Extending interior-point methods to non-linear second-order cone programming: application to finite-strain elastoplasticity

Chadi El Boustani, Jeremy Bleyer, Mathieu Arquier, Karam Sab

Extending interior-point methods to non-linear second-order cone programming: application to finite-strain elastoplasticity

Chadi El Boustani[a,b], Jeremy Bleyer[a,*], Mathieu Arquier[b], Karam Sab[a]

[a]*Laboratoire Navier, Ecole des Ponts ParisTech, Univ Gustave Eiffel, CNRS, 77455 Marne-la-Vallée, FRANCE*
[b]*Strains, 75012 Paris, FRANCE*

Abstract

Interior-point methods are well suited for solving convex non-smooth optimization problems which arise for instance in problems involving plasticity or contact conditions. This work attempts at extending their field of application to optimization problems involving either smooth but non-convex or non-smooth but convex objectives or constraints. A typical application for such kind of problems is finite-strain elastoplasticity which we address using a total Lagrangian formulation based on logarithmic strain measures. The proposed interior-point algorithm is implemented and tested on 3D examples involving plastic collapse and geometrical changes. Comparison with classical Newton-Raphson/return mapping methods shows that the interior-point method exhibits good computational performance, especially in terms of convergence robustness. Similarly to what is observed for convex small-strain plasticity, the interior-point method is able to converge for much larger load steps than classical methods.

Keywords: non-linear optimization, interior-point method, finite-strain elastoplasticity, logarithmic strain measure

1. Introduction

In the field of non-smooth mechanics, interior-point methods (IPM) have recently emerged as interesting alternative resolution strategies e.g. for problems involving contact [1–6], elastoplasticity [1, 7–9], limit analysis [10–14], granular materials [15, 16] or even viscoplastic fluids [17, 18]. They are indeed well suited to model optimization problems involving non-smooth constraints [19], in particular when they can be expressed using self-dual second-order Lorentz cone or positive semi-definite matrix cones, yielding, respectively, problems belonging to the class of Second-Order Cone Programming (SOCP) [20, 21] or Semi-Definite Programming (SDP). Initially, interior-point methods have been developed for solving Linear Programming (LP) problems [22], offering much better complexity

*Corresponding author
Email addresses: chadi.el-boustani@enpc.fr (Chadi El Boustani), jeremy.bleyer@enpc.fr (Jeremy Bleyer), mathieu.arquier@strains.fr (Mathieu Arquier), karam.sab@enpc.fr (Karam Sab)

than the simplex algorithm [23]. Interestingly, robustness and efficiency observed in the LP case can be transferred to the non-linear setting provided that the problem can be reformulated in the SOCP or SDP format [24–26]. Quite recently, extensions of the IPM towards problems involving non self-dual convex cones (e.g. the exponential and power cones) showed extremely promising results [27]. Off-the-shelf interior point solvers usually impose a standard input format so that mechanical problems must be reformulated into the required form [19]. This usually induces the introduction of additional auxiliary variables, thereby increasing the problem size [17]. However, such reformulations may not always be needed in a custom dedicated interior point solver [9, 18]. For instance, elastoplastic or viscoplastic problems contain a smooth term (related to the elastic or viscous part) and a non-smooth term (the plastic part). The latter can be treated using non-smooth SOCP constraints whereas the former can be kept as such in the objective function, its gradient and Hessian matrix entering the residual and tangent KKT system computation.

The present work aims at exploring one step further in the direction of extending IPM to other mechanical problems by tackling the case of finite-strain plasticity. Building upon previous works on a custom IPM solver including smooth convex terms [9, 18], we investigate the case of problems containing smooth but non-convex terms. Obviously, proofs of convergence of the IPM algorithm will necessarily be lost in the non-convex case. However, our heuristic reasoning is that we will restrict to a case in which difficulties will be decoupled. On the one hand, non-smoothness is present only in conic constraints which we still consider to be convex, while, on the other hand, non-convexity of some objective terms or constraints will concern only terms which we assume to be smooth. Since the IPM can be seen as a Newton method with continuation along the so-called central path, we hope that smooth terms will be properly handled by the Newton method and that continuation along the central path will sill handle properly the non-smooth but convex conic constraints. Our proposed algorithm is therefore a simple extension of a classical IPM to the previously mentioned non-convex case.

We apply the proposed framework to the specific case of logarithmic strain elastoplasticity [28]. As it will be discussed later, the use of the logarithmic strain setting enables a simple extension of classical small-strain elastoplastic constitutive laws to the finite-strain setting. In particular, the additive decomposition between elastic and plastic strain is preserved and elastic energy densities and plastic dissipation potentials are still convex with respect to the corresponding strains. Non-convexity only arises due to the non-linear relation between displacement and total strain. Using such a framework, we indeed obtain a problem in which non-smoothness (due to the plastic dissipation) can still be expressed using convex constraints whereas non-convexity involves smooth terms (strain/displacement relation). Benchmark 3D examples will validate our implementation. Comparison is made against standard Newton-Raphson methods based on a return mapping inner procedure enabling to assess the computational cost and convergence robustness of the IPM solver.

The main contributions of the present manuscript can be summarized as follows:

- classical IPM implementation can be easily adapted to the proposed non-convex

setting without much difficulty;

- logarithmic strain elastoplasticity perfectly fits into the considered framework;
- numerical results show that finite-strain logarithmic elastoplasticity can be efficiently solved using the proposed IPM solver;
- more precisely, although the total number of iterations per load steps is usually larger than Newton-type methods, the IPM exhibits much better robustness properties, enabling to converge for much larger load steps.

The manuscript is organized as follows: section 2 is devoted to presenting the considered optimization problem setting and the proposed IPM algorithm; section 3 details the logarithmic strain plasticity formulation, the corresponding variational principle and some practical implementation details; section 4 is dedicated to the validation on numerical examples.

2. Primal-dual interior point method for nonlinear optimisation problems with cone constraints

2.1. Nonlinear optimisation problems with second-order cone constraints

Let us consider the following constrained nonlinear optimization problem with second-order cone constraints:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & f(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{x} \in \mathcal{K} \end{array} \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $f(\boldsymbol{x})$ is a scalar-valued and $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^p$ a vector-valued function, both functions are assumed to be sufficiently smooth. Linear equality constraints are modelled using matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of rank $m \leq n$ and non-linear second-order cone constraints (SOC) are represented by $\mathcal{K}$, a Cartesian product of self-dual second-order cones i.e. $\mathcal{K} = \mathcal{K}_1 \times \ldots \times \mathcal{K}_q$, such as:

- the positive orthant:

$$\mathbb{R}^m_+ = \{\boldsymbol{z} \in \mathbb{R}^m \text{ s.t. } z_j \geq 0 \quad \forall j = 1, \ldots, m\} \tag{2.2}$$

- the Lorentz second-order cone:

$$\mathcal{Q}^{m+1} = \left\{\mathbf{z} = (z^0, \bar{\mathbf{z}}) \in \mathbb{R} \times \mathbb{R}^m \text{ s.t. } z^0 \geq \|\bar{\mathbf{z}}\|\right\} \tag{2.3}$$

- the rotated Lorentz second-order cone:

$$\mathcal{Q}_r^{m+2} = \left\{\mathbf{z} = (z^0, z^1, \tilde{\mathbf{z}}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \text{ s.t. } 2z^1 z^0 \geq \|\tilde{\mathbf{z}}\|^2\right\} \tag{2.4}$$

Note that standard second-order cone programming problems correspond to the particular case of (2.1) in which f is linear and $\boldsymbol{g}$ is absent (or linear, in such a case it can be appended to matrix $\boldsymbol{A}$).

One important feature of problem (2.1) is that non-linear constraints appear in two very different forms: general but smooth non-linear constraints via $\boldsymbol{g}$ and non-smooth but specific SOC constraints via $\mathcal{K}$. In particular, both types of constraints will be treated differently in the solution algorithm by extending classical primal-dual interior point methods (IPM) dedicated to tackling non-smooth conic constraints to the case of additional smooth non-linear constraints. Similarly, the objective function is assumed to be smooth. Modeling the presence of non-smooth terms such as $\|\boldsymbol{x}\|_2$ would require appropriate conic reformulation to treat it using SOC constraints.

2.2. Optimality conditions

Let $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{s})^T$ be the concatenation of all variables and the constraints Lagrange multipliers. The Lagrangian function of problem (2.1) is defined as follows:

$$\mathcal{L}(\boldsymbol{w}) = f(\boldsymbol{x}) + \boldsymbol{y}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{z}^T\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{s}^T\boldsymbol{x} \tag{2.5}$$

Then Karush-Kuhn-Tucker (KKT) first-order conditions for optimality of problem (2.1) are given by the following:

$$\boldsymbol{r}(\boldsymbol{w}) = \begin{pmatrix} \nabla_x\mathcal{L} \\ \nabla_y\mathcal{L} \\ \nabla_z\mathcal{L} \\ \boldsymbol{x}^T\boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\nabla}_x f + \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{G}^T\boldsymbol{z} - \boldsymbol{s} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{g}(\boldsymbol{x}) \\ \boldsymbol{x}^T\boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \quad \text{and } \boldsymbol{x} \in \mathcal{K},\ \boldsymbol{s} \in \mathcal{K}^* \tag{2.6}$$

where $\mathcal{K}^*$ is the dual cone and $\boldsymbol{G}(\boldsymbol{x}) \in \mathbb{R}^{p\times n}$ being the collection of the gradients of the nonlinear constraints:

$$\boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\nabla}_x g_1 \\ \vdots \\ \boldsymbol{\nabla}_x g_p \end{bmatrix} \tag{2.7}$$

The main idea of the IPM consists in finding a solution to the KKT conditions given by (2.6) by following the neighbourhood of a curve called the *central path* which consists of a sequence of iterates $\boldsymbol{w}(\eta) = (\boldsymbol{x}(\eta), \boldsymbol{y}(\eta), \boldsymbol{z}(\eta), \boldsymbol{s}(\eta))$ parameterized by a *barrier parameter* $\eta \geq 0$. These iterates are *interior points* i.e. they satisfy the conic constraints $\boldsymbol{x} \in \mathcal{K}$ and $\boldsymbol{s} \in \mathcal{K}^*$. The central path is defined as the solution to the perturbed KKT system or *barrier* KKT system (BKKT) defined as follows:

$$\boldsymbol{r}(\boldsymbol{w}, \eta) = \begin{pmatrix} \nabla_x\mathcal{L} \\ \nabla_y\mathcal{L} \\ \nabla_z\mathcal{L} \\ \boldsymbol{x}^T\boldsymbol{s} - \eta\boldsymbol{e} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\nabla}_x f + \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{G}^T\boldsymbol{z} - \boldsymbol{s} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{g}(\boldsymbol{x}) \\ \boldsymbol{x} \circ \boldsymbol{s} - \eta\boldsymbol{e} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \quad \text{and } \boldsymbol{x} \in \mathcal{K},\ \boldsymbol{s} \in \mathcal{K}^* \tag{2.8}$$

in which the operator $\circ$ and $\boldsymbol{e}$ depend on the types of the cones defining $\mathcal{K}$ (see for instance appendix A where $\boldsymbol{e} = (1, \boldsymbol{0})$ of dimension $m+1$ is defined for a second order cone $\mathcal{Q}^{m+1}$).

The introduction of the perturbed KKT system, the main idea of the IPM, is designed in order to tackle the complementarity constraint $\boldsymbol{x}^T\boldsymbol{s} = 0$. All other constraints, especially those involving $\boldsymbol{g}$, remain unchanged. As it will be discussed later, the proposed algorithm will therefore be similar to a Newton-method on the smooth non-linear residuals and a modified Newton-method on the conic residuals, exactly as in the standard SOCP case.

2.3. Solving the perturbed KKT system

As mentioned earlier, the aim of the IPM is to find a series of iterates $\boldsymbol{w}(\eta) = (\boldsymbol{x}(\eta), \boldsymbol{y}(\eta), \boldsymbol{z}(\eta), \boldsymbol{s}(\eta))$ while driving η to 0, yielding, at convergence, a solution to the original unperturbed KKT system (2.6). For that, a *primal-dual* method will be used in which both primal and dual variables remain unknown.

At each iteration (k), a Newton step on the perturbed KKT system (2.8) is computed towards the central path for a fixed value of $\eta^{(k)}$. The solution is updated after a step-length calculation and the barrier parameter is reduced by some amount. The process is repeated until the residuals fall under a certain tolerance. This will result in a series of iterates $\boldsymbol{w}^{(k)} = (\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{z}^{(k)}, \boldsymbol{s}^{(k)})$ which remain feasible with respect to the conic constraints.

