# $\texttt{PorousLab}$: Plasticity theory manual

Danilo Cavalcanti

2025-06

# Introduction

This manual summarizes the small-strain rate-independent plasticity formulation used as the plasticity theory part of the `PorousLab` theory manual. The presentation follows the computational plasticity framework of Simo and Hughes [2] and de Souza Neto et al. [1].

# 1 Small-strain rate-independent plasticity

At small strain, the total strain tensor is additively decomposed into elastic and plastic parts,

$$\varepsilon = \varepsilon^{\mathrm{e}} + \varepsilon^{\mathrm{p}}. \tag{1}$$

The elastic response is assumed linear in the elastic strain,

$$\boldsymbol{\sigma} = \mathbb{C} : \varepsilon^{\mathrm{e}} = \mathbb{C} : (\varepsilon - \varepsilon^{\mathrm{p}}), \tag{2}$$

where $\mathbb{C}$ is the fourth-order elasticity tensor. For isotropic elasticity,

$$\mathbb{C} : \varepsilon^{\mathrm{e}} = 2\,\mu\ \operatorname{dev}\varepsilon^{\mathrm{e}} + K\ \operatorname{tr}(\varepsilon^{\mathrm{e}})\mathbf{I},$$

with shear modulus $\mu$, bulk modulus $K$, and second-order identity $\mathbf{I}$.

The material state is represented by the Cauchy stress $\boldsymbol{\sigma}$ and by internal variables $\mathbf{q}$. The elastic domain in generalized stress space is

$$\mathbb{E}_{\sigma} = \{(\boldsymbol{\sigma}, \mathbf{q}) \mid f(\boldsymbol{\sigma}, \mathbf{q}) \le 0\}, \tag{3}$$

where $f(\boldsymbol{\sigma}, \mathbf{q}) = 0$ defines the yield surface. States with $f(\boldsymbol{\sigma}, \mathbf{q}) < 0$ are strictly elastic.

The evolution of the plastic strain is governed by the flow rule as

$$\dot{\varepsilon}^{\mathrm{p}} = \gamma\,\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q}), \tag{4}$$

where $\gamma$ is a non-negative parameter named the plastic multiplier and $\mathbf{n}$ is the flow direction.

The evolution of the internal variables is governed by the hardening/softening law,

$$\dot{\mathbf{q}} = \gamma\,\mathbf{h}(\boldsymbol{\sigma}, \mathbf{q}), \tag{5}$$

The parameter $\gamma$ is a nonnegative function, called the consistency parameter, or, plastic multiplier. It is assumed that it obeys the Kuhn–Tucker complementary conditions:

$$\begin{aligned} \gamma &\geq 0, \\ f(\boldsymbol{\sigma}, \mathbf{q}) &\leq 0, \\ \gamma f(\boldsymbol{\sigma}, \mathbf{q}) &= 0. \end{aligned} \tag{6}$$

In addition to the KKT conditions presented in equation (6), the plastic multiplier also satisfies the consistency condition:

$$\gamma \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0. \tag{7}$$

## 1.1 Interpretation of the Kuhn–Tucker conditions

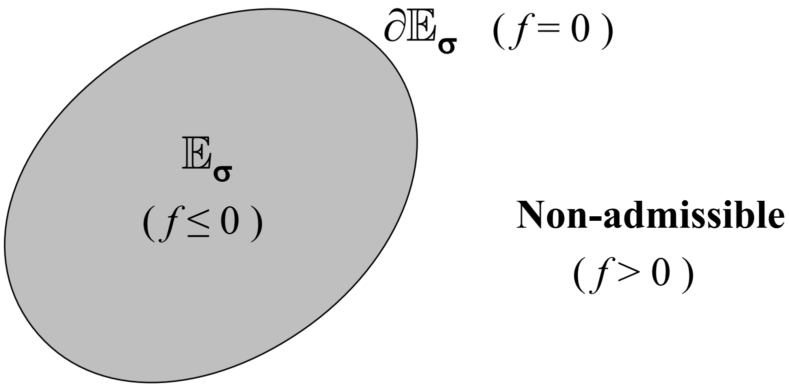


Figure 1: Elastic domain and admissible states in stress space.

**Case 1: Instantaneously elastic response**

Consider a state $\{\boldsymbol{\sigma}, \mathbf{q}\} \in \text{int}(\mathbb{E}_{\sigma})$, so that $f(\boldsymbol{\sigma}, \mathbf{q}) < 0$. With the Kuhn–Tucker complementarity condition, (6), we conclude that:

$$\gamma f(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad \text{and} \quad f(\boldsymbol{\sigma}, \mathbf{q}) < 0 \Rightarrow \gamma = 0.$$

Then, from the evolution rules, (4) and (5):

$$\dot{\varepsilon}^{\text{p}} = \mathbf{0} \quad \text{and} \quad \dot{\mathbf{q}} = \mathbf{0}.$$

These imply that the stress increment is purely elastic:

$$f(\boldsymbol{\sigma}, \mathbf{q}) < 0 \Rightarrow \begin{cases} \gamma = 0, \\ \dot{\varepsilon}^{\text{p}} = \dot{\mathbf{q}} = \mathbf{0}, \\ \dot{\boldsymbol{\sigma}} = \mathbf{C} : \dot{\varepsilon}. \end{cases}$$

Now suppose that $\{\boldsymbol{\sigma}, \mathbf{q}\} \in \partial(\mathbb{E}_{\sigma})$, so that $f(\boldsymbol{\sigma}, \mathbf{q}) = 0$. Then Kuhn–Tucker complementarity condition, (6), is automatically satisfied even if $\gamma > 0$. In this case, two conditions can arise, the plastic multiplier can be zero or positive. The value of the plastic multiplier is defined using the consistency condition, (7).

**Case 2.a: Unloading from a plastic state**

If $\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) < 0$, from the consistency condition:

$$\gamma \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad \text{and} \quad \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) < 0 \Rightarrow \gamma = 0.$$

Then, from the evolution rules, (4) and (5):

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \mathbf{0} \quad \text{and} \quad \dot{\mathbf{q}} = \mathbf{0}.$$

These imply that the stress increment is purely elastic, meaning that unloading is happening.

**Case 2.b: Plastic and neutral loading**

If $\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0$, from the consistency condition:

$$\gamma \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad \text{and} \quad \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0 \Rightarrow \gamma \geq 0.$$

If $\gamma > 0$, then a plastic loading is happening, with

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \neq \mathbf{0} \quad \text{and} \quad \dot{\mathbf{q}} \neq \mathbf{0}.$$

If $\gamma = 0$, then a neutral loading is happening, with:

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \mathbf{0} \quad \text{and} \quad \dot{\mathbf{q}} = \mathbf{0}.$$

To summarize this discussion:

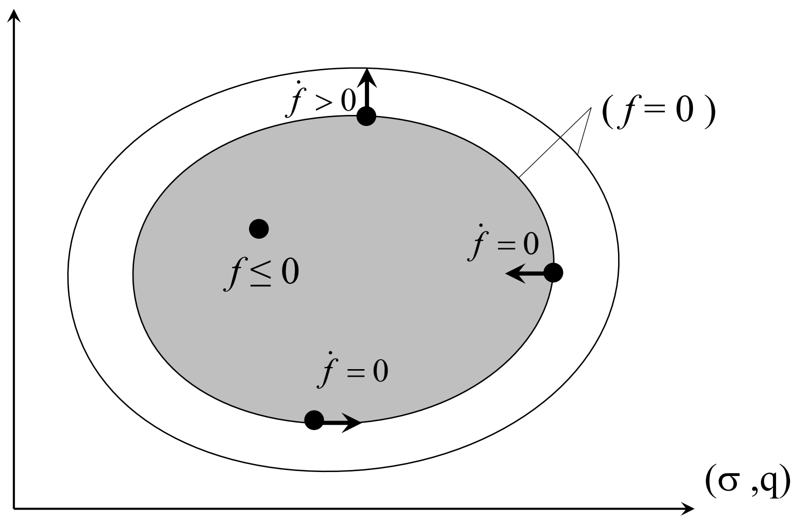


Figure 2: Geometric interpretation of the elastic domain, yield surface, and consistency condition in stress–internal-variable space.

## 1.2 Plastic multiplier and continuum elastoplastic tangent

The parameter $\gamma$ defines the rate of variation of the plastic deformation when $f = 0$. This evolution happens to maintain the condition $f = 0$. Then, $\dot{f} = 0$, as it is stated in the consistency condition, (7).

