



HAL
open science

Advances in the simulation of viscoplastic fluid flows using interior-point methods

Jeremy Bleyer

► **To cite this version:**

Jeremy Bleyer. Advances in the simulation of viscoplastic fluid flows using interior-point methods. *Computer Methods in Applied Mechanics and Engineering*, 2018, 330, pp.368-394. 10.1016/j.cma.2017.11.006 . hal-01672045

HAL Id: hal-01672045

<https://enpc.hal.science/hal-01672045>

Submitted on 22 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Advances in the simulation of viscoplastic fluid flows using interior-point methods

Jeremy Bleyer^{a,*}

^a*Ecole des Ponts ParisTech, Laboratoire Navier UMR 8205 (ENPC-IFSTTAR-CNRS)
Université Paris-Est, Cité Descartes, 6-8 av Blaise Pascal, 77455 Champs-sur-Marne, FRANCE*

Abstract

We present a primal-dual interior point algorithm for the resolution of steady-state viscoplastic fluid flows formulated as a conic optimization problem. We give a complete description of the algorithm including some advanced aspects such as a predictor-corrector and scaling scheme to improve its efficiency. Our interior-point approach is shown to be largely more efficient than Augmented Lagrangian (AL) approaches which are traditionally used to solve such problems. In particular, the interior-point approach is roughly 5 times faster than the modern accelerated version of AL algorithms. The yield surfaces are shown to be accurately predicted and various examples ranging from channel flows to three-dimensional flows through a porous medium demonstrate its efficiency.

Keywords: yield stress fluids, viscoplasticity, Bingham model, interior point method, conic programming

1. Introduction

Viscoplastic or yield-stress fluids are materials which behave like a solid below a critical yield stress and flow like a viscous fluid for stresses higher than this threshold. Such materials are now commonly found in civil engineering, petroleum or food industries. Although many models have been proposed to describe the behavior of such materials, the Bingham model is the most well-known and very simple as the material is considered to be rigid below a yield criterion described by von Mises criterion and is linearly viscous above the yield stress. Such a simple model is enough to reproduce interesting physics such as the existence of stopped and rigid-body plug regions in flows or the return to rest in a finite time. For recent reviews on yield stress fluids, we refer to [1, 2].

* *Correspondence to:* J. Bleyer, Laboratoire Navier, 6-8 av Blaise Pascal, Cité Descartes, 77455 Champs-sur-Marne, France, Tel : +33 (0)1 64 15 37 40

Email address: jeremy.bleyer@enpc.fr (Jeremy Bleyer)

URL: <https://sites.google.com/site/bleyerjeremy/> (Jeremy Bleyer)

Regarding their numerical simulation, viscoplastic fluids are very challenging because of the separation between flowing and non flowing regimes. This non-smoothness in the expression of the constitutive relation necessitates to resort to more advanced techniques than those traditionally used for Newtonian fluids for instance. Regularized models have first been proposed to replace the non-smooth viscoplastic constitutive law by a smooth purely viscous model [3, 4]. Augmented Lagrangian (AL) approaches have then emerged as an interesting alternative to the use of regularized models to solve viscoplastic fluid flows, it is now one of the most popular method to solve such problems [5–9]. However, it still suffers from a slow convergence rate so that three-dimensional simulations are still extremely expensive. Recently, it has been pointed out that such AL schemes can be accelerated quite easily, yielding a faster convergence rate [10]. For a recent review on the simulation of viscoplastic fluids and various extensions, we refer to [11] and references herein.

In [12], it has first been proposed to use interior-point algorithms to solve the velocity minimum principle satisfied by the solution to a viscoplastic fluid flow. It follows the general trend that has been observed in the field of computational limit analysis/yield design theory in which interior-point methods (IPM) have been largely and successfully used for estimating the collapse load of various structures (continuum, beams, plates, shells, etc.) made of different materials (steel, masonry, soils, etc.). It is interesting to note that this field experienced a similar evolution regarding numerical techniques used to solve the rigid-perfectly plastic problem, which can be explained by the strong similarities shared with viscoplastic fluids. Regularization methods have indeed first been proposed [13, 14] until Augmented Lagrangian techniques became more appropriate [5, 15, 16]. Since the development of the IPM in the mathematical programming community [17], linear and non-linear convex optimization problems can now be efficiently solved and it became the state-of-the-art method in the field of computational limit analysis [18–21].

More precisely, interior-point algorithms are very efficient for *second-order cone programming* (SOCP), a specific class of problems which can be reformulated using second-order cone constraints. It encompasses linear programming problems and many non-linear problems, such as quadratic programming (see [22] for different illustrative applications). In the field of limit analysis, many yield criteria can be reformulated using such SOCP constraints and, in particular, the von Mises criterion. In [12], it has been shown that the quadratic viscous term for the Bingham model could also be easily reformulated so that the associated optimization problems also falls in the SOCP framework. It was therefore proposed to use the industrial interior-point software MOSEK [23] as a black-box solver for computing the viscoplastic flows. To be solved by a dedicated solver such as MOSEK, SOCP problems often need to be formulated in a standard format. For instance, the objective function can only be linear, which requires reformulating the quadratic viscous term using second-order cone constraints. Another requirement of the standard format is that a given optimization variable can belong to one cone only. For these reasons, the resulting optimization problems present a much larger number of variables or constraints compared to the initial formulation, leading to higher computational costs. The extension to a Herschel-Bulkley power-law

behavior also introduces similar problems.

In the present paper, we follow the general idea of the IPM, as usually implemented in IPM solvers, to derive an efficient algorithm for computing viscoplastic flows. The work presented here will serve many purposes:

- give all the necessary information to implement a home-made interior point algorithm and share the associated code for a larger diffusion of IPM in the field of viscoplastic fluids (part of the codes used in this paper can be found at <https://doi.org/10.5281/zenodo.1038519> [24]);
- further assess its efficiency compared to Augmented Lagrangian algorithms;
- further validate the solutions computed with the IPM, in particular regarding the accuracy of yield surfaces;
- provide an efficient way to resolve the limitations introduced by standard formats of black-box solvers.

To answer all these goals, the manuscript will first rapidly recall Augmented Lagrangian approaches and their accelerated versions in section 2, since they will be used for numerical comparison with IPM results. In section 3, the SOCP framework and the general idea of the IPM will be given. Advanced aspects of the IPM will be detailed in section 4 which will lead to a much more efficient algorithm. To ease the notation, the presentation of the first sections will be made in the context of antiplane problems with a Bingham model, 2D and 3D problems as well as extensions to other constitutive models will be discussed in section 5. Finally, numerical results and illustrative applications are presented in section 6.