2.3.1. The Newton system

Given an iteration (k) at which a point $\boldsymbol{w}^{(k)}$ satisfying the conic constraints is known and a value for the barrier parameter $\eta^{(k)}$ has been chosen, the next iterate is calculated by computing a new point near the central path. This is obtained by performing one iteration of the Newton method when linearising the residual equations such as:

$$\boldsymbol{r}^{(k+1)}(\boldsymbol{w}, \eta) = \boldsymbol{r}^{(k)}(\boldsymbol{w}) + \boldsymbol{r'}^{(k)}(\boldsymbol{w}, \eta).\boldsymbol{\Delta w} = 0 \tag{2.9}$$

thus solving the system $\boldsymbol{J}^{(k)}.\Delta\boldsymbol{w} = -\boldsymbol{r}^{(k)}(\boldsymbol{w})$ with the jacobian matrix $\boldsymbol{J}^{(k)} = \boldsymbol{r'}^{(k)}(\boldsymbol{w})$ and where $\Delta\boldsymbol{w} = (\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{z}, \Delta\boldsymbol{s})$ is a descent direction. More precisely, one has:

$$\boldsymbol{J}^{(k)}.\Delta\boldsymbol{w} = \begin{bmatrix} \boldsymbol{H}^{(k)} & \boldsymbol{A}^T & (\boldsymbol{G}^{(k)})^T & -\boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{G}^{(k)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{S}^{(k)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{X}^{(k)} \end{bmatrix} \begin{Bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{z} \\ \Delta\boldsymbol{s} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{r}_d^{(k)} \\ -\boldsymbol{r}_p^{(k)} \\ -\boldsymbol{r}_p^{\prime(k)} \\ -\boldsymbol{r}_c^{(k)} \end{Bmatrix} = -\boldsymbol{r}^{(k)}(\boldsymbol{w}, \eta) \tag{2.10}$$

in which $\boldsymbol{H}^{(k)}$ is the Hessian of the Lagrangian with respect to $\boldsymbol{x}$ given by:

$$\boldsymbol{H}^{(k)} = \boldsymbol{\nabla}^2_{xx}\mathcal{L}^{(k)} = \boldsymbol{\nabla}^2_{xx}f^{(k)} - \sum_{i=1}^{p} z_i^{(k)}\boldsymbol{\nabla}^2_{xx}g_i^{(k)} \tag{2.11}$$

and $\boldsymbol{X}^{(k)} = \boldsymbol{diag}(\ldots, \boldsymbol{mat}(\boldsymbol{x}_i^{(k)}), \ldots)$ and $\boldsymbol{S}^{(k)} = \boldsymbol{diag}(\ldots, \boldsymbol{mat}(\boldsymbol{s}_i^{(k)}), \ldots)$ and the vector $\boldsymbol{r}^{(k)}$ corresponding to the vector of residuals for the linearized equations of the KKT system:

$$\boldsymbol{r}_d^{(k)} = \boldsymbol{\nabla}_x f^{(k)} + \boldsymbol{A}^T \boldsymbol{y}^{(k)} + (\boldsymbol{G}^{(k)})^T \boldsymbol{z}^{(k)} - \boldsymbol{s}^{(k)} \tag{2.12a}$$

$$\boldsymbol{r}_p^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b} \tag{2.12b}$$

$$\boldsymbol{r'}_p^{(k)} = \boldsymbol{g}(\boldsymbol{x}^{(k)}) \tag{2.12c}$$

$$\boldsymbol{r}_c^{(k)} = \eta^{(k)}\boldsymbol{e} - \boldsymbol{X}^{(k)}\boldsymbol{S}^{(k)}\boldsymbol{e} \tag{2.12d}$$

2.3.2. Reduced linear system

Most IPM implementations further reduce the system size by eliminating the conic variables. A specific rescaling procedure on the primal-dual pair $(\boldsymbol{x}, \boldsymbol{s})$ is usually adopted to yield a symmetric reduced system and to improve the algorithm convergence. In the following, we use the Nesterov-Todd rescaling which has been described at length in [25, 29],.

The final obtained reduced system is of the following form:

$$\begin{bmatrix} \boldsymbol{H}^{(k)} + (\boldsymbol{F}^{(k)})^2 & \boldsymbol{A}^T & (\boldsymbol{G}^{(k)})^T \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{G}^{(k)} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{z} \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{r'}_d^{(k)} \\ -\boldsymbol{r}_p^{(k)} \\ -\boldsymbol{r'}_p^{(k)} \end{Bmatrix} \tag{2.13}$$

where:

$$\boldsymbol{r'}_d^{(k)} = \boldsymbol{\nabla}_x f^{(k)} + \boldsymbol{A}^T \boldsymbol{y}^{(k)} + (\boldsymbol{G}^{(k)})^T \boldsymbol{z}^{(k)} - \boldsymbol{s}^{(k)} - (\boldsymbol{F}\boldsymbol{V}^{-1})^{(k)} \boldsymbol{r}_c^{(k)} \tag{2.14}$$

2.4. Interior-point method implementation details

2.4.1. Step-length calculation

The next iterate $\boldsymbol{w}^{k+1}$ is obtained by performing a line-search on the maximum allowable step length α in direction $\boldsymbol{\Delta w}$, i.e. $\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \alpha \boldsymbol{\Delta w}$ with $\alpha \in [0, 1]$. This step should be chosen such that the iterate still satisfies the conic constraints. Practically, α is chosen such as $\alpha \leq \alpha_{max} < 1$ with $\alpha_{max} = 0.995$ typically, ensuring that the next iterate does not fall exactly on the feasible region boundary.

2.4.2. Choice of the barrier parameter

Following standard IPM implementations, the barrier parameter η^{k+1} is chosen using the average value of the complementarity gap $\mathrm{g}_i^{k+1} = (\boldsymbol{s}_i^{k+1})^T \cdot \boldsymbol{x}_i^{k+1}$ for each cone:

$$\eta^{k+1} = \gamma \left(\frac{1}{m} \sum_{i=1}^{m} \mathrm{g}_i^{k+1} \right) = \gamma \overline{\mathrm{g}}^{k+1} \tag{2.15}$$

where m is the total number of cones and $\gamma \in [0, 1]$ is a centering parameter.

2.4.3. Predictor-corrector scheme and adaptive choice of the centering parameter

Our implementation also adopts the predictor-corrector scheme proposed by Mehrotra [30] which greatly improves the IPM convergence. The predictor-corrector scheme amounts

to computing two different directions, taking advantage of having to factorize the KKT matrix $\boldsymbol{J}$ only once:

1. the first step, called *affine step*, corresponds to solving (2.8) with $\eta = 0$ (unperturbed KKT system), and its solution is denoted $\boldsymbol{\Delta w}^a$. The maximum step length for this step is noted α^a_{max}.
2. The centering parameter is then chosen using the following heuristic [31]:

$$\gamma = (1 - \alpha^a_{max})^3 \tag{2.16}$$

   therefore allowing full centering steps in order to quickly attain the central path if the affine step is small.
3. the final step direction $\boldsymbol{\Delta w}$ is then computed using the value of η^k calculated using the equation (2.15) and a reduced residual vector depending on γ, ensuring that residuals and complementarity gaps are reduced at the same rate.

Further details about the choice of the parameters and the reduced residual expressions can be found in [30, 32, 33].

2.5. The complete primal-dual interior point algorithm

Algorithm 1 Predictor-corrector primal-dual interior point algorithm

Initialization ($k = 0$): all variable are
while $\|\boldsymbol{r'}_d^{(k)}, \boldsymbol{r}_p^{(k)}, \boldsymbol{r'}_p^{(k)}\| \geq \epsilon$ and $\bar{\mathsf{g}}^{(k)} \geq \epsilon$ **do**
    compute scaling matrices $\boldsymbol{F}$ and $\boldsymbol{V}$ ▷ Appendix B
    form reduced Newton system $\boldsymbol{J}^{(k)}$ ▷ Section 2.3
    form affine residual vector $\boldsymbol{r}^a = \boldsymbol{r}^{(k)}(\boldsymbol{w}, \eta = 0)$ ▷ Section 2.4.3
    solve affine step $\boldsymbol{J}^{(k)}.\Delta\boldsymbol{w}^a = -\boldsymbol{r}^a$ and calculate the affine solution $\Delta\boldsymbol{w}^a$
    compute maximum affine step length α^a_{max} ▷ Section 2.4.1
    estimate centering parameter γ using equation (2.16)
    set new barrier parameter $\eta^{(k)} = \gamma\bar{\mathsf{g}}^{(k)}$ ▷ Section 2.4.2
    form corrector residual vector $\boldsymbol{r}_c^{(k)}$ using equation (2.12d) and reduced residuals
    solve corrector step $\boldsymbol{J}^{(k)}.\Delta\boldsymbol{w} = -\boldsymbol{r}^{(k)}$ by reusing the factorisation of $\boldsymbol{J}^{(k)}$
    compute maximum step length α ▷ Section 2.4.1
    update variables $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \alpha\Delta\boldsymbol{w}$
    compute new complementarity gap $\bar{\mathsf{g}}^{(k+1)}$
    if $k > N_{iter,max}$ or $\alpha \leq \epsilon$ **then**
        break ▷ Algorithm failed
    end if
    $k \leftarrow k + 1$
end while

3. Application to logarithmic strain elastoplastic problems

3.1. Kinematics

Let $\mathcal{B}^0$ be an elastoplastic body that occupies the space Ω^0 in the reference configuration. Its external boundary $\Gamma^0 := \partial\mathcal{B}^0$ is split into two distinct parts $\Gamma^0 = \Gamma^0_{\boldsymbol{u}} \cup \Gamma^0_{\boldsymbol{t}}$ and $\Gamma^0_{\boldsymbol{u}} \cap \Gamma^0_{\boldsymbol{t}} = \emptyset$ where:

- displacements $\boldsymbol{u}$ are imposed to $\boldsymbol{u}_d$ on $\Gamma^0_{\boldsymbol{u}}$;
- surface tractions are imposed to $\boldsymbol{t}_d$ on $\Gamma^0_{\boldsymbol{t}}$.

The deformation process maps every point $\boldsymbol{X} \in \Omega^0$ in the reference configuration to a point $\boldsymbol{x}_{(t)} = \boldsymbol{X} + \boldsymbol{u}_{(\boldsymbol{X},t)}$ in the equilibrium configuration. The deformation gradient is then given by $\boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{\nabla}_{\mathrm{X}}\, \boldsymbol{x} = \boldsymbol{I} + \boldsymbol{\nabla}_{\mathrm{X}}\, \boldsymbol{u}$ with $J = \det(\boldsymbol{F}) > 0$. The right polar decomposition of the deformation gradient $\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$ allows the definition of two fundamental tensors: the material stretch tensor $\boldsymbol{U}$ and the rotation tensor $\boldsymbol{R}$ with $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$.

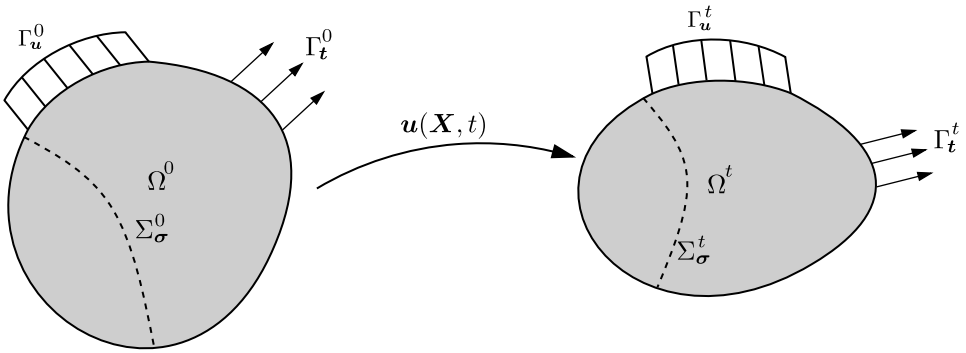


Figure 1: Reference model

3.2. The logarithmic strain measure and its work-conjugate stress

In this work, we will adopt the logarithmic strain framework proposed in [28] which has been shown to be well suited for describing finite-strain metal plasticity. In this framework, the chosen total strain measure is the Hencky logarithmic strain:

$$\boldsymbol{E} = \frac{1}{2}\ln(\boldsymbol{F}^T\boldsymbol{F}) = \ln(\boldsymbol{U}) \tag{3.1}$$

An attractive feature of using logarithmic strain measures is that classical small strain constitutive relations can be naturally extended to a finite-strain setting. In particular, the total (Hencky) strain can be split additively into many contributions (elastic, plastic, thermal, swelling, etc.). Its trace is also linked with the volume change $J = \exp(\mathrm{tr}(\boldsymbol{E}))$. Classical von Mises plasticity can therefore be used in the space of logarithmic strains, in particular the total strain will still be assumed to consist of the sum of an elastic and a plastic part:

$$\boldsymbol{E} = \boldsymbol{E}^e + \boldsymbol{E}^p \tag{3.2}$$

We will denote by $\boldsymbol{T}$ the stress measure associated with $\boldsymbol{E}$ with respect to the power density of internal forces i.e. $p_{\text{int}} = \boldsymbol{T} : \dot{\boldsymbol{E}}$. This pair of variables can be expressed to other classical stress/strain measures such as the Green-Lagrange strain measure $\boldsymbol{E}_{GL} = \frac{1}{2}(\boldsymbol{U}^2 - \boldsymbol{I})$ and its work-conjugate stress, the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$. One has for instance:

$$p_{\text{int}} = \boldsymbol{S} : \dot{\boldsymbol{E}}_{GL} = \boldsymbol{T} : \dot{\boldsymbol{E}} = \boldsymbol{T} : \left(\mathbb{M} : \dot{\boldsymbol{E}}_{GL}\right) \tag{3.3}$$

where

$$\mathbb{M} = \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{E}_{GL}} \tag{3.4}$$

is a fourth order geometrical tensor mapping both strain rate measures. We, therefore, also have:

$$\boldsymbol{S} = \boldsymbol{T} : \mathbb{M} \tag{3.5}$$

or equivalently:

$$\boldsymbol{T} = \boldsymbol{S} : \mathbb{M}^{-1} \tag{3.6}$$

Constitutive tangent stiffness operators can also be expressed using the two different stress/strain pairs and are related as follows:

$$\mathbb{D}_{GL} = \mathbb{M} : \mathbb{D} : \mathbb{M} + \boldsymbol{T} : \mathfrak{L} \tag{3.7}$$

where $\mathbb{D}_{GL}$ is the tangent operator in the Green-Lagrange setting i.e. such that $\dot{\boldsymbol{S}} = \mathbb{D}_{GL} : \dot{\boldsymbol{E}}_{GL}$, $\mathbb{D}$ is the tangent operator in the logarithmic setting i.e. such that $\dot{\boldsymbol{T}} = \mathbb{D} : \dot{\boldsymbol{E}}$ and finally $\mathfrak{L}$ is sixth-order mapping tensor defined as:

$$\mathfrak{L} = \frac{\partial^2 \boldsymbol{E}}{\partial \boldsymbol{E}_{GL} \partial \boldsymbol{E}_{GL}} \tag{3.8}$$

In fact both Green-Lagrange and Hencky strain measures belong to the more general family of Seth-Hill strain measures. General details concerning these strain measures and the geometric mappings existing between them, especially the general expression for $\mathbb{M}$ and $\mathfrak{L}$, can be found in appendix C and in [28, 34, 35]. Although our focus is mostly on the use of the Hencky strain measure, our implementation has been based on the Seth-Hill framework, allowing to change easily the chosen strain measure (keeping in mind that choosing another strain measure than the Hencky strain for describing finite-strain plasticity may not be appropriate).