We use the consistency condition to define the plastic multiplier, $\gamma$. Using the chain rule:

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{\partial f}{\partial \mathbf{q}} \cdot \dot{\mathbf{q}} \leq 0.$$

Using the rate form of the elastic law,

$$\dot{\boldsymbol{\sigma}} = \mathbb{C} : (\dot{\varepsilon} - \dot{\varepsilon^{\mathrm{p}}}),$$

one obtains

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : (\dot{\varepsilon} - \dot{\varepsilon^{\mathrm{p}}}) + \frac{\partial f}{\partial \mathbf{q}} \cdot \dot{\mathbf{q}} \leq 0.$$

Equivalently,

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon} - \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon^{\mathrm{p}}} + \frac{\partial f}{\partial \mathbf{q}} \cdot \dot{\mathbf{q}} \leq 0.$$

Substituting the flow and the hardening rules:

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon} - \gamma \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} + \gamma \frac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h} \leq 0.$$

Thus,

$$\dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon} - \gamma \left( \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \frac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h} \right) \leq 0.$$

Defining the scalars $A$ and $B$:

$$A = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon},$$

$$B = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \frac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h}.$$

The condition $A \geq 0$ has a geometric interpretation. Considering $A$ as a dot product, then $A = a \cos\theta$. The condition $A > 0$ implies that yielding occurs with $0 \leq \theta \leq \pi/2$, and neutral loading occurs with $\theta = \pi/2$, when the rate of the strain tensor occurs in a direction tangent to the yield surface.

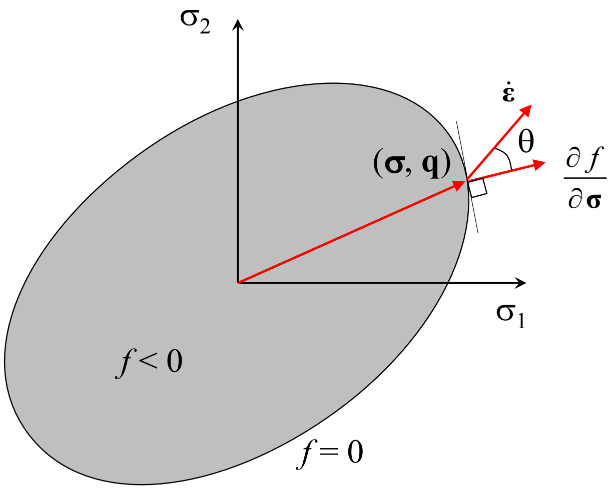


Figure 3: Elastic predictor and plastic corrector interpretation of the fully implicit return-mapping algorithm.

Considering first the case of plastic loading, for which $\gamma > 0$, the consistency condition gives

$$\frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon} - \gamma \left( \frac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \frac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h} \right) = 0.$$

Therefore,

$$\gamma = \frac{\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\varepsilon}}{\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \dfrac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h}}.$$

With the definitions of $A$ and $B$ above, the plastic multiplier for plastic loading is then given by

$$\gamma = \frac{A}{B}.$$

However, this expression was obtained under the assumption that $\gamma > 0$. It cannot be used directly for unloading, because in that case $A < 0$ and the expression above would give a negative plastic multiplier, which is inadmissible. The plastic multiplier must satisfy the Kuhn–Tucker condition $\gamma \geq 0$. Therefore, the positive part of $A$ is taken by introducing the Macaulay brackets,

$$\langle x \rangle = \frac{x + |x|}{2}.$$

Thus, assuming $B > 0$, the loading/unloading form on the yield surface may be written as

$$\gamma = \frac{\langle A \rangle}{B}.$$

This expression enforces the unilateral character of the plastic flow. If $A > 0$, then $\gamma = A/B > 0$ and plastic loading occurs. If $A = 0$, then $\gamma = 0$ and the response is neutral loading. If $A < 0$, then $\langle A \rangle = 0$, so that $\gamma = 0$, corresponding to elastic unloading from the yield surface.

Substituting the plastic multiplier in the rate form of the stress,

$$\dot{\boldsymbol{\sigma}} = \mathbb{C} : (\dot{\varepsilon} - \gamma\, \mathbf{n}),$$

gives

$$\dot{\boldsymbol{\sigma}} = \mathbb{C} : \left( \dot{\varepsilon} - \frac{\partial_{\boldsymbol{\sigma}} f : \mathbb{C} : \dot{\varepsilon}}{\partial_{\boldsymbol{\sigma}} f : \mathbb{C} : \mathbf{n} - \partial_{\mathbf{q}} f \cdot \mathbf{h}} \mathbf{n} \right).$$

Remember that

$$\mathbf{a}\,(\mathbf{b} : \mathbf{u}) = (\mathbf{a} \otimes \mathbf{b}) : \mathbf{u}.$$

Then,

$$\mathbf{n}\,[(\partial_{\boldsymbol{\sigma}} f : \mathbb{C}) : \dot{\varepsilon}] = [\mathbf{n} \otimes (\partial_{\boldsymbol{\sigma}} f : \mathbb{C})] : \dot{\varepsilon}.$$

Finally, the tangent elastoplastic moduli tensor is defined:

$$\dot{\boldsymbol{\sigma}} = \mathbb{C}^{\text{ep}} : \dot{\varepsilon},$$

with

$$\mathbb{C}^{\text{ep}} = \begin{cases} \mathbb{C}, & \text{if } \gamma = 0, \\[2ex] \mathbb{C} - \dfrac{(\mathbb{C} : \mathbf{n}) \otimes \left( \dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} \right)}{\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \dfrac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h}}, & \text{if } \gamma > 0. \end{cases}$$

Considering an ideal associative flow rule:

$$\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q}) = \frac{\partial f}{\partial \boldsymbol{\sigma}}(\boldsymbol{\sigma}, \mathbf{q}),$$

$$\mathbf{h} = \mathbf{0}.$$

Then,

$$\mathbb{C}^{\mathrm{ep}} = \begin{cases} \mathbb{C}, & \text{if } \gamma = 0, \\ \mathbb{C} - \dfrac{\mathbb{C} : \dfrac{\partial f}{\partial \boldsymbol{\sigma}} \otimes \dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C}}{\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dfrac{\partial f}{\partial \boldsymbol{\sigma}}}, & \text{if } \gamma > 0. \end{cases}$$

For this particular case, the tangent elastoplastic moduli is symmetric. For general hardening/-softening or non-associative plasticity, the tangent need not be symmetric.

# 2 Implicit integration algorithm

## 2.1 Strain-driven constitutive update

The local constitutive update at a Gauss point is strain driven.

At the beginning of the step the known state is

$$\{\varepsilon_n, \varepsilon_n^{\mathrm{p}}, \mathbf{q}_n\}. \tag{8}$$

The total strain at the new step is prescribed by the global finite element iteration,

$$\varepsilon_{n+1} = \varepsilon_n + \Delta\varepsilon_{n+1}. \tag{9}$$

In a small-strain finite element discretization,

$$\Delta\varepsilon_{n+1} = \mathbf{B}\,\Delta\mathbf{u}_{n+1}, \tag{10}$$

where **B** is the standard strain-displacement matrix.

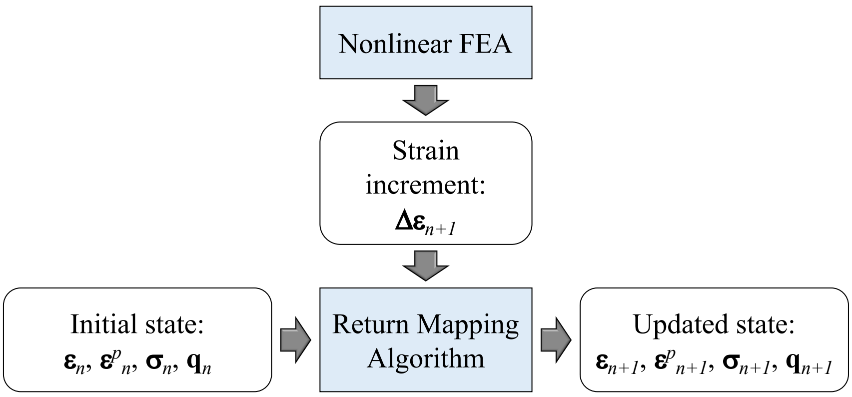


Figure 4: Interaction between the global Newton iteration and the local constitutive update at Gauss points.

## 2.2 Elastic predictor

The backward-Euler algorithm first freezes the plastic variables and computes the trial elastic state,

$$\varepsilon_{n+1}^{\mathrm{e,trial}} = \varepsilon_{n+1} - \varepsilon_n^{\mathrm{p}}, \tag{11}$$

$$\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}} = \mathbb{C} : \varepsilon_{n+1}^{\mathrm{e,trial}}, \qquad \mathbf{q}_{n+1}^{\mathrm{trial}} = \mathbf{q}_n, \tag{12}$$

$$f_{n+1}^{\mathrm{trial}} = f(\boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}, \mathbf{q}_n). \tag{13}$$

The elastic admissibility check is

$$f_{n+1}^{\mathrm{trial}} \le 0. \tag{14}$$

If (14) holds, the step is elastic:

$$\Delta\gamma = 0, \qquad \boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}, \qquad \varepsilon_{n+1}^{\mathrm{p}} = \varepsilon_n^{\mathrm{p}}, \qquad \mathbf{q}_{n+1} = \mathbf{q}_n. \tag{15}$$

In numerical implementation the inequality is evaluated with a tolerance.

