New Numerical Solver for Elastoplastic Problems based on the Moreau-Yosida Theorem

Peter Gruber, Jan Valdman*

Abstract

We discuss a new solution algorithm for solving elastoplastic problems with hardening. The one time-step elastoplastic problem can be formulated as a convex minimization problem with a continuous but non-smooth functional depending on unknown displacement smoothly and on the plastic strain non-smoothly. It is shown that the functional structure allows the application of the Moreau-Yosida Theorem known in convex analysis. It guarantees that the substitution of the non-smooth plastic-strain as a function of the linear strain which depends on the displacement only yields an already smooth functional in the displacement only. Moreover, the second derivative of such functional exists in all continuum points apart from interfaces where elastic and plastic zones intersect. This allows the efficient implementation of a Newton-like method. For easy implementation most essential Matlab© functions are provided. Numerical experiments in two dimensions state quadratic convergence of the Newton-like method as long as the elastoplastic interface is detected sufficiently precisely.

1 Introduction

We consider the quasi-static initial-boundary value problem for small strain elastoplasticity with an isotropic hardening. Starting from the classical formulation, combining the equilibrium of forces with elastoplastic isotropic hardening law under the assumption of small deformations, we can formulate the time-dependent variational inequality. The uniqueness of a solution of such inequality has been for instance proved in [Joh76] utilizing results for general variation inequalities [DL76].

The traditional numerical methods for solving the time-dependent variational inequality were based on the explicit Euler time-discretization with respect to the loading history. In this case the idea of implicit return mapping discretization [SH98] turned out fruitful for calculations. By implicit Euler time-discretization on the other side, the time-dependent inequality is approximated by a sequence of

*Special Research Program SFB F013 'Numerical and Symbolic Scientific Computing', Johannes Kepler University Linz, Altenbergerstrasse 69, A-4040 Linz, Austria, email: Jan.Valdman@sfb013.uni-linz.ac.at

time-independent variational inequalities [KL84] for the unknown displacement u and the plastic strain p. Each of these inequalities is equivalent to a minimization problem with the convex but non-smooth functional [HR99]. We introduce a new algorithm for solving such minimization problem. Our algorithm is of the Newton type and it utilizes the dependence $p = p(\varepsilon(u))$ of the plastic strain on the total strain $\varepsilon(u)$ [AC00]. This makes it possible to reformulate the energy minimization problem $e(u) \to \min$ for the unknown displacement u only. Since the dependencies of the minimization functional $e(u, p)$ on the plastic strain p, and of the plastic strain p on the total strain $\varepsilon(u)$ are continuous but non-smooth, the Fréchet derivate $De(u)$ seems not to exist. The main theoretical result here is to show that $e(u)$ is in fact differentiable. More precisely, we show that the structure of the functional $e(u)$ satisfies the assumptions of the Moreau-Yoshida theorem from convex analysis and therefore $e(u)$ must be (Fréchet) differentiable.

For the space-discretization, the finite element method of the lowest order with the nodal linear displacement and the piece-wise constant plastic strain is used. The unknown discretized displacement u satisfies the necessary condition $De(u) = 0$, which represents the system of nonlinear equations. It is shown that the discretized second derivative $D^2e(u)$ exists everywhere apart from the elastoplastic interface, i.e., apart from the discrete points, which disjoin elastic zones from plastic zones. The measure of the set of interface points is known to be zero in the continuous case. Therefore, it is believed that the Newton-like method would also converge in the discrete case.

Numerical experiments in Matlab© justify the theoretical expectations. Three numerical examples in two dimensions are presented. First two examples, the L-shape and wrench examples which include positive hardening parameters, provide the following conclusions:

- The number of iteration steps is (almost) independent of the size of the discretization.
- The Newton-like method converges quadratically after the elastoplastic zones are identified sufficiently. This remark has also been made independently in the convergence analysis of [Bla97].

The last example of the plate with a hole serves as a benchmark problem in perfect plasticity, where theoretical conclusions mentioned above do not need to hold anymore. The Newton-like method oscillates for finer meshes and additional damping or nested iterations techniques are necessary in order obtain convergence.

The paper is organized as follows. Section 2 recalls the mathematical modeling of elastoplasticity and also addresses the Moreau-Yoshida Theorem. Finite elements discretization and the implementation of the Newton-like method are discussed in Section 3. Numerical examples in Section 4 illustrate the behavior of the Newton-like method.

2 Mathematical Modeling

Let $d \in \mathbb{N}$ be the space dimension, $\Omega \in \mathbb{R}^d$ be an open domain with a Lipschitz-continuous boundary $\Gamma := \partial\Omega$. Further let Γ be split into two distinct parts Γ_D (*Dirichlet boundary*) and Γ_N (*Neumann boundary*), such that $\Gamma_D \cup \Gamma_N = \Gamma$. The set Θ be some time interval (e.g., we assume $\Theta = [0, T]$ for $T \in \mathbb{R}^+$), and $\mathbb{R}^+ := \{x \in \mathbb{R} \mid x > 0\}$. The *matrix-scalar-product* $:$ is defined for two equal size matrices $A = (a_{ij})_{ij}$ and $B = (b_{ij})_{ij}$ as $A : B = \sum_{ij} a_{ij} b_{ij}$. The *Frobenius-norm* of matrix A reads $\|A\|_F := \sqrt{A : A}$. Let I denote the (square) identity matrix. The *trace* and the *deviator* of a matrix $A \in \mathbb{R}^{d\times d}$ are defined by $\operatorname{tr} A := A : I$ and $\operatorname{dev} A := A - \frac{\operatorname{tr} A}{d} I$.

2.1 Classical Formulation of Elastoplasticity

The equilibrium of forces reads

$$-\operatorname{div}(\sigma) = f \quad \text{in } \Omega\,, \tag{1}$$

where the $d \times d$ sized tensor $\sigma(x, t)$ denotes the stress tensor and $f(x, t)$ describes volume forces acting in each material point $x \in \Omega$ at the time $t \in \Theta$. The $d \times d$ sized (linearized) strain tensor ε describes the local deformation defined as

$$\varepsilon(u) := \frac{1}{2}\left(\nabla u + (\nabla u)^T\right), \tag{2}$$

where $u(x, t)$ denotes the body displacement. The plastic part of the strain is denoted by $p(x, t)$. The relation between stress and strain is given by Hook's law

$$\sigma = \mathbb{C}\,(\varepsilon - p)\,, \tag{3}$$

where the fourth-order *elasticity tensor* $\mathbb{C} \in \mathbb{R}^{d\times d}_{d\times d}$ is defined by $\mathbb{C}_{ijkl} := \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$. Here $\lambda, \mu \in \mathbb{R}^+$ denote the *Lamé*-constants, and δ_{ij} denotes the *Kronecker-symbol*. As an alternative, one uses another material parameters *Young's modulus* $E = \mu\frac{3\lambda+2\mu}{\lambda+\mu}$ and the *Poisson ratio* $\nu = \frac{\lambda}{2(\lambda+\mu)}$. Further we assume the initial conditions

$$u(x, 0) = 0 \quad \text{and} \quad \sigma(x, 0) = 0 \qquad \text{for all } x \in \Omega\,, \tag{4}$$

and the boundary conditions

$$u = u_D \qquad \text{on } \Gamma_D\,, \tag{5}$$

$$\sigma \cdot n = g \qquad \text{on } \Gamma_N\,, \tag{6}$$

where $n(x, t)$ denotes the exterior unit normal, $u_D(x, t)$ denotes a prescribed displacement and $g(x, t)$ denotes a prescribed surface tension. Purely elastic behavior of a body is given by the expressions (1) - (6) and $p \equiv 0$. In order to model plasticity we need another two restrictions, which incorporate the time development of p. We introduce the hardening parameter α, which is a scalar identifier in case of isotropic hardening [ACFK02]. The tuple (σ, α) is called *generalized stress*. A generalized stress is called *admissible*, if a *dissipation functional* φ with

$$\varphi(\sigma,\alpha) := \begin{cases} 0 & \text{if } \phi(\sigma,\alpha) \leq 0, \\ \infty & \text{if } \phi(\sigma,\alpha) > 0, \end{cases} \tag{7}$$

satisfies

$$\varphi(\sigma,\alpha) < \infty. \tag{8}$$

The function ϕ is convex and called the *yield function*. In case of isotropic hardening it is defined

$$\phi(\sigma,\alpha) := \begin{cases} \|\operatorname{dev}\sigma\|_F - \sigma_y(1+H\alpha) & \text{if } \alpha \geq 0\ , \\ \infty & \text{if } \alpha < 0\ , \end{cases} \tag{9}$$

The material constants $\sigma_y > 0$ and $H > 0$ are called *yield stress* and *modulus of hardening*. All admissible generalized stresses are characterized by $\phi(\sigma,\alpha) \leq 0$. In this sense we understand, that the hardening parameter α in $\varphi(\sigma,\alpha) < \infty$ controls the (convex) set of admissible stresses σ. Finally the Prandtl-Reuss normality law states, that for all generalized stresses (τ,β) there holds

$$\dot{p} : (\tau - \sigma) - \dot{\alpha}\,(\beta - \alpha) \ \leq\ \varphi(\tau,\beta) - \varphi(\sigma,\alpha), \tag{10}$$

where $\dot{p}$ and $\dot{\alpha}$ denote the first time derivative of p and α.
Problem 1 (classical formulation)**.** *Find (u,p,α), such that expressions (1)–(6), (8) and (10) are satisfied.*

We will transform Problem 1 to a dual classical formulation. In order to do so, we have to summarize some convex analysis theory.
Definition 1 (conjugate function)**.** *For a function $f : X \to [-\infty,\infty]$ we define the* conjugate function *$f^* : X^* \to [-\infty,\infty]$ by*

$$f^*(x^*) = \sup_{x\in X}(\langle x^*, x\rangle - f(x)).$$

Definition 2 (subdifferential)**.** *Let f be a convex function on X. For any $x \in X$ the* sub-differential *$\partial f(x)$ of x is the possibly empty subset of X^* defined by*

$$\partial f(x) = \{x^* \in X^* : \langle x^*, y - x\rangle \leq f(y) - f(x) \quad \forall y \in X\}.$$

Definition 3 (lower semicontinuity)**.** *A function $f : X \to [-\infty,+\infty]$ is called* lower semi-continuous *if*

$$\{x_n\}_{n\in\mathbb{N}} \to x \Rightarrow \liminf_{n\to\infty} f(x_n) \geq f(x).$$

Definition 4 (proper function)**.** *A function $f : X \to [-\infty,+\infty]$ is called* proper *if there exists a point $x \in X$ such that $f(x) < \infty$.*
Theorem 1. *Let X be a Banach space, and $f : X \to [-\infty,\infty]$ be a proper, convex, lower semi-continuous function. Then*

$$x^* \in \partial f(x) \Leftrightarrow x \in \partial f^*(x^*).$$

Proof. See [Kos91]. □

Applying this theorem to (10) the following equivalences hold:

$$
\begin{array}{rrcll}
& \dot{p} : (\tau - \sigma) - \dot{\alpha}\,(\beta - \alpha) & \leq & \varphi(\tau, \beta) - \varphi(\sigma, \alpha) & (\forall \tau, \beta) \\
\Leftrightarrow & \langle (\dot{p}, -\dot{\alpha}), (\tau, \beta) - (\sigma, \alpha) \rangle & \leq & \varphi(\tau, \beta) - \varphi(\sigma, \alpha) & (\forall \tau, \beta) \\
\Leftrightarrow & (\dot{p}, -\dot{\alpha}) & \in & \partial \varphi(\sigma, \alpha) & \\
\Leftrightarrow & (\sigma, \alpha) & \in & \partial \varphi^*(\dot{p}, -\dot{\alpha}) & \\
\Leftrightarrow & \langle (\sigma, \alpha), (q, \beta) - (\dot{p}, -\dot{\alpha}) \rangle & \leq & \varphi^*(q, \beta) - \varphi^*(\dot{p}, -\dot{\alpha}) & (\forall q, \beta) \\
\Leftrightarrow & \sigma : (q - \dot{p}) + \alpha\,(\beta + \dot{\alpha}) & \leq & \varphi^*(q, \beta) - \varphi^*(\dot{p}, -\dot{\alpha}) & (\forall q, \beta).
\end{array}
$$