2. Augmented Lagrangian approach for viscoplastic fluids

In this section, we will first recall the Augmented Lagrangian formulation which is now widely used for computing viscoplastic flows along with accelerated versions of the corresponding optimization algorithms.

2.1. Augmented Lagrangian approach

Let us consider the following minimum principle satisfied by the steady-state velocity field $u(\underline{x})$ for the antiplane flow through a cylindrical channel of section Ω lying in the Oxy -frame and subjected to a pressure gradient $f(\underline{x})$:

$$\min_{u \in \mathcal{V}} \int_{\Omega} \left(\frac{\eta}{2} \|\underline{\nabla} u\|^2 + \tau_0 \|\underline{\nabla} u\| \right) - \int_{\Omega} f u \quad (1)$$

where η represents the viscosity and τ_0 the yield stress of the Bingham model. We assume no-slip boundary conditions on some part $\partial\Omega_u$ of the domain boundary and \mathcal{V} denotes the appropriate functional space for u .

By introducing the strain rate \underline{d} as an additional optimization variable, the non-smoothness of the yield stress term and the derivative arising from the gradient operator are decoupled in the objective function and treated as a linear equality constraint:

$$\min_{u, \underline{d}} \int_{\Omega} \left(\frac{\eta}{2} \|\underline{d}\|^2 + \tau_0 \|\underline{d}\| \right) - \int_{\Omega} f u \quad (2a)$$

$$\text{s.t.} \quad \underline{\nabla} u = \underline{d} \quad (2b)$$

The previous formulation fits in the general framework of minimizing the sum of two convex functions $G(x) + H(y)$ constrained by a linear equation of the form $Ax + By = b$:

$$\min_{x, y} G(x) + H(y) \quad (3a)$$

$$\text{s.t.} \quad Ax + By = b \quad (3b)$$

In our case, with $x = \underline{d}$ and $y = u$, we have $G(x) = \int_{\Omega} g(x)$ with $g(x) = \frac{\eta}{2} \|x\|^2 + \tau_0 \|x\|$ and $H(y) = -\langle f, y \rangle + \iota_{\mathcal{V}}(y)$ with $\iota_{\mathcal{V}}$ being the indicator function of \mathcal{V} , while $A = \text{Id}$, $B = -\underline{\nabla}$ and $b = 0$. Two common methods for solving problems of the form (3) are the Alternating Direction Method of Multipliers (ADMM) and the Alternating Minimization Algorithm (AMA) which are Augmented Lagrangian approaches combined with an Uzawa-like algorithm. Both of them perform two separate minimization steps, once in x and once in y , before updating the Lagrangian multiplier associated with the linear constraint:

ADMM

$$x_{k+1} = \arg \min_x G(x) - \langle \lambda_k, Ax \rangle + \frac{r}{2} \|Ax + By_k - b\|^2 \quad (4a)$$

$$y_{k+1} = \arg \min_y H(y) - \langle \lambda_k, By \rangle + \frac{r}{2} \|Ax_{k+1} + By - b\|^2 \quad (4b)$$

$$\lambda_{k+1} = \lambda_k + r(b - Ax_{k+1} - By_{k+1}) \quad (4c)$$

Whereas for AMA, the first step is performed without the quadratic regularization term if G is strictly convex (which is the case here since $\eta > 0$):

AMA

$$x_{k+1} = \arg \min_x G(x) - \langle \lambda_k, Ax \rangle \quad (5a)$$

$$y_{k+1} = \arg \min_y H(y) - \langle \lambda_k, By \rangle + \frac{r}{2} \|Ax_{k+1} + By - b\|^2 \quad (5b)$$

$$\lambda_{k+1} = \lambda_k + r(b - Ax_{k+1} - By_{k+1}) \quad (5c)$$

where in both cases $r > 0$ is the augmentation parameter of the AL approach.

Identifying the Lagrange multiplier λ_k with a stress vector $\underline{\sigma}_k$, the first step amounts to the following update of the strain rate variable:

$$\underline{d}_{k+1} = \begin{cases} \frac{\tilde{\underline{\sigma}}_k}{\tilde{\eta}} \left(1 - \frac{\tau_0}{\|\tilde{\underline{\sigma}}_k\|} \right) & \text{if } \|\tilde{\underline{\sigma}}_k\| \geq \tau_0 \\ \underline{0} & \text{otherwise} \end{cases} \quad (6)$$

with

$$\begin{cases} \tilde{\underline{\sigma}}_k = \underline{\sigma}_k + r \nabla u_k, \tilde{\eta} = \eta + r & \text{for ADMM} \\ \tilde{\underline{\sigma}}_k = \underline{\sigma}_k, \tilde{\eta} = \eta & \text{for AMA} \end{cases} \quad (7)$$

The second step amounts, for both algorithms, to solving the following Stokes problem:

Find $u_{k+1} \in \mathcal{V}$ such that:

$$\int_{\Omega} r \nabla u_{k+1} \cdot \nabla v = \int_{\Omega} f v - \int_{\Omega} (\underline{\sigma}_k - r \underline{d}_{k+1}) \cdot \nabla v \quad \forall v \in \mathcal{V} \quad (8)$$

Finally, the Lagrange multiplier update reduces, for both algorithms, to:

$$\underline{\sigma}_{k+1} = \underline{\sigma}_k + r(\nabla u_{k+1} - \underline{d}_{k+1}) \quad (9)$$

Note that it is possible to invert the first two steps in both ADMM and AMA yielding slightly different algorithms with very similar efficiency.

2.2. Accelerated versions of Augmented Lagrangian methods

Both ADMM and AMA algorithms require solving the same Stokes problem at each iteration and perform local updates only. The cost per iteration is hence quite small since the Stokes stiffness matrix can be factorized once and for all at the beginning of the iterations. Despite this nice feature, the number of iterations required to reach convergence has often been reported to be very large, the theoretical convergence rate of the residuals being in $O(1/k)$.