## 2.3 Plastic corrector and discrete Kuhn–Tucker conditions

If $f_{n+1}^{\text{trial}} > 0$, the trial state is inadmissible. Then, to update the stress state we adopt Simo et al. [2] plastic correction, solving the system of equations:

$$\begin{cases} \varepsilon_{n+1}^{\text{p}} = \varepsilon_n^{\text{p}} + \Delta\gamma_{n+1}\,\mathbf{n}_{n+1}, \\ \mathbf{q}_{n+1} = \mathbf{q}_n + \Delta\gamma_{n+1}\,\mathbf{h}_{n+1}, \\ f(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) = 0. \end{cases} \tag{16}$$

where

$$\mathbf{n}_{n+1} = \mathbf{n}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}), \qquad \mathbf{h}_{n+1} = \mathbf{h}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}). \tag{17}$$

The directions are evaluated at the corrected state, not at the trial state, unless an explicit or cutting-plane variant is being derived.

The Kuhn–Tucker conditions must also be satisfied:

$$\Delta\gamma_{n+1} \geq 0, \qquad f(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \leq 0, \qquad \Delta\gamma_{n+1}\, f(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) = 0. \tag{18}$$

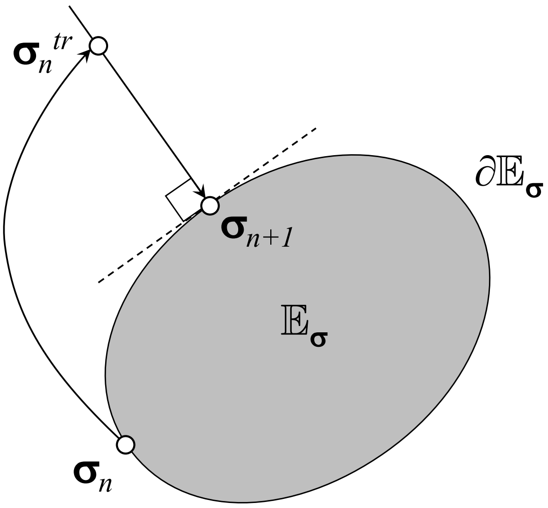


Figure 5: Local Newton iteration for the fully implicit return-mapping problem.

## 2.4 Fully implicit return mapping algorithm

The fully implicit return mapping is obtained by solving the nonlinear system associated with the plastic corrector. Following the closest point projection derivation, first write the residual equations in terms of the plastic strain:

$$\mathbf{r}_{1,n+1} \coloneqq -\varepsilon_{n+1}^{\text{p}} + \varepsilon_n^{\text{p}} + \Delta\gamma_{n+1}\,\mathbf{n}_{n+1}, \tag{19}$$

$$\mathbf{r}_{2,n+1} \coloneqq -\mathbf{q}_{n+1} + \mathbf{q}_n + \Delta\gamma_{n+1}\,\mathbf{h}_{n+1}, \tag{20}$$

$$r_{3,n+1} \coloneqq f(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}). \tag{21}$$

The first equation is the backward-Euler plastic flow rule, the second equation is the backward-Euler hardening/softening law, and the third equation imposes the active yield constraint.

The stress and plastic strain are related by the elastic law,

$$\boldsymbol{\sigma}_{n+1} = \mathbb{C}_{n+1} : (\varepsilon_{n+1} - \varepsilon^{\mathrm{p}}_{n+1}), \tag{22}$$

and therefore

$$-\varepsilon^{\mathrm{p}}_{n+1} = \mathbb{C}^{-1}_{n+1} : \boldsymbol{\sigma}_{n+1} - \varepsilon_{n+1}. \tag{23}$$

Substitution of (23) into $\mathbf{r}_{1,n+1}$ gives:

$$\mathbf{r}_{1,n+1} \coloneqq \mathbb{C}^{-1}_{n+1} : \boldsymbol{\sigma}_{n+1} - \varepsilon_{n+1} + \varepsilon^{\mathrm{p}}_{n} + \Delta\gamma_{n+1}\,\mathbf{n}_{n+1}, \tag{24}$$

$$\mathbf{r}_{2,n+1} \coloneqq -\mathbf{q}_{n+1} + \mathbf{q}_{n} + \Delta\gamma_{n+1}\,\mathbf{h}_{n+1}, \tag{25}$$

$$r_{3,n+1} \coloneqq f(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}). \tag{26}$$

The residual vector is

$$\mathbf{r}_{n+1} \coloneqq \begin{bmatrix} \mathbf{r}_{1,n+1} \\ \mathbf{r}_{2,n+1} \\ r_{3,n+1} \end{bmatrix}. \tag{27}$$

Linearizing the residual vector gives

$$\mathbf{r}_{n+1} \simeq \mathbf{r}^{(k)}_{n+1} + \begin{bmatrix} \dfrac{\partial \mathbf{r}_{1,n+1}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial \mathbf{r}_{1,n+1}}{\partial \mathbf{q}} & \dfrac{\partial \mathbf{r}_{1,n+1}}{\partial (\Delta\gamma)} \\ \dfrac{\partial \mathbf{r}_{2,n+1}}{\partial \boldsymbol{\sigma}} & \dfrac{\partial \mathbf{r}_{2,n+1}}{\partial \mathbf{q}} & \dfrac{\partial \mathbf{r}_{2,n+1}}{\partial (\Delta\gamma)} \\ \left(\dfrac{\partial r_{3,n+1}}{\partial \boldsymbol{\sigma}}\right)^{\top} & \left(\dfrac{\partial r_{3,n+1}}{\partial \mathbf{q}}\right)^{\top} & \dfrac{\partial r_{3,n+1}}{\partial (\Delta\gamma)} \end{bmatrix}^{(k)} \begin{bmatrix} \boldsymbol{\sigma}^{(k+1)}_{n+1} - \boldsymbol{\sigma}^{(k)}_{n+1} \\ \mathbf{q}^{(k+1)}_{n+1} - \mathbf{q}^{(k)}_{n+1} \\ \Delta\gamma^{(k+1)}_{n+1} - \Delta\gamma^{(k)}_{n+1} \end{bmatrix}. \tag{28}$$

Defining the variables increments at the $k$-th iteration as

$$\begin{aligned} \delta\boldsymbol{\sigma}^{(k)}_{n+1} &\coloneqq \boldsymbol{\sigma}^{(k+1)}_{n+1} - \boldsymbol{\sigma}^{(k)}_{n+1}, \\ \delta\mathbf{q}^{(k)}_{n+1} &\coloneqq \mathbf{q}^{(k+1)}_{n+1} - \mathbf{q}^{(k)}_{n+1}, \\ \delta(\Delta\gamma^{(k)}_{n+1}) &\coloneqq \Delta\gamma^{(k+1)}_{n+1} - \Delta\gamma^{(k)}_{n+1}. \end{aligned} \tag{29}$$

Using the Newton–Raphson method, setting $\mathbf{r}_{n+1} = \mathbf{0}$, so that

$$\begin{bmatrix} \mathbb{C}^{-1}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}_{n+1} & \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{n}_{n+1} & \mathbf{n}_{n+1} \\ \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{h}_{n+1} & -\mathbf{I} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{h}_{n+1} & \mathbf{h}_{n+1} \\ (\partial_{\boldsymbol{\sigma}} f_{n+1})^{\top} & \left(\partial_{\mathbf{q}} f_{n+1}\right)^{\top} & 0 \end{bmatrix}^{(k)} \begin{bmatrix} \delta\boldsymbol{\sigma}^{(k)}_{n+1} \\ \delta\mathbf{q}^{(k)}_{n+1} \\ \delta(\Delta\gamma^{(k)}_{n+1}) \end{bmatrix} = - \begin{bmatrix} \mathbf{r}^{(k)}_{1,n+1} \\ \mathbf{r}^{(k)}_{2,n+1} \\ r^{(k)}_{3,n+1} \end{bmatrix}. \tag{30}$$

All functions and derivatives in (30) are evaluated at $(\boldsymbol{\sigma}^{(k)}_{n+1}, \mathbf{q}^{(k)}_{n+1}, \Delta\gamma^{(k)}_{n+1})$.