Thus Problem 1 is equivalent to

Problem 2 (dual classical formulation)**.** *Find* (u, p, α), *such that*

$$
\begin{aligned}
-\operatorname{div}(\sigma) &= f \quad \textit{in } \Omega, && (11) \\
\varepsilon(u) &= \frac{1}{2}\left(\nabla u + (\nabla u)^T\right), \\
\varepsilon(u) &= \mathbb{C}^{-1}\sigma + p, \\
\sigma \cdot n &= g \quad \textit{on } \Gamma_N, \\
u &= u_D \quad \textit{on } \Gamma_D, \\
\varphi(\sigma, \alpha) &< \infty, \\
\sigma : (q - \dot{p}) + \alpha\,(\beta + \dot{\alpha}) &\leq \varphi^*(q, \beta) - \varphi^*(\dot{p}, -\dot{\alpha}) \qquad (\forall q, \beta). && (12)
\end{aligned}
$$

2.2 Variational Formulation of Elastoplasticity

Let $V := [H^1(\Omega)]^d$, $V_D := [H^1_D(\Omega)]^d$, $V_0 := [H^1_0(\Omega)]^d$, and $W := [L_2(\Omega)]^{d\times d}_{\mathrm{sym}} \times L_2(\Omega)$. We multiply (11) with test functions $v \in V_0$, integrate (11) and (12) over Ω, partial integrate (11), and obtain a variational problem

Problem 3 (Variational formulation)**.** *Find* $(u, p, \alpha) \in V_D \times W$, *such that for all* $(v, q, \beta) \in V_0 \times W$ *there hold*

$$
\int_\Omega \mathbb{C}\,(\varepsilon(u) - p) : \varepsilon(v)\,\mathrm{d}x = \int_\Omega f \cdot v\,\mathrm{d}x + \int_{\Gamma_N} g \cdot v\,\mathrm{d}S(x)
$$

and

$$
\int_\Omega \left[\mathbb{C}\,(\varepsilon(u) - p) : (q - \dot{p}) + \alpha\,(\beta + \dot{\alpha})\right]\,\mathrm{d}x \leq \int_\Omega \varphi^*(q, \beta)\,\mathrm{d}x - \int_\Omega \varphi^*(\dot{p}, -\dot{\alpha})\,\mathrm{d}x.
$$

We discretize in time by backward Euler with the discretization parameter k, precisely by the substitution of

$$
u = u_1,\ p = p_1,\ \alpha = \alpha_1,\ \dot{p} = \frac{p_1 - p_0}{k},\ \dot{\alpha} = \frac{\alpha_1 - \alpha_0}{k},
$$

where the initial value α_0 has to satisfy $\varphi(\cdot, \alpha_0) < \infty$, such that due to the definition of φ and ϕ there must hold $\alpha_0 \geq 0$. We obtain a one time-step problem

Problem 4 (One time-step). *Find* $(u_1, p_1, \alpha_1) \in V_D \times W$*, such that for all* $(v, q, \beta) \in V_0 \times W$ *there holds*

$$\int_\Omega \mathbb{C}\left(\varepsilon(u_1) - p_1\right) : \varepsilon(v) \,\mathrm{d}x = \int_\Omega f \cdot v \,\mathrm{d}x + \int_{\Gamma_N} g \cdot v \,\mathrm{d}S(x),$$

and

$$\begin{aligned}&\int_\Omega \left[\mathbb{C}\left(\varepsilon(u_1) - p_1\right) : (kq - p_1 + p_0) + \alpha_1\left(k\beta + \alpha_1 - \alpha_0\right)\right] \,\mathrm{d}x \\ &\quad\le k\int_\Omega \varphi^*(q,\beta)\,\mathrm{d}x - k\int_\Omega \varphi^*\Big(\frac{p_1 - p_0}{k}, \frac{\alpha_0 - \alpha_1}{k}\Big)\,\mathrm{d}x.\end{aligned}$$

Problem 4 can be expressed in a more abstract way. Therefore, let $x_1 \in V$, $y_1 \in V$, $(x_2, x_3) \in W$, $(y_2, y_3) \in W$, $X := (x_1, x_2, x_3)$ and $Y := (y_1, y_2, y_3)$. Further let

$$\begin{aligned}
a_1(X,Y) &:= \int_\Omega \mathbb{C}\left(\varepsilon(x_1) - x_2\right) : \varepsilon(y_1)\,\mathrm{d}x, \\
a_2(X,Y) &:= \int_\Omega \left[x_3 y_3 - \mathbb{C}\left(\varepsilon(x_1) - x_2\right) : y_2\right]\,\mathrm{d}x, \\
a(X,Y) &:= a_1(X,Y) + a_2(X,Y), && (13)\\
L(X) &:= \int_\Omega f\cdot x_1\,\mathrm{d}x + \int_{\Gamma_N} g\cdot x_1\,\mathrm{d}S(x), && (14)\\
\Psi(X) &:= k\int_\Omega \varphi^*\Big(\frac{x_2 - p_0}{k}, \frac{\alpha_0 - x_3}{k}\Big)\,\mathrm{d}x. && (15)
\end{aligned}$$

Note, that $a(X,Y) = \int_\Omega \mathbb{C}\left(\varepsilon(x_1) - x_2\right) : (\varepsilon(y_1) - y_2) + x_3 y_3 \,\mathrm{d}x$ is a symmetric, positive definite bilinear-form. Further L is linear and Ψ is convex in X. With the special choice of $x_1 := u_1$, $x_2 := p_1$, $x_3 := \alpha_1$, $y_1 := u_1 - v$, $y_2 := p_0 + kq$ and $y_3 := \alpha_0 - k\beta$ Problem 4 reads

Problem 5. *Find* $X \in V_D \times W$*, such that for all* $Y \in V_D \times W$ *there holds:*

$$\begin{aligned}
a_1(X, X - Y) &= L(X - Y), && (16)\\
a_2(X, X - Y) &\le \Psi(Y) - \Psi(X). && (17)
\end{aligned}$$

Summation of (16) and (17) leads to

Problem 6. *Find* $X \in V_D \times W$*, such that for all* $Y \in V_D \times W$ *there holds:*

$$a(X, X - Y) \le L(X - Y) + \Psi(Y) - \Psi(X). \tag{18}$$

Lemma 1. *Problems 5 and 6 are equivalent.*

Proof. Problem 5 $\Rightarrow$ Problem 6: trivial by adding (16) and (17).
Problem 6 $\Rightarrow$ Problem 5: Let $X := (x_1, x_2, x_3)$ solve (18) for all $Y = (y_1, y_2, y_3)$. Particularly, for the choice $Y := (x_1, y_2, y_3)$ there follows that X solves (17) for all $Y := (x_1, y_2, y_3)$. Since the bilinear form $a_2(\cdot,\cdot)$ and $\Psi(\cdot)$ are independent of y_1, X also solves (17) for arbitrary $Y \in V_D \times W$. Similarly, X solves

$$a_1(X, Y - X) \le L(Y - X). \tag{19}$$

for the special choice $Y := (y_1, x_2, x_3)$. Since $a_1(\cdot,\cdot)$ and $L(\cdot)$ are independent of y_2 and y_3, X solves (19) for arbitrary $Y \in V_D \times W$. By the substitution $Z = Y - X$ in (19) one obtains the inequality

$$a_1(X, Z) \leq L(Z)$$

for all $Z \in V_0 \times W$. The reversed inequality is then formulated by replacing Z by $-Z$. Thus the equality (16) must be satisfied. $\square$

Definition 5 (energy functional in elastoplasticity)**.** *Let $X \in V_D \times W$, and let $a(\cdot,\cdot)$, $\Psi(\cdot)$ and $L(\cdot)$ be defined as in (13)–(15). Then we define*

$$e(X) := \frac{1}{2}a(X, X) + \Psi(X) - L(X)$$

which is called the energy functional in elastoplasticity.

Lemma 2. *Let $a(\cdot,\cdot)$, $L(\cdot)$ and $\Psi(\cdot)$ be defined as in (13)–(15). Further let $e(\cdot)$ be defined as in Definition 5. Then expressions (i) and (ii) are equivalent:*

(i) Find $X \in V_D \times W$ such that for all $Y \in V_D \times W$ there holds

$$L(Y - X) \leq a(X, Y - X) + \Psi(Y) - \Psi(X)\ .$$

(ii) Find $X \in V_D \times W$ such that

$$e(X) = \min_{Y \in V_D \times W} e(Y).$$

Proof. $(ii) \Rightarrow (i)$: Let $Y \in V_D \times W$ and $\theta \in (0, 1)$ be arbitrary and fixed. Expression (ii) implies

$$e(X + \theta(Y - X)) \geq e(X).$$

Hence

$$\theta a(X, Y-X) + \frac{1}{2}\theta^2 a(Y-X, Y-X) + \underbrace{\Psi(X + \theta(Y - X)) - \Psi(X)}_{\leq \theta(\Psi(Y)-\Psi(X))} - \theta L(Y-X) \geq 0,$$

and thus

$$a(X, Y - X) + \frac{1}{2}\theta a(Y - X, Y - X) + \Psi(Y) - \Psi(X) - L(Y - X) \geq 0.$$

Taking the limit $\theta \downarrow 0$ leads to expression (i).