However, it is possible to accelerate the convergence of both algorithms to reach an optimal $O(1/k^2)$ rate of the residuals [25]. Most importantly, these accelerated versions are obtained at a completely negligible cost, except for keeping in memory the fields from the previous iteration. Following Nesterov's predictor-corrector scheme [25, 26], the previous steps (6)-(9) are now replaced by:

$$\underline{d}_{k+1} = \begin{cases} \frac{\tilde{\underline{\sigma}}_k}{\tilde{\eta}} \left(1 - \frac{\tau_0}{\|\tilde{\underline{\sigma}}_k\|} \right) & \text{if } \|\tilde{\underline{\sigma}}_k\| \geq \tau_0 \\ \underline{0} & \text{otherwise} \end{cases} \quad (10)$$

with

$$\begin{cases} \tilde{\underline{\sigma}}_k = \underline{\sigma}_k + r \nabla u_k, \tilde{\eta} = \eta + r & \text{for accelerated-ADMM} \\ \tilde{\underline{\sigma}}_k = \underline{\sigma}_k, \tilde{\eta} = \eta & \text{for accelerated-AMA} \end{cases} \quad (11)$$

Find $u_{k+1}^{\text{pred}} \in \mathcal{V}$ such that:

$$\int_{\Omega} r \nabla u_{k+1}^{\text{pred}} \cdot \nabla v = \int_{\Omega} f v - \int_{\Omega} (\underline{\sigma}_k - r \underline{d}_{k+1}) \cdot \nabla v \quad \forall v \in \mathcal{V} \quad (12)$$

$$\underline{\sigma}_{k+1}^{\text{pred}} = \underline{\sigma}_k + r(\underline{d}_{k+1} - \nabla u_{k+1}^{\text{pred}}) \quad (13)$$

$$t_{k+1} = \left(1 + \sqrt{1 + 4t_k^2} \right) / 2 \quad (14a)$$

$$u_{k+1} = u_{k+1}^{\text{pred}} + \frac{t_k - 1}{t_{k+1}} (u_{k+1}^{\text{pred}} - u_k^{\text{pred}}) \quad (14b)$$

$$\underline{\sigma}_{k+1} = \underline{\sigma}_{k+1}^{\text{pred}} + \frac{t_k - 1}{t_{k+1}} (\underline{\sigma}_{k+1}^{\text{pred}} - \underline{\sigma}_k^{\text{pred}}) \quad (14c)$$

where $t_0 = 1$ and $u_0^{\text{pred}} = u_0$, $\underline{\sigma}_0^{\text{pred}} = \underline{\sigma}_0$ for the initialization.

It is clear that the first predictor steps are exactly the same as before. The only difference comes from the corrector step (14) which changes the previous predictions for the next iteration using some inertial terms accounting for the values obtained at the last iteration. Thus, contrary to the unaccelerated scheme, we must keep in memory the predictors of the last iteration k to perform the correction at iteration $k + 1$. We also observe that an additional scalar parameter t has been introduced which corresponds to an adaptively chosen relaxation step. We recover the unaccelerated version by fixing $t_k = 1$.

Remark:. Convergence of the accelerated ADMM version is ensured only if both functions G and H are strongly convex [25]. In the present case, H is only weakly convex and a restart procedure would need to be implemented to ensure convergence. This has not been done here since we always observed convergence of the accelerated ADMM algorithm, at least for antiplane simulations. Convergence of the accelerated AMA algorithm is however always ensured [25].

The use of such accelerated schemes in the context of viscoplastic fluid flows has recently been proposed in [10]. We share the point of view of this paper to use such accelerated variants in the future when implementing Augmented Lagrangian algorithms because of the small amount of work needed to accelerate an already existing AL code.

3. Primal-dual interior point method for viscoplastic fluids

We now describe the primal-dual interior point approach proposed for computing viscoplastic flows with a much higher efficiency than AL approaches. This section concentrates on the main ideas and most important ingredients of the method. More advanced aspects to obtain an efficient algorithm will be discussed in section 4.

3.1. Second-order cone programming formulation and optimality conditions

The interior-point approach can be applied to any linear or non-linear convex optimization problem but has been shown to be particularly efficient for solving specific classes of convex programming: linear programming (LP), second-order cone programming¹ (SOCP) and semi-definite programming (SDP). It usually requires reformulating the considered problem as the minimization of a linear function under specific convex constraints: linear equality/inequality constraints for LP, second-order cone constraints for SOCP and semi-definite constraints for SDP. As shown in [12], the viscoplastic problem can be rewritten into a second-order cone program by introducing additional optimization variables and conic constraints to fit the standard formulation required by industrial IP softwares like MOSEK. However, we will take advantage of the particular structure of the problem, namely an objective function which can be decomposed into a smooth (viscous) and a non-smooth (plastic) part to reformulate only the non-smooth part using second-order cone constraints and keep the non-linear but smooth part of the objective function as such. This leads to an important saving on the number of auxiliary variables compared to what has been proposed in [12].

By characterizing the subgradient of the objective function in (1), its solution u satisfies the following conditions [8, 27]:

Find $(u, t, \underline{\lambda})$ such that:

$$\int_{\Omega} (\eta \underline{\nabla} u + \tau_0 \underline{\lambda}) \cdot \underline{\nabla} v = \int_{\Omega} f v \quad \forall v \in \mathcal{V} \quad (15a)$$

$$\underline{\nabla} u = t \underline{\lambda} \quad , \quad t(1 - \|\underline{\lambda}\|) = 0, \quad t \geq 0, \quad \|\underline{\lambda}\| \leq 1 \quad (15b)$$

which expresses the mechanical equilibrium for a stress state of the form $\underline{\sigma} = \eta \underline{\nabla} u + \tau_0 \underline{\lambda}$, that is a sum of a viscous part and a stress state $\tau_0 \underline{\lambda}$ which is inside the yield criterion or at its boundary. Besides, the strain rate is associated to the plastic part by the flow rule $\underline{\nabla} u = t \underline{\lambda}$ with either $t = 0$ if $\|\underline{\lambda}\| < 1$ or $\|\underline{\lambda}\| = 1$ if $t > 0$.

Conditions (15b) can also be equivalently reformulated as the following primal-dual second order cone constraints:

$$\begin{aligned} & \underline{\nabla} u = t \underline{\lambda}, \quad t - \underline{\nabla} u \cdot \underline{\lambda} = 0, \quad \|\underline{\nabla} u\| \leq t, \quad \|\underline{\lambda}\| \leq 1 \\ \iff & (t, \underline{\nabla} u) \circ (1, -\underline{\lambda}) = (0, \underline{0}), \quad (t, \underline{\nabla} u) \in \mathcal{Q}^3, \quad (1, -\underline{\lambda}) \in \mathcal{Q}^3 \end{aligned} \quad (16)$$

where the circular product is defined in Appendix A and with \mathcal{Q}^n denoting the quadratic Lorentz cone of dimension n . The Karush-Kuhn-Tucker (KKT) optimality conditions can now be rewritten in the form of linear and conic constraints:

Find $(u, t, \underline{\lambda})$ such that:

$$\int_{\Omega} (\eta \underline{\nabla} u + \tau_0 \underline{\lambda}) \cdot \underline{\nabla} v = \int_{\Omega} f v \quad \forall v \in \mathcal{V} \quad (17a)$$

$$(t, \underline{\nabla} u) \circ (1, -\underline{\lambda}) = (0, \underline{0}) \quad (17b)$$

$$(t, \underline{\nabla} u) \in \mathcal{Q}^3 \quad , \quad (1, -\underline{\lambda}) \in \mathcal{Q}^3 \quad (17c)$$

¹which encompasses quadratic programming

Remark:. The generalization of the above derivation to Herschel-Bulkley fluids is immediate, one just has to replace the viscous quadratic term in (1) by the corresponding power-law potential, transforming the linear variational problem (17a) into a non-linear one.