To simplify the nomenclature, define the block operators

$$\left(\boldsymbol{\Psi}^{(k)}_{n+1}\right)^{-1} \coloneqq \begin{bmatrix} \mathbb{C}^{-1}_{n+1} + \Delta\gamma^{(k)}_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}^{(k)}_{n+1} & \Delta\gamma^{(k)}_{n+1}\,\partial_{\mathbf{q}}\mathbf{n}^{(k)}_{n+1} \\ \Delta\gamma^{(k)}_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{h}^{(k)}_{n+1} & -\mathbf{I} + \Delta\gamma^{(k)}_{n+1}\,\partial_{\mathbf{q}}\mathbf{h}^{(k)}_{n+1} \end{bmatrix}, \tag{31}$$

$$\boldsymbol{\rho}^{(k)}_{n+1} \coloneqq \begin{bmatrix} \mathbf{r}^{(k)}_{1,n+1} \\ \mathbf{r}^{(k)}_{2,n+1} \end{bmatrix}, \qquad \boldsymbol{\delta}^{(k)}_{n+1} \coloneqq \begin{bmatrix} \delta\boldsymbol{\sigma}^{(k)}_{n+1} \\ \delta\mathbf{q}^{(k)}_{n+1} \end{bmatrix}, \tag{32}$$

and

$$\mathbf{g}_{n+1}^{(k)\top} := \begin{bmatrix} \partial_{\boldsymbol{\sigma}} f_{n+1}^{(k)} & \partial_{\mathbf{q}} f_{n+1}^{(k)} \end{bmatrix}, \qquad \mathbf{e}_{n+1}^{(k)} := \begin{bmatrix} \mathbf{n}_{n+1}^{(k)} \\ \mathbf{h}_{n+1}^{(k)} \end{bmatrix}. \tag{33}$$

With these definitions, the Newton system can be written in the compact form

$$\begin{bmatrix} \left(\boldsymbol{\Psi}_{n+1}^{(k)}\right)^{-1} & \mathbf{e}_{n+1}^{(k)} \\ \mathbf{g}_{n+1}^{(k)\top} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{n+1}^{(k)} \\ \delta(\Delta\gamma_{n+1}^{(k)}) \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\rho}_{n+1}^{(k)} \\ f_{n+1}^{(k)} \end{bmatrix}. \tag{34}$$

The first block row of (34) is

$$\left(\boldsymbol{\Psi}_{n+1}^{(k)}\right)^{-1} \boldsymbol{\delta}_{n+1}^{(k)} + \mathbf{e}_{n+1}^{(k)}\, \delta(\Delta\gamma_{n+1}^{(k)}) = -\boldsymbol{\rho}_{n+1}^{(k)}, \tag{35}$$

and the second block row is

$$\mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\delta}_{n+1}^{(k)} = -f_{n+1}^{(k)}. \tag{36}$$

Premultiplying (35) by $\mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\Psi}_{n+1}^{(k)}$ gives

$$\mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\delta}_{n+1}^{(k)} + \mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\Psi}_{n+1}^{(k)}\, \mathbf{e}_{n+1}^{(k)}\, \delta(\Delta\gamma_{n+1}^{(k)}) = -\mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\Psi}_{n+1}^{(k)}\, \boldsymbol{\rho}_{n+1}^{(k)}. \tag{37}$$

Substituting (36) into (37), the Newton correction for the plastic multiplier increment is

$$\delta(\Delta\gamma_{n+1}^{(k)}) = \frac{f_{n+1}^{(k)} - \mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\Psi}_{n+1}^{(k)}\, \boldsymbol{\rho}_{n+1}^{(k)}}{\mathbf{g}_{n+1}^{(k)\top}\, \boldsymbol{\Psi}_{n+1}^{(k)}\, \mathbf{e}_{n+1}^{(k)}}. \tag{38}$$

Then the stress and internal-variable corrections follow from (35):

$$\boldsymbol{\delta}_{n+1}^{(k)} = -\boldsymbol{\Psi}_{n+1}^{(k)} \left(\boldsymbol{\rho}_{n+1}^{(k)} + \mathbf{e}_{n+1}^{(k)}\, \delta(\Delta\gamma_{n+1}^{(k)})\right). \tag{39}$$

The closest point projection algorithm can therefore be stated as follows:

1) Initialize $k = 0$, $\boldsymbol{\sigma}_{n+1}^{(0)} = \boldsymbol{\sigma}_{n+1}^{\mathrm{trial}}$, $\mathbf{q}_{n+1}^{(0)} = \mathbf{q}_n$, and $\Delta\gamma_{n+1}^{(0)} = 0$. If plastic strain is stored during the iteration, set $\varepsilon_{n+1}^{p,(0)} = \varepsilon_n^{\mathrm{p}}$.

2) Evaluate $f_{n+1}^{(k)} = f(\boldsymbol{\sigma}_{n+1}^{(k)}, \mathbf{q}_{n+1}^{(k)})$, $\boldsymbol{\rho}_{n+1}^{(k)}$, $\mathbf{e}_{n+1}^{(k)}$, and $\mathbf{g}_{n+1}^{(k)}$. Stop when

$$\left\|\boldsymbol{\rho}_{n+1}^{(k)}\right\| \leq \mathrm{tol}_r, \qquad |f_{n+1}^{(k)}| \leq \mathrm{tol}_f. \tag{40}$$

3) Compute $\boldsymbol{\Psi}_{n+1}^{(k)}$ from (31) and determine $\delta(\Delta\gamma_{n+1}^{(k)})$ from (38).

4) Compute $\boldsymbol{\delta}_{n+1}^{(k)}$ from (39).

5) Update the state variables:

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + \delta\boldsymbol{\sigma}_{n+1}^{(k)}, \tag{41}$$

$$\mathbf{q}_{n+1}^{(k+1)} = \mathbf{q}_{n+1}^{(k)} + \delta\mathbf{q}_{n+1}^{(k)}, \tag{42}$$

$$\Delta\gamma_{n+1}^{(k+1)} = \Delta\gamma_{n+1}^{(k)} + \delta(\Delta\gamma_{n+1}^{(k)}). \tag{43}$$

If plastic strain is stored during the iteration, update it from the elastic relation at fixed total strain:

$$\varepsilon_{n+1}^{p,(k+1)} = \varepsilon_{n+1}^{p,(k)} - \mathbb{C}_{n+1}^{-1} : \delta\boldsymbol{\sigma}_{n+1}^{(k)}. \tag{44}$$

6) Set $k \leftarrow k+1$ and return to step 2).

After convergence, the accepted state must satisfy the discrete Kuhn–Tucker conditions (18); in particular, $\Delta\gamma_{n+1} \geq 0$.

# 3 Consistent tangent constitutive matrix

Previously, the continuum elastoplastic tangent tensor, $\mathbb{C}^{\mathrm{ep}}$, was defined from the rate relation

$$\dot{\boldsymbol{\sigma}} = \mathbb{C}^{\mathrm{ep}} : \dot{\boldsymbol{\varepsilon}}, \tag{45}$$

with

$$\mathbb{C}^{\mathrm{ep}} = \begin{cases} \mathbb{C}, & \text{if } \gamma = 0, \\ \mathbb{C} - \dfrac{(\mathbb{C} : \mathbf{n}) \otimes \left(\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C}\right)}{\dfrac{\partial f}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \mathbf{n} - \dfrac{\partial f}{\partial \mathbf{q}} \cdot \mathbf{h}}, & \text{if } \gamma > 0. \end{cases} \tag{46}$$

This tensor gives an exact relation between differential stress and strain increments. However, in an algorithmic process the stress update is obtained from finite increments through the return-mapping equations. The tangent used in the global Newton–Raphson method must therefore be consistent with the linearization of the discrete stress-update algorithm [3]. Thus, a consistent, or algorithmic, tangent constitutive tensor is defined by

$$\mathrm{d}\boldsymbol{\sigma}_{n+1} = \mathbb{C}^{\mathrm{alg}}_{n+1} : \mathrm{d}\boldsymbol{\varepsilon}_{n+1}, \qquad \mathbf{D}^{\mathrm{alg}}_{n+1} = \left[\mathbb{C}^{\mathrm{alg}}_{n+1}\right]_{\mathrm{Voigt}}. \tag{47}$$

To derive this tensor, consider the converged plastic corrector system

$$\begin{cases} \boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} = \boldsymbol{\varepsilon}^{\mathrm{p}}_{n} + \Delta\gamma_{n+1}\,\mathbf{n}_{n+1}, \\ \mathbf{q}_{n+1} = \mathbf{q}_n + \Delta\gamma_{n+1}\,\mathbf{h}_{n+1}, \\ f_{n+1} = 0. \end{cases} \tag{48}$$

Taking differentials gives

$$\begin{cases} \mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} = \mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}_{n} + \mathrm{d}(\Delta\gamma_{n+1})\,\mathbf{n}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{n}_{n+1} \cdot \mathrm{d}\mathbf{q}_{n+1}, \\ \mathrm{d}\mathbf{q}_{n+1} = \mathrm{d}\mathbf{q}_n + \mathrm{d}(\Delta\gamma_{n+1})\,\mathbf{h}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{h}_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{h}_{n+1} \cdot \mathrm{d}\mathbf{q}_{n+1}, \\ \partial_{\boldsymbol{\sigma}} f_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \partial_{\mathbf{q}} f_{n+1} \cdot \mathrm{d}\mathbf{q}_{n+1} = 0. \end{cases} \tag{49}$$

The variables at step $n$ are known during the local linearization; therefore

$$\mathrm{d}\boldsymbol{\varepsilon}^{\mathrm{p}}_{n} = \mathbf{0}, \qquad \mathrm{d}\mathbf{q}_n = \mathbf{0}. \tag{50}$$

Moreover, since

$$\boldsymbol{\sigma}_{n+1} = \mathbb{C}_{n+1} : \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1}\right), \qquad -\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} = \mathbb{C}^{-1}_{n+1} : \boldsymbol{\sigma}_{n+1} - \boldsymbol{\varepsilon}_{n+1}, \tag{51}$$

one obtains

$$\mathrm{d}\varepsilon^{\mathrm{p}}_{n+1} = \mathrm{d}\varepsilon_{n+1} - \mathbb{C}^{-1}_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1}. \tag{52}$$