$(i) \Rightarrow (ii)$: Let $X \in V_D \times W$ solve (i), and $Y \in V_D \times W$ be arbitrary and fixed.

$$\begin{aligned} e(Y) &= e(X + (Y - X)) \\ &= \frac{1}{2}a(X, X) + a(X, Y - X) + \frac{1}{2}a(Y - X, Y - X) + \Psi(X + Y - X) \\ &\quad -L(X) - L(Y - X) \\ &= e(X) + \underbrace{a(X, Y - X) + \Psi(Y) - \Psi(X) - L(Y - X)}_{\geq 0} \geq e(X). \end{aligned}$$

Hence, there holds $e(X) = \min_{Y \in V_D \times W} e(Y)$. $\square$

2.3 Minimization Problem

Thanks to Lemma 2, Problem 6 is equivalent to
Problem 7 (Minimization problem)**.** *Find* $(u_1, p_1, \alpha_1) \in V_D \times W$*, such that*

$$e(u_1, p_1, \alpha_1) = \frac{1}{2} \int_\Omega \mathbb{C}\,(\varepsilon(u_1) - p_1) : (\varepsilon(u_1) - p_1) + \alpha_1^2 + 2k\,\varphi^*(\frac{p_1 - p_0}{k}, \frac{\alpha_0 - \alpha_1}{k})\,\mathrm{d}x$$
$$- \int_\Omega f \cdot u_1\,\mathrm{d}x - \int_{\Gamma_N} g \cdot u_1\,\mathrm{d}S(x) \rightarrow\ min.$$

Lemma 3. *Let* $\mathcal{R} := \mathbb{R}^{d\times d} \times \mathbb{R}$*, the tuple* $(p, \alpha) \in \mathcal{R}$*, and the convex yield function* ϕ *be defined as in (9). Then there holds*

$$\varphi^*(p, \alpha) = \begin{cases} \sigma_y \|p\|_F & \textit{if} \quad (\operatorname{tr} p = 0) \wedge (\alpha + H\sigma_y\|p\|_F \leq 0)\,, \\ \infty & \textit{else.} \end{cases} \tag{20}$$

Proof. Let $\mathcal{M} := \{(q, \beta) \in \mathcal{R} \mid (\beta \geq 0) \wedge (\|\operatorname{dev} q\|_F - \sigma_y\,(1 + H\beta) \leq 0)\}$. The definition of a conjugate function (Definition 1) yields

$$\varphi^*(p, \alpha) = \sup_{(q,\beta)\in\mathcal{R}} (q : p + \beta\alpha - \varphi(q, \beta))\,. \tag{21}$$

If the supremum differs from $-\infty$, it can only be attained if $\varphi(q, \beta) < \infty$. Thus, due to the definitions of φ in (7) and ϕ in (9) there holds

$$\varphi^*(p, \alpha) = \sup_{(q,\beta)\in\mathcal{M}} (q : p + \beta\alpha)\ .$$

In the first instance, we will show

$$\varphi^*(p, \alpha) \geq \begin{cases} \sigma_y \|p\|_F & \text{if} \quad (\operatorname{tr} p = 0) \wedge (\alpha + H\sigma_y\|p\|_F \leq 0)\ , \\ \infty & \text{else}\ , \end{cases}$$

and then finally,

$$(\operatorname{tr} p = 0) \wedge (\alpha + H\sigma_y\|p\|_F \leq 0) \quad \Rightarrow \quad \varphi^*(p, \alpha) \leq \sigma_y\|p\|_F\ .$$

Let $c \in \mathbb{R}$. We choose $(q, \beta) = (cI, 0)$, which is element in $\mathcal{M}$, since

$$\|\operatorname{dev}(cI)\|_F = 0 \quad \text{and} \quad \sigma_y \geq 0\ .$$

The choice of (q, β) yields

$$\varphi^*(p, \alpha) \geq \sup_{c\in\mathbb{R}}\ c\,\underbrace{p : I}_{=\operatorname{tr} p}\,,$$

and thus there holds

$$\varphi^*(p, \alpha) \geq \begin{cases} 0 & \text{if}\ \operatorname{tr} p = 0\ , \\ +\infty & \text{else}\ . \end{cases}$$

Let $\theta := \frac{\sigma_y(1+H\beta)}{\|p\|_F}$ and $\operatorname{tr} p = 0$. We choose $(q, \beta) = (\theta p, \beta)$, which is element in $\mathcal{M}$, since

$$\|\operatorname{dev}(\theta p)\|_F = \theta \|\operatorname{dev} p\|_F = \theta \|p\|_F = \sigma_y(1 + H\beta) \ .$$

The certain choice of (q, β) yields

$$\begin{aligned}
\varphi^*(p, \alpha) &\geq \sup_{\beta \geq 0} (p : \theta p + \alpha\beta) = \sup_{\beta \geq 0} (\sigma_y (1 + H\beta) \|p\|_F + \alpha\beta) \\
&= \sigma_y \|p\|_F + \sup_{\beta \geq 0} ((\sigma_y H \|p\|_F + \alpha) \beta) \ ,
\end{aligned}$$

and thus there holds

$$\varphi^*(p, \alpha) \geq \begin{cases} \sigma_y \|p\|_F & \text{if } (\operatorname{tr} p = 0) \wedge (\sigma_y H \|p\|_F + \alpha \leq 0) \ , \\ +\infty & \text{else .} \end{cases}$$

Let $\operatorname{tr} p = 0$ and $\sigma_y H \|p\|_F + \alpha \leq 0$. There holds

$$\begin{aligned}
\varphi^*(p, \alpha) &= \sup_{(q,\beta) \in \mathcal{M}} (p : q + \alpha\beta) = \sup_{(q,\beta) \in \mathcal{M}} \left((\operatorname{dev} q) : p + \frac{\operatorname{tr} q}{\dim(q)} \underbrace{I : p}_{=0} + \alpha\beta \right) \\
&\leq \sup_{(q,\beta) \in \mathcal{M}} (\|\operatorname{dev} q\|_F \|p\|_F + \alpha\beta) \\
&\leq \sup_{\beta \geq 0} (\sigma_y (1 + H\beta) \|p\|_F + \alpha\beta) \\
&= \sigma_y \|p\|_F + \sup_{\beta \geq 0} ((\sigma_y H \|p\|_F + \alpha) \beta) = \sigma_y \|p\|_F \ ,
\end{aligned}$$

so the proposition is true. □

Combining the definition of φ^* in (20) and the minimal value condition of the energy functional in Problem 7 it is necessary to guarantee the condition

$$\varphi^*(\frac{p_1 - p_0}{k}, \frac{\alpha_0 - \alpha_1}{k}) < +\infty.$$

Due to Lemma 3 we have to determine p_1 and α_1 such that $\operatorname{tr}(p_1 - p_0) = 0$ and $\alpha_1 \geq \alpha_0 + \sigma_y H \|p_1 - p_0\|_F$. Under this condition we can find the minimizer of α_1 in Problem 7 by setting $\alpha_1 = \alpha_0 + \sigma_y H \|p_1 - p_0\|_F$, which leads to a minimization problem in u_1 and p_1.

Problem 8. *Find* $(u_1, p_1) \in V_D \times [L_2(\Omega)]^{d \times d}_{sym}$ *such that*

$$\begin{aligned}
e(u_1, p_1) = \frac{1}{2} \int_\Omega \mathbb{C}(\varepsilon(u_1) - p_1) : (\varepsilon(u_1) - p_1) \, \mathrm{d}x + \frac{1}{2} \int_\Omega (\alpha_0 + \sigma_y H \|p_1 - p_0\|_F)^2 \, \mathrm{d}x \\
+ \int_\Omega \sigma_y \|p_1 - p_0\|_F \, \mathrm{d}x - \int_\Omega f \cdot u_1 \, \mathrm{d}x - \int_{\Gamma_N} g \cdot u_1 \, \mathrm{d}S(x) \to \mathit{min}.
\end{aligned}$$

The minimizer in p_1 of Problem 8 can be calculated analytically (for a proof see [ACZ99]):

$$p_1 = \frac{(\|\operatorname{dev} A\|_F - \beta)_+}{2\mu + \sigma_y^2 H^2} \frac{\operatorname{dev} A}{\|\operatorname{dev} A\|_F} + p_0,$$

where A, β and the operator $(\cdot)_+$ are defined as

$$A := \mathbb{C}\left[\varepsilon(u_1) + p_0\right], \qquad \beta := \sigma_y(1+\alpha_0 H) \quad \text{and} \quad (\cdot)_+ = \begin{cases} \cdot & \text{if } \cdot > 0, \\ 0 & \text{else.} \end{cases}$$

Problem 8 is equivalent to the following minimization problem, which depends on the displacement u_1 only.

Problem 9. *Find $u_1 \in V_D$ such that*

$$\begin{aligned} e(u_1) = \frac{1}{2}\int_\Omega & \mathbb{C}(\varepsilon(u_1) - p_1(\varepsilon(u_1))) : (\varepsilon(u_1) - p_1(\varepsilon(u_1)))\,\mathrm{d}x \\ &+ \frac{1}{2}\int_\Omega (\alpha_0 + \sigma_y H\|p_1(\varepsilon(u_1)) - p_0\|_F)^2 + \sigma_y\|p_1(\varepsilon(u_1)) - p_0\|_F\,\mathrm{d}x \\ &- \int_\Omega f\cdot u_1\,\mathrm{d}x - \int_{\Gamma_N} g\cdot u_1\,\mathrm{d}S(x) \to min, \end{aligned} \tag{22}$$

with

$$p_1(\varepsilon(u_1)) = \frac{(\|\operatorname{dev} A\|_F - \beta)_+}{2\mu + \sigma_y^2 H^2}\frac{\operatorname{dev} A}{\|\operatorname{dev} A\|_F} + p_0, \tag{23}$$

where

$$A = \mathbb{C}\left[\varepsilon(u_1) + p_0\right] \quad \textit{and} \quad \beta = \sigma_y(1+\alpha_0 H).$$

2.4 The Moreau-Yosida Theorem

We will make use of an abstract formulation of (22). Let

$$\begin{aligned} \|B\|_{\mathbb{C}} &:= \left(\int_\Omega \mathbb{C}B(x) : B(x)\,\mathrm{d}x\right)^{\frac{1}{2}}, \text{ and} \\ \psi(p_1) &:= \frac{1}{2}\int_\Omega (\alpha_0 + \sigma_y H\|p_1 - p_0\|_F)^2\,\mathrm{d}x + \int_\Omega \sigma_y\|p_1 - p_0\|_F\,\mathrm{d}x \end{aligned}$$

define a new matrix scalar product and a (convex) functional. Expression (22) then rewrites as

$$e(u_1) = \frac{1}{2}\|\varepsilon(u_1) - p_1(\varepsilon(u_1))\|_{\mathbb{C}}^2 + \psi(p_1(\varepsilon(u_1))) - L(u_1) \;\to\; \min. \tag{24}$$

Minimizing functional $e(u_1)$ can be done by finding the root of its first derivative $De(u_1)$. The next theorem shows, that $e(u_1)$ is indeed smooth, no matter the dependency ψ on p is non-smooth.

Theorem 2 (Moreau-Yosida)**.** *Let H be a Hilbert space with scalar product $\langle\cdot,\cdot\rangle_H$, H^* its dual space, $\psi : H \to \mathbb{R}$ convex, and function f be defined as*

$$f : H \to \mathbb{R},\; y \mapsto \min_{x\in H}\left[\frac{1}{2}\|y-x\|_H^2 + \psi(x)\right].$$

Further let $\tilde{x}(y)$ denote the (unique) function, which yields

$$f(y) = \frac{1}{2}\|y - \tilde{x}(y)\|_H^2 + \psi(\tilde{x}(y)).$$

for all $y \in H$. Then there holds:

1. f is convex.

2. f is Fréchet-differentiable with $Df(y) = y - \tilde{x}(y) \ \in H^*$.