3.3. The global incremental variational problem for elastoplastic media

We place ourselves in the framework of standard generalized materials [36, 37], namely we postulate the existence of a Helmholtz free energy function of the following form:

$$\psi(\boldsymbol{E}, \boldsymbol{E}^p, p) = \psi^e(\boldsymbol{E}^e) + \psi^p(p) \tag{3.9}$$

where ψ^e is the stored elastic energy density and $\psi^p(p)$ the hardening energy density with p being an internal state variable. In this decomposition, it is assumed ψ^p is convex and ψ^e

polyconvex [38]. Besides, we postulate the existence of a convex positively-homogeneous plastic dissipation potential $\phi(\dot{\boldsymbol{E}}^p)$.

We now focus on an incremental formulation over $[t_n, t_{n+1}]$ among the total time interval $[0, T]$, assuming that all mechanical fields are known at time t_n. Following [28], and in the same spirit as in previous works [6, 9], the unknown fields at time t_{n+1} can be obtained from the solution to the following incremental variational formulation [28]:

$$(\boldsymbol{u}_{n+1}, \boldsymbol{E}^p_{n+1}, p_{n+1}) = \underset{\boldsymbol{u}, \boldsymbol{E}^p, p}{\arg\min} \int_{\Omega^0} \int_{t_n}^{t_{n+1}} (\dot{\psi}(\boldsymbol{E}, \boldsymbol{E}^p, p) + \phi(\dot{\boldsymbol{E}}^p)) dt d\Omega - \int_{t_n}^{t_{n+1}} P_{ext}(\dot{\boldsymbol{u}}) dt \quad (3.10)$$

where P_{ext} is the power of external loads.

Restricting the above minimization to radial evolutions (see [9]) of $\dot{\boldsymbol{E}}^p(t)$ over the time interval, we obtain the following incremental minimization principle:

$$(\boldsymbol{u}_{n+1}, \boldsymbol{E}^p_{n+1}, p_{n+1}) = \underset{\boldsymbol{u}, \boldsymbol{E}^p, p}{\arg\min} \int_{\Omega^0} \Psi_n^{n+1}(\boldsymbol{E}, \boldsymbol{E}^p, p) d\Omega - P_{ext}(\boldsymbol{u}_{n+1} - \boldsymbol{u}_n) \quad (3.11)$$

in which we assumed that external forces are constant and where the incremental pseudo-energy density is given by:

$$\begin{aligned} \Psi_n^{n+1}(\boldsymbol{E}, \boldsymbol{E}^p, p) = \psi^e(\boldsymbol{E}, \boldsymbol{E}^p) + \psi^p(p) - \psi^e(\boldsymbol{E}_n, \boldsymbol{E}^p_n) - \psi^p(p_n) \\ + \phi(\boldsymbol{E}^p - \boldsymbol{E}^p_n) \end{aligned} \quad (3.12)$$

3.4. The case of von Mises plastic with linear isotropic hardening

We now particularize the problem to von Mises plasticity with linear isotropic hardening. In this case:

$$\psi(\boldsymbol{E}, \boldsymbol{E}^p, p) = \psi^e(\boldsymbol{E}, \boldsymbol{E}^p) + \psi^p(p) = \frac{1}{2}(\boldsymbol{E} - \boldsymbol{E}^p) : \mathbb{D} : (\boldsymbol{E} - \boldsymbol{E}^p) + \frac{1}{2} E_h p^2 \quad (3.13)$$

where $\mathbb{D}$ is the elastic modulus tensor of small-strain isotropic linear elasticity:

$$\mathbb{D} = \lambda \boldsymbol{I} \otimes \boldsymbol{I} + 2\mu \mathbb{I} \quad (3.14)$$

E_h is the hardening modulus and $p(t) = \int_0^t \sqrt{\frac{2}{3}} \|\dot{\boldsymbol{E}}^p\| dt$ is the accumulated plastic strain where $\|\boldsymbol{a}\| = \sqrt{a_{ij} a_{ij}}$.

The plastic dissipation potential is here:

$$\phi(\dot{\boldsymbol{E}}^p) = \begin{cases} \sqrt{\frac{2}{3}} \sigma_0 \|\dot{\boldsymbol{E}}^p\| & \text{if } \operatorname{tr}(\dot{\boldsymbol{E}}^p) = 0 \\ +\infty & \text{otherwise} \end{cases} \quad (3.15)$$

Introducing the notation $\bar{\star} = \star - \star_n$ to denote the variable increment over the current

time step, (3.12) becomes:

$$\Psi_n^{n+1}(\boldsymbol{E},\boldsymbol{E}^p,p) = \frac{1}{2}(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p):\mathbb{D}:(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p) + \boldsymbol{T}_n:(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p) + \frac{1}{2}E_h(\bar{p})^2 + E_h p_n \bar{p} + \sqrt{\frac{2}{3}}\sigma_0\|\bar{\boldsymbol{E}}^p\| \tag{3.16}$$

with $\mathrm{tr}(\bar{\boldsymbol{E}}^p) = \mathrm{tr}(\boldsymbol{E}^p) = 0$.

Since $\dot{p} = \sqrt{\frac{2}{3}}\|\dot{\boldsymbol{E}}^p\|$, we also have $\bar{p} = \sqrt{\frac{2}{3}}\|\bar{\boldsymbol{E}}^p\|$ therefore

$$E_h p_n \bar{p} + \sqrt{\frac{2}{3}}\sigma_0\|\bar{\boldsymbol{E}}^p\| = \sqrt{\frac{2}{3}}\sigma_{Y,n}\|\bar{\boldsymbol{E}}^p\|$$

with $\sigma_{Y,n} = \sigma_0 + E_h.p_n$ being the yield stress at the previous time step. The minimization problem (3.11) therefore reads as:

$$\min_{\boldsymbol{u},\bar{\boldsymbol{E}}^p,p} \int_{\Omega^0} \left(\frac{1}{2}(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p):\mathbb{D}:(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p) + \boldsymbol{T}_n:(\bar{\boldsymbol{E}}-\bar{\boldsymbol{E}}^p) + \frac{1}{2}E_h(\bar{p})^2 + \sqrt{\frac{2}{3}}\sigma_{y,n}\|\bar{\boldsymbol{E}}^p\| \right) d\Omega - P_{ext}(\bar{\boldsymbol{u}}) \tag{3.17}$$

under the following constraints:

$$\bar{\boldsymbol{E}} + \boldsymbol{E}_n = \frac{1}{2}\ln\left(\boldsymbol{F}^T\cdot\boldsymbol{F}\right) \tag{3.18a}$$

$$\boldsymbol{F} = \boldsymbol{F}_n + \nabla_X \bar{\boldsymbol{u}} \tag{3.18b}$$

$$\bar{\boldsymbol{u}} + \boldsymbol{u}_n = \boldsymbol{u}_{d,n+1} \quad \text{on } \Gamma_{\boldsymbol{u}}^0 \tag{3.18c}$$

$$\mathrm{tr}(\bar{\boldsymbol{E}}^p) = 0 \tag{3.18d}$$

Following [8], the traceless constraint $\mathrm{tr}(\bar{\boldsymbol{E}}^p) = 0$ can be removed by introducing directly in the elastic energy every occurrences of $\bar{\boldsymbol{E}}^p$ by $\mathrm{dev}(\bar{\boldsymbol{E}}^p) = \mathbb{K}:\bar{\boldsymbol{E}}^p$ where $\mathbb{K}$ is the projector over deviatoric tensors.

Moreover, in the objective function of (3.17), the only non-smooth term is $\sqrt{\frac{2}{3}}\sigma_{y,n}\|\bar{\boldsymbol{E}}^p\|$ which can be transformed into a second-order cone constraint by introducing an additional variable $\bar{\gamma}$ such that $\|\bar{\boldsymbol{E}}^p\| \leq \bar{\gamma}$. Replacing also the quadratic term $\frac{1}{2}E_h(\bar{p})^2$ with this new variable, one finally obtains:

$$\underset{\bar{\boldsymbol{u}},\bar{\boldsymbol{E}}^p,\bar{\gamma}}{\text{minimize}} \quad \int_{\Omega^0} \left[\frac{1}{2}(\bar{\boldsymbol{E}} - \mathbb{K} : \bar{\boldsymbol{E}}^p) : \mathbb{D} : (\bar{\boldsymbol{E}} - \mathbb{K} : \bar{\boldsymbol{E}}^p) + \boldsymbol{T}_n : (\bar{\boldsymbol{E}} - \mathbb{K} : \bar{\boldsymbol{E}}^p) \right. \tag{3.19a}$$

$$\left. + \frac{1}{3}E_h(\bar{\gamma})^2 + \sqrt{\frac{2}{3}}\sigma_{Y,n}\bar{\gamma} \right] d\Omega - P_{ext}(\bar{\boldsymbol{u}})$$

$$\text{subject to} \quad \bar{\boldsymbol{E}} + \boldsymbol{E}_n = \frac{1}{2}\ln\left(\boldsymbol{F}^T \cdot \boldsymbol{F}\right) \qquad \text{in } \Omega^0 \tag{3.19b}$$

$$\boldsymbol{F} = \boldsymbol{F}_n + \nabla_X \bar{\boldsymbol{u}} \qquad \text{in } \Omega^0 \tag{3.19c}$$

$$\bar{\boldsymbol{u}} + \boldsymbol{u}_n = \boldsymbol{u}_{d,n+1} \qquad \text{on } \Gamma^0_{\boldsymbol{u}} \tag{3.19d}$$

$$\|\bar{\boldsymbol{E}}^p\| \leq \bar{\gamma} \qquad \text{in } \Omega^0 \tag{3.19e}$$

which fits into the non-linear second-order cone programming format of (2.1) where the last constraint is expressed as a second-order Lorentz cone constraint $(\bar{\gamma}, \bar{\boldsymbol{E}}^p) \in \mathcal{Q}^7$. Let us finally remark that constraints (3.19b) and (3.19c) can be eliminated and that one can replace $\bar{\boldsymbol{E}}$ by its non-linear expression as a function $\bar{\boldsymbol{u}}$:

$$\bar{\boldsymbol{E}} = \bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}) = \frac{1}{2}\ln\left((\boldsymbol{F}_n + \nabla_X \bar{\boldsymbol{u}})^T \cdot (\boldsymbol{F}_n + \nabla_X \bar{\boldsymbol{u}})\right) - \boldsymbol{E}_n \tag{3.20}$$

so that the final problem expressed in the format of (2.1) involves the unknowns $\boldsymbol{x} = (\bar{\boldsymbol{u}}, \bar{\boldsymbol{E}}^p, \bar{\gamma})$, linear constraints associated with the kinematic boundary condition (3.19d) later expressed as $\boldsymbol{A}\bar{\boldsymbol{u}} = \boldsymbol{b}$, no non-linear constraints $\boldsymbol{g}(\boldsymbol{x})$ and the non-linear smooth objective function:

$$f(\boldsymbol{x}) = \frac{1}{2}(\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}) - \mathbb{K} : \bar{\boldsymbol{E}}^p) : \mathbb{D} : (\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}) - \mathbb{K} : \bar{\boldsymbol{E}}^p) + \boldsymbol{T}_n : (\bar{\boldsymbol{E}}(\bar{\boldsymbol{u}}) - \mathbb{K} : \bar{\boldsymbol{E}}^p) \tag{3.21}$$

$$+ \frac{1}{3}E_h(\bar{\gamma})^2 + \sqrt{\frac{2}{3}}\sigma_{Y,n}\bar{\gamma} - P_{ext}(\bar{\boldsymbol{u}})$$

3.5. Residuals and KKT system

Let us now detail the expression of the residuals and the associated KKT around a current iterate k. We will denote by $\star^{(k)}$ the value of a quantity $\star$ at this iterate e.g. $\boldsymbol{F}^{(k)} = \boldsymbol{F}_n + \boldsymbol{\nabla}_{\mathrm{X}}\, \bar{\boldsymbol{u}}^{(k)}$. First, the variation of the total logarithmic strain in direction $\delta\bar{\boldsymbol{u}}$ is given by:

$$\delta\bar{\boldsymbol{E}} = \mathbb{M}^{(k)} : \delta\bar{\boldsymbol{E}}_{GL} = \mathbb{M}^{(k)} : \mathrm{sym}((\boldsymbol{F}^{(k)})^T \cdot \boldsymbol{\nabla}_{\mathrm{X}}\, \delta\bar{\boldsymbol{u}}) \tag{3.22}$$

The variation of the objective function $f(\boldsymbol{x})$ in direction $\delta\boldsymbol{x}$ is given by:

$$(\nabla_{\boldsymbol{x}} f, \delta\boldsymbol{x}) = (\nabla_{\bar{\boldsymbol{u}}} f, \delta\bar{\boldsymbol{u}}) + (\nabla_{\bar{\boldsymbol{E}}^p} f, \delta\bar{\boldsymbol{E}}^p) + (\nabla_{\bar{\gamma}} f, \delta\bar{\gamma}) \tag{3.23}$$

where:

$$
\begin{aligned}
(\nabla_{\bar{\boldsymbol{u}}} f, \delta\bar{\boldsymbol{u}}) &= \int_{\Omega^0} \left(\mathbb{D} : (\bar{\boldsymbol{E}}^{(k)} - \mathbb{K} : \bar{\boldsymbol{E}}^{p,(k)}) + \boldsymbol{T}_n\right) : \delta\bar{\boldsymbol{E}} d\Omega - P_{ext}(\delta\bar{\boldsymbol{u}}) \\
&= \int_{\Omega^0} \boldsymbol{T}^{(k)} : \mathbb{M}^{(k)} : \mathrm{sym}((\boldsymbol{F}^{(k)})^T \cdot \boldsymbol{\nabla}_{\mathrm{X}}\, \delta\bar{\boldsymbol{u}}) d\Omega - P_{ext}(\delta\bar{\boldsymbol{u}}) \qquad (3.24) \\
(\nabla_{\bar{\boldsymbol{E}}^p} f, \delta\bar{\boldsymbol{E}}^p) &= \int_{\Omega^0} \left(-\mathbb{D} : (\bar{\boldsymbol{E}}^{(k)} - \mathbb{K} : \bar{\boldsymbol{E}}^{p,(k)}) - \boldsymbol{T}_n\right) : \mathbb{K} : \delta\bar{\boldsymbol{E}}^p d\Omega \\
&= \int_{\Omega^0} -\mathrm{dev}(\boldsymbol{T}^{(k)}) : \delta\bar{\boldsymbol{E}}^p d\Omega \qquad (3.25) \\
(\nabla_{\bar{\gamma}} f, \delta\bar{\gamma}) &= \int_{\Omega^0} \left(\frac{2}{3} E_h \bar{\gamma}^{(k)} + \sqrt{\frac{2}{3}}\sigma_{Y,n}\right) \delta\bar{\gamma} d\Omega \qquad (3.26)
\end{aligned}
$$

where we introduced $\boldsymbol{T}^{(k)}$ the value of the stress tensor at iterate k given by $\boldsymbol{T}^{(k)} = \boldsymbol{T}_n + \mathbb{D} : (\bar{\boldsymbol{E}}^{(k)} - \mathbb{K} : \bar{\boldsymbol{E}}^{p,(k)})$.