Substituting (50) and (52) into (49) gives

$$\begin{cases} \mathrm{d}\varepsilon_{n+1} = \left(\mathbb{C}^{-1}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}_{n+1}\right) : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{n}_{n+1}\cdot \mathrm{d}\mathbf{q}_{n+1} + \mathbf{n}_{n+1}\ \mathrm{d}(\Delta\gamma_{n+1}), \\ \mathbf{0} = \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{h}_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \left(-\mathbf{I} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{h}_{n+1}\right)\cdot \mathrm{d}\mathbf{q}_{n+1} + \mathbf{h}_{n+1}\ \mathrm{d}(\Delta\gamma_{n+1}), \\ 0 = \partial_{\boldsymbol{\sigma}} f_{n+1} : \mathrm{d}\boldsymbol{\sigma}_{n+1} + \partial_{\mathbf{q}} f_{n+1}\cdot \mathrm{d}\mathbf{q}_{n+1}. \end{cases} \tag{53}$$

In matrix notation,

$$\underbrace{\begin{bmatrix} \mathbb{C}^{-1}_{n+1} + \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}_{n+1} & \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{n}_{n+1} & \mathbf{n}_{n+1} \\ \Delta\gamma_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{h}_{n+1} & -\mathbf{I} + \Delta\gamma_{n+1}\,\partial_{\mathbf{q}}\mathbf{h}_{n+1} & \mathbf{h}_{n+1} \\ \left(\partial_{\boldsymbol{\sigma}} f_{n+1}\right)^{\top} & \left(\partial_{\mathbf{q}} f_{n+1}\right)^{\top} & 0 \end{bmatrix}}_{\mathbf{K}_{\mathrm{alg},n+1}} \begin{bmatrix} \mathrm{d}\boldsymbol{\sigma}_{n+1} \\ \mathrm{d}\mathbf{q}_{n+1} \\ \mathrm{d}(\Delta\gamma_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathrm{d}\varepsilon_{n+1} \\ \mathbf{0} \\ 0 \end{bmatrix}. \tag{54}$$

All functions and derivatives in (54) are evaluated at the converged state $(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\gamma_{n+1})$. The inverse relation is written as

$$\begin{bmatrix} \mathrm{d}\boldsymbol{\sigma}_{n+1} \\ \mathrm{d}\mathbf{q}_{n+1} \\ \mathrm{d}(\Delta\gamma_{n+1}) \end{bmatrix} = \mathbf{K}^{-1}_{\mathrm{alg},n+1} \begin{bmatrix} \mathrm{d}\varepsilon_{n+1} \\ \mathbf{0} \\ 0 \end{bmatrix}. \tag{55}$$

Equivalently, define the block inverse

$$\mathbf{K}^{-1}_{\mathrm{alg},n+1} = \begin{bmatrix} \mathbb{D}_{11,n+1} & \mathbf{D}_{12,n+1} & \mathbf{d}_{13,n+1} \\ \mathbf{D}_{21,n+1} & \mathbf{D}_{22,n+1} & \mathbf{d}_{23,n+1} \\ \mathbf{d}^{\top}_{31,n+1} & \mathbf{d}^{\top}_{32,n+1} & d_{33,n+1} \end{bmatrix}. \tag{56}$$

Since the right-hand side of (55) contains only $\mathrm{d}\varepsilon_{n+1}$ in its first block, the stress differential is

$$\mathrm{d}\boldsymbol{\sigma}_{n+1} = \mathbb{D}_{11,n+1} : \mathrm{d}\varepsilon_{n+1}. \tag{57}$$

Therefore, the consistent tangent constitutive tensor is

$$\mathbb{C}^{\mathrm{alg}}_{n+1} = \mathbb{D}_{11,n+1}, \qquad \mathbf{D}^{\mathrm{alg}}_{n+1} = \left[\mathbb{D}_{11,n+1}\right]_{\mathrm{Voigt}}. \tag{58}$$

The value of $\Delta\gamma_{n+1}$ appearing in (54) is the converged plastic multiplier increment obtained from the return-mapping algorithm.

# 4 PorousLab implementation of mechanical laws

The mechanical constitutive models in PorousLab are organized as a small class hierarchy. The base class MechanicalLaw provides tensor-vector utilities and the invariant operations used by the yield functions. In particular, it defines the hydrostatic stress, deviatoric stress, stress invariants, and the gradients and Hessians used in the return mapping. These definitions are collected in Appendix A.

The class MechanicalLinearElastic adds the linear elastic law. It computes the elastic constitutive matrix $\mathbf{D}^e$, the elastic flexibility matrix $\mathbf{C}^e$, the shear modulus $G$, and the bulk modulus $K$. The elastic trial stress used by the elastoplastic models is

$$\boldsymbol{\sigma}_{n+1}^{\text{trial}} = \mathbf{D}^e \left(\varepsilon_{n+1} - \varepsilon_n\right) + \boldsymbol{\sigma}_n. \tag{59}$$

This is the form implemented by the common elastoplastic evaluator.

The class MechanicalElastoPlastic is the shared implicit elastoplastic driver. It declares the model-dependent methods:

- yieldCondition,
- yieldStressGradient,
- flowVector,
- flowVectorGradient,
- hardening,
- hardeningStressGradient.

The Von Mises, Drucker–Prager, and Mohr–Coulomb classes provide these model-specific pieces. The common driver then performs the same return-mapping loop for each model.

For the implicit path, the state variables used by the integration point are the total strain, previous strain, previous stress, and previous plastic strain. The residual solved in the common return mapping is the plastic-strain update residual

$$\mathbf{r}_{n+1} = -\varepsilon_{n+1}^{\text{p}} + \varepsilon_n^{\text{p}} + \Delta\gamma_{n+1}\,\mathbf{n}_{n+1}. \tag{60}$$

This sign is consistent with the convention adopted in this document: $\varepsilon_{n+1}^{\text{p}} = \varepsilon_n^{\text{p}} + \Delta\gamma_{n+1}\mathbf{n}_{n+1}$. Models with scalar hardening contribute a scalar $h$ to the denominator of the plastic-multiplier correction. Perfectly plastic models return $h = 0$.

Listing 1 summarizes the shared implicit implementation.

Listing 1: Shared implicit return mapping used by elastoplastic laws.

```
De = elasticConstitutiveMatrix(material, ip)
Ce = elasticFlexibilityMatrix(material, ip)
Dt = De

stress = De * (ip.strain - ip.strainOld) + ip.stressOld
f = yieldCondition(material, ip, stress)

if f < 0
    % Elastic step.
    return
end

lambda = 0
ep = ip.plasticstrainOld
epOld = ip.plasticstrainOld
r = zeros(4, 1)

while abs(f) > yieldTolerance || norm(r) > residualTolerance
    n  = flowVector(material, ip, stress)
    df = yieldStressGradient(material, ip, stress)
    dn = flowVectorGradient(material, ip, stress)
    h  = hardening(material, ip, stress)

    Psi = Ce + lambda * dn
    dlambda = (f - df' * (Psi \ r)) / (df' * (Psi \ n) + h)

    dstress = -Psi \ (r + dlambda * n)
    stress = stress + dstress
    lambda = lambda + dlambda

    ep = epOld + lambda * n
    f = yieldCondition(material, ip, stress)
    r = -ep + epOld + lambda * n
end

ip.plasticstrain = ep
Psi = inv(Ce + lambda * dn)
Dt = Psi - (Psi * (n * df') * Psi) / (df' * (Psi * n))
```

The final matrix $\mathbf{D}_t$ is the algorithmic tangent returned to the finite element assembly. In the notation of the preceding sections,

$$\mathbf{\Psi} = \left(\mathbf{C}^e + \Delta\gamma_{n+1}\,\partial_\sigma \mathbf{n}_{n+1}\right)^{-1}, \qquad \mathbf{D}_t = \mathbf{\Psi} - \frac{\mathbf{\Psi}\,\mathbf{n}_{n+1}\,(\partial_\sigma f_{n+1})^\top\,\mathbf{\Psi}}{(\partial_\sigma f_{n+1})^\top\,\mathbf{\Psi}\,\mathbf{n}_{n+1}}. \tag{61}$$

Drucker–Prager and Mohr–Coulomb also include optional alternative integration paths. These paths are specialized closed-form or principal-space returns and are described in the corresponding model sections.

# 5 Von Mises plasticity

The PorousLab implementation is given in MechanicalElastoPlasticVonMises. It uses the stress vector convention and invariant definitions summarized in Appendix A. The von Mises equivalent stress is

$$\sigma_{\mathrm{VM}} = \sqrt{3J_2}. \tag{62}$$

The yield function implemented in the code is

$$f(\boldsymbol{\sigma}) = \sigma_{\mathrm{VM}} - \sigma_{y0}, \tag{63}$$

where $\sigma_{y0}$ is the material parameter sy0. The material parameter Kp is returned by the model-specific hardening method. In the current implementation, the yield stress is not updated from a stored hardening variable; the commented term $\sigma_y = \sigma_{y0} + K_p\alpha$ is not active in the yield function.