Proof. ad 1 (convexity): Let $y_1, y_2 \in H$, $t \in (0,1)$ be arbitrary and fixed. Further let $\overline{x} := \tilde{x}((1-t)y_1 + ty_2)$, $\overline{x}_1 := \tilde{x}(y_1)$ and $\overline{x}_2 := \tilde{x}(y_2)$. Due to the definition of f there holds

$$f((1-t)y_1 + ty_2) = \frac{1}{2}\|(1-t)y_1 + ty_2 - \overline{x}\|_H^2 + \psi(\overline{x}) \ .$$

Since $\overline{x}$ is the minimizer, the expression is certainly not getting any lower if $\overline{x}$ is substituted by any other element in H. Thus

$$f((1-t)y_1 + ty_2) \leq \frac{1}{2}\|(1-t)y_1 + ty_2 - (1-t)\overline{x}_1 - t\overline{x}_2\|_H^2 + \psi((1-t)\overline{x}_1 + t\overline{x}_2) \ .$$

Triangle inequality and convexity of ψ yield

$$f((1-t)y_1 + ty_2) \leq (1-t)\left[\frac{1}{2}\|y_1 - \overline{x}_1\|_H^2 + \psi(\overline{x}_1)\right] + t\left[\frac{1}{2}\|y_2 - \overline{x}_2\|_H^2 + \psi(\overline{x}_2)\right] \ ,$$

and thus

$$f((1-t)y_1 + ty_2) \leq (1-t)f(y_1) + tf(y_2).$$

ad 2 (differentiability): Let $y \in H$ and $\Delta y \in H$ be arbitrary and fixed. All subgradients $g \in H^*$ of f yield

$$f(y + \Delta y) \geq f(y) + \langle g(y), \Delta y\rangle_H \ . \tag{25}$$

On the other hand there holds

$$\begin{aligned} f(y+\Delta y) &= \min_{x\in H}\left[\frac{1}{2}\|y + \Delta y - x\|_H^2 + \psi(x)\right] \\ &\leq \frac{1}{2}\|y + \Delta y - \tilde{x}(y)\|_H^2 + \psi(\tilde{x}(y)) \\ &= \frac{1}{2}\langle y + \Delta y - \tilde{x}(y),\ y + \Delta y - \tilde{x}(y)\rangle_H + \psi(\tilde{x}(y)) \ . \end{aligned}$$

Hence,

$$f(y+\Delta y) \leq f(y) + \langle y - \tilde{x}(y),\ \Delta y\rangle_H + \frac{1}{2}\|\Delta y\|_H^2 \ . \tag{26}$$

Subtracting expression (25) from (26) one obtains

$$0 \leq \langle y - \tilde{x}(y), \Delta y\rangle_H - \langle g(y), \Delta y\rangle_H + \frac{1}{2}\|\Delta y\|_H^2 \ .$$

The same inequality must be valid, if we replace Δy by $-\Delta y$, such that there holds

$$-\frac{1}{2}\|\Delta y\|_H^2 \leq \langle y - \tilde{x}(y) - g(y), \Delta y\rangle_H \leq \frac{1}{2}\|\Delta y\|_H^2 \ .$$

Hence, one obtains $y - \tilde{x}(y) - g(y) = 0$ resp. $g(y) = y - \tilde{x}(y)$ (proof by contradiction, draft: assume, that $z := y - \tilde{x}(y) - g(y)$ would not equal zero; choose $\gamma \in (0,1)$;

choose $\Delta y = \gamma z$; contradiction). The sub-differential $g(y)$ is uniquely defined for all $y \in H$, thus a Fréchet-derivative, which is denoted by the symbol $Df := g$, or explicitly

$$Df(y) = \langle y - \tilde{x}(y), \cdot \rangle_H \ .$$

$\square$

Theorem 2 is essential in the so called *Moreau-Yosida regularization* (see [Mor65] and [Yos94]), which has its origin in convex analysis. Hence in this paper we are calling it *Moreau-Yosida theorem*. Applying Theorem 2 and the chain rule to the energy function (24), we can now build its Gâteaux-differential.

$$\begin{aligned} De(u_1, v) &= \langle \varepsilon(u_1) - p_1(\varepsilon(u_1)) \, , \varepsilon(v) \rangle_{\mathbb{C}} - L(v) \\ &= \int_\Omega \mathbb{C} \left(\varepsilon(u_1) - p_1(\varepsilon(u_1)) \right) : \varepsilon(v) \ \mathrm{d}x - L(v), \end{aligned} \tag{27}$$

with p_1 and L defined as in (23) and (14).

3 Discretization and Implementation

Subsections 3.1 and 3.2 are based on [ACFK02].

3.1 Discretization in Space

We decompose the polygonal 2D domain Ω into a triangulation with $\mathrm{N}_\mathcal{T} \sim h^{-d}$ triangles and $\mathrm{N}_\mathcal{N}$ nodes $x_i \in \{1, \dots, \mathrm{N}_\mathcal{N}\}$. Here $\mathrm{N}_\mathcal{T}$ means *number of elements* and $\mathrm{N}_\mathcal{N}$ means *number of nodes*. Let $\mathcal{T}$ be such a domain decomposition in 2D where all $T \in \mathcal{T}$ are triangles with nodes x_i for $i \in \{1, \dots, \mathrm{N}_\mathcal{N}\}$. Let $\mathcal{E}$ be a set of edges $E \in \mathcal{E}$ and let $\mathcal{E}_N$ be its intersection with the Neumann boundary Γ_N. We approximate the infinite-dimensional space V by a finite-dimensional subspace $V_h := \{u_{1h} \in V \mid u_{1h|_T}$ is a linear polynomial $\forall T \in \mathcal{T}\}$. The base functions $\eta_{i,j} \in V_h$, $i \in \{1, \dots, \mathrm{N}_\mathcal{N}\}$, $j \in \{1, 2\}$ are of the form $\eta_{i,j}(x) := \varphi_i(x) e_j$, where $\varphi_i(x)$ is a 1D linear nodal shape (hut) function and e_j is the j-th unit vector. Therefore, u_h can be expressed $u_h(x) := \sum_{i,j} u_{i,j} \eta_{i,j}(x)$, where $u_{i,j} := (u(x_i))_j$. We are interested in finding $u_{1h} \in V_{hD} := V_h \cap V_D$ such that $De(u_{1h}) = 0$. For implementation, u_{1h} is stored as a vector $\mathbf{u} = ((u_{i,j})_{j=1}^2)_{i=1}^{\mathrm{N}_\mathcal{N}} \in \mathbb{R}^{2\mathrm{N}_\mathcal{N}}$. Analogously a test function v_h is represented by the vector $\mathbf{v}$. Let $(k_1, k_2, k_3) := ((x_1, y_1), (x_2, y_2), (x_3, y_3))$ be the vertices of one single element $T \in \mathcal{T}$. For linear elements there holds

$$\nabla \begin{pmatrix} \varphi_{k1} \\ \varphi_{k2} \\ \varphi_{k3} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} .$$

3.2 Vector Representation

We consider the so called *plain model*, which assumes the strain ε or the stress σ to have zero components in direction, where the domain Ω is thin (*plain strain model* or *plain stress model*). The following formulations hold for the plain strain model only, a modification for the plain stress model can be done as well. We assume the total strain ε, the plastic strain p and the stress tensor σ in forms

$$\varepsilon = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & 0 \\ \varepsilon_{12} & \varepsilon_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad p = \begin{pmatrix} p_{11} & p_{12} & 0 \\ p_{12} & p_{22} & 0 \\ 0 & 0 & p_{33} \end{pmatrix}, \quad \sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{12} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{pmatrix}.$$

The information about ε can be saved in the vector $\overrightarrow{\varepsilon} := (\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12})^T$. Since p is trace-free, there must hold $p_{33} = -p_{11} - p_{22}$. Therefore it is sufficient to store p in the vector $\overrightarrow{p} := (p_{11}, p_{22}, p_{12})^T$. Due to $\sigma_{33} = \sigma_{11} + \sigma_{22}$ the stress σ can be saved in the vector $\overrightarrow{\sigma} := (\sigma_{11}\sigma_{22}, \sigma_{12})^T$ too. For the calculation of the energy functional derivative we will make an extent use of identifiers in vector representation. Table 1 summarizes in which way these identifiers will transform. It follows for instance $\sigma_\varepsilon : \varepsilon = (\overrightarrow{\sigma}_\varepsilon)^T \overrightarrow{\varepsilon}$ and $\sigma_p : \varepsilon = (\overrightarrow{\sigma}_p)^T \overrightarrow{\varepsilon}$. Let R_T and R_E be element and edge restriction operators which map the global vector $\mathbf{u}$ on the local element $\mathbf{u}_T$ or edge $\mathbf{u}_E$ vectors

$$\mathbf{u}_T = R_T \mathbf{u}\ , \quad \mathbf{u}_E = R_E \mathbf{u}\ . \tag{28}$$

Since $\varepsilon(u_{1h})$ is constant on each element T there holds

$$\varepsilon_V(u_{1h}(x)_{|T}) = \begin{pmatrix} \partial_x\varphi_{k_1} & 0 & \partial_x\varphi_{k_2} & 0 & \partial_x\varphi_{k_3} & 0 \\ 0 & \partial_y\varphi_{k_1} & 0 & \partial_y\varphi_{k_2} & 0 & \partial_y\varphi_{k_3} \\ \partial_y\varphi_{k_1} & \partial_x\varphi_{k_1} & \partial_y\varphi_{k_2} & \partial_x\varphi_{k_2} & \partial_y\varphi_{k_3} & \partial_x\varphi_{k_3} \end{pmatrix} \begin{pmatrix} u_{k_1,x} \\ u_{k_1,y} \\ u_{k_2,x} \\ u_{k_2,y} \\ u_{k_3,x} \\ u_{k_3,y} \end{pmatrix}$$

or, in a more compact way,

$$\varepsilon_V(u_{1h}(x)_{|T}) = B_T\, \mathbf{u}_T\ . \tag{29}$$

The implementation of B_T in Matlab© reads:

```
function [B,area] = elem_B(vertices)
F = [ones(1,3);vertices'];
area = det(F)/2;
phiGrad = F\[zeros(1,2);eye(2)];
B([1,3],[1,3,5]) = phiGrad'; B([3,2],[2,4,6]) = phiGrad';
end
```

Integration over body and surface forces may be realized by the midpoint rule. We approximate f and g by