The global conic constraints $\boldsymbol{x} \in \mathcal{K}$ correspond here to $\bar{\boldsymbol{u}} \in \mathbb{R}^n$ (free variable) and $(\bar{\gamma}, \bar{\boldsymbol{E}}^p) \in \mathcal{Q}^7$ so that the dual conic variable $\boldsymbol{s} = (\boldsymbol{s}_{\bar{\boldsymbol{u}}}, \boldsymbol{s}_{\bar{\boldsymbol{E}}^p}, \boldsymbol{s}_{\bar{\gamma}}) \in \mathcal{K}$ are such that $\boldsymbol{s}_{\bar{\boldsymbol{u}}} = 0$ and $(\boldsymbol{s}_{\bar{\gamma}}, \boldsymbol{s}_{\bar{\boldsymbol{E}}^p}) \in \mathcal{Q}^7$. As a result, one obtain the following expressions for the first two residuals of (2.6):

$$
\boldsymbol{r}_d^{(k)} = \begin{Bmatrix} \boldsymbol{r}_{d,\bar{\boldsymbol{u}}}^{(k)} \\ \boldsymbol{r}_{d,\bar{\boldsymbol{E}}^p}^{(k)} \\ \boldsymbol{r}_{d,\bar{\gamma}}^{(k)} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\nabla}_{\bar{\boldsymbol{u}}} f^{(k)} + \boldsymbol{A}^T \boldsymbol{y}^{(k)} - \boldsymbol{s}_{\bar{\boldsymbol{u}}}^{(k)} \\ \boldsymbol{\nabla}_{\bar{\boldsymbol{E}}^p} f^{(k)} - \boldsymbol{s}_{\bar{\boldsymbol{E}}^p}^{(k)} \\ \boldsymbol{\nabla}_{\bar{\gamma}} f^{(k)} - \boldsymbol{s}_{\bar{\gamma}}^{(k)} \end{Bmatrix} \qquad (3.27)
$$

$$
\boldsymbol{r}_p^{(k)} = \boldsymbol{A}\bar{\boldsymbol{u}}^{(k)} - \boldsymbol{b} \qquad (3.28)
$$

Expressing that the residuals should be zero for the solution, the three blocks $\boldsymbol{r}_d^{(k)}$ respectively yield:

$$
\int_{\Omega^0} \boldsymbol{T}^{(k)} : \mathbb{M}^{(k)} : \mathrm{sym}((\boldsymbol{F}^{(k)})^T \cdot \boldsymbol{\nabla}_{\mathrm{X}}\, \delta\bar{\boldsymbol{u}}) d\Omega - P_{ext}(\delta\bar{\boldsymbol{u}}) + \int_{\Gamma_{\boldsymbol{u}}^0} \boldsymbol{y}^{(k)} \cdot \delta\bar{\boldsymbol{u}} dS = 0 \quad \forall \delta\bar{\boldsymbol{u}} \qquad (3.29)
$$

$$
\boldsymbol{s}_{\bar{\boldsymbol{E}}^p}^{(k)} = -\mathrm{dev}(\boldsymbol{T}^{(k)}) \qquad (3.30)
$$

$$
\boldsymbol{s}_{\bar{\gamma}}^{(k)} = \sqrt{\frac{2}{3}}\sigma_{Y,n} + \frac{2}{3} E_h \bar{\gamma}^{(k)} \qquad (3.31)
$$

The first block expresses the virtual work principle whereas the last two, combined with the conic constraint $\|\boldsymbol{s}_{\bar{\boldsymbol{E}}^p}\| \leq \boldsymbol{s}_{\bar{\gamma}}$ express the plastic yield criterion:

$$
\sqrt{\frac{3}{2}} \|\mathrm{dev}(\boldsymbol{T}^{(k)})\| \leq \sigma_{Y,n} + \sqrt{\frac{2}{3}} E_h \bar{\gamma}^{(k)} \qquad (3.32)
$$

One can also easily check that, combined with the complementarity condition $\boldsymbol{x}^T\boldsymbol{s} = 0$ of the Lorentz cone (A.6), these relations are equivalent to the plastic consistency condition and plastic flow rule:

$$\bar{\gamma}^{(k)}\left(\left(\sigma_{Y,n} + \sqrt{\frac{2}{3}}E_h\bar{\gamma}^{(k)}\right)^2 - \frac{3}{2}\|\mathrm{dev}(\boldsymbol{T}^{(k)})\|^2\right) = 0 \tag{3.33}$$

$$\bar{\boldsymbol{E}}^{p,(k)} = \frac{\sqrt{\frac{3}{2}}\bar{\gamma}^{(k)}}{\sigma_{Y,n} + \sqrt{\frac{2}{3}}E_h\bar{\gamma}^{(k)}}\,\mathrm{dev}(\boldsymbol{T}) \tag{3.34}$$

where it is clear that $\bar{\gamma}^{(k)} = \|\bar{\boldsymbol{E}}^{p,(k)}\| = \sqrt{\frac{2}{3}}\bar{p}^{(k)}$ during a plastic evolution.

Finally, the jacobian system (2.10) reads in this case as:

$$\boldsymbol{J}^{(k)} = \begin{bmatrix} \boldsymbol{H}^{(k)} & \boldsymbol{A}^T & -\boldsymbol{I} \\ \boldsymbol{A} & 0 & 0 \\ \boldsymbol{S}^{(k)} & 0 & \boldsymbol{X}^{(k)} \end{bmatrix} \tag{3.35}$$

in which the Hessian $\boldsymbol{H}^{(k)}$ is obtained using the chain rule resulting in a sum of five different contributions, the first four corresponding to the material contribution to the stiffness matrix and the fifth being the geometrical stiffness matrix:

$$\begin{aligned}
\Delta\bar{\boldsymbol{x}}\cdot\boldsymbol{H}^{(k)}\cdot\delta\bar{\boldsymbol{x}} = &\int_{\Omega^0}\Big[\mathrm{sym}((\boldsymbol{F}^{(k)})^T\cdot\boldsymbol{\nabla}_{\mathrm{X}}\,\Delta\bar{\boldsymbol{u}}) : \mathbb{D}^{(k)}_{GL} : \mathrm{sym}((\boldsymbol{F}^{(k)})^T\cdot\boldsymbol{\nabla}_{\mathrm{X}}\,\delta\bar{\boldsymbol{u}})\Big]d\Omega \\
&-\int_{\Omega^0}\Big[\Delta\bar{\boldsymbol{E}}^p : (\mathbb{K} : \mathbb{D} : \mathbb{M}^{(k)}) : \mathrm{sym}((\boldsymbol{F}^{(k)})^T\cdot\boldsymbol{\nabla}_{\mathrm{X}}\,\delta\bar{\boldsymbol{u}})\Big]d\Omega \\
&-\int_{\Omega^0}\Big[\mathrm{sym}((\boldsymbol{F}^{(k)})^T\cdot\boldsymbol{\nabla}_{\mathrm{X}}\,\Delta\bar{\boldsymbol{u}}) : (\mathbb{M}^{(k)} : \mathbb{D} : \mathbb{K}) : \delta\bar{\boldsymbol{E}}^p\Big]d\Omega \\
&+\int_{\Omega^0}\Big[\Delta\bar{\boldsymbol{E}}^p : (\mathbb{K} : \mathbb{D} : \mathbb{K}) : \delta\bar{\boldsymbol{E}}^p\Big]d\Omega \\
&+\int_{\Omega^0}\Big[\boldsymbol{T}^{(k)}_n : \mathbb{M}^{(k)}_n : ((\boldsymbol{\nabla}_{\mathrm{X}}\,\delta\bar{\boldsymbol{u}})^T\cdot\boldsymbol{\nabla}_{\mathrm{X}}\,\Delta\bar{\boldsymbol{u}})\Big]d\Omega
\end{aligned} \tag{3.36}$$

with the tangent Green-Lagrange stiffness $\mathbb{D}^{(k)}_{GL}$ being given by (3.7):

$$\mathbb{D}^{(k)}_{GL} = \mathbb{M}^{(k)} : \mathbb{D} : \mathbb{M}^{(k)} + \boldsymbol{T}^{(k)} : \mathfrak{L}^{(k)} \tag{3.37}$$

3.6. Finite-element implementation

The finite-element discretization of (3.19) is quite standard and very close to the kinematic approach discussed in [9]. In particular, we used 10-noded quadratic tetrahedra for the displacement field interpolation. The logarithmic plastic strain unknowns correspond to its six components expressed at all quadrature points (4 Gauss points per tetrahedron in the present case) and similarly to the additional scalar variable $\bar{\gamma}$. As discussed in [9], the equations involving $\bar{\boldsymbol{E}}^p$ and $\bar{\gamma}$ in the KKT system are all of local nature i.e. they are expressed at the quadrature point level and are all uncoupled. They can therefore be easily condensed when forming system (B.4) yielding a further reduced system involving only the displacement variables and the boundary condition Lagrange multipliers as final unknowns. As a consequence, the proposed primal-dual IPM exhibits the same computational cost as regards linear system resolutions as a standard Newton-Raphson method.

3.7. Initialization points of the IPM

Because of the incremental aspect of the mechanical problem, once the solution of the incremental problem for a specific load step, say, t_n, is obtained, loading conditions as well as state variables, e.g. $\sigma_{Y,n}$ are updated before solving the new incremental problem for the next step t_{n+1}. It is usual that the change in data of these problems is small so that solutions are often close from each other between consecutive time steps. Exploiting a good initial guess of the solution, also known as *warm-start*, is a current challenge for interior point methods due to the fact that conic variables should initially be feasible and ideally far from the feasible region boundary.

In our implementation, all variables are initialized with the converged values of the previous load step, except the conic variables $(\boldsymbol{x}, \boldsymbol{s})$. Indeed, in order to have a starting point located far from the feasible region boundary, the initial value for the next load step is taken as follows:

$$\left.\begin{array}{l}(x_0, \bar{\mathbf{x}})_{n+1}^{(0)} = (x_0, \beta\bar{\mathbf{x}})_n \\ (s_0, \bar{\mathbf{s}})_{n+1}^{(0)} = (s_0, \beta\bar{\mathbf{s}})_n\end{array}\right\} \quad \text{with } \beta = 0.7 \text{ typically} \tag{3.38}$$

4. Illustrative applications

4.1. Membrane effect in a fully-clamped elastoplastic beam

In this example, we consider a fully clamped rectangular beam of length $L = 2.0\,\mathrm{m}$ oriented in direction x and of height $h = 0.1\,\mathrm{m}$ and width $b = 0.04\,\mathrm{m}$. The beam consists of a von Mises plastic material with no hardening and is subject to a uniformly distributed body force $\boldsymbol{b} = -f\boldsymbol{e}_z$. The material parameter values are summarized in table 1.

The present implementation results have been compared with computations using the commercial finite-element software Abaqus [39] and also using the open-source finite-element platform FEniCS [40–42] coupled with MFront [43] for the constitutive behaviour integration.

The FEniCS/MFront implementation consists in a total-Lagrangian implementation of logarithmic plasticity using a standard Newton-Raphson/return mapping procedure. The

Table 1: Beam material properties

Material parameter	
Young modulus	$E = 210\,\text{GPa}$
Poisson's ratio	$\nu = 0.30$
Initial yield stress	$\sigma_0 = 250\,\text{MPa}$
Hardening modulus	$h = 0.00\,\text{MPa}$

coupling between both libraries relies on the `MFrontGenericInterfaceSupport` project [43].

The Abaqus implementation relies on an updated-Lagrangian formulation using the Cauchy stress tensor and its work-conjugate rate of deformation. The integration technique for the total deformation gives the logarithmic strains (`LE` in Abaqus notation) which is used in the case of metal plasticity. For more details, one can refer to sections 1.4 and 1.5 in the Abaqus theory manual [39].