3.2. General idea

The idea of the interior-point method (IPM) is to find a solution to (17) by following, more or less closely, a curve called the *central path* given by $(u(\mu), t(\mu), \underline{\lambda}(\mu))$ and parametrized by a *barrier parameter* $\mu \geq 0$. The central path is defined as the unique solution to the following perturbation of the KKT system (17) [28, 29]:

Find $(u, t, \underline{\lambda})$ such that :

$$\int_{\Omega} (\eta \underline{\nabla} u + \tau_0 \underline{\lambda}) \cdot \underline{\nabla} v = \int_{\Omega} f v \quad \forall v \in \mathcal{V} \quad (18a)$$

$$(t, \underline{\nabla} u) \circ (1, -\underline{\lambda}) = (\mu, \underline{0}) \quad (18b)$$

$$(t, \underline{\nabla} u) \in \mathcal{Q}^3 \quad , \quad (1, -\underline{\lambda}) \in \mathcal{Q}^3 \quad (18c)$$

For $\mu = 0$, we have exactly (17). From the perturbed equation (18b), we obtain for the solution to (18) that:

$$t(\mu) = \frac{\mu}{2} + \frac{1}{2} \sqrt{\mu^2 + 4 \|\underline{\nabla} u\|^2} > \|\underline{\nabla} u\| \quad (19a)$$

$$\underline{\lambda}(\mu) = \underline{\nabla} u \left(\frac{\mu}{2} + \frac{1}{2} \sqrt{\mu^2 + 4 \|\underline{\nabla} u\|^2} \right)^{-1} \quad (19b)$$

so that the stress state is given by $\underline{\sigma} = \eta \underline{\nabla} u + \tau_0 \underline{\nabla} u \left(\frac{\mu}{2} + \frac{1}{2} \sqrt{\mu^2 + 4 \|\underline{\nabla} u\|^2} \right)^{-1}$. As a consequence the solution of (18) for a *given* $\mu > 0$ can be interpreted as the solution of a regularized viscoplastic behavior, as pointed out recently in [11]. However, we will later discuss how the proposed method is fundamentally different, at least from the user point of view, from traditional regularized approaches.

A nice property of the central path is that it defines a continuous set of strictly feasible points which are well-centred, i.e. far from the boundary of the feasible region (except when approaching the optimum for very small μ) so that large descent steps can be taken when minimizing the objective function from points located near the central path (see Figure 1). Indeed, when computing descent directions from the unperturbed KKT equations (17), the iterates usually attain the feasible region boundary very rapidly so that only very small steps can be taken close to the boundary to preserve feasibility of the subsequent iterates, yielding a small convergence rate. If one is able to compute solutions to (18), the central path defines a preferential way to converge to the solution of the viscoplastic problem by driving μ to 0.

From the KKT conditions (18b), different strategies for finding a solution to (18) can be chosen. *Primal interior point methods* amount at eliminating the dual conic variables $\underline{\lambda}$

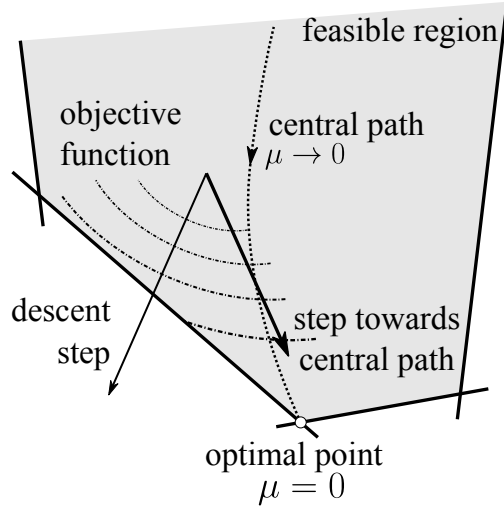


Figure 1: General idea of the interior point method. The central path defines a set of well-centered points located far from the feasible region boundaries. A descent step rapidly ends near the feasible region boundary. By using steps which follow the central path, larger steps can be taken to avoid the feasible region boundary.

using (19b), working only on the primal variable u . The auxiliary variable t plays no role anymore and the second-order cone constraints are automatically satisfied. In this case, the interior-point method shares strong similarities with the barrier method and regularization approaches. We will not detail primal interior point methods here since their practical performance is not very good compared to *primal-dual* methods which will be described later. For such an approach, $\underline{\lambda}$ and t remain as unknowns and (18) is then as difficult to solve as (17). However, as will be made clear in section 3.4, we will never aim at finding points exactly on the central path but just use it as a guide towards the solution.

3.3. Finite element discretization

Before going into the general steps of the primal-dual IPM, let us first introduce the finite-element discretization of problem (17). We consider a linear interpolation of the fluid velocity field u on triangular finite elements. The strain rate vector, the auxiliary variable t associated with its norm as well as the dual normalized stress vector $\underline{\lambda}$ are all interpolated as piecewise constants on the same triangular mesh which will enable to satisfy everywhere all the constraints of (17). We adopt here the same mixed stress-velocity finite element approximation as traditionally used in Augmented Lagrangian approaches. Denoting the corresponding vectors of unknowns using bold symbols, the discrete version of the perturbed KKT conditions are given by:

Find $(\mathbf{u}, \mathbf{t}, \mathbf{d}, \boldsymbol{\lambda})$ such that:

$$\mathbf{K}\mathbf{u} + \tau_0 \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{f} \quad (20a)$$

$$\mathbf{B}\mathbf{u} - \mathbf{d} = \mathbf{0} \quad (20b)$$

$$(t_i, \mathbf{d}_i) \circ (1, -\boldsymbol{\lambda}_i) = \mu \mathbf{e} \quad \forall i = 1, \dots, N \quad (20c)$$

$$(t_i, \mathbf{d}_i) \in \mathcal{Q}^3, \quad (1, -\boldsymbol{\lambda}_i) \in \mathcal{Q}^3 \quad (20d)$$

where $\mathbf{e} = (1, 0, 0)$, $\mathbf{B} = \nabla \mathbf{N}_u$ denotes the velocity shape function derivatives and \mathbf{K} is the stiffness matrix arising from the viscosity term. As is classically done in standard interior-point conic solvers, an additional strain rate variable \mathbf{d} has been introduced to avoid any linear relationship inside the cones so that they are all decoupled from each other whereas coupling is restored by the additional linear constraint $\mathbf{B}\mathbf{u} - \mathbf{d} = \mathbf{0}$. The conic constraints and complementarity equations are written for all elements $i = 1, \dots, N$ with N being the total number of elements.