The yield-function gradient is

$$\partial_{\boldsymbol{\sigma}} f = \partial_{\boldsymbol{\sigma}}\sigma_{\mathrm{VM}} = \frac{\sqrt{3}}{2\sqrt{J_2}}\,\partial_{\boldsymbol{\sigma}} J_2. \tag{64}$$

For numerical robustness, the code evaluates this expression with $J_2 \leftarrow \max(J_2, 10^{-12})$ in the gradient.

The flow vector returned by the implementation is the normalized deviatoric stress vector,

$$\mathbf{n}_{\mathrm{VM}} = \frac{\mathbf{s}}{\|\mathbf{s}\|}, \qquad \mathbf{s} = \boldsymbol{\sigma} - p\,\mathbf{m}, \tag{65}$$

where $p = I_1/3$ and $\mathbf{m} = [1, 1, 1, 0]^\top$. The corresponding flow-vector gradient used by the code is the Hessian of the von Mises equivalent stress,

$$\partial_{\boldsymbol{\sigma}}\mathbf{n}_{\mathrm{VM}} = \partial^2_{\boldsymbol{\sigma}\boldsymbol{\sigma}}\sigma_{\mathrm{VM}}, \tag{66}$$

with the expression given in Appendix A. The hydrostatic axis, $J_2 = 0$, is a singular point of this direction and must be regularized in implementation.

The scalar hardening terms supplied to the common return-mapping algorithm are

$$h = K_p, \qquad \partial_{\boldsymbol{\sigma}} h = \mathbf{0}. \tag{67}$$

The inherited implicit stress update uses the residual

$$\mathbf{r}_{n+1} = -\boldsymbol{\varepsilon}^{\mathrm{p}}_{n+1} + \boldsymbol{\varepsilon}^{\mathrm{p}}_{n} + \Delta\gamma_{n+1}\,\mathbf{n}_{\mathrm{VM},n+1} = \mathbf{0}. \tag{68}$$

At iteration $k$, the scalar plastic-multiplier correction is

$$\delta(\Delta\gamma^{(k)}_{n+1}) = \frac{f^{(k)}_{n+1} - \left(\partial_{\boldsymbol{\sigma}} f^{(k)}_{n+1}\right)^\top \boldsymbol{\Psi}^{(k)}_{n+1}\mathbf{r}^{(k)}_{n+1}}{\left(\partial_{\boldsymbol{\sigma}} f^{(k)}_{n+1}\right)^\top \boldsymbol{\Psi}^{(k)}_{n+1}\mathbf{n}^{(k)}_{\mathrm{VM},n+1} + K_p}, \tag{69}$$

where

$$\left(\boldsymbol{\Psi}^{(k)}_{n+1}\right)^{-1} = \mathbb{C}^{-1}_{n+1} + \Delta\gamma^{(k)}_{n+1}\,\partial_{\boldsymbol{\sigma}}\mathbf{n}^{(k)}_{\mathrm{VM},n+1}. \tag{70}$$

# 6 Drucker–Prager plasticity

The PorousLab implementation is given in MechanicalElastoPlasticDruckerPrager. The model is written in terms of the hydrostatic stress $p = I_1/3$ and the second deviatoric invariant $J_2$. The smooth Drucker–Prager yield function is

$$f(\boldsymbol{\sigma}) = \sqrt{J_2} + \eta\, p - \xi\, c, \tag{71}$$

where $c$ is the cohesion. The plastic potential is

$$g(\boldsymbol{\sigma}) = \sqrt{J_2} + \bar{\eta}\, p, \tag{72}$$

so the flow is nonassociated when $\bar{\eta} \neq \eta$.

The coefficients $\eta$, $\xi$, and $\bar{\eta}$ are obtained by matching the Drucker–Prager cone to a Mohr–Coulomb cone. For the inner approximation,

$$\xi = \frac{6\cos\phi}{\sqrt{3}(3+\sin\phi)}, \qquad \eta = \frac{6\sin\phi}{\sqrt{3}(3+\sin\phi)}, \qquad \bar{\eta} = \frac{6\sin\psi}{\sqrt{3}(3+\sin\psi)}. \tag{73}$$

For the outer approximation,

$$\xi = \frac{6\cos\phi}{\sqrt{3}(3-\sin\phi)}, \qquad \eta = \frac{6\sin\phi}{\sqrt{3}(3-\sin\phi)}, \qquad \bar{\eta} = \frac{6\sin\psi}{\sqrt{3}(3-\sin\psi)}. \tag{74}$$

For the plane-strain approximation,

$$\xi = \frac{3}{\sqrt{9+12\tan^2\phi}}, \qquad \eta = \frac{3\tan\phi}{\sqrt{9+12\tan^2\phi}}, \qquad \bar{\eta} = \frac{3\tan\psi}{\sqrt{9+12\tan^2\psi}}. \tag{75}$$

The yield-function gradient is

$$\partial_{\boldsymbol{\sigma}} f = \frac{\eta}{3}\,\partial_{\boldsymbol{\sigma}} I_1 + \frac{1}{2\sqrt{J_2}}\,\partial_{\boldsymbol{\sigma}} J_2, \tag{76}$$

and the flow vector is

$$\mathbf{n} = \partial_{\boldsymbol{\sigma}} g = \frac{\bar{\eta}}{3}\,\partial_{\boldsymbol{\sigma}} I_1 + \frac{1}{2\sqrt{J_2}}\,\partial_{\boldsymbol{\sigma}} J_2. \tag{77}$$

The implementation sets the third component of these vectors to zero in plane stress. The flow-vector gradient is the Hessian of $\sqrt{J_2}$:

$$\partial_{\boldsymbol{\sigma}}\mathbf{n} = \frac{1}{2\sqrt{J_2}}\,\partial^2_{\boldsymbol{\sigma}\boldsymbol{\sigma}} J_2 - \frac{1}{4J_2^{3/2}}\,(\partial_{\boldsymbol{\sigma}} J_2)\,(\partial_{\boldsymbol{\sigma}} J_2)^\top\,. \tag{78}$$

At $J_2 = 0$, the code sets the invariant derivatives of $\sqrt{J_2}$ to zero in the implicit path. The model is perfectly plastic in the current implementation:

$$h = 0, \qquad \partial_{\boldsymbol{\sigma}} h = \mathbf{0}. \tag{79}$$

The inherited implicit integration path uses the common closest-point Newton structure with $\partial_{\boldsymbol{\sigma}} f$, $\mathbf{n}$, and $\partial_{\boldsymbol{\sigma}}\mathbf{n}$ from (76)–(78). The class also contains an alternative smooth-cone return. In that algorithm, for an inadmissible trial state,

$$\Delta\gamma_{n+1} = \frac{f^{\text{trial}}_{n+1}}{G + K\eta\bar{\eta}}, \tag{80}$$

where $K$ and $G$ are the bulk and shear moduli. The smooth-cone return is

$$\mathbf{s}_{n+1} = \left(1 - \frac{G\,\Delta\gamma_{n+1}}{\sqrt{J_2^{\text{trial}}}}\right)\mathbf{s}_{n+1}^{\text{trial}}, \qquad p_{n+1} = p_{n+1}^{\text{trial}} - K\bar{\eta}\,\Delta\gamma_{n+1}. \tag{81}$$

If the update reaches the apex, the implemented stress is hydrostatic:

$$\boldsymbol{\sigma}_{n+1} = \frac{\xi c}{\eta}\,\mathbf{m}. \tag{82}$$

# 7 Mohr–Coulomb plasticity

The `PorousLab` implementation is given in `MechanicalElastoPlasticMohrCoulomb`. Two integration paths are available. The inherited implicit path uses a smooth invariant form of the Mohr–Coulomb surface, while the alternative path performs a return in principal-stress space with main-plane, edge, and apex branches.

## 7.1 Smooth invariant form

Let

$$J = \sqrt{J_2}, \qquad x = -\frac{3\sqrt{3}}{2}\frac{J_3}{J^3}, \qquad \theta = \frac{1}{3}\arcsin(x), \tag{83}$$

with $x$ clamped to the interval $(-1 + 10^{-10}, 1 - 10^{-10})$ in the code. For an angle $\beta$, define

$$A_\beta(\theta) = \cos\theta - \frac{\sin\theta\sin\beta}{\sqrt{3}}. \tag{84}$$

The implemented yield function is

$$f(\boldsymbol{\sigma}) = \frac{\sin\phi}{3} I_1 + \sqrt{J_2}\,A_\phi(\theta) - c\cos\phi, \tag{85}$$

where $\phi$ is the friction angle and $c$ is the cohesion. The nonassociated plastic potential is obtained by replacing $\phi$ with the dilation angle $\psi$ and omitting the cohesion term:

$$g(\boldsymbol{\sigma}) = \frac{\sin\psi}{3} I_1 + \sqrt{J_2}\,A_\psi(\theta). \tag{86}$$

The flow is associative only when $\psi = \phi$.