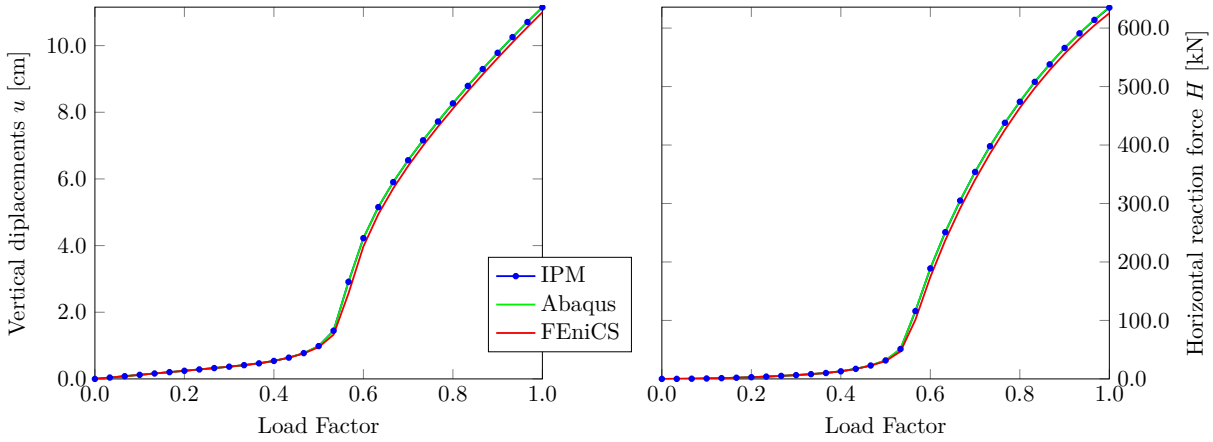
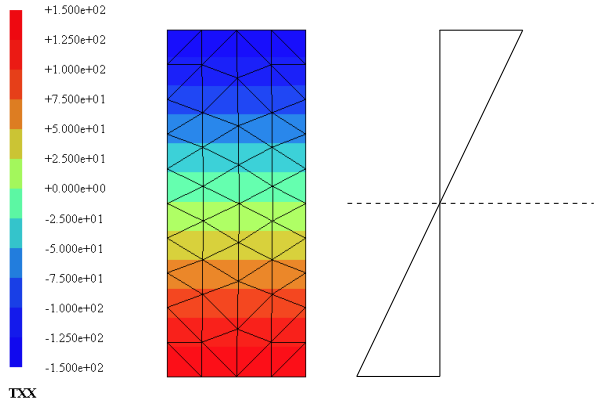
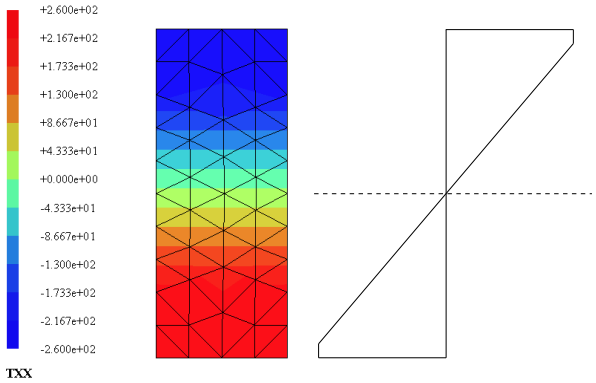


Figure 2: Evolution of the mid-span deflection u and the horizontal support reaction H

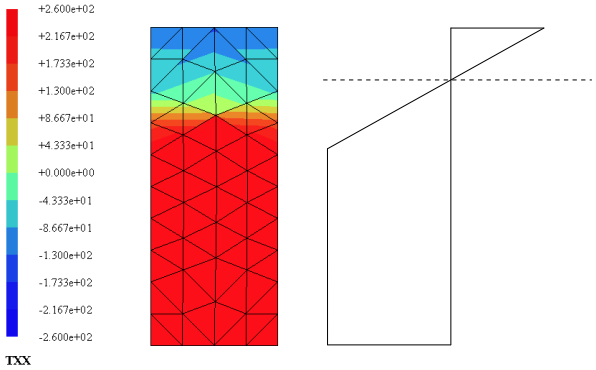
We monitor the evolution of the mid-span deflection u and the horizontal support reaction H when increasing the body force up to $f = 50\,\text{MN/m}^3$. The evolutions of u and H have been represented in Figure 2. It can first be observed that all three different implementations yield very similar results, the slight difference observed with respect to the FEniCS/MFront computations can be attributed to the fact that a different mesh, although of similar element size, was used. The obtained results clearly exhibit a first elastic then plastic stage (for a load factor below 0.5) when geometrical non-linear effects do not play an important role. A secondary stiffening stage (for load factor larger than 0.5) is then observed due to membrane catenary effect (see the increase of the horizontal reaction force) when geometrical non-linear effects become more and more important. This is further confirmed when inspecting the normal stress diagram along the mid-span cross-section (Figure 3) showing an elastic stage solution (Fig. 3a), the onset of cross-section



(a) Load factor = 0.33



(b) Load factor = 0.50



(c) Load factor = 1.0

Figure 3: Generalised normal stress component T_{XX} in the middle-section for different load factors

yielding in pure bending (Fig. 3b) and finally a membrane-dominated plastic stage (Fig. 3c). The final deformed configuration has also been represented in Figure 4.

In order to assess the numerical solution procedure, we will compare the FEniCS solution with the proposed IPM solution since both approaches rely on a total-Lagrangian formulation. In particular, we compared the number of iterations per load step to reach convergence using the same relative residual tolerance. It must be recalled that, apart from the way boundary conditions are handled, the linear system size, and hence the cost per iteration, is similar for both methods.

Results are reported in figure 5 where it can be observed that the required number of iterations is much larger for the IPM than for the Newton method used in FEniCS for 30 load steps. This is by no means surprising due to the quadratic convergence of the Newton method close to a solution. It can also be observed that the required number of iterations increases in the second stage of the problem where plasticity and geometrically non-linear effects become much more dominant.

However, an extremely interesting feature of the IPM is its robustness over large load steps. Indeed, the Newton method was unable to converge with less than 20 load steps

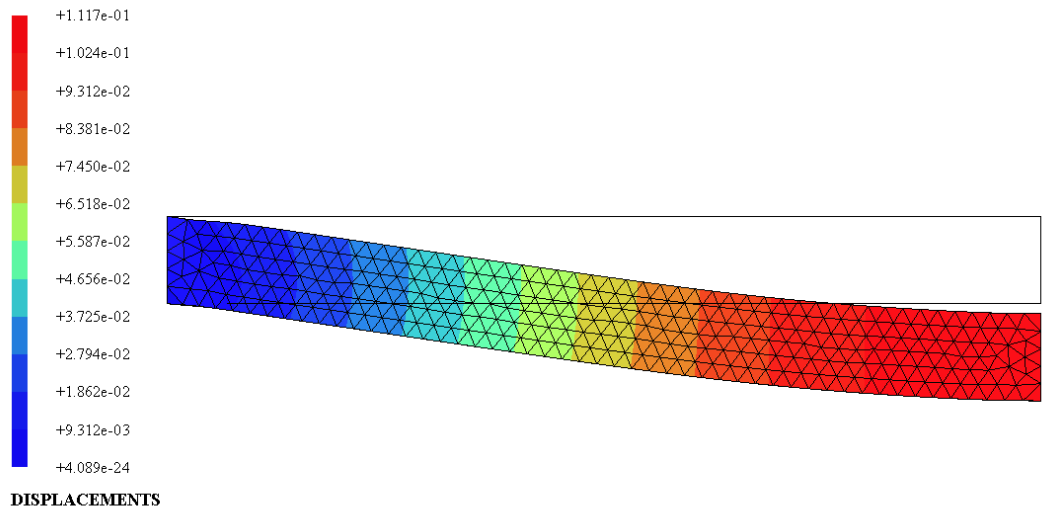


Figure 4: Deformed configuration and displacement isovalues in m (Load factor = 1.0)

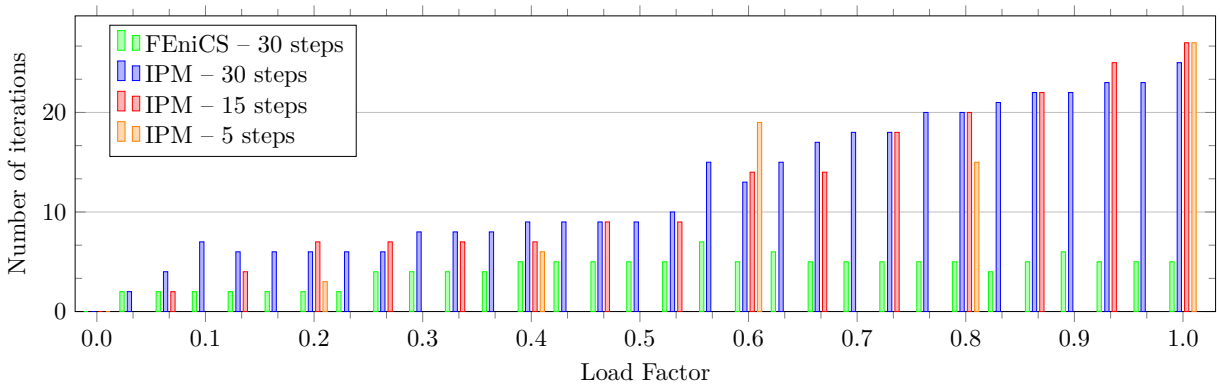


Figure 5: Number of iterations per load step

whereas the IPM method could converge using only 5 load steps. Besides, this robustness does not seem to deteriorate the convergence quality since roughly the same number of iterations is required for the same load level when using smaller load steps. Overall, the total number of iterations (table 2) using 5 load steps becomes competitive compared with the Newton method, whereas the Newton method is more efficient than the IPM with similar load-stepping. Moreover, figure 6 clearly shows that using fewer load steps yields similar values for the displacement and reaction forces.

4.2. Necking of a rod

We consider the rod-necking problem, a standard benchmark problem of finite plasticity that have been used by various authors [28, 44, 45]. The goal of this example is to compare the results given by the IPM algorithm with reference results of [28] and using an Abaqus implementation. The initial length of the rod is $l = 53.34\,\text{mm}$, the radius $r_0 = 6.4135\,\text{mm}$. The necking is triggered by an initial imperfection of the rod in the

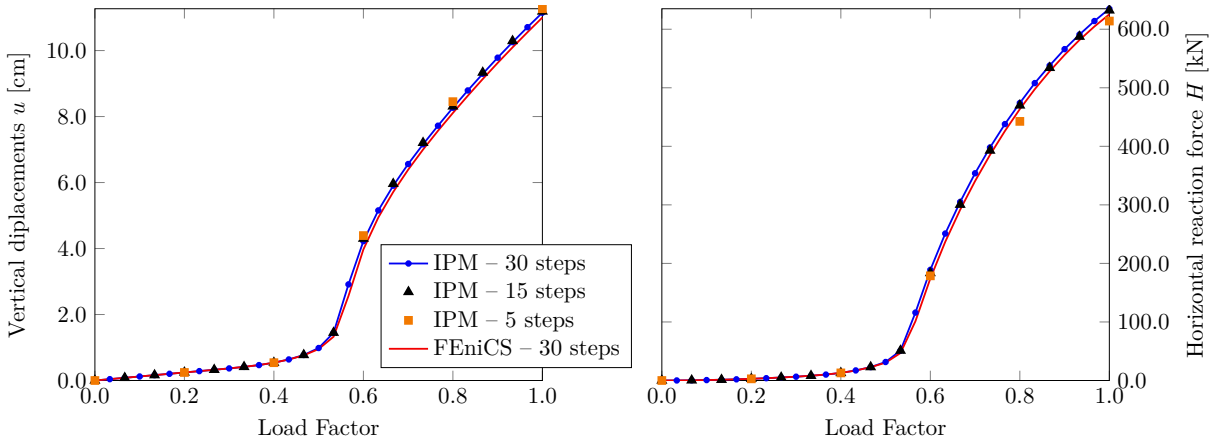


Figure 6: Comparison of the evolution of the mid-span deflection u and the horizontal support reaction H using 5 load steps

Table 2: Total number of iterations for the different methods and load-stepping

Method	Total iterations
FEniCS (30 steps)	128
IPM (30 steps)	385
IPM (15 steps)	192
IPM (5 steps)	70

middle section represented by a continuous decrease of the radius to $r = 0.982r_0$ over a length of 8.98 mm.

The constitutive response of the material is characterized by the logarithmic strain plasticity formulation using a von Mises yield criterion and isotropic hardening. The following saturation-type non-linear isotropic hardening is considered:

$$\sigma_Y(p) = hp + (\sigma_\infty - \sigma_0)(1 - \exp[-\omega p]) \tag{4.1}$$

As a first approach, this non-linear hardening model has been represented in our linear hardening model implementation by a piecewise-linear model. Namely, at each time step, the new value of the elastic yield limit is calculated, and the linear hardening modulus E_h is set to:

$$E_h = \left.\frac{d\sigma_Y(p)}{dp}\right|_{p=p_n} = h + (\sigma_\infty - \sigma_0)\omega \exp[-\omega p_n] \tag{4.2}$$

Note, however, that it would have been possible to model directly the non-linear hardening law in format (2.1).

The material parameter values are summarized in table 3.

Due to the problem symmetry, we consider only one sixteenth of the specimen, dis-

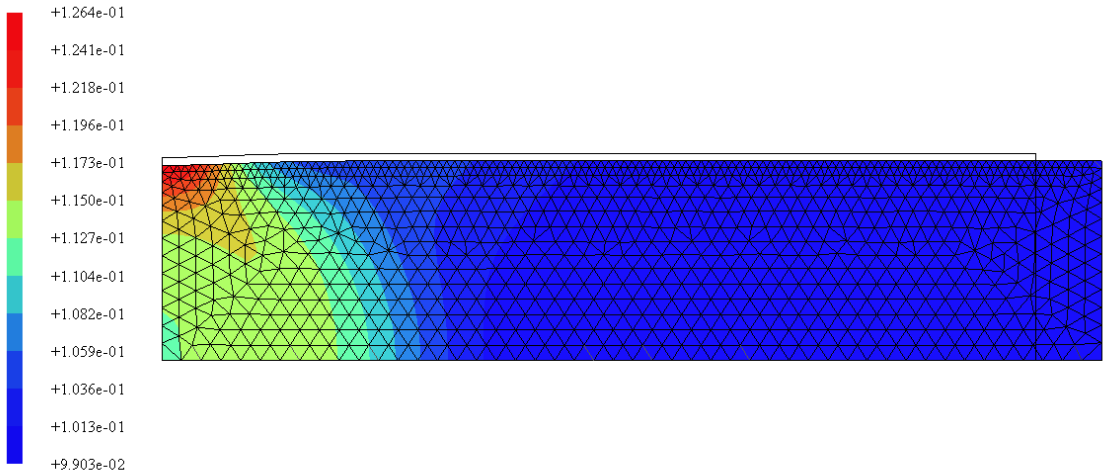
Table 3: Rod material properties

Material parameter	
Young modulus	$E = 207.9\,\text{GPa}$
Poisson ratio	$\nu = 0.29$
Initial yield stress	$\sigma_0 = 450\,\text{MPa}$
Infinite yield stress	$\sigma_\infty = 715\,\text{MPa}$
Hardening modulus	$h = 129.24\,\text{MPa}$
Saturation parameter	$\omega = 16.93$

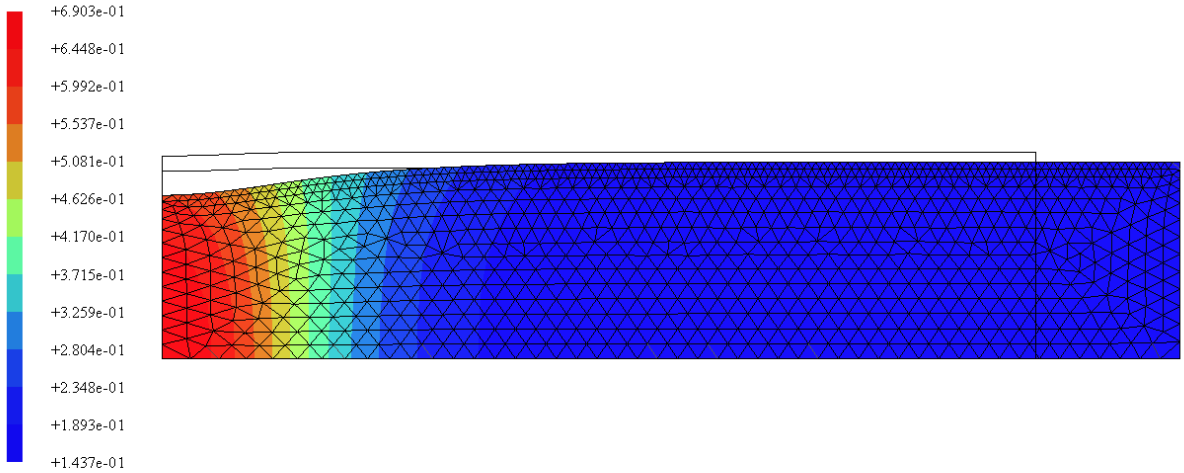
cretized with 44 000 quadratic tetrahedra. Boundary conditions consist of symmetry conditions and a displacement-driven condition $u_x = u(t)$ where a total imposed displacement of $u = 4.5\,\text{mm}$ applied in 65 uniform load increments (approximately 0.07 mm per increment).