3.4. Primal-dual interior point algorithm

We are now ready to describe the different steps of the primal-dual IPM. As mentioned earlier, we do not need to find exactly points lying on the central path but will only try to remain close to it. Hence, at each iteration k , a Newton step on the perturbed system (20) is computed towards the central path for a fixed value of μ^k . From this new point, the barrier parameter is decreased by some amount and the process is repeated until the different residuals fall below a prescribed tolerance. This method generates a sequence of iterates $(\mathbf{u}^k, \mathbf{t}^k, \mathbf{d}^k, \boldsymbol{\lambda}^k)$ which need to remain *strictly* feasible regarding the conic constraints i.e. $\|\mathbf{d}_i^k\| < t^k$ and $\|\boldsymbol{\lambda}_i^k\| < 1$.

3.4.1. Solving the perturbed KKT system

Let us consider a given iteration k at the beginning of which a point $\mathbf{z}^k = (\mathbf{u}^k, \mathbf{t}^k, \mathbf{d}^k, \boldsymbol{\lambda}^k)$ satisfying the conic constraints is known and a value for the barrier parameter $\mu^{(k)}$ has been chosen. The next iterate is obtained by computing an approximation of the point on the central path for $\mu = \mu^{(k)}$. This approximation is obtained by using one iteration of the Newton method. Hence, a descent direction $\Delta \mathbf{z} = (\Delta \mathbf{u}, \Delta \mathbf{t}, \Delta \mathbf{d}, \Delta \boldsymbol{\lambda})$ is computed by solving the linearization of the perturbed KKT system around \mathbf{z}^k :

$$\mathbf{A}\Delta \mathbf{z} = \begin{bmatrix} \mathbf{K} & \mathbf{0} & \mathbf{0} & \tau_0 \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{C}_\lambda^k & -\mathbf{C}_d^k \\ \mathbf{0} & -\mathbf{diag}(\boldsymbol{\lambda}^k) & \mathbf{I} & -\mathbf{C}_t^k \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{t} \\ \Delta \mathbf{d} \\ \Delta \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_d^k \\ \mathbf{r}_p^k \\ \mathbf{r}_c^k \\ \mathbf{r}_{c'}^k \end{Bmatrix} = \mathbf{r}^k \quad (21)$$

where $\mathbf{C}_\lambda^k = \mathbf{diag}(\dots, (\boldsymbol{\lambda}_i^k)^T, \dots)$, $\mathbf{C}_d^k = \mathbf{diag}(\dots, (\mathbf{d}_i^k)^T, \dots)$ and $\mathbf{C}_t^k = \mathbf{diag}(\dots, \begin{bmatrix} t_i^k \\ t_i^k \end{bmatrix}, \dots)$ arise from the linearization of (20c).

The vector \mathbf{r}^k corresponds to a vector of residuals for the different linearized equations of the KKT system: $\mathbf{r}_d^k = \mathbf{f} - \mathbf{K}\mathbf{u}^k - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^k$ is the *dual* residual, $\mathbf{r}_p^k = \mathbf{d}^k - \mathbf{B}\mathbf{u}^k$ is the *primal* residual and $\mathbf{r}_c^k = \mu^{(k)} - \mathbf{t}^k + \mathbf{C}_\lambda^k \mathbf{d}^k$, $\mathbf{r}_{c'}^k = \mathbf{C}_t^k \boldsymbol{\lambda}^k - \mathbf{d}^k$ are the *centering* residuals. Due to the linearity of the first two equations of (20), it is possible that the primal and dual residuals are always zero if the initial point $\mathbf{z}^0 = (\mathbf{u}^0, \mathbf{t}^0, \mathbf{d}^0, \boldsymbol{\lambda}^0)$ is chosen so that $\mathbf{r}_p^0 = \mathbf{r}_d^0 = \mathbf{0}$. If finding an initial point satisfying the linear constraints for a general interior-point problem may be difficult, in our case a simple initialization using the solutions of the

corresponding Stokes flow and $\boldsymbol{\lambda}^0 = 0$ satisfies these conditions. The case where $\mathbf{r}_p^k = \mathbf{r}_d^k = 0$ for all iterations is usually referred to as a *feasible* interior-point method in contrast to an *infeasible* interior-point method where the initial point does not satisfy the primal and dual residual equations. However, one has to recall that, in any case, each iterate will always need to be feasible with respect to the conic constraints. Although it may seem more efficient to use a feasible approach, numerical experiments show that infeasible approaches are almost as efficient so that the overall performance is quite insensitive to the choice of the initial point.

3.4.2. Step-length computation

The next iterate is given by $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha \Delta \mathbf{z}$ where $\alpha \in (0, 1]$ is a step-length which is chosen so that \mathbf{z}^{k+1} still satisfies all conic constraints. In practice, one can adopt for instance a simple line-search procedure. A fraction-to-boundary rule is used to avoid that the next iterates lie exactly on the boundary of the feasible domain. In practice, this rule consists in choosing $\alpha = \theta \alpha_{max}$ with $\theta = 0.99 - 0.995$.

The primal and dual residuals are then reduced by a factor $1 - \alpha$. The efficiency of the interior-point approach is due to the fact that this step-length will in general remain quite large i.e. close to 1, because the iterates tend to follow the central path which is located far from the boundaries of the conic-feasible region. On the contrary, a traditional descent direction along the gradient of the objective function will reach the feasible region boundaries at early stages of the iterations and the subsequent step lengths will be very small in order to maintain feasibility, exhibiting then only little progress.

3.4.3. Choice of the barrier parameter

A key point of the interior-point method lies in the choice of the barrier parameter μ for each iteration. In order to solve the initial unperturbed problem, the method aims at decreasing this parameter from one iteration to another. At the end of iteration k , a feasible point \mathbf{z}^{k+1} has been computed and a common choice for $\mu^{(k+1)}$ makes use of the value of the *complementarity gap* $g_i^{k+1} = t_i^{k+1} - \boldsymbol{\lambda}_i^{k+1} \cdot \mathbf{d}_i^{k+1}$ for each point i :

$$\mu^{(k+1)} = \gamma \left(\frac{1}{N} \sum_{i=1}^N g_i^{k+1} \right) = \gamma \bar{g}^{k+1} \quad (22)$$

i.e. $\mu^{(k+1)}$ is equal to the global average complementarity gap \bar{g}^{k+1} times a scalar parameter $\gamma \in (0, 1]$ which will control the amount of centering for the next iteration. The idea of this *centering* parameter is that for $\gamma = 0$ the KKT equations of the next iteration will aim at reducing the complementarity gap but maybe to the detriment of going away from the central path. On the contrary, a value of $\gamma = 1$ means that the complementarity gap will not be reduced (see Appendix B) but the next iterate will be closer to the central path so that a larger step may be taken at the next iteration. We will see in the next section that this centering parameter can be chosen in an adaptive manner.