$$f_T := f(\overline{x}_T) \quad \text{and} \quad g_E := g(\overline{x}_E),$$

Common (Tensor) Representation	Vector Representation
$\varepsilon := \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & 0 \\ \varepsilon_{12} & \varepsilon_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\vec{\varepsilon} := \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\,\varepsilon_{12} \end{pmatrix}$
$p := \begin{pmatrix} p_{11} & p_{12} & 0 \\ p_{12} & p_{22} & 0 \\ 0 & 0 & -(p_{11}+p_{22}) \end{pmatrix}$	$\vec{p} := \begin{pmatrix} p_{11} \\ p_{22} \\ p_{12} \end{pmatrix}$ with $\Vert p\Vert_F^2 = \vec{p}^{\,T} N\, \vec{p}$ where $N := \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$
$\sigma_\varepsilon := \mathbb{C}\,\varepsilon = \begin{pmatrix} \sigma_{\varepsilon,11} & \sigma_{\varepsilon,12} & 0 \\ \sigma_{\varepsilon,12} & \sigma_{\varepsilon,22} & 0 \\ 0 & 0 & \sigma_{\varepsilon,33} \end{pmatrix}$ with $\mathbb{C} := \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$	$\vec{\sigma}_\varepsilon := \begin{pmatrix} \sigma_{\varepsilon,11} \\ \sigma_{\varepsilon,22} \\ \sigma_{\varepsilon,12} \end{pmatrix} = C\,\vec{\varepsilon}$ with $C := \begin{pmatrix} \lambda+2\,\mu & \lambda & 0 \\ \lambda & \lambda+2\,\mu & 0 \\ 0 & 0 & \mu \end{pmatrix}$ and $\sigma_{\varepsilon,33} = \underbrace{\frac{\lambda}{2\;(\lambda+\mu)}}_{=\nu} \begin{pmatrix}1 & 1 & 0\end{pmatrix}\,\vec{\sigma}_\varepsilon$
$\sigma_p := \mathbb{C}\,p = 2\,\mu\,p + \lambda\,\underbrace{\mathrm{tr}(p)}_{=0}\,I = 2\,\mu\,p$	$\vec{\sigma}_p := \begin{pmatrix} \sigma_{p,11} \\ \sigma_{p,22} \\ \sigma_{p,12} \end{pmatrix} = 2\,\mu\,\vec{p}$ and $\sigma_{p,33} = -\begin{pmatrix}1 & 1 & 0\end{pmatrix}\,\vec{\sigma}_p$
$\sigma := \mathbb{C}\;(\varepsilon - p) = \sigma_\varepsilon - \sigma_p$	$\vec{\sigma} = \vec{\sigma}_\varepsilon - \vec{\sigma}_p$ and $\sigma_{33} = \sigma_{\varepsilon,33} - \sigma_{p,33}$
$\mathrm{tr}\,\sigma_\varepsilon := \sum_i \sigma_{\varepsilon,ii}$	$\mathrm{tr}\,\sigma_\varepsilon = \frac{\nu+1}{\nu}\,\sigma_{\varepsilon,33}$
$\mathrm{dev}\,\sigma_\varepsilon := \sigma_\varepsilon - \frac{\mathrm{tr}\,\sigma_\varepsilon}{\dim(\sigma_\varepsilon)}\,I$	$\overrightarrow{\mathrm{dev}\,\sigma_\varepsilon} := \begin{pmatrix} (\mathrm{dev}\,\sigma_\varepsilon)_{11} \\ (\mathrm{dev}\,\sigma_\varepsilon)_{22} \\ (\mathrm{dev}\,\sigma_\varepsilon)_{12} \end{pmatrix} = \vec{\sigma}_\varepsilon - \frac{\mathrm{tr}\,\sigma_\varepsilon}{\dim(\sigma_\varepsilon)} \begin{pmatrix}1\\1\\0\end{pmatrix}$ $\Rightarrow \quad \overrightarrow{\mathrm{dev}\,\sigma_\varepsilon} = \underbrace{\left(I - \frac{\nu+1}{\dim(\sigma_\varepsilon)} \begin{pmatrix}1 & 1 & 0\\1 & 1 & 0\\0 & 0 & 0\end{pmatrix}\right)}_{=:K} \vec{\sigma}_\varepsilon$
$\mathrm{dev}\,\sigma = \mathrm{dev}\,\sigma_\varepsilon - \underbrace{\mathrm{dev}\,\sigma_p}_{=\sigma_p}$	$\overrightarrow{\mathrm{dev}\,\sigma} = \overrightarrow{\mathrm{dev}\,\sigma_\varepsilon} - \vec{\sigma}_p$ $(\mathrm{dev}\,\sigma)_{33} = -\begin{pmatrix}1 & 1 & 0\end{pmatrix}\,\vec{\sigma}$
$\Vert\,\mathrm{dev}\,\sigma\Vert_F^2 := \sum_{i,j}\,(\mathrm{dev}\,\sigma)_{ij}^2$	$\Vert\,\mathrm{dev}\,\sigma\Vert_F^2 = \left(\overrightarrow{\mathrm{dev}\,\sigma}\right)^T N\,\overrightarrow{\mathrm{dev}\,\sigma}$

Table 1: Table of Vector Representation

where $\overline{x}_T$, and $\overline{x}_E$ respectively, denote the center of mass of the element T, and the edge E respectively. Defining

$$\mathbf{f}_T := \frac{|T|}{3}\, R_T^T\, f_T \ , \quad \text{and} \quad \mathbf{g}_E := \frac{|E|}{2}\, R_E^T\, g_E \ ,$$

on each $T \in \mathcal{T}$ and on each $E \in \mathcal{E}$ there hold

$$\int_T f \cdot v \, \mathrm{d}x \approx \mathbf{f}_T^T \mathbf{v} \ , \quad \text{and} \quad \int_E g \cdot v \, \mathrm{d}S(x) \approx \mathbf{g}_E^T \mathbf{v} \ . \tag{30}$$

This can be realized by the following two Matlab© functions:

```
function f_ = elem_volumeforce(vertices)
T = det([ones(3,1),vertices]);
fs = f(sum(vertices)/3)';
f_ = [fs; fs; fs]*T/6;
end
```

```
function g_ = elem_surfaceforce(vertices)
n = (vertices(2,:) - vertices(1,:))*[0,-1; 1,0];
T = norm(n);
gs = g_neumann(sum(vertices)/2,n/norm(n))';
g_ = [gs; gs]*T/2;
end
```

3.3 Calculation of the Discrete Formulation and a Newton-like Method

The Gâteaux-derivative in (27) will now be lead into its discrete analogon. First, the whole integral over Ω will be split into a sum of integrals on single finite elements. Combining (28), (29) and (30) we obtain the discrete formulation of the Gâteaux-differential

$$De(\mathbf{u},\mathbf{v}) = \sum_{T \in \mathcal{T}} \left[|T| \ \left(C\, B_T \mathbf{u}_T - 2\mu\, \overrightarrow{p_1}(B_T \mathbf{u}_T) \right)^T \ B_T\, R_T - \mathbf{f}_T^T \right] \mathbf{v} - \sum_{E \in \mathcal{E}_N} \mathbf{g}_E^T\, \mathbf{v}$$

with

$$\overrightarrow{p_1}(B_T \mathbf{u}_T) = \begin{cases} \overrightarrow{p_0} & \text{if } \|\operatorname{dev} A\| < \beta, \\ \frac{1}{\delta}\, \overrightarrow{\operatorname{dev} A} \left(1 - \frac{\beta}{\|\operatorname{dev} A\|} \right) + \overrightarrow{p_0} & \text{else,} \end{cases}$$

where

$$\begin{aligned} \overrightarrow{\operatorname{dev} A} &= KC\, B_T\, \mathbf{u}_T + 2\, \mu\, \overrightarrow{p_0} \ , \\ \|\operatorname{dev} A\| &= \left(\left(\overrightarrow{\operatorname{dev} A} \right)^T N\, \overrightarrow{\operatorname{dev} A} \right)^{\frac{1}{2}} =: \|\overrightarrow{\operatorname{dev} A}\|_N, \\ \beta &= \sigma_y \left(1 + \alpha_0 H \right), \\ \delta &= 2\mu + \sigma_y^2 H^2. \end{aligned}$$

Since $De(\mathbf{u}, \mathbf{v})$ is linear in $\mathbf{v}$ we can obtain the Fréchet-derivative

$$De(\mathbf{u}) = \sum_{T\in\mathcal{T}} \Big(|T| \; (CB_T\mathbf{u}_T - 2\mu\, \overrightarrow{p_1}(B_T\mathbf{u}_T))^T \; B_T\, R_T - \mathbf{f}_T \Big) - \sum_{E\in\mathcal{E}_N} \mathbf{g}_E, \tag{31}$$

which represent the discrete form of (27). Note that the second derivative $D^2e(\mathbf{u})$ can be calculated everywhere apart from the the material points satisfying the condition $\|\operatorname{dev} A\| \neq \beta$ and reads

$$\begin{aligned} D^2e(\mathbf{u}) &= D\left(\sum_{T\in\mathcal{T}} |T| \; (CB_T\mathbf{u}_T - 2\mu\, \overrightarrow{p_1}(B_T\mathbf{u}_T))^T \; B_T\, R_T \right) \\ &= \sum_{T\in\mathcal{T}} |T|\, R_T^T\, B_T^T\, (C - 2\mu\, D_{B_T\mathbf{u}_T}\, \overrightarrow{p_1}(B_T\mathbf{u}_T))^T \; B_T\, R_T \;, \end{aligned}$$

with

$$D_{B_T\mathbf{u}_T}\; \overrightarrow{p_1}(B_T\mathbf{u}_T) = \begin{cases} 0 & \text{if } \|\operatorname{dev} A\| < \beta \\ \frac{1}{\delta}\, w' - \frac{\beta}{\delta} \left(\frac{w'}{(w^T N w)^{\frac{1}{2}}} - \frac{w\, w^T N w'}{(w^T N w)^{\frac{3}{2}}} \right) & \text{if } \|\operatorname{dev} A\| > \beta, \end{cases} \tag{32}$$

where $w(\mathbf{u}) := \overrightarrow{\operatorname{dev} A}$ and $w' := D_{B_T\mathbf{u}_T}\overrightarrow{\operatorname{dev} A} = K\, C$. In other words, the second derivative exists in purely elastic material points as well as in purely plastic material points. It only does not exist in the elastoplastic interface points where $\|\operatorname{dev} A\| = \beta$. For numerical computations we will use the value corresponding to the case $\|\operatorname{dev} A\| > \beta$ also in the critical interface case, where $\|\operatorname{dev} A\| = \beta$.

The Newton-like method is applied for the calculation of $\mathbf{u} \in \mathbf{R}^{d\cdot \mathrm{N}_{\mathcal{N}}}$ such that $De(\mathbf{u}) = 0$ and $\mathbf{u}$ satisfies the Dirichlet boundary condition:

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u}_i \qquad (\forall i \in \mathbb{N}), \tag{33}$$

where $\Delta\mathbf{u}_i$ solves

$$-D^2e(\mathbf{u}_{i-1})\; \Delta\mathbf{u}_i = De(\mathbf{u}_{i-1}) \;.$$

The assembly of the stiffness matrix `EnergyDD` (it stands for $D^2e(u)$) and the right hand side `EnergyD` ($De(u)$) is implemented in the Matlab© as:

```
function [EnergyD,EnergyDD] = energy_derivatives(u,p0Initial, ...
nodes,elements,neumann,N,K,C,param)
SU = size(u,1);
EnergyD = zeros(SU,1); EnergyDD = sparse(SU,SU);
for j = 1:size(elements,1)
    vertices = nodes(elements(j,:),:);
    I = 2*elements(j,[1,1,2,2,3,3]) - [1,0,1,0,1,0];
    [B,area] = elem_B(vertices);
    eps = B*u(I);
    p0 = p0Initial(:,j);
```

```
    [p,pD_eps] = elem_p(j,eps,p0,N,K,C,param);
    ED(I) = ED(I) + area*B'*(C*eps - 2*param.mu*p);
    ED(I) = ED(I) - elem_volumeforce(vertices);
    EDD(I,I) = EDD(I,I) + area*B'*(C - 2*param.mu*pD_eps)*B;
end
for j = 1:size(neumann,1)
    vertices = nodes(neumann(j,:),:);
    I = 2*neumann(j,[1,1,2,2]) - [1,0,1,0];
    ED(I) = ED(I) - elem_surfaceforce(vertices);
end
```

The calculation of $p(\varepsilon(u))$ and $D_{\varepsilon(u)}\, p(\varepsilon(u))$ on the $j-th$ element T_j can be realized by the Matlab© function `elem_p`:

```
function [p,pD_eps] = elem_p(j,eps,p0,N,K,C,param)
p = p0;
pD_eps = zeros(3,3);
devAD_eps = K*C;
devA = devAD_eps*eps - 2*param.mu*p0;
norm_devA = sqrt(devA'*N*devA);
posFac = norm_devA - param.beta(j);
if (posFac > 0)
    p = devA*posFac/(param.delta*norm_devA) + p0;
    pD_eps = (devAD_eps - (param.beta(j)/norm_devA)* devAD_eps ...
          - (devA*((devA'*N)*devAD_eps)) norm_devA^2))/param.delta;
end
```