Figures 7a and 7b depict two deformed meshes for $u = 2.0\,\text{mm}$ and $u = 4.50\,\text{mm}$ with the equivalent plastic strain isovalues. One can clearly see in the first figure the accumulation of plastic strain on the top of the reduced section where the necking will onset. Figure 8b represents the load-deflection curve as well as the radial contraction at the center section and the end sections of the rod. The reference model, the Abaqus model and the IPM solution all provide very similar results, especially for $u \leq 3.0\,\text{mm}$, i.e. before necking occurs. In the necking stage, small differences are observed which may be attributed to different algorithm tolerances or the incrementation process. However, the difference remains less than 1% which is clearly very satisfying. Figure 9a shows a number of iterations for the IPM ranging from 7 to 15 per increment. Iteration number slightly increases with increasing geometrical non-linearities but remains at a reasonable level. This result further confirms the robustness of the IPM with problem complexity. This aspect is one of the main advantages of the classical IPM when applied to convex problems which seems to be conserved in the present non-convex case.

Finally, we further tested the robustness of the IPM with respect to large load steps. Obviously, one can expect that this problem is challenging for any solver due to the striction regime occurring around 3 mm. We were able to obtain a converged solution using only 5 load steps for $u \leq 2.5\,\text{mm}$, 10 load steps for u between 2.5 mm and 3.5 mm and 2 load steps between 3.5 mm and 4.5 mm. Figure 9b shows the number of iterations for each of these 17 load steps. Interestingly, the number of iteration is quite similar to those of Figure 9a with smaller load steps. The total number of iterations in this case is 166 against 543 for the initial load-stepping using 65 load steps. This further confirms that the IPM is particularly robust in terms of convergence properties to large load steps.

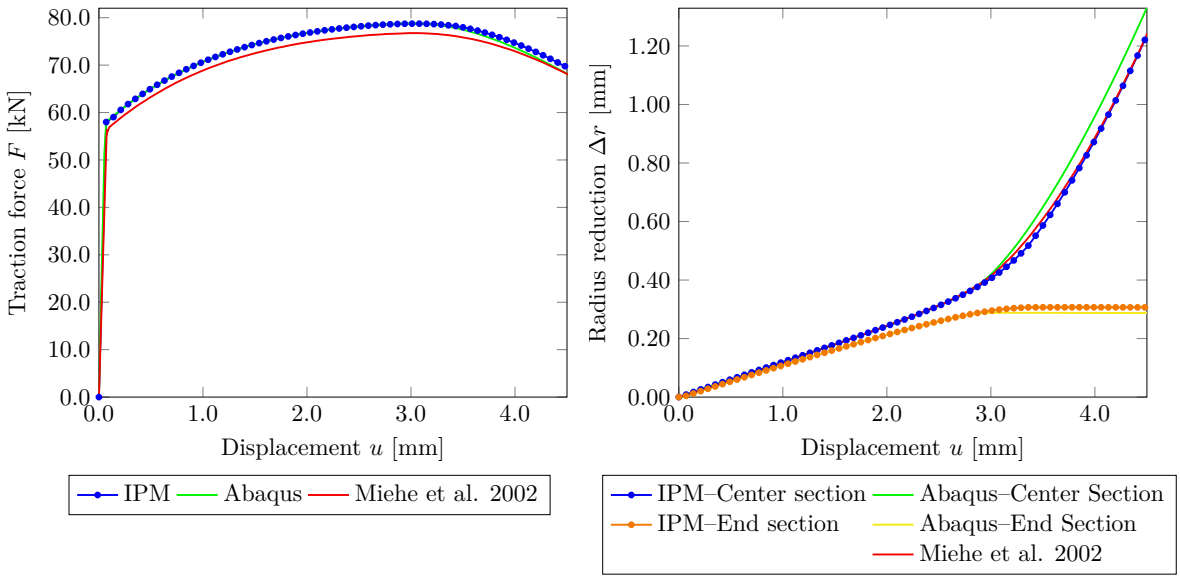


(a) $u = 2.0\,\mathrm{mm}$



(b) $u = 4.5\,\mathrm{mm}$

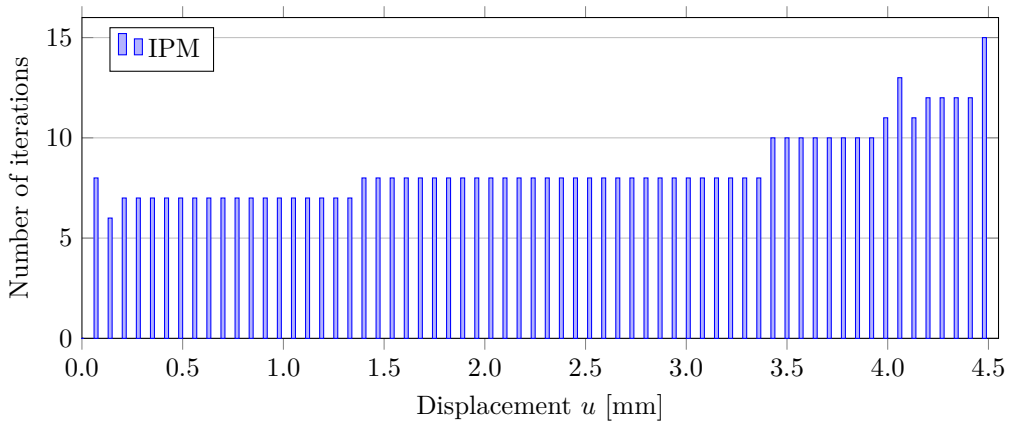
Figure 7: Deformed geometry and equivalent plastic strain isovalues for different elongation values u



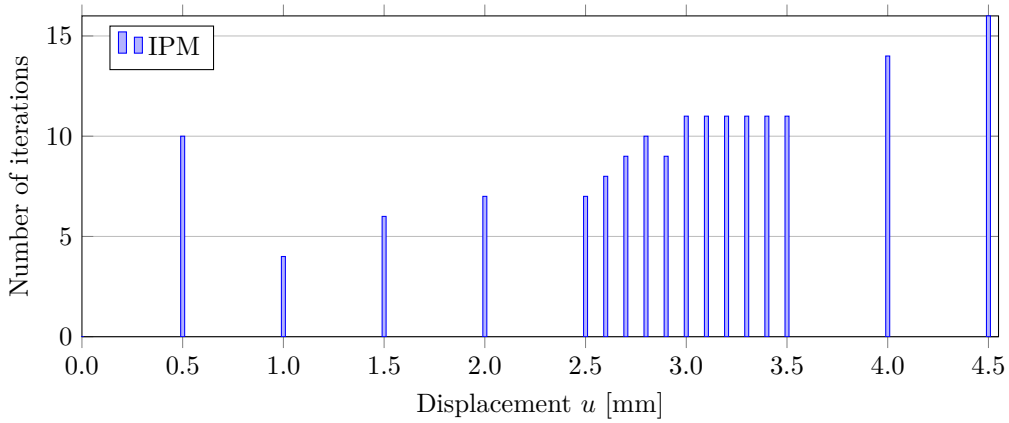
(a) Load displacement curve

(b) Radial reduction Δr of the mid and end section

Figure 8: Comparison of the results



(a) with 65 increments



(b) with 17 increments

Figure 9: Number of iterations per displacement increment

5. Conclusion

In this work, we investigated the use of a primal-dual interior-point algorithm for solving second-order cone programming (SOCP) problems involving non-convex objectives or constraints. A key assumption in our method relies on the fact that the additional non-convex terms are smooth, inducing additional contributions to the KKT system tangent matrix and residuals. In particular, the non-smooth terms, i.e. the conic constraints, are still assumed to be convex as for standard convex SOCP problems. Our proposed algorithm is therefore a straightforward extension to the standard primal-dual IPM.

The resolution of such non-convex optimization problems has been illustrated in the case of finite-strain elastoplasticity problems relying on a logarithmic strain framework. Indeed, in such models the elastoplastic constitutive law exhibits the same expression as in the small strain-case whereas only the total strain/displacement relationship is modified. The non-smooth characteristic of the plastic law can therefore still be reformulated as conic constraints whereas the non-convex strain/displacement expression is smooth. As a result, the corresponding incremental variational problem of logarithmic strain plasticity fits into the considered optimization problem format.

Implementation has been validated on numerical benchmarks and compared to standard Newton-type procedures. Since both approaches require the resolution of a Newton system of similar size, computational efficiency can be assessed by comparing the required number of iterations. Our results show that the IPM exhibits a good convergence behaviour with respect to the problem complexity and the load step size. Indeed, convergence robustness and relative stability of the number of iterations is one of the key interesting features which are classically observed with IPM in the convex setting. Our results seem to extend this observation in the present non-convex setting. As a result, it was possible to compute elasto-plastic solutions with only a few large load steps when Newton-type methods failed to converge in such situations. The total number of iterations of the IPM became competitive with respect to Newton methods when increasing the load step size, without impairing too much the solution quality.

Nevertheless further work is needed to improve the efficiency of the solution procedure. First, it is known that IPM cannot fully exploit the knowledge of points close to the solution. More efficient warm-start strategies than the simple strategy taken here could therefore improve the method convergence, especially when considering small load steps. Second, our results indicate that Newton-type method become more interesting with small load-stepping discretization but exhibit less robustness than IPM for larger load steps. A potentially effective strategy would then to use an IPM algorithm in the first iterations and switch to a Newton-type method for the final iterations, thereby benefiting from the Newton-method quadratic convergence near the solution. Some implementation details may also be worth investigating for improving the solver efficiency. For instance, some IPM implementations use merit functions to measure the quality of the next iterate. Finally, in Mehrotra's predictor-corrector scheme the complementarity gap linearization accuracy is improved by taking advantage of the affine step solution. Investigating if a similar approach can be used for improving the other non-linear terms would also be interesting to explore.

Appendix A Second-order cones

General results concerning Jordan algebra over the Lorentz second-order cone following the paper of Alizadeh and Goldfrab [20] are given here. For a vector $\boldsymbol{v} = (v^0, \overline{\boldsymbol{v}}) \in \mathcal{Q}^{m+1}$, we define:

$$\boldsymbol{mat}(\boldsymbol{v}) = \begin{bmatrix} v^0 & \overline{\boldsymbol{v}}^T \\ \overline{\boldsymbol{v}} & v^0 \boldsymbol{I}_m \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)} \tag{A.1}$$

$$\det(\boldsymbol{v}) = \det(\boldsymbol{mat}(\boldsymbol{v})) = (v^0)^2 - \|\overline{\boldsymbol{v}}\|^2 \tag{A.2}$$

$$\boldsymbol{Q} = \boldsymbol{diag}(1, -\boldsymbol{I}_m) \tag{A.3}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{Q}\boldsymbol{v} = (v^0, -\overline{\boldsymbol{v}}) \tag{A.4}$$

$$(\boldsymbol{mat}(\boldsymbol{v}))^{-1} = \frac{1}{\det(\boldsymbol{v})} \begin{bmatrix} v^0 & -\overline{\boldsymbol{v}}^T \\ -\overline{\boldsymbol{v}} & \dfrac{\det(\boldsymbol{v})\boldsymbol{I}_m + \overline{\boldsymbol{v}}\overline{\boldsymbol{v}}^T}{v^0} \end{bmatrix} \tag{A.5}$$

The last equation is valid only if $\boldsymbol{v}$ is located strictly inside $\mathcal{Q}^{m+1}$, i.e. $\boldsymbol{v} \in int(\mathcal{Q}^{m+1})$, which is equivalent to $\det(\boldsymbol{v}) > 0$.

Let $\boldsymbol{X} = \boldsymbol{mat}(\boldsymbol{x})$ and $\boldsymbol{S} = \boldsymbol{mat}(\boldsymbol{s})$, the complementarity condition $\boldsymbol{x}^T\boldsymbol{s} = 0$ for the second-order Lorentz cone can be rewritten as:

$$\boldsymbol{x} \circ \boldsymbol{s} = \begin{Bmatrix} x^0 s^0 + \overline{\boldsymbol{x}}^T\overline{\boldsymbol{s}} \\ x^0\overline{\boldsymbol{s}} + s^0\overline{\boldsymbol{x}} \end{Bmatrix} = \boldsymbol{X}\boldsymbol{s} = \boldsymbol{S}\boldsymbol{x} = \boldsymbol{X}\boldsymbol{S}\boldsymbol{e} \tag{A.6}$$

with $\boldsymbol{e} = (1, \boldsymbol{0})$ of dimension $m + 1$.