3.5. Initialization and convergence criteria

For the initialization of the algorithm, we chose to take $\mathbf{u} = 0$ (except when non-zero imposed velocity boundary conditions are present, see example 6.2), $\mathbf{d} = \boldsymbol{\lambda} = 0$ and $\mathbf{t} = 1$. This choice ensures that the initial point is strictly feasible regarding both conic constraints $(\mathbf{t}_i, \mathbf{d}_i) \in \mathcal{Q}^3$ and $(1, \boldsymbol{\lambda}_i) \in \mathcal{Q}^3$. Besides, it also satisfies the strain rate-velocity relation $\mathbf{B}\mathbf{u} = \mathbf{d}$. As a consequence, the primal residual \mathbf{r}_p will always be zero during the iterative process.

Another possible initialization would be to use the solution $\mathbf{u}_{\text{Stokes}}$ of the corresponding Stokes problem as a first guess for \mathbf{u} and $\mathbf{B}\mathbf{u}_{\text{Stokes}}$ as a first guess for \mathbf{d} . In this case, t has to be chosen to be strictly greater than $\|\mathbf{B}\mathbf{u}_{\text{Stokes},i}\|$ everywhere. We did not see a significant improvement in the efficiency of the algorithm by choosing this initialization so we decided to keep the first choice.

For the chosen initialization, the initial complementarity gap is equal to 1 everywhere. As the residual vectors and the complementarity gap are both reduced at the same rate, we stopped the iterative process when the following conditions are met:

$$\bar{g}^k < \epsilon \quad \text{and} \quad \|(\mathbf{r}_p^k, \mathbf{r}_d^k)\| < \epsilon \quad (23)$$

with a tolerance of typically $\epsilon = 10^{-8}$. A point satisfying these conditions will then be considered as an optimal solution to the initial *unperturbed* problem (17).

The algorithm is also stopped if the iteration step size becomes too small or if a maximum number of iterations (e.g. $N_{\text{iter,max}} = 200$) is exceeded meaning that the algorithm failed to find an optimal solution. In all our numerical tests, this situation has never been encountered.

A simple version of the primal-dual interior point algorithm is summarized by Algorithm 1.

4. Advanced aspects of the primal-dual interior-point method

4.1. Predictor-corrector scheme and adaptive choice of the barrier parameter

One major difficulty in using the previous simple version of the primal-dual IPM lies in the optimal choice of the centering parameter γ . As mentioned before, the choice of γ represents a trade-off between improving centering and reducing the complementarity gap. One major improvement in the efficiency of primal-dual IPM has been realized by Mehrotra [30] who proposed an adaptive choice of the centering parameter by using a predictor-corrector scheme. This scheme takes advantage of the fact that factorizing the KKT matrix \mathbf{A} of equation (21) is the most demanding step for each iteration. Once this factorization has been found, it is quite inexpensive to solve a second linear system with the same matrix.

Algorithm 1 A simple primal-dual interior point algorithm

- 1: Initialization ($k = 0$): choose $\mathbf{z}^0 = (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0})$, $\gamma \in (0, 1]$ and $\epsilon > 0$
 - 2: $(\mathbf{r}_p^0, \mathbf{r}_d^0) = (\mathbf{f}, \mathbf{0})$
 - 3: $\bar{g}^0 = \frac{1}{N} \sum_{i=1}^N (t_i^0 - \boldsymbol{\lambda}_i^0 \cdot \mathbf{d}_i^0) = 1$
 - 4: $\mu^{(0)} = \gamma \bar{g}^0$
 - 5: **while** $\|(\mathbf{r}_p^k, \mathbf{r}_d^k)\| > \epsilon$ or $\bar{g}^k > \epsilon$ **do**
 - 6: form residual vector \mathbf{r}^k and Newton matrix \mathbf{A}
 - 7: solve $\mathbf{A}\Delta\mathbf{z} = \mathbf{r}^k$
 - 8: compute maximum step length α
 - 9: update variables $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha\Delta\mathbf{z}$
 - 10: compute new barrier parameter $\mu^{(k+1)} = \gamma\bar{g}^{k+1}$
 - 11: $k \leftarrow k + 1$
 - 12: **if** $k > N_{iter,max}$ or $\alpha \leq \epsilon$ **then break** "Algorithm failed."
 - 13: **end if**
 - 14: **end while**
-

More precisely, the predictor-corrector scheme uses the fact that:

$$\begin{aligned}
(t_i^+, \mathbf{d}_i^+) \circ (1, -\boldsymbol{\lambda}_i^+) &= (t_i, \mathbf{d}_i) \circ (1, -\boldsymbol{\lambda}_i) \\
&\quad + (\Delta t_i - \boldsymbol{\lambda}_i^T \Delta \mathbf{d}_i - \mathbf{d}_i^T \Delta \boldsymbol{\lambda}_i, \Delta \mathbf{d}_i - t_i \Delta \boldsymbol{\lambda}_i - \Delta t_i \boldsymbol{\lambda}_i) \\
&\quad - (\Delta \mathbf{d}_i^T \Delta \boldsymbol{\lambda}_i, \Delta t_i \Delta \boldsymbol{\lambda}_i)
\end{aligned}$$

with the notation $\mathbf{z}^+ = \mathbf{z} + \Delta\mathbf{z}$. When linearizing this quantity to form the Newton step of the perturbed system, the last quadratic term $(\Delta \mathbf{d}_i^T \Delta \boldsymbol{\lambda}_i, \Delta t_i \Delta \boldsymbol{\lambda}_i)$ is neglected. Mehrotra suggested estimating this term from a first pure Newton step on the unperturbed system. Hence, (21) is first solved with $\mu^{(k)} = 0$, the corresponding right-hand side being given by:

$$\mathbf{r}_d^k = \mathbf{f} - \mathbf{K}\mathbf{u}^k - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^k \quad (24a)$$

$$\mathbf{r}_p^k = \mathbf{d}^k - \mathbf{B}\mathbf{u}^k \quad (24b)$$

$$\mathbf{r}_c^k = -\mathbf{t}^k + \mathbf{C}_\lambda^k \mathbf{d}^k \quad (24c)$$

$$\mathbf{r}_{c'}^k = \mathbf{C}_t^k \boldsymbol{\lambda}^k - \mathbf{d}^k \quad (24d)$$

