
Norm of the Error in partial derivatives to Yield Surface


10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
Relative Error
10^{-16}
10^{-14}
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^0
relative step size, h_r
$df/d\sigma$
$d^2f/d\sigma^2$
Im $df/d\sigma$
Im $d^2f/d\sigma^2$