Appendix B Nesterov-Todd scaling and system reduction

Starting from the linearized form of the complementarity condition written as:

$$(\boldsymbol{x} \circ \boldsymbol{s})^{(k+1)} \simeq \boldsymbol{X}^{(k)}\boldsymbol{S}^{(k)}\boldsymbol{e} + \boldsymbol{X}^{(k)}\Delta\boldsymbol{s} + \boldsymbol{S}^{(k)}\Delta\boldsymbol{x} \tag{B.1}$$

It can be shown that for $\boldsymbol{x}, \boldsymbol{s} \in \text{int}(\mathcal{Q}^{m+1})$, there exists a unique matrix $\boldsymbol{F}$, depending on $\boldsymbol{x}$ and $\boldsymbol{s}$, such that

$$\boldsymbol{F}\boldsymbol{x} = \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{s}} = \boldsymbol{F}^{-T}\boldsymbol{S} \text{ and } \boldsymbol{x} \circ \boldsymbol{s} = \tilde{\boldsymbol{x}} \circ \tilde{\boldsymbol{s}} \tag{B.2}$$

Let $\boldsymbol{V} = \boldsymbol{mat}(\tilde{\boldsymbol{x}}) = \boldsymbol{mat}(\tilde{\boldsymbol{s}})$ the associated matrix for the scaled point $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{s}}$. Using this symmetrical Nesterov-Todd scaling, the linearized complementarity condition can be rewritten as

$$(\boldsymbol{x} \circ \boldsymbol{s})^{(k+1)} \simeq (\boldsymbol{x} \circ \boldsymbol{s})^{(k)} + \boldsymbol{V}\boldsymbol{F}^{-T}\Delta\boldsymbol{s} + \boldsymbol{V}\boldsymbol{F}\Delta\boldsymbol{x} \tag{B.3}$$

In the specific case of the second-order Lorentz cone, $\boldsymbol{F}^{-T} = \boldsymbol{F}^{-1}$. Further details and the expression for matrix $\boldsymbol{F}$ can be found in [20, 25, 29, 46]. The linearized KKT system

(2.10) can then be efficiently reduced:

$$\boldsymbol{J}^{(k)}\Delta\boldsymbol{w} = \begin{bmatrix} \boldsymbol{H}^{(k)} & \boldsymbol{A}^T & \boldsymbol{G}^{T(k)} & -\boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{G}^{(k)} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ (\boldsymbol{V}\boldsymbol{F})^{(k)} & \boldsymbol{0} & \boldsymbol{0} & (\boldsymbol{V}\boldsymbol{F}^{-T})^{(k)} \end{bmatrix} \begin{Bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{z} \\ \Delta\boldsymbol{s} \end{Bmatrix} = \begin{Bmatrix} -r_d^{(k)} \\ -r_p^{(k)} \\ -r_p'^{(k)} \\ -r_c^{(k)} \end{Bmatrix} = -\boldsymbol{r}^{(k)}(\boldsymbol{w},\eta) \tag{B.4}$$

The next step to reducing the system is to eliminate all conic slack variables $\Delta\boldsymbol{s}$ using static condensation:

$$\Delta\boldsymbol{s} = -(\boldsymbol{F}\boldsymbol{V}^{-1})^{(k)} r_c^{(k)} - (\boldsymbol{F}^T\boldsymbol{F})^{(k)}\Delta\boldsymbol{x} \tag{B.5}$$

one finally obtains the following reduced KKT system combining only the primal variables and the Lagrange multipliers for the equality constraints:

$$\boldsymbol{J}^{(k)}\Delta\boldsymbol{w} = \begin{bmatrix} \boldsymbol{H}^{(k)} + (\boldsymbol{F}^T\boldsymbol{F})^{(k)} & \boldsymbol{A}^T & \boldsymbol{G}^{T(k)} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{G}^{(k)} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{z} \end{Bmatrix} = \begin{Bmatrix} -r_d'^{(k)} \\ -r_p^{(k)} \\ -r_p'^{(k)} \end{Bmatrix} \tag{B.6}$$

where:

$$r_d'^{(k)} = \boldsymbol{\nabla}_x f^{(k)} + \boldsymbol{A}^T\boldsymbol{y}^{(k)} + \boldsymbol{G}^{T(k)}\boldsymbol{z}^{(k)} - \boldsymbol{s}^{(k)} - (\boldsymbol{F}\boldsymbol{V}^{-1})^{(k)} r_c^{(k)} \tag{B.7a}$$

$$r_p^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b} \tag{B.7b}$$

$$r_p'^{(k)} = g(\boldsymbol{x}^{(k)}) \tag{B.7c}$$

Appendix C Geometrical tensors

C.1 The generalised strain measure

Within an infinitesimal neighbourhood of a generic material particle, pure rotations can be distinguished from pure stretching by means of the polar decomposition of the deformation gradient. Under the action of pure rotations, the distances between particles within this neighbourhood remain fixed. Under stretching, we say that the region surrounding ı is strained. To quantify straining, i.e. to evaluate how much $\boldsymbol{U}$ departs from $\boldsymbol{I}$ being a rigid deformation, some kind of strain measure needs to be defined.

Since $\boldsymbol{U}$ is symmetrical, it follows that they admit a unique spectral decomposition

$$\boldsymbol{U} = \sum_{i=1}^{3} \lambda_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i = \sum_{i=1}^{3} \lambda_i \boldsymbol{M}_{ii} \tag{C.1}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are the eigenvalues and the triad $\{\boldsymbol{N}_1, \boldsymbol{N}_2, \boldsymbol{N}_3\}$ the Lagrangian triad or Lagrangian principle directions. The same triad can be expressed using $\boldsymbol{M}_{ij}$ which defines the full-symmetric spectral basis tensors:

$$\boldsymbol{M}_{ij} = \begin{cases} \boldsymbol{N}_i \otimes \boldsymbol{N}_i & \text{if } i = j \\ \frac{1}{2}(\boldsymbol{N}_i \otimes \boldsymbol{N}_j + \boldsymbol{N}_j \otimes \boldsymbol{N}_i) & \text{if } i \neq j \end{cases} \tag{C.2}$$

The generalised Lagragian strain measure is defined through an isotropic function of the pure stretching tensor $\boldsymbol{U}$ defined in its spectral space along the principal Lagrangian directions:

$$\boldsymbol{E}^* = f^*(\boldsymbol{U}) = \sum_{i=1}^{3} f^*(\lambda_i)\boldsymbol{N}_i \otimes \boldsymbol{N}_i = \sum_{i=1}^{3} f^*(\lambda_i)\boldsymbol{M}_{ii} \tag{C.3}$$

One of the most common families of strain measures is the Seth-Hill family defined as follows:

$$\boldsymbol{E}^{(m)} = \sum_{i=1}^{3} \frac{1}{m}(\lambda_i^m - 1)\boldsymbol{M}_{ii} \tag{C.4}$$

This family contains the most common strain measures used in literature:

$$\boldsymbol{E}_{GL} = \boldsymbol{E}^{(2)} = \sum_{i=1}^{3} \frac{1}{2}(\lambda_i^2 - 1)\boldsymbol{M}_{ii} \qquad \text{The Green-Lagrange deformation tensor} \tag{C.5}$$

$$\boldsymbol{E}^{(0)} = \sum_{i=1}^{3} \frac{1}{2}\ln(\lambda_i)\boldsymbol{M}_{ii} \qquad \text{The material Hencky deformation tensor} \tag{C.6}$$

C.2 Fourth and sixth order mapping tensors

Since all strain measures given by the generalised definition, or more specifically by the Seth-Hill family, are unique for a given deformation gradient, there is a one-to-one mappings that allows the passage between the different measures [28, 34, 35]. Let $\boldsymbol{E}^*$ and $\boldsymbol{E}^\dagger$ be two different generalised material strain measures as functions of the material stretch tensor $\boldsymbol{U}$. The fourth order geometrical mapping tensor $\mathbb{M}_{\boldsymbol{E}^\dagger}^{\boldsymbol{E}^*}$ such as $\boldsymbol{E}^* = \mathbb{M}_{\boldsymbol{E}^\dagger}^{\boldsymbol{E}^*} : \boldsymbol{E}^\dagger$ is given by:

$$\mathbb{M}_{\boldsymbol{E}^\dagger}^{\boldsymbol{E}^*} = \sum_{i=1}^{3} \frac{f^*(\lambda_i)}{f^\dagger(\lambda_i)}\boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ii} \tag{C.7}$$

one can easily verify that $\boldsymbol{E}^\dagger = \mathbb{M}_{\boldsymbol{E}^*}^{\boldsymbol{E}^\dagger} : \boldsymbol{E}^* = (\mathbb{M}_{\boldsymbol{E}^\dagger}^{\boldsymbol{E}^*})^{-1} : \boldsymbol{E}^*$ with:

$$\mathbb{M}_{\boldsymbol{E}^*}^{\boldsymbol{E}^\dagger} = (\mathbb{M}_{\boldsymbol{E}^\dagger}^{\boldsymbol{E}^*})^{-1} = \sum_{i=1}^{3} \frac{f^\dagger(\lambda_i)}{f^*(\lambda_i)}\boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ii} \tag{C.8}$$

In a similar way, there is a one-to-one mapping between the rates of each measure such that

$$\dot{\boldsymbol{E}}^* = \mathbb{M}_{\dot{\boldsymbol{E}}^\dagger}^{\dot{\boldsymbol{E}}^*} : \dot{\boldsymbol{E}}^\dagger \tag{C.9}$$

$$\mathbb{M}_{\dot{\boldsymbol{E}}^\dagger}^{\dot{\boldsymbol{E}}^*} = \frac{\partial \boldsymbol{E}^*}{\partial \boldsymbol{E}^\dagger} = \sum_{i=1}^{3} \frac{df^*(\lambda_i)/d\lambda_i}{df^\dagger(\lambda_i)/d\lambda_i}\boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ii} + \sum_{i=1}^{3}\sum_{j\neq i}^{3} \frac{f^*(\lambda_j) - f^*(\lambda_i)}{f^\dagger(\lambda_j) - f^\dagger(\lambda_i)}\boldsymbol{M}_{ij} \otimes \boldsymbol{M}_{ij} \tag{C.10}$$

In most cases, the reference strain measure used is the Green-Lagrange measure $\dot{\boldsymbol{E}}^\dagger = \dot{\boldsymbol{E}}_{GL}$, and the geometrical mapping tensor taken according to this measure:

$$\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} = \sum_{i=1}^{3} \frac{df^*(\lambda_i)/d\lambda_i}{\lambda_i} \boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ii} + \sum_{i=1}^{3}\sum_{j\neq i}^{3} \frac{2(f^*(\lambda_j) - f^*(\lambda_i))}{\lambda_j^2 - \lambda_i^2} \boldsymbol{M}_{ij} \otimes \boldsymbol{M}_{ij} \tag{C.11}$$

Note that all the defined geometrical tensors verify both the major and the minor symmetry conditions:

$$\mathbb{M}_{ijkl} = \mathbb{M}_{klij} \quad \text{and} \quad \mathbb{M}_{ijkl} = \mathbb{M}_{jikl} = \mathbb{M}_{ijlks} \tag{C.12}$$

and that

$$\mathbb{M}^{\dot{\boldsymbol{E}}_{GL}}_{\dot{\boldsymbol{E}}^*} = \left(\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}\right)^{-1} \tag{C.13}$$

In a general constitutive equation, the constitutive tangent moduli $\mathbb{D}^*$ relating the generalised strain increments and the generalised stress increments is required. Note that time derivatives for Lagrangian measures are always objective. This tangent moduli can also be written in terms of the classical tangent moduli $\mathbb{D}_{GL}$ relating the Green-Lagrange strain increments and the second Piola-Kirchhoff stress increments using the geometric mapping tensors. Assuming that we have derived the constitutive tangent moduli $\mathbb{D}^*$ for a generalised strain measure $\boldsymbol{E}^*$ and its work conjugate stress tensor $\boldsymbol{T}^*$ such that:

$$\dot{\boldsymbol{T}}^* = \mathbb{D}^* : \dot{\boldsymbol{E}}^* \tag{C.14}$$

the equivalent tangent moduli $\mathbb{D}_{GL}$ such that $\dot{\boldsymbol{S}} = \mathbb{D}_{GL} : \dot{\boldsymbol{E}}_{GL}$ can be calculated as follows:

$$\begin{aligned}
\dot{\boldsymbol{S}} &= \dot{\boldsymbol{T}}^* : \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} + \boldsymbol{T}^* : \dot{\mathbb{M}}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} \\
\dot{\boldsymbol{S}} &= \left(\frac{\partial \boldsymbol{T}^*}{\partial \boldsymbol{E}^*} : \frac{\partial \boldsymbol{E}^*}{\partial \boldsymbol{E}_{GL}} : \dot{\boldsymbol{E}}_{GL}\right) : \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} + \boldsymbol{T}^* : \left(\frac{\partial \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}}{\partial \boldsymbol{E}_{GL}} : \dot{\boldsymbol{E}}_{GL}\right) \\
\dot{\boldsymbol{S}} &= \mathbb{D}^* : \left(\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} : \dot{\boldsymbol{E}}_{GL}\right) : \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} + \boldsymbol{T}^* : \left(\mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} : \dot{\boldsymbol{E}}_{GL}\right) \\
\dot{\boldsymbol{S}} &= \left(\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} : \mathbb{D}^* : \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} + \boldsymbol{T}^* : \mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}\right) : \dot{\boldsymbol{E}}_{GL}
\end{aligned} \tag{C.15}$$

Thus we have

$$\mathbb{D}_{GL} = \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} : \mathbb{D}^* : \mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} + \boldsymbol{T}^* : \mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} \tag{C.16}$$

The previous relation is found using the major symmetry of $\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}$ and the introduction of the sixth-order geometric tensor $\mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}$ relating the rate of $\mathbb{M}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} = \partial \boldsymbol{E}^*/\partial \boldsymbol{E}_{GL}$ and the rate of $\boldsymbol{E}_{GL}$:

$$\mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} = \frac{\partial^2 \boldsymbol{E}^*}{\partial \boldsymbol{E}_{GL} \partial \boldsymbol{E}_{GL}} \tag{C.17}$$

Both geometrical tensors are required in order to formally map the tangent moduli associated to one strain measure to the tangent moduli associated to the other strain measure.