The stress gradient of a scalar function

$$\Phi_\beta(\boldsymbol{\sigma}) = \frac{\sin\beta}{3} I_1 + \sqrt{J_2}\,A_\beta(\theta) \tag{87}$$

is written by the chain rule as

$$\frac{\partial\Phi_\beta}{\partial\boldsymbol{\sigma}} = \frac{\partial\Phi_\beta}{\partial I_1}\frac{\partial I_1}{\partial\boldsymbol{\sigma}} + \frac{\partial\Phi_\beta}{\partial J_2}\frac{\partial J_2}{\partial\boldsymbol{\sigma}} + \frac{\partial\Phi_\beta}{\partial J_3}\frac{\partial J_3}{\partial\boldsymbol{\sigma}}. \tag{88}$$

The scalar derivatives with respect to $I_1$, $J_2$, and $J_3$ are given in Appendix A. In the implementation, $\partial f/\partial\boldsymbol{\sigma}$ is obtained from $\beta = \phi$, while the flow vector is

$$\mathbf{n} = \frac{\partial g}{\partial\boldsymbol{\sigma}} = \left.\frac{\partial\Phi_\beta}{\partial\boldsymbol{\sigma}}\right|_{\beta=\psi}. \tag{89}$$

The flow-vector gradient is the Hessian of $\Phi_\psi$, also summarized in Appendix A. The current smooth invariant implementation is perfectly plastic:

$$h = 0, \qquad \partial_{\boldsymbol{\sigma}} h = \mathbf{0}. \tag{90}$$

## 7.2 Principal-stress return branches

The alternative Mohr–Coulomb integration path works in principal stress space. Let the trial principal stresses be ordered as $\sigma_1^{\text{trial}} \geq \sigma_2^{\text{trial}} \geq \sigma_3^{\text{trial}}$. The trial yield value on the main plane is

$$f^{\text{trial}} = \left(\sigma_1^{\text{trial}} - \sigma_3^{\text{trial}}\right) + \left(\sigma_1^{\text{trial}} + \sigma_3^{\text{trial}}\right)\sin\phi - 2c\cos\phi. \tag{91}$$

If $f^{\text{trial}} \leq 0$, the step is elastic. Otherwise the main-plane plastic multiplier is

$$\Delta\gamma_{n+1} = \frac{f^{\text{trial}}}{4G\left(1 + \dfrac{\sin\psi\sin\phi}{3}\right) + 4K\sin\psi\sin\phi}. \tag{92}$$

The corresponding principal-stress update is

$$\sigma_1 = \sigma_1^{\text{trial}} - \left[2G\left(1 + \frac{\sin\psi}{3}\right) + 2K\sin\psi\right]\Delta\gamma_{n+1}, \tag{93}$$

$$\sigma_2 = \sigma_2^{\text{trial}} + \left(\frac{4G}{3} - 2K\right)\sin\psi\,\Delta\gamma_{n+1}, \tag{94}$$

$$\sigma_3 = \sigma_3^{\text{trial}} + \left[2G\left(1 - \frac{\sin\psi}{3}\right) - 2K\sin\psi\right]\Delta\gamma_{n+1}. \tag{95}$$

This branch is accepted only if the updated ordering $\sigma_1 \geq \sigma_2 \geq \sigma_3$ is satisfied. If it is not, the code tests an edge return by solving a $2\times 2$ system for two plastic multipliers. If the edge return is not admissible, the update goes to the apex,

$$\sigma_1 = \sigma_2 = \sigma_3 = c\cot\phi. \tag{96}$$

The principal-space consistent tangents for the main plane, edge, and apex branches are then rotated back to the global Voigt basis by the spectral projectors used in the implementation.

# A Stress invariants and derivatives used in PorousLab

The mechanical laws discussed in Sections 5– 7 use the PorousLab stress vector convention

$$\boldsymbol{\sigma} = \begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{xy}\end{bmatrix}, \qquad \mathbf{m} = \begin{bmatrix}1\\1\\1\\0\end{bmatrix}. \tag{97}$$

The shear component is the tensorial shear stress $\sigma_{xy}$. The hydrostatic stress and deviatoric stress vector are

$$p = \frac{I_1}{3}, \qquad \mathbf{s} = \boldsymbol{\sigma} - p\,\mathbf{m}. \tag{98}$$

## A.1 Invariant definitions

The stress invariants implemented in `MechanicalLaw` are

$$I_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}, \tag{99}$$
$$I_2 = \sigma_{xx}\sigma_{yy} + \sigma_{yy}\sigma_{zz} + \sigma_{xx}\sigma_{zz} - \sigma_{xy}^2, \tag{100}$$
$$I_3 = \sigma_{xx}\sigma_{yy}\sigma_{zz} - \sigma_{zz}\sigma_{xy}^2. \tag{101}$$

The deviatoric invariants are implemented as functions of $(I_1, I_2, I_3)$:

$$J_2(I_1, I_2) = \frac{I_1^2}{3} - I_2, \tag{102}$$
$$J_3(I_1, I_2, I_3) = \frac{2I_1^3}{27} - \frac{I_1 I_2}{3} + I_3. \tag{103}$$

In the implementation, $J_2$ is projected to a non-negative value with $J_2 \leftarrow \max(J_2, 0)$.

## A.2 Base invariant derivatives

The derivative of $I_1$ with respect to the stress vector is

$$\frac{\partial I_1}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{104}$$

The derivative of $I_2$ is

$$\frac{\partial I_2}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{yy} + \sigma_{zz} \\ \sigma_{xx} + \sigma_{zz} \\ \sigma_{xx} + \sigma_{yy} \\ -2\sigma_{xy} \end{bmatrix}, \tag{105}$$

and the derivative of $I_3$ is

$$\frac{\partial I_3}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{yy}\,\sigma_{zz} \\ \sigma_{xx}\,\sigma_{zz} \\ \sigma_{xx}\,\sigma_{yy} - \sigma_{xy}^2 \\ -2\,\sigma_{zz}\,\sigma_{xy} \end{bmatrix}. \tag{106}$$

The Hessian of $I_1$ is zero. The Hessians of $I_2$ and $I_3$ are

$$\frac{\partial^2 I_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}, \tag{107}$$

and

$$\frac{\partial^2 I_3}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} = \begin{bmatrix} 0 & \sigma_{zz} & \sigma_{yy} & 0 \\ \sigma_{zz} & 0 & \sigma_{xx} & 0 \\ \sigma_{yy} & \sigma_{xx} & 0 & -2\sigma_{xy} \\ 0 & 0 & -2\sigma_{xy} & -2\sigma_{zz} \end{bmatrix}. \tag{108}$$

## A.3 Derivatives of J2

The scalar derivatives of $J_2(I_1, I_2)$ are

$$\frac{\partial J_2}{\partial I_1} = \frac{2I_1}{3}, \qquad \frac{\partial J_2}{\partial I_2} = -1, \qquad \frac{\partial^2 J_2}{\partial I_1^2} = \frac{2}{3}. \tag{109}$$

Therefore, by the chain rule,

$$\frac{\partial J_2}{\partial \boldsymbol{\sigma}} = \frac{\partial J_2}{\partial I_1}\frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial J_2}{\partial I_2}\frac{\partial I_2}{\partial \boldsymbol{\sigma}}. \tag{110}$$

Substitution gives the vector used in the code:

$$\frac{\partial J_2}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} \dfrac{2\sigma_{xx} - \sigma_{yy} - \sigma_{zz}}{3} \\ \dfrac{2\sigma_{yy} - \sigma_{xx} - \sigma_{zz}}{3} \\ \dfrac{2\sigma_{zz} - \sigma_{yy} - \sigma_{xx}}{3} \\ 2\sigma_{xy} \end{bmatrix}. \tag{111}$$

The Hessian follows from the second-order chain rule:

$$\begin{aligned} \frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} &= \frac{\partial^2 J_2}{\partial I_1^2}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^\top + \frac{\partial J_2}{\partial I_2}\frac{\partial^2 I_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} \\ &= \frac{2}{3}\mathbf{m}\,\mathbf{m}^\top - \frac{\partial^2 I_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}}. \end{aligned} \tag{112}$$

Thus,

$$\frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}. \tag{113}$$

## A.4 Derivatives of J3

The scalar derivatives of $J_3(I_1, I_2, I_3)$ are

$$\frac{\partial J_3}{\partial I_1} = \frac{2I_1^2}{9} - \frac{I_2}{3}, \qquad \frac{\partial J_3}{\partial I_2} = -\frac{I_1}{3}, \qquad \frac{\partial J_3}{\partial I_3} = 1, \tag{114}$$

$$\frac{\partial^2 J_3}{\partial I_1^2} = \frac{4I_1}{9}, \qquad \frac{\partial^2 J_3}{\partial I_1 \partial I_2} = -\frac{1}{3}. \tag{115}$$

All other second derivatives with respect to $I_1$, $I_2$, and $I_3$ are zero. The gradient of $J_3$ is therefore

$$\frac{\partial J_3}{\partial \boldsymbol{\sigma}} = \frac{\partial J_3}{\partial I_1}\frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial J_3}{\partial I_2}\frac{\partial I_2}{\partial \boldsymbol{\sigma}} + \frac{\partial J_3}{\partial I_3}\frac{\partial I_3}{\partial \boldsymbol{\sigma}}. \tag{116}$$