This first step is called the *affine step* and its solution is denoted by $\Delta\mathbf{z}^a = (\Delta\mathbf{u}^a, \Delta\mathbf{t}^a, \Delta\mathbf{d}^a, \Delta\boldsymbol{\lambda}^a)$. We denote by α_{max}^a the maximum step length possible for this search direction. The centering parameter γ is then estimated from this maximum step length using the following heuristic [31]:

$$\gamma = (1 - \alpha_{max}^a) \cdot \min \{0.5; (1 - \alpha_{max}^a)^2\} \quad (25)$$

This heuristic enables to choose a centering parameter which is small if a large affine step can be taken (α_{max}^a close to 1) so that the adopted step is close to a pure Newton step. Otherwise, centering is improved by choosing a larger value of γ , yielding a more conservative direction

towards the central path. The final search direction is then computed as the solution $\Delta \mathbf{z}$ of (21) with the following residuals:

$$\mathbf{r}_d^k = (1 - \gamma) (\mathbf{f} - \mathbf{K}\mathbf{u}^k - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^k) \quad (26a)$$

$$\mathbf{r}_p^k = (1 - \gamma) (\mathbf{d}^k - \mathbf{B}\mathbf{u}^k) \quad (26b)$$

$$\mathbf{r}_c^k = \mu^{(k)} - \mathbf{t}^k + \mathbf{C}_\lambda^k \mathbf{d}^k \quad (26c)$$

$$\mathbf{r}_{c'}^k = \mathbf{C}_t^k \boldsymbol{\lambda}^k - \mathbf{d}^k + \mathbf{C}_{\Delta \mathbf{t}^a}^k \Delta \boldsymbol{\lambda}^a \quad (26d)$$

with $\mu^{(k)} = \gamma \bar{g}^k$ and where the quadratic correction $\mathbf{C}_{\Delta \mathbf{t}^a}^k \Delta \boldsymbol{\lambda}^a$ has been added to the centering residuals based on the affine step prediction (the quadratic correction term of (26c) is in fact zero, see Appendix B). We remark that a factor $(1 - \gamma)$ has been added to the primal and dual residuals, this is motivated by the fact that the new primal and dual residuals corresponding to the point $\mathbf{z} + \alpha \Delta \mathbf{z}$ are then given by $(\mathbf{r}_p^{k+1}, \mathbf{r}_d^{k+1}) = (1 - \alpha(1 - \gamma))(\mathbf{r}_p^k, \mathbf{r}_d^k)$ and that the new average complementarity gap is also given by $\bar{g}^{k+1} = (1 - \alpha(1 - \gamma))\bar{g}^k$ (see again Appendix B). This ensures that each quantity is decreased at the same rate which is beneficial for the algorithm efficiency.

4.2. Reduced linear system

As previously mentioned, the main computation cost of the IPM is the resolution of the Newton linear system (21) (the factorization of \mathbf{A} in case of a predictor-corrector scheme) at each iteration. It can be observed from the structure of system (21) that \mathbf{A} is a sparse non-symmetric matrix and is quite large compared to the size of the stiffness matrix \mathbf{K} of a classical Stokes problem. However, it is possible to reduce the previous system to a smaller system, of the size of \mathbf{K} , to solve (21) more efficiently, although the obtained matrix will be non-symmetric matrix if condensing matrix \mathbf{A} directly. A specific rescaling procedure has, therefore, to be adopted to obtain a symmetric reduced system but also to ensure convergence of the interior-point algorithm.

To do so, let us first rewrite the complementarity conditions as follows:

$$(t^i, \mathbf{d}^i) \circ (1, -\boldsymbol{\lambda}^i) = \mathbf{X}^i \mathbf{s}^i = \mathbf{X}^i \mathbf{S}^i \mathbf{e} = \mathbf{S}^i \mathbf{X}^i \mathbf{e} = \mathbf{S}^i \mathbf{x}^i \quad (27a)$$

$$\text{where } \mathbf{x}^i = \begin{Bmatrix} t^i \\ \mathbf{d}^i \end{Bmatrix} \quad (27b)$$

$$\mathbf{s}^i = \begin{Bmatrix} 1 \\ -\boldsymbol{\lambda}^i \end{Bmatrix} \quad (27c)$$

$$\mathbf{X}^i = \text{mat}(\mathbf{x}^i) = \begin{bmatrix} t_i & (\mathbf{d}^i)^T \\ \mathbf{d}^i & t_i \mathbf{I} \end{bmatrix} \quad (27d)$$

$$\mathbf{S}^i = \text{mat}(\mathbf{s}^i) = \begin{bmatrix} 1 & -(\boldsymbol{\lambda}^i)^T \\ -\boldsymbol{\lambda}^i & \mathbf{I} \end{bmatrix} \quad (27e)$$

which are standard notations for conic programming problems. Denoting $\mathbf{X} = \mathbf{diag}(\dots, \mathbf{X}^i, \dots)$ and $\mathbf{S} = \mathbf{diag}(\dots, \mathbf{S}^i, \dots)$ the corresponding global block-diagonal matrices, equation (18b)

can be rewritten as $\mathbf{X}\mathbf{S}\hat{\mathbf{e}} = \mu\hat{\mathbf{e}}$ where $\hat{\mathbf{e}}$ is the global concatenation of N \mathbf{e} vectors. The linearization of (27a) is then rewritten as:

$$\mathbf{S}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{s} = \mathbf{S} \begin{Bmatrix} \Delta\mathbf{t} \\ \Delta\mathbf{d} \end{Bmatrix} + \mathbf{X} \begin{Bmatrix} 0 \\ -\Delta\boldsymbol{\lambda} \end{Bmatrix} \quad (28)$$

We now introduce a symmetrization of the complementarity equations and use one of the most famous called the *Nesterov-Todd scaling*, described in Appendix C, which instead of considering the consistency equation $\mathbf{x}^i \circ \mathbf{s}^i = 0$ replaces it with $\mathbf{v}^i \circ \mathbf{v}^i = 0$ where \mathbf{v}^i is a scaled point depending on \mathbf{x}^i and \mathbf{s}^i and is such that $(\mathbf{x}^i)^T \mathbf{s}^i = \|\mathbf{v}^i\|^2$ (but with $\mathbf{x}^i \circ \mathbf{s}^i \neq \mathbf{v}^i \circ \mathbf{v}^i$). In particular, there exists a symmetric matrix \mathbf{F}^i such that $\mathbf{v}^i = (\mathbf{F}^i)^{-1} \mathbf{s}^i = \mathbf{F}^i \mathbf{x}^i$. The linearized complementarity equation (28) is then replaced by the following:

$$\mathbf{V}\mathbf{F}\Delta\mathbf{x} + \mathbf{V}\mathbf{F}^{-1}\Delta\mathbf{s} = \mathbf{R}_c^k = \mu\hat{\mathbf{e}} - \mathbf{V}^2\hat{\mathbf{e}} \quad (29)$$

where \mathbf{V} (resp. \mathbf{F}) is the block matrix made of each $\mathbf{V}^i = \mathbf{mat}(\mathbf{v}^i)$ (resp. \mathbf{F}^i).