Note, that $\mathbf{u}_i$ must satisfy (generally inhomogeneous) Dirichlet boundary conditions for all $i \in \mathbb{N}$. Therefore it is sufficient for the initial approximation $\mathbf{u}_0$ to solve the inhomogeneous Dirichlet conditions, and for $\Delta\mathbf{u}_i$ to solve the homogeneous Dirichlet conditions. For the termination of the Newton-like method we check, whether the relative error of the discrete approximation

$$\frac{|u_i - u_{i-1}|_\varepsilon}{|u_i|_\varepsilon + |u_{i-1}|_\varepsilon} \tag{34}$$

is smaller than a given prescribed bound $\epsilon \in \mathbb{R}^+$. Note that the seminorm

$$|\cdot|_\varepsilon := \Big(\int_\Omega ||\varepsilon(\cdot)||^2 \,\mathrm{d}x\Big)^{1/2}$$

is more easily computable than an equivalent H_1 norm

$$||\cdot||_1 := \Big(\int_\Omega (||\cdot||^2 + ||\nabla\cdot||^2)\,\mathrm{d}x\Big)^{1/2}.$$

The quality of the iterated solutions is measured by the relative error of the energy as

$$\frac{|e(u_i) - e(u_{i-1})|}{|e(u_i)| + |e(u_{i-1})|}. \tag{35}$$

The implementation of (34) reads:

```
function norm_delta_u = termination_criterion(u,u_old,Delta_u,...
nodes, elements)
N_eps = [1,0,0; 0,1,0; 0,0,0.5];
norm_u_old_sq = 0; norm_u_sq = 0; norm_du_sq = 0; norm_ED_sq = 0;
for j = 1:size(elements,1)
    vertices = nodes(elements(j,:),:);
    [B,area] = elem_B(vertices);
    eps_u_old = B*u_old(I); eps_u = B*u(I);
    eps_Delta_u = B*Delta_u(I);
    norm_u_old_sq = norm_u_old_sq + ...
                          area*(eps_u_old'*N_eps*eps_u_old);
    norm_u_sq = norm_u_sq + area*(eps_u'*N_eps*eps_u);
    norm_Delta_u_sq = norm_Delta_u_sq + ...
                      area*(eps_Delta_u'*N_eps*eps_Delta_u);
end
norm_Delta_u = sqrt(norm_Delta_u_sq);
norm_u = sqrt(norm_u_sq);
norm_u_old = sqrt(norm_u_old_sq);
norm_delta_u = norm_Delta_u / (norm_u_old + norm_u);
```

The implementation of the Newton-like method is realized via the Matlab© file fem.m.

```
d = [1; 1; 0];
K = eye(3) - d*d'*(1 + param.nu)/3;
N = [2,1,0;1,2,0;0,0,2];
C = param.lambda*[1,1,0;1,1,0;0,0,0] + param.mu*[2,0,0;0,2,0;0,0,1];
u = zeros(2*size(nodes,1),1); Delta_u = u; ED = u;
p0Initial = zeros(size(elements,1),3);
isPlasticElement = zeros(1,size(elements,1));
EDD = sparse(size(u,1),size(u,1));
[I,J] = separate_into_prescribed_and_free_nodes(...)
u = apply_dirichlet_boundary_conditions(...)
newton_epsilon = 1e-12; newton_iterations = 0;
norm_delta_u = newton_epsilon;
while (norm_delta_u >= newton_epsilon)
    [ED,EDD] = energy_derivatives(u,p0Initial,nodes,elements, ...
                neumann,N,K,C,param);
    ED(J) = ED(J) + EDD(J,:)*u(J);
    u_old = u;
    Delta_u(I) = solve_linear_system(EDD(I,I),ED(I),Delta_u(I));
    u(I) = u(I) - Delta_u(I);
    newton_iterations = newton_iterations + 1;
    norm_delta_u = termination_criterion(u_old, u, Delta_u);
end
post_processing
end
```

3.4 Other techniques used

There are two additional numerical techniques implemented for the solution of the elastoplastic problem with perfectly plastic material ($H = 0$) in Example 3.

Nested Iteration Technique In order to obtain a reasonable initial approximation for the Newton-like method on the finer mesh, one can prolongate the solution u from the coarser mesh and take this value as the initial approximation u_0. This so called *nested iteration technique*, for a detailed info see [Hac85].

Damping Technique The idea of damping is to replace the upgrade of $\mathbf{u}_i$ defined in (33) by

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \alpha_i \Delta \mathbf{u}_i$$

with a *damping* parameter $\alpha_i \leq 1$. The following strategy to determine α_i is based on the comparison of energies: α_i is originally set to 1 and halfened until the energy functional is reduced, i.e., it holds

$$e(u_{i-1} + \alpha_i \Delta u_i) < e(u_{i-1}). \tag{36}$$

4 Numerical Examples

The following tests were calculated on a computer with 2.4 GHz CPU, 2 Gb RAM using Matlab© version 7.0 on Linux OS. The quality of the discrete solution is measured by a global error estimator η defined as

$$\eta = \frac{\sqrt{\sum_{T \in \mathcal{T}_h} \eta_T^2}}{\sqrt{\sum_{T \in \mathcal{T}_h} \int_T \sigma_h^* : \mathbb{C}^{-1} \sigma_h^* \, \mathrm{d}x}}\,, \tag{37}$$

where

$$\eta_T^2 = \int_T (\sigma_h - \sigma_h^*) : \mathbb{C}^{-1} \left(\sigma_h - \sigma_h^*\right) \, \mathrm{d}x\,,$$

and σ_h^* is a nodal Clement interpolation of the piecewise constant σ_h defined as

$$\sigma_h^*(v) = \frac{\sum_{\{k | v \in \overline{T_k}\}} |T_k| \sigma_h(T_k)}{\sum_{\{k | v \in \overline{T_k}\}} |T_k|}.$$

We define 'DOF' as the short form of *degrees of freedom*, and 'VPZ' to be the short form of *variation in plastic zones* which is calculated as follows: In the i-th iteration step the boolean vector $\mathbf{w}^i$ stores the information about which elements are plastic and which are not by defining its components

$$w_j^i := \begin{cases} 1 & \text{if } T_j \text{ is a plastic element,} \\ 0 & \text{else.} \end{cases}$$

Let the starting vector $\mathbf{w}^0 = 0$. Variation in plastic zones VPZ_{i-1}^i from the $(i-1)$-st to the i-th iteration step is defined by

$$\mathrm{VPZ}_{i-1}^i = \frac{100}{\mathrm{N}_{\mathcal{T}}} \sum_{j=1}^{n} |w_j^i - w_j^{i-1}|.$$

It all numerical examples, the termination bound $\epsilon = 1e - 12$ was used.

Example 1 (L-Shape). *This example is taken from [ACFK02] and its geometry and the coarse grid triangulation are displayed in Figure 1. We assume nonhomogeneous Dirichlet boundary conditions in polar coordinates r, θ*

$$\begin{aligned} u_r(r,\theta) =& \frac{1}{2\mu} r^\alpha \left[-(\alpha+1)\cos((\alpha+1)\,\theta) + (\mathrm{C}_2 - (\alpha+1))\mathrm{C}_1 \cos((\alpha-1)\,\theta) \right], \\ u_\theta(r,\theta) =& \frac{1}{2\mu} r^\alpha \left[(\alpha+1)\sin((\alpha+1)\,\theta) + (\mathrm{C}_2 + (\alpha-1))\mathrm{C}_1 \sin((\alpha-1)\,\theta) \right]. \end{aligned} \tag{38}$$

The critical exponent $\alpha \approx 0.544483737$ is the solution of the equation

$$\alpha \sin(2\omega) + sin(2\omega\alpha) = 0$$

with $\omega = \frac{3\pi}{4}$ and $\mathrm{C}_1 = -(\cos((\alpha+1)\omega))/\cos((\alpha-1)\omega), \mathrm{C}_2 = (2(\lambda+2\mu))/(\lambda+\mu)$. It can be shown that the formulae (38) describe the solution of the purely elastic problem with the same nonhomogeneous Dirichlet boundary conditions also in the interior of the Lshape domain. Thus there is an strain-singularity in the reentrant corner, which can also be expected for the elastoplastic case. The material parameters are defined as

$$E = 1e5,\ \nu = 0.3,\ \sigma_Y = 2.2,\ H = 1.$$

Figure 2 shows the yield function (right) and the elastoplastic zones (left), where purely elastic zones are colored green (light gray in case of a non-color print respectively), and elastoplastic zones are colored pink (dark grey respectively). The domain's displacement is multiplied by factor 3e3. *Table 2 reports on convergence behavior of Newton-like method for graduated uniform meshes.*

Example 2 (Wrench). *This example simulates the deformation of a screw-wrench under pressure. Problem geometry is shown in Figure 3: A screw-wrench* sticks *on a screw (homogeneous Dirichlet boundary condition) and a surface load g is applied to a part of the wrench's handhold in interior normal direction (Neumann boundary condition). The material parameters are set*

$$E = 2e8,\ \nu = 0.3,\ \sigma_Y = 2e6,\ H = 0.001$$

and the problem was calculated traction $g(x) = 6e4$. Figure 4 shows the yield function (right) and the elastoplastic zones (left), where purely elastic zones are colored green (light gray in case of a non-color print respectively), and elastoplastic zones are colored pink (dark grey respectively). The displacement of the domain is multiplied by factor 10. *Table 3 reports on the convergence of the Newton-like method for graduated uniform meshes.*

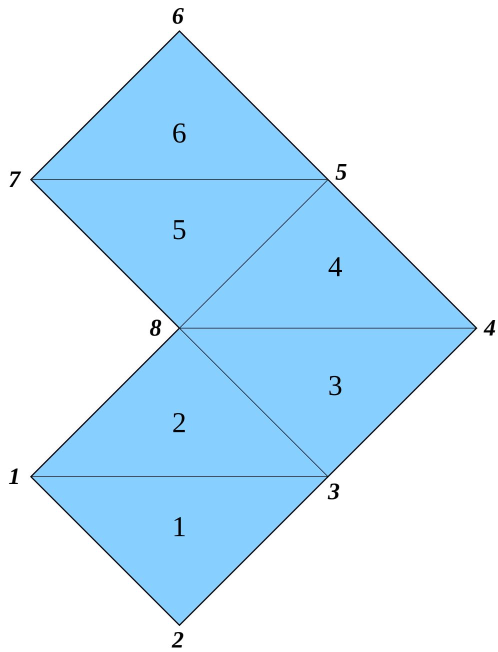

Figure 1: Problem geometry and the coarse triangulation of Example 1. The L-shape domain Ω is described by the polygon $(-1,-1)$, $(0,-2)$, $(2,0)$, $(0,2)$, $(-1,1)$, $(0,0)$.