The code-level component form is

$$\frac{\partial J_3}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \tag{117}$$

where

$$\begin{aligned}
a_1 &= \frac{2\sigma_{xx}^2 - 2\sigma_{xx}\sigma_{yy} - 2\sigma_{xx}\sigma_{zz} + 3\sigma_{xy}^2 - \sigma_{yy}^2 + 4\sigma_{yy}\sigma_{zz} - \sigma_{zz}^2}{9}, \\
a_2 &= \frac{-\sigma_{xx}^2 - 2\sigma_{xx}\sigma_{yy} + 4\sigma_{xx}\sigma_{zz} + 3\sigma_{xy}^2 + 2\sigma_{yy}^2 - 2\sigma_{yy}\sigma_{zz} - \sigma_{zz}^2}{9}, \\
a_3 &= \frac{-\sigma_{xx}^2 + 4\sigma_{xx}\sigma_{yy} - 2\sigma_{xx}\sigma_{zz} - 6\sigma_{xy}^2 - \sigma_{yy}^2 - 2\sigma_{yy}\sigma_{zz} + 2\sigma_{zz}^2}{9}, \\
a_4 &= \frac{2\sigma_{xy}\left(\sigma_{xx} + \sigma_{yy} - 2\sigma_{zz}\right)}{3}.
\end{aligned} \tag{118}$$

The Hessian is assembled modularly as

$$\begin{aligned}
\frac{\partial^2 J_3}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} = {}& \frac{\partial^2 J_3}{\partial I_1^2}\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^{\top} + \frac{\partial^2 J_3}{\partial I_1 \partial I_2}\left[\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_2}{\partial \boldsymbol{\sigma}}\right)^{\top} + \frac{\partial I_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial I_1}{\partial \boldsymbol{\sigma}}\right)^{\top}\right] \\
& + \frac{\partial J_3}{\partial I_2}\frac{\partial^2 I_2}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}} + \frac{\partial J_3}{\partial I_3}\frac{\partial^2 I_3}{\partial \boldsymbol{\sigma}\,\partial \boldsymbol{\sigma}}.
\end{aligned} \tag{119}$$

This is the modular form of the matrix returned by `hessianJ3`.

## A.5 Von Mises equivalent stress derivatives

For

$$\sigma_{\mathrm{VM}}(J_2) = \sqrt{3J_2}, \tag{120}$$

the scalar derivatives are

$$\frac{\partial \sigma_{\mathrm{VM}}}{\partial J_2} = \frac{\sqrt{3}}{2\sqrt{J_2}}, \qquad \frac{\partial^2 \sigma_{\mathrm{VM}}}{\partial J_2^2} = -\frac{\sqrt{3}}{4\,J_2^{3/2}}. \tag{121}$$

The chain rule gives

$$\frac{\partial \sigma_{\mathrm{VM}}}{\partial \boldsymbol{\sigma}} = \frac{\partial \sigma_{\mathrm{VM}}}{\partial J_2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}, \tag{122}$$

$$\frac{\partial^2 \sigma_{\mathrm{VM}}}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} = \frac{\partial^2 \sigma_{\mathrm{VM}}}{\partial J_2^2}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right)^{\top} + \frac{\partial \sigma_{\mathrm{VM}}}{\partial J_2}\frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}}. \tag{123}$$

Equivalently,

$$\frac{\partial \sigma_{\mathrm{VM}}}{\partial \boldsymbol{\sigma}} = \frac{\sqrt{3}}{2\sqrt{J_2}}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}, \tag{124}$$

$$\frac{\partial^2 \sigma_{\mathrm{VM}}}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} = \frac{\sqrt{3}}{2\sqrt{J_2}}\frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}\partial \boldsymbol{\sigma}} - \frac{\sqrt{3}}{4J_2^{3/2}}\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial \boldsymbol{\sigma}}\right)^{\top}. \tag{125}$$

## A.6 Lode-dependent Mohr–Coulomb derivatives

The smooth Mohr–Coulomb implementation uses

$$x = -\frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{3/2}}, \qquad \theta = \frac{1}{3}\arcsin(x), \qquad J = \sqrt{J_2}. \tag{126}$$

For a generic angle $\beta$, define

$$\Phi_\beta = \frac{\sin\beta}{3} I_1 + J\, A_\beta(\theta), \tag{127}$$

where

$$A_\beta = \cos\theta - \frac{\sin\theta\sin\beta}{\sqrt{3}}, \quad \frac{\partial A_\beta}{\partial\theta} = -\sin\theta - \frac{\cos\theta\sin\beta}{\sqrt{3}}, \quad \frac{\partial^2 A_\beta}{\partial\theta^2} = -\cos\theta + \frac{\sin\theta\sin\beta}{\sqrt{3}}. \tag{128}$$

The derivative of $\Phi_\beta$ with respect to the stress vector is assembled from the invariant derivatives as

$$\frac{\partial\Phi_\beta}{\partial\boldsymbol{\sigma}} = \frac{\partial\Phi_\beta}{\partial I_1}\frac{\partial I_1}{\partial\boldsymbol{\sigma}} + \frac{\partial\Phi_\beta}{\partial J_2}\frac{\partial J_2}{\partial\boldsymbol{\sigma}} + \frac{\partial\Phi_\beta}{\partial J_3}\frac{\partial J_3}{\partial\boldsymbol{\sigma}}. \tag{129}$$

The scalar coefficients used in the code are

$$\frac{\partial\Phi_\beta}{\partial I_1} = \frac{\sin\beta}{3}, \qquad \frac{\partial\Phi_\beta}{\partial J_2} = \frac{A_\beta - \dfrac{\partial A_\beta}{\partial\theta}\tan(3\theta)}{2J}, \qquad \frac{\partial\Phi_\beta}{\partial J_3} = -\frac{\sqrt{3}\dfrac{\partial A_\beta}{\partial\theta}}{2J_2\cos(3\theta)}. \tag{130}$$

When $\cos(3\theta)$ is below the numerical tolerance, the code freezes the Lode-angle dependence and uses $\partial\Phi_\beta/\partial J_2 = A_\beta/(2J)$ and $\partial\Phi_\beta/\partial J_3 = 0$.

The Hessian is also written by the chain rule. Let

$$C_2 = \frac{\partial\Phi_\beta}{\partial J_2}, \qquad C_3 = \frac{\partial\Phi_\beta}{\partial J_3}, \qquad C_4 = \frac{\partial^2 A_\beta}{\partial\theta^2} + 3\tan(3\theta)\frac{\partial A_\beta}{\partial\theta}. \tag{131}$$

The second scalar derivatives used in the code are

$$\frac{\partial^2\Phi_\beta}{\partial J_2^2} = -\frac{A_\beta - \tan^2(3\theta)C_4 - 3\tan(3\theta)\dfrac{\partial A_\beta}{\partial\theta}}{4J^3}, \tag{132}$$

$$\frac{\partial^2\Phi_\beta}{\partial J_2\partial J_3} = \frac{\left(\dfrac{1}{2}\tan(3\theta)C_4 + \dfrac{\partial A_\beta}{\partial\theta}\right)\sqrt{3}}{2J_2^2\cos(3\theta)}, \tag{133}$$

$$\frac{\partial^2\Phi_\beta}{\partial J_3^2} = \frac{3C_4}{4J_2^{5/2}\cos^2(3\theta)}. \tag{134}$$

Thus,

$$\begin{aligned}\frac{\partial^2\Phi_\beta}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} &= C_2\frac{\partial^2 J_2}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} + C_3\frac{\partial^2 J_3}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} + \frac{\partial^2\Phi_\beta}{\partial J_2^2}\frac{\partial J_2}{\partial\boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial\boldsymbol{\sigma}}\right)^\top \\ &\quad + \frac{\partial^2\Phi_\beta}{\partial J_2\partial J_3}\left[\frac{\partial J_2}{\partial\boldsymbol{\sigma}}\left(\frac{\partial J_3}{\partial\boldsymbol{\sigma}}\right)^\top + \frac{\partial J_3}{\partial\boldsymbol{\sigma}}\left(\frac{\partial J_2}{\partial\boldsymbol{\sigma}}\right)^\top\right] \\ &\quad + \frac{\partial^2\Phi_\beta}{\partial J_3^2}\frac{\partial J_3}{\partial\boldsymbol{\sigma}}\left(\frac{\partial J_3}{\partial\boldsymbol{\sigma}}\right)^\top.\end{aligned} \tag{135}$$

# References


[1] Eduardo A. de Souza Neto, Djordje Perić, and David R. J. Owen. *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons, Chichester, 2008.

[2] Juan C. Simo and Thomas J. R. Hughes. *Computational Inelasticity*. Springer, New York, 1998.

[3] Juan C Simo and Robert Leroy Taylor. Consistent tangent operators for rate-independent elastoplasticity. *Computer methods in applied mechanics and engineering*, 48(1):101–118, 1985.