In practice, computing the six-order tensor $\mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}$ then doing the double contraction is not efficient therefore formulas for calculating directly the double contraction $\boldsymbol{T}^* : \mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}}$ have been derived:

$$\begin{aligned}\boldsymbol{T}^* : \mathfrak{L}^{\dot{\boldsymbol{E}}^*}_{\dot{\boldsymbol{E}}_{GL}} &= \sum_{i=1}^{3} F(\lambda_i) T^*_{ii} \boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ii} \\ &+ \sum_{i=1}^{3} \sum_{j\neq i} G(\lambda_i, \lambda_j) T^*_{ii} \boldsymbol{M}_{ij} \otimes \boldsymbol{M}_{ij} \\ &+ \sum_{i=1}^{3} \sum_{j\neq i} G(\lambda_i, \lambda_j) T^*_{ij} \left(\boldsymbol{M}_{ii} \otimes \boldsymbol{M}_{ij} + \boldsymbol{M}_{ij} \otimes \boldsymbol{M}_{ii}\right) \\ &+ \sum_{i=1}^{3} \sum_{j\neq i} \sum_{k\neq j\neq i} H(\lambda_i, \lambda_j, \lambda_k) T^*_{ik} \left(\boldsymbol{M}_{ij} \otimes \boldsymbol{M}_{jk} + \boldsymbol{M}_{jk} \otimes \boldsymbol{M}_{ij}\right)\end{aligned} \tag{C.18}$$

where

$$T^*_{ij} = \boldsymbol{T}^* : \boldsymbol{M}_{ij} \tag{C.19}$$

$$F(\lambda_i) = -\frac{2}{\lambda_i^4} \tag{C.20}$$

$$G(\lambda_i, \lambda_j) = \frac{8(f^*(\lambda_j) - f^*(\lambda_i)) - 4\Lambda_{ij}/\lambda_i}{\Lambda_{ij}^2} \tag{C.21}$$

$$H(\lambda_i, \lambda_j, \lambda_k) = 8\frac{-\Lambda_{jk} f^*(\lambda_i) - \Lambda_{ki} f^*(\lambda_j) - \Lambda_{ij} f^*(\lambda_k)}{\Lambda_{jk}\Lambda_{jk}\Lambda_{ij}} \tag{C.22}$$

$$\Lambda_{ij} = \lambda_j^2 - \lambda_i^2 \tag{C.23}$$

Note that $H(\lambda_i, \lambda_j, \lambda_k) = H(\lambda_i, \lambda_k, \lambda_j) = H(\lambda_k, \lambda_i, \lambda_j)$ but $G(\lambda_i, \lambda_j) \neq G(\lambda_j, \lambda_i)$. Also, when two or three principal stretches converge to the same value we have:

$$H(\lambda_i, \lambda_j, \lambda_k \to \lambda_j) = G(\lambda_i, \lambda_j) \tag{C.24}$$

$$H(\lambda_i, \lambda_j \to \lambda_i, \lambda_k \to \lambda_i) = G(\lambda_i, \lambda_j \to \lambda_i) = F(\lambda_i) \tag{C.25}$$

Note that all the corresponding formulas for the logarithmic strain measure and their geometrical mapping tensors can be found by replacing $f^*(\lambda_i)$ with $\ln(\lambda_i)$.

References

[1] Y. Kanno, Nonsmooth mechanics and convex optimization, CRC Press, Boca Raton, FL, 2011.

[2] Y. Kanno, J. A. C. Martins, A. Pinto da Costa, Three-dimensional quasi-static frictional contact by using second-order cone linear complementarity problem, International Journal for Numerical Methods in Engineering 65 (1) (2006) 62–83. doi:10.1002/nme.1493.

[3] R. Kučera, J. Machalová, H. Netuka, P. Ženčák, An interior-point algorithm for the minimization arising from 3d contact problems with friction, Optimization methods and software 28 (6) (2013) 1195–1217.

[4] I. Temizer, M. Abdalla, Z. Gürdal, An interior point method for isogeometric contact, Computer Methods in Applied Mechanics and Engineering 276 (2014) 589–611.

[5] D. Mangoni, A. Tasora, R. Garziera, A primal–dual predictor–corrector interior point method for non-smooth contact dynamics, Computer Methods in Applied Mechanics and Engineering 330 (2018) 351–367.

[6] C. E. Boustani], J. Bleyer, M. Arquier, M.-K. Ferradi, K. Sab, Dual finite-element analysis using second-order cone programming for structures including contact, Engineering Structures 208 (2020) 109892. doi:https://doi.org/10.1016/j.engstruct.2019.109892.
URL http://www.sciencedirect.com/science/article/pii/S0141029619317766

[7] K. Krabbenhoft, A. Lyamin, S. Sloan, P. Wriggers, An interior-point algorithm for elastoplasticity, International Journal for Numerical Methods in Engineering 69 (3) (2007) 592–626.

[8] K. Yonekura, Y. Kanno, Second-order cone programming with warm start for elastoplastic analysis with von Mises yield criterion, Optimization and Engineering 13 (2) (2012) 181–218. doi:10.1007/s11081-011-9144-4.
URL https://link.springer.com/article/10.1007/s11081-011-9144-4

[9] C. El Boustani, J. Bleyer, M. Arquier, M.-K. Ferradi, K. Sab, Elastoplastic and limit analysis of 3D steel assemblies using second-order cone programming and dual finite-elements, Engineering Structures (2020) 38.

[10] A. Makrodimopoulos, C. M. Martin, Lower bound limit analysis of cohesive-frictional materials using second-order cone programming, International Journal for Numerical Methods in Engineering 66 (4) (2006) 604–634. doi:10.1002/nme.1567.
URL http://doi.wiley.com/10.1002/nme.1567

[11] A. Makrodimopoulos, C. M. Martin, Upper bound limit analysis using simplex strain elements and second-order cone programming, International Journal for Numerical and Analytical Methods in Geomechanics 31 (6) (2007) 835–865. doi:10.1002/nag.567.
URL http://doi.wiley.com/10.1002/nag.567

[12] C. M. Martin, A. Makrodimopoulos, Finite-element limit analysis of Mohr–Coulomb materials in 3d using semidefinite programming, Journal of Engineering Mechanics 134 (4) (2008) 339–347.

[13] H. Vincent, M. Arquier, J. Bleyer, P. de Buhan, Yield design-based numerical analysis of three-dimensional reinforced concrete structures, International Journal for Numerical and Analytical Methods in Geomechanics 42 (18) (2018) 2177–2192.

[14] J. Bleyer, G. Hassen, Automated formulation and resolution of limit analysis problems (2020). arXiv:2005.04779.

[15] K. Krabbenhoft, A. Lyamin, J. Huang, M. V. da Silva, Granular contact dynamics using mathematical programming methods, Computers and Geotechnics 43 (2012) 165–176.

[16] X. Zhang, K. Krabbenhoft, D. Sheng, Particle finite element analysis of the granular column collapse problem, Granular Matter 16 (4) (2014) 609–619.

[17] J. Bleyer, M. Maillard, P. De Buhan, P. Coussot, Efficient numerical computations of yield stress fluid flows using second-order cone programming, Computer Methods in Applied Mechanics and Engineering 283 (2015) 599–614.

[18] J. Bleyer, Advances in the simulation of viscoplastic fluid flows using interior-point methods, Computer Methods in Applied Mechanics and Engineering.

[19] J. Bleyer, Automating the formulation and resolution of convex variational problems: applications from image processing to computational mechanics, arXiv preprint arXiv:1911.13185.

[20] F. Alizadeh, D. Goldfarb, Second-order cone programming, Mathematical Programming 95 (1) (2003) 3–51. doi:10.1007/s10107-002-0339-5.

[21] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear algebra and its applications 284 (1-3) (1998) 193–228.

[22] N. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proceedings of the sixteenth annual ACM symposium on Theory of computing, 1984, pp. 302–311.

[23] G. B. Dantzig, Origins of the simplex method, in: A history of scientific computing, ACM Press, 1990, pp. 141–151.

[24] Y. Nesterov, A. Nemirovskii, Y. Ye, Interior-point polynomial algorithms in convex programming, Vol. 13, SIAM, 1994.

[25] Y. E. Nesterov, M. J. Todd, Primal-dual interior-point methods for self-scaled cones, SIAM Journal on optimization 8 (2) (1998) 324–364.

[26] M. Wright, The interior-point revolution in optimization: history, recent developments, and lasting consequences, Bulletin of the American mathematical society 42 (1) (2005) 39–56.

[27] R. Chares, Cones and interior-point algorithms for structured convex optimization involving powers and exponentials, Ph.D. thesis, Ph. D. Thesis, UCL-Université Catholique de Louvain, Louvain-la-Neuve, Belgium (2009).

[28] C. Miehe, N. Apel, M. Lambrecht, Anisotropic additive plasticity in the logarithmic strain space: modular kinematic formulation and implementation based on incremental minimization principles for standard materials, Computer Methods in Applied Mechanics and Engineering 191 (47-48) (2002) 5383–5425. doi:10.1016/S0045-7825(02)00438-3.
URL http://linkinghub.elsevier.com/retrieve/pii/S0045782502004383

[29] E. Andersen, C. Roos, T. Terlaky, On implementing a primal-dual interior-point method for conic quadratic optimization, Mathematical Programming 95 (2) (2003) 249–277. doi:10.1007/s10107-002-0349-3.

[30] S. Mehrotra, On the Implementation of a Primal-Dual Interior Point Method, SIAM Journal on Optimization 2 (4) (1992) 575–601. doi:10.1137/0802028.
URL http://epubs.siam.org/doi/10.1137/0802028

[31] M. Salahi, J. Peng, T. Terlaky, On mehrotra-type predictor-corrector algorithms, SIAM Journal on Optimization 18 (4) (2008) 1377–1397.

[32] E. Andersen, On formulating quadratic functions in optimization problems, Technical Report TR-1-2013, MOSEK (2013).

[33] Mosek, Mosek Modeling Manual, User manual, Mosek (2013).
URL http://www.mosek.com/

[34] M. A. Caminero, F. J. Montáns, K.-J. Bathe, Modeling large strain anisotropic elastoplasticity with logarithmic strain and stress measures, Computers & Structures 89 (11-12) (2011) 826–843. doi:10.1016/j.compstruc.2011.02.011.
URL https://linkinghub.elsevier.com/retrieve/pii/S0045794911000447

[35] M. Latorre, F. J. Montáns, Stress and strain mapping tensors and general work-conjugacy in large strain continuum mechanics, Applied Mathematical Modelling 40 (5-6) (2016) 3938–3950. doi:10.1016/j.apm.2015.10.045.
URL https://linkinghub.elsevier.com/retrieve/pii/S0307904X1500709X

[36] B. Halphen, Q. Son Nguyen, Sur les matériaux standard généralisés, Journal de Mécanique 14 (1975) 39–63.
URL https://hal.archives-ouvertes.fr/hal-00105514

[37] J. Lemaitre, J.-L. Chaboche, Mechanics of solid materials, Cambridge university press, 1994.

[38] M. Abbas, A. Ern, N. Pignet, Hybrid high-order methods for finite deformations of hyperelastic materials, Computational Mechanics 62 (4) (2018) 909–928.

[39] M. Smith, ABAQUS/Standard User's Manual, Version 6.9, Simulia, 2009.

[40] A. Logg, G. N. Wells, Dolfin: Automated finite element computing, ACM Transactions on Mathematical Software 37 (2). doi:10.1145/1731022.1731030.

[41] A. Logg, K.-A. Mardal, G. Wells, Automated solution of differential equations by the finite element method: The FEniCS book, Vol. 84, Springer Science & Business Media, 2012.

[42] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The fenics project version 1.5, Archive of Numerical Software 3 (100). doi:10.11588/ans.2015.100.20553.

[43] T. Helfer, J. Bleyer, T. Frondelius, I. Yashchuk, T. Nagel, D. Naumov, The 'mfront-genericinterfacesupport' project, Journal of Open Source Software 5 (48) (2020) 2003. doi:10.21105/joss.02003.
URL https://doi.org/10.21105/joss.02003

[44] J. Simo, Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory, Computer Methods in Applied Mechanics and Engineering 99 (1) (1992) 61–112. doi:10.1016/0045-7825(92)90123-2.
URL https://linkinghub.elsevier.com/retrieve/pii/0045782592901232

[45] P. Papadopoulos, J. Lu, A general framework for the numerical solution of problems in finite elasto-plasticity, Computer Methods in Applied Mechanics and Engineering 159 (1-2) (1998) 1–18. doi:10.1016/S0045-7825(98)80101-1.
URL https://linkinghub.elsevier.com/retrieve/pii/S0045782598801011

[46] S. P. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, Cambridge, UK ; New York, 2004.

[47] T. Helfer, B. Michel, J.-M. Proix, M. Salvo, J. Sercombe, M. Casella, Introducing the open-source mfront code generator: Application to mechanical behaviours and material knowledge management within the pleiades fuel element modelling platform, Computers & Mathematics with Applications 70 (5) (2015) 994–1023.

[48] H. Yamashita, H. Yabe, A primal–dual interior point method for nonlinear optimization over second-order cones, Optimization Methods and Software 24 (3) (2009) 407–426. doi:10.1080/10556780902752447.
URL http://www.tandfonline.com/doi/abs/10.1080/10556780902752447

[49] J. Kleinert, B. Simeon, M. Obermayr, An inexact interior point method for the large-scale simulation of granular material, Computer Methods in Applied Mechanics and Engineering 278 (2014) 567–598.

[50] H. Zhang, J. Li, S. Pan, New second-order cone linear complementarity formulation and semi-smooth newton algorithm for finite element analysis of 3d frictional contact problem, Computer Methods in Applied Mechanics and Engineering 200 (1-4) (2011) 77–88.

[51] A. Klarbring, A mathematical programming approach to three-dimensional contact problems with friction, Computer Methods in Applied Mechanics and Engineering 58 (2) (1986) 175–200.