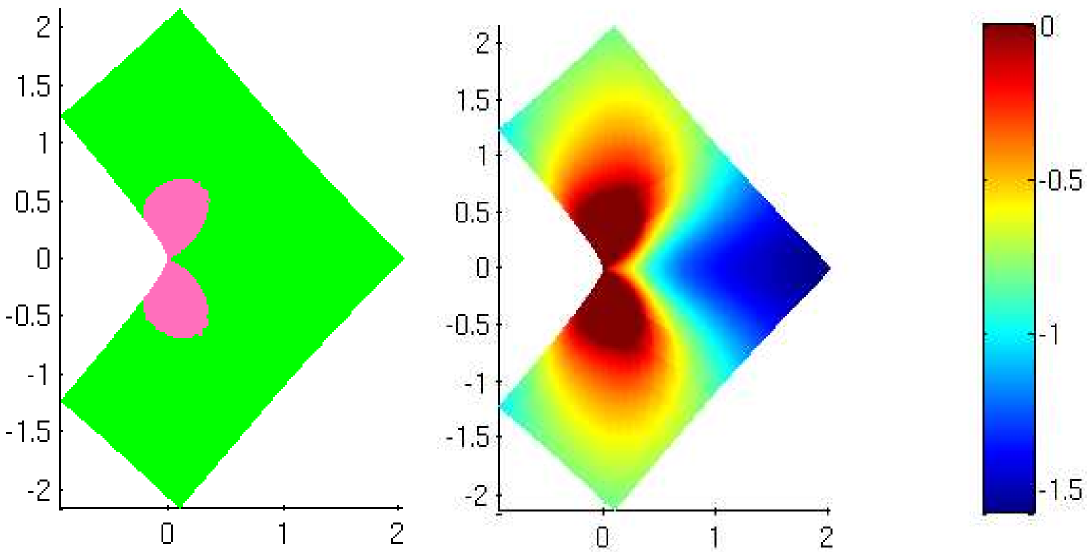

Figure 2: Elastoplastic zones (left) and yield function (right) of the deformed domain in Example 1. The displacement is magnified by factor $3e3$.

Level	1	2	...	6	7	8
DOF	10	66	...	20466	97282	391170
relative error:						
step 1	2.8383e-02	3.9827e-02	...	7.2243e-02	7.0236e-02	6.8321e-02
step 2	1.0467e-04	1.2352e-03	...	1.1004e-02	1.1063e-02	1.1022e-02
step 3	2.3781e-09	6.1409e-07	...	1.1453e-03	1.2746e-03	1.3552e-03
step 4	1.0944e-16	2.9589e-13	...	2.0826e-05	4.0743e-05	5.9611e-05
step 5			...	6.8005e-09	5.1957e-08	2.0693e-07
step 6			...	5.2211e-15	1.3866e-13	4.3361e-12
step 7			...			1.8774e-14
VPZ (%):						
step 0-1	16.67	10.42	...	10.59	10.61	10.62
step 1-2	0	2.083	...	2.873	2.816	2.752
step 2-3	0	0	...	0.2686	0.2218	0.1638
step 3-4	0	0	...	0.04069	0.02848	0.01882
step 4-5			...	0	0	0
step 5-6			...	0	0	0
step 6-7			...			0
Time (sec.)	2.00537	2.25042	...	142.29	590.106	2692.87
Error est.	0.401437	0.2482	...	0.0409114	0.0272622	0.0181553

Table 2: Convergence table in Example 1 **(Lshape)**. The table displays the relative error in displacements (34) and the variation of plastic zones (VPZ) per iteration step for various uniformly refined meshes. The quality of the discrete solutions is measured by the global error estimator (37).

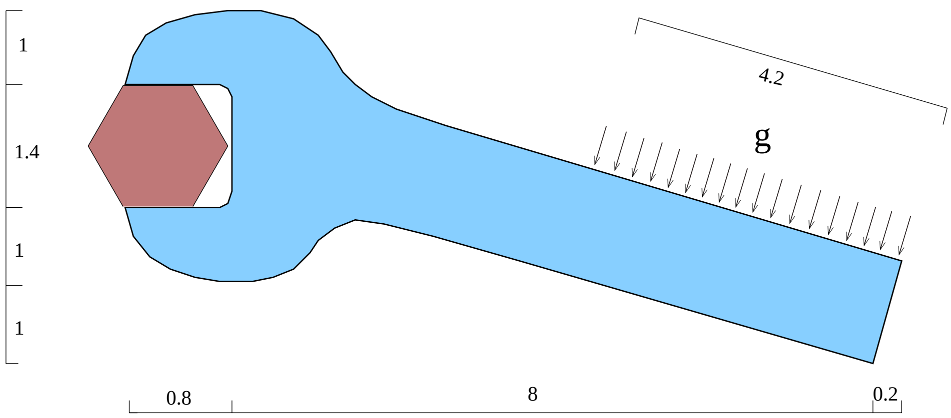

Figure 3: Problem geometry in Example 2.

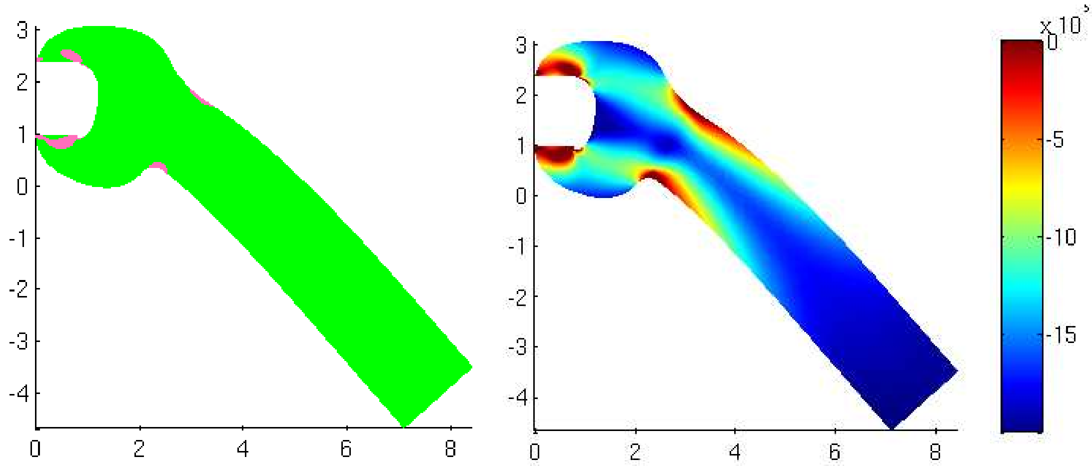

Figure 4: Elastoplastic zones (left) and yield function (right) of the deformed domain in Example 2. The displacement is magnified by factor 10.

Level	0	1	...	5	6	7
DOF	60	202	...	41662	165246	658174
relative error:						
step 1	2.3834e-14	3.6169e-03	...	1.3194e-01	1.4872e-01	1.5846e-01
step 2		2.3598e-06	...	5.6966e-02	6.9302e-02	7.9603e-02
step 3		1.5324e-11	...	7.5805e-03	1.3223e-02	2.9909e-02
step 4		4.5752e-15	...	4.0307e-04	2.4344e-03	3.5626e-03
step 5			...	5.9665e-06	2.1840e-04	1.2013e-04
step 6			...	2.9485e-10	1.5089e-05	1.0364e-05
step 7			...	7.8696e-14	3.8914e-09	1.1642e-09
step 8			...		1.5508e-13	2.9988e-13
VPZ (%):						
step 0-1	0	1.25	...	1.819	1.83	1.828
step 1-2		0	...	0.9741	1.168	1.27
step 2-3		0	...	0.3564	0.5591	0.7588
step 3-4		0	...	0.05127	0.1501	0.1418
step 4-5			...	0.002441	0.02563	0.02319
step 5-6			...	0	0.00183	0.004425
step 6-7			...	0	0	0
step 7-8			...		0	0
Time (sec.)	1.31385	2.58625	...	262.304	1177.64	4892
Error est.	0.780432	0.53131	...	0.0868307	0.0758023	0.0421956

Table 3: Convergence table in Example 2 **(wrench)**. The table displays the relative error in displacements (34) and the variation of plastic zones (VPZ) per iteration step for various uniformly refined meshes. The quality of the discrete solutions is measured by the global error estimator (37).

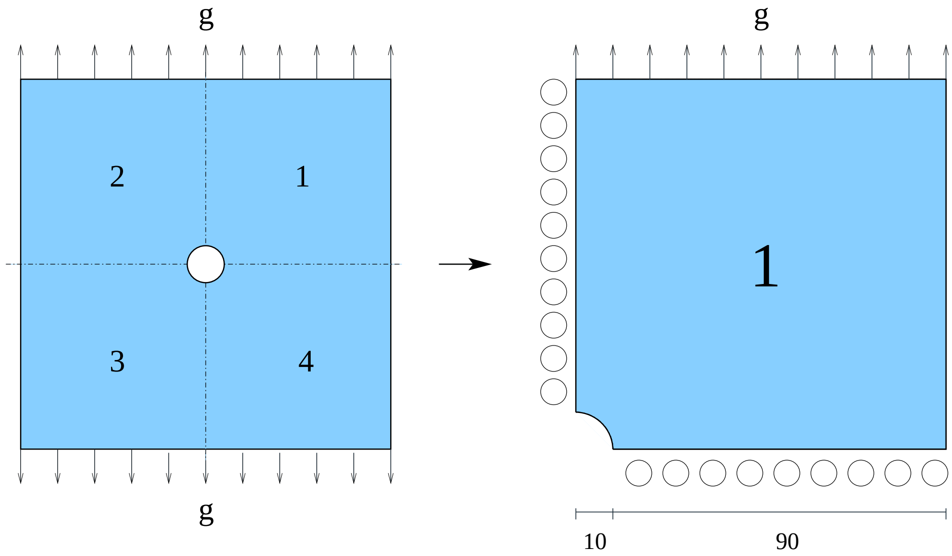

Figure 5: Problem geometry in Example 3.

Example 3 (Plate with a Hole)**.** *This example is taken from [ea02] and serves as a benchmark problem in computational plasticity. The example domain is a thin plate represented by the square* $(-10, 10)\times(-10, 10)$ *with a circular hole of the radius* $r = 1$ *in the middle, as can be seen in Figure 5. A surface load* g *is applied on the plate's upper and lower edge. Due to symmetric geometry only the right upper quarter of the domain is discretized. Therefore it is necessary to incorporate homogeneous Dirichlet boundary conditions in the normal direction (gliding conditions) to both symmetric edges. The material parameters are set*

$$E = 206900,\ \nu = 0.29,\ \sigma_Y = \sqrt{\frac{2}{3}}\,450,\ H = 0.$$

It means our model is a perfect plasticity model. Figure 6 shows the yield function (right) and the elastoplastic zones, where purely elastic zones are colored green (light gray in case of a non-color print respectively), and elastoplastic zones are colored pink (dark grey respectively). The domains's displacement is multiplied by 100*. Table 4 reports on the convergence of the Newton-like method for graduated uniform meshes. It turns out that the non-nested iteration technique which was successful in other previous example does not work here for finest mesh. Therefore, the nested iteration technique or the damping technique are used, see Table 5 for more details.*

Acknowledgments

The authors are pleased to acknowledge support by the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' for their support under grant SFB F013/F1306 in Linz, Austria. The idea of looking at the elasto-plastic formulation in terms of the Moreau-Yosida Theorem came out during working progresses with H. Gfrerer, J. Kienesberger and U. Langer.

Level	0	1	...	3	4	5
DOF	245	940	...	14560	57920	231040
Non-nested iteration technique						
relative error:						
step 1	2.4308e-02	5.4158e-02	...	8.0495e-02	8.4982e-02	8.6934e-02
step 2	9.8924e-03	2.6174e-02	...	7.0273e-02	8.2874e-02	9.7932e-02
step 3	1.0180e-03	4.1482e-03	...	2.6452e-02	3.8208e-02	1.4122e-01
step 4	1.1566e-06	4.6425e-04	...	3.5268e-03	6.1751e-03	4.6951e-01
step 5	4.0331e-12	1.6937e-05	...	1.7461e-04	4.5520e-04	8.3690e-01
step 6	4.3289e-16	1.3371e-09	...	3.6869e-06	6.0711e-05	9.5657e-01
step 7		9.3330e-16	...	2.4166e-10	4.5021e-07	9.8160e-01
step 8			...	3.6081e-15	5.4823e-12	...
step 9			...		7.3296e-15	diverging
VPZ (%):						
step 0-1	2.667	4.778	...	5.299	5.345	5.356
step 1-2	3.11	5	...	7.104	7.252	7.498
step 2-3	0.4444	2.444	...	4.674	5.045	5.509
step 3-4	0	0.6667	...	1.639	2.08	0.5994
step 4-5	0	0.1111	...	0.3472	0.4948	1.815
step 5-6	0	0	...	0.02778	0.07465	0.3416
step 6-7		0	...	0	0.006944	0.1141
step 7-8			...	0	0	...
step 8-9			...		0	diverging
Time (sec.)	3.23081	8.16373	...	106.88	468.615	-
Nested iteration technique						
relative error:						
step 1	2.4308e-02	5.5612e-02	...	5.1909e-02	3.9135e-02	2.4310e-02
step 2	9.8924e-03	1.8466e-02	...	1.0428e-02	7.3208e-03	3.1321e-03
step 3	1.0180e-03	2.6082e-03	...	1.2247e-03	1.6236e-03	5.9788e-04
step 4	1.1566e-06	2.1786e-04	...	7.0150e-05	1.6808e-04	1.4709e-04
step 5	4.0330e-12	1.6854e-05	...	6.1389e-08	3.3952e-06	3.1330e-06
step 6	5.6042e-16	1.3203e-09	...	8.1421e-14	2.4807e-10	1.1140e-09
step 7		7.9697e-16	...		6.8650e-15	1.5225e-14
VPZ (%):						
step 0-1	2.667	6.333	...	14.7	15.24	15.28
step 1-2	3.11	3.889	...	2.257	1.307	0.6021
step 2-3	0.4444	2.111	...	0.8125	0.3594	0.1402
step 3-4	0	0.3333	...	0.1458	0.1181	0.03472
step 4-5	0	0.1111	...	0.006944	0.0191	0.007378
step 5-6	0	0	...	0	0	0
step 6-7		0	...		0	0
Time (sec.)	3.2685	11.1333	...	119.328	492.936	2119.71
Error est.	0.0519797	0.0456903	...	0.0244116	0.0153209	0.00881961

Table 4: Convergence table in Example 3 **(plate with a hole)**. The table displays the relative error in displacements (34) and the variation of plastic zones (VPZ) per iteration step for various uniformly refined meshes. The quality of the discrete solutions is measured by the global error estimator (37).

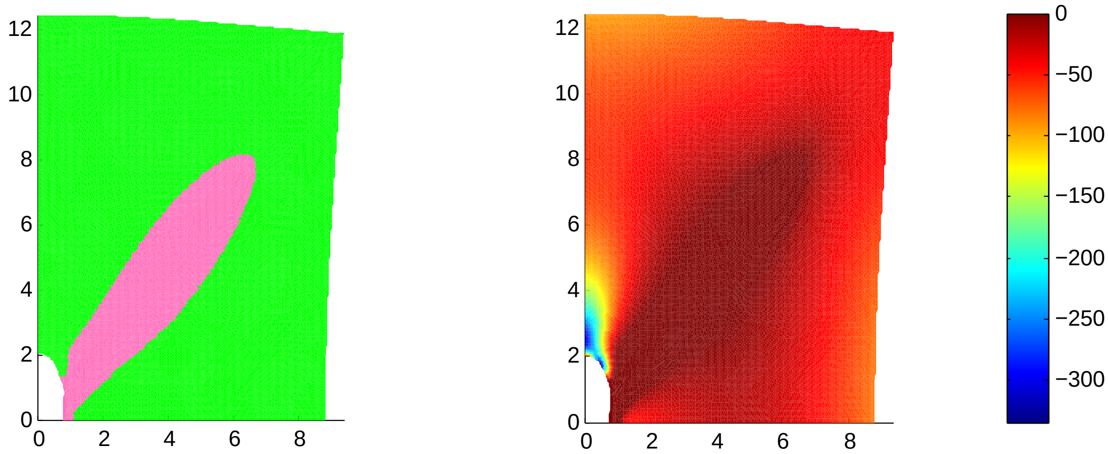

Figure 6: Elastoplastic zones (left) and yield function (right) of the deformed domain in Example 3 . The displacement is magnified by factor 100.

Problem 'Plate with a Hole' at Level 5 (DOF=231040)					
	Standard Method		Damped Method		
	Error in $e(u)$	Error in u	Error in $e(u)$	Error in u	k
step 1	-3.1619e-01	8.6934e-02	-3.4320e-03	8.6934e-02	0
step 2	-5.1283e-02	9.7932e-02	-5.5442e-04	9.7932e-02	0
step 3	2.8336e-01	1.4122e-01	-6.2037e-04	1.4517e-01	2
step 4	2.1854e+00	4.6951e-01	-1.3852e-04	4.8643e-02	0
step 5	7.2304e+00	8.3690e-01	-1.9002e-04	7.9731e-02	2
step 6	2.1073e+01	9.5657e-01	-6.3455e-05	2.3192e-02	1
step 7	5.9547e+01	9.8160e-01	-2.0840e-05	1.1869e-02	1
step 8	1.3702e+02	9.8480e-01	-1.2660e-05	7.7685e-03	1
step 9	3.6256e+02	9.8221e-01	-6.1992e-06	2.3371e-03	0
step 10	9.0941e+02	9.8513e-01	-6.5443e-08	5.3965e-04	0
step 11	1.5903e+03	9.8493e-01	-9.1647e-09	2.9142e-04	0
step 12	3.1730e+03	9.7935e-01	-4.5144e-11	2.3221e-05	0
step 13	1.0887e+04	9.7633e-01	-1.3790e-15	1.3469e-07	0
step 14	...	...	-2.8347e-15	4.4192e-12	0
step 15	not converging		9.9597e-16	1.4070e-14	0

Table 5: Complementary convergence table in Example 3 **(plate with a hole)**. The standard Newton-like method with a zero initial approximation does not converge in Example 3 for refinement level 5 with 231040 degrees of freedom. Thus, the damping technique is applied. Both the standard (non-damped with $\alpha = 1$) and the damped Newton-like method ($\alpha_i = 2^{-k}$, where k denotes the number of damping steps) are compared: the relative errors in the displacement u_i and the relative errors in the energy $e(u_i)$ at the i-th iteration step are computed by (34) and (35). The right most column reports on how many damping steps k have been necessary to guarantee the energy reduction (36).

References

[AC00] J. Alberty and C. Carstensen, *Numerical analysis of time-depending primal elastoplasticity with hardening*, SIAM J. Numer. Anal. **37** (2000), no. 4, 1271–1294.

[ACFK02] J. Alberty, C. Carstensen, S. A. Funken, and R. Klose, *Matlab implementation of the finite element method in elasticity*, Computing **69** (2002), no. 3, 239–263. MR 1 954 562

[ACZ99] J. Alberty, C. Carstensen, and D. Zarrabi, *Adaptive numerical analysis in primal elastoplasticity with hardening*, Comput. Methods Appl. Mech. Eng. **171** (1999), no. 3-4, 175–204.

[Bla97] R. Blaheta, *Numerical methods in elasto-plasticity*, Comp. Meth. Appl. Mech. Engrg. **147** (1997), 167–185.

[DL76] G. Duvaut and Lions J. L., *Numerical analysis of variational inequalities*, Springer-Verlag Berlin Heidelberg, 1976.

[ea02] Stein et al. (ed.), *Error-controlled adaptive finite elements in solid mechanics*, John Wiley & Sons, 2002.

[Hac85] W. Hackbusch, *Multi-grid methods and applications*, Springer-Verlag Berlin, Heidelberg, 1985.

[HR99] W. Han and B.D. Reddy, *Plasticity: Mathematical theory and numerical analysis*, Springer-Verlag New York, 1999.

[Joh76] C. Johnson, *Existence theorems for plasticity problems*, J. math. pures et appl. **55** (1976), 431–444.

[KL84] V. G. Korneev and U. Langer, *Approximate solution of plastic flow theory problems*, Teubner-Texte zur Mathematik, vol. 69, Teubner-Verlag, Leipzig, 1984.

[Kos91] P. Kosmol, *Optimierung und Approximation*, Walter de Gruyter, 1991.

[Mor65] J.J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bulletin de la Société Mathématique de France **93** (1965), 273–299.

[SH98] J.C. Simo and T.J.R. Hughes, *Computational inelasticity*, Springer-Verlag New York, 1998.

[Yos94] K. Yosida, *Functional Analysis*, Springer, 1994, Reprint of the 6th edition.

d	$\in \mathbb{N}$, space dimension
Ω	$\subset \mathbb{R}^d$, open domain
Γ	$= \partial\Omega$, domain boundary
Γ_D	$\subset \Gamma$, Dirichlet boundary (prescribed deformations)
Γ_N	$\subset \Gamma$, Neumann boundary (prescribed surface forces)
n	outer normal of Γ
σ	$\in C^1(\Omega)^{d\times d}$, stress
ε	$\in C^1(\Omega)^{d\times d}$, elastic strain
u	$\in C^2(\Omega)^d$, deformation
f	$\in C(\Omega)^d$, body forces
u_D	$\in C(\Gamma_D)^d$, prescribed deformations on Γ_D
g	$\in C(\Gamma_N)^d$, prescribed surface forces on Γ_N
λ	$\in \mathbb{R}^+$, "Lamé modulus", Lamé constant
μ	$\in \mathbb{R}^+$, "sheer modulus", Lamé constant
E	$\in \mathbb{R}^+$, "Young's modulus"
ν	$\in \left[0, \frac{1}{2}\right]$, "Poisson ratio"
δ_{ij}	"Kronecker delta"
$\mathbb{C}$	$\in \mathbb{R}^{d\times d}_{d\times d}$ with $\mathbb{C}_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$, elasticity tensor
p	$\in C^1(\Omega)^{d\times d}$, plastic strain
α	$\in C(\Omega, \mathbb{R}^+)$, hardening parameter (function)
H	$\in \mathbb{R}^+$, "modulus of hardening"
σ_y	$\in \mathbb{R}^+$, yield stress
φ	dissipation functional
ϕ	yield function
$\dot{f}$	$= \frac{\partial f}{\partial t}$, time derivative of a function f
∇f	$= \left(\frac{\partial f_i}{\partial x_j}\right)_{i,j}$, gradient of a (vector) function f
Δf	$= \sum_i \frac{\partial^2 f_i}{\partial^2 x_i}$, Laplace of a vector function f
Df	Fréchet Derivative of function f
$\operatorname{div} A$	$= \left(\sum_j \frac{\partial a_{ij}}{\partial x_j}\right)_i$, divergence of a matrix A
$\operatorname{tr} A$	$= \sum_i a_{ii}$, trace of a matrix A
$\operatorname{dev} A$	$= A - \frac{\operatorname{tr} A}{\dim(A)} I$, deviator of a matrix A
$\\|A\\|_F$	$= \sqrt{\sum_{i,j} a_{ij}^2}$, Frobenius norm of a matrix A

Table 6: Used Symbols