We now rewrite the linearized KKT system (21) as follows:

$$\mathbf{A}\Delta\mathbf{z} = \begin{bmatrix} \mathbf{K} & \mathbf{0} & \tau_0\mathbf{B}^T\mathbf{J} \\ \mathbf{B} & \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}\mathbf{F} & \mathbf{V}\mathbf{F}^{-1} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{x} \\ \Delta\mathbf{s} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_d \\ \mathbf{r}_p \\ \mathbf{R}_c \end{Bmatrix} \quad (30)$$

where we dropped the iteration counter k and with $\mathbf{J} = \mathbf{diag}(\dots, [0 \quad -\mathbf{I}_2], \dots)$ so that $\mathbf{J}\mathbf{s} = -\boldsymbol{\lambda}$. Since \mathbf{V} is an invertible matrix for a strictly feasible primal-dual point (\mathbf{x}, \mathbf{s}) (see Appendix C), the last two rows can be easily condensed, using only pointwise computations at the finite element level, to form the following final reduced system (see Appendix D):

$$\underbrace{\left(\mathbf{K} + \tau_0\mathbf{B}^T(\overline{\mathbf{F}^{-2}})^{-1}\mathbf{B} \right)}_{\mathbf{A}'} \Delta\mathbf{u} = \mathbf{r}_d + \tau_0\mathbf{B}^T(\overline{\mathbf{F}^{-2}})^{-1}(\mathbf{r}_p + \overline{\mathbf{F}^{-1}\mathbf{V}^{-1}\mathbf{R}_c}) = \mathbf{r}' \quad (31a)$$

Therefore, the reduced linear system form amounts only to solving the linear system (31a) $\mathbf{A}'\Delta\mathbf{u} = \mathbf{r}'$ which is still sparse but is now much smaller and symmetric positive definite since $(\overline{\mathbf{F}^{-2}})^{-1}$ is (see (C.18)). The corresponding matrix \mathbf{A}' can be interpreted as the sum of a purely viscous contribution with constant viscosity \mathbf{K} and a locally variable contribution due to the yield stress term $\tau_0\mathbf{B}^T(\overline{\mathbf{F}^{-2}})^{-1}\mathbf{B}$.

Besides, Mehrotra's predictor-corrector scheme can still be applied with this reduced form since the linear system matrix \mathbf{A} , and therefore \mathbf{A}' , does not change between the affine and the corrector step. One just has to adapt the corresponding values of the reduced residual vectors for each step. The quadratic correction term for the second step of Mehrotra's scheme is here given by $\mathbf{F}_i\Delta\mathbf{x}_i^a \circ \mathbf{F}_i^{-1}\Delta\mathbf{s}_i^a$ where $\Delta\mathbf{x}^a, \Delta\mathbf{s}^a$ are the directions solutions to the affine step.

Finally, let us highlight that computing the reduced residual vectors, the reduced matrix \mathbf{A}' or the back-substitutions for the auxiliary variables involves the computation of \mathbf{F}^{-1} , \mathbf{V}^{-1} or $(\overline{\mathbf{F}^{-2}})^{-1}$ which are given by closed-form formulae in Appendix C.

4.3. Summary of the final algorithm

We now summarize the different steps of the final primal-dual interior point algorithm (Algorithm 2) incorporating Mehrotra's predictor-corrector scheme and a reduced linear system form.

Algorithm 2 Predictor-corrector primal-dual interior point algorithm

- 1: Initialization ($k = 0$): choose $\mathbf{z}^0 = (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0})$ and $\epsilon > 0$
 - 2: $(\mathbf{r}_p^0, \mathbf{r}_d^0) = (\mathbf{f}, \mathbf{0})$
 - 3: $\bar{g}^0 = \frac{1}{N} \sum_{i=1}^N (t_i^0 - \boldsymbol{\lambda}_i^0 \cdot \mathbf{d}_i^0) = 1$
 - 4: **while** $\|(\mathbf{r}_p^k, \mathbf{r}_d^k)\| > \epsilon$ or $\bar{g}^k > \epsilon$ **do**
 - 5: compute scaling matrices \mathbf{F}, \mathbf{V} from the current point \mathbf{z}^k
 - 6: form reduced Newton matrix \mathbf{A}' from (31a)
 - 7: form affine residual vector \mathbf{r}'_a corresponding to the reduced version of (24)
 - 8: solve affine step $\mathbf{A}'\Delta\mathbf{u}^a = \mathbf{r}'_a$
 - 9: back-substitute for $(\Delta\mathbf{t}^a, \Delta\mathbf{d}^a, \Delta\boldsymbol{\lambda}^a)$ from (D.1b),(D.3b)
 - 10: compute maximum affine step length α_{max}^a
 - 11: estimate centering parameter γ through (25)
 - 12: set $\mu^{(k)} = \gamma\bar{g}^k$
 - 13: form corrector residual vector \mathbf{r}'_c corresponding to the reduced version of (26)
 - 14: solve corrector step $\mathbf{A}'\Delta\mathbf{u} = \mathbf{r}'_c$ (reuse factorization of \mathbf{A}')
 - 15: back-substitute for $(\Delta\mathbf{t}, \Delta\mathbf{d}, \Delta\boldsymbol{\lambda})$ from (D.1b),(D.3b)
 - 16: compute maximum step length α
 - 17: update variables $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha\Delta\mathbf{z}$
 - 18: compute new complementarity gap \bar{g}^{k+1}
 - 19: **if** $k > N_{iter,max}$ or $\alpha \leq \epsilon$ **then break** "Algorithm failed."
 - 20: **end if**
 - 21: $k \leftarrow k + 1$
 - 22: **end while**
-

4.4. Numerical implementation

The numerical implementation of the proposed IPM algorithm for viscoplastic fluid flows has been realized using FEniCS [32] as regards the finite-element discretization and building of the KKT linear systems. Thanks to the flexibility offered by FEniCS in the choice of the linear system solver, we used the direct solver MUMPS [33, 34] which enables to solve large sparse linear systems. Computations have been realized on a Intel Core i7-4600U (2.1 GHz).

5. Comments on some extensions

5.1. Two and three-dimensional flows

Let us quickly consider here the case of two or three-dimensional flows, the overall steps of the IPM are exactly the same as for the antiplane case with only small differences mainly

due to the incompressibility constraint. Indeed, the solution of a steady viscoplastic flow in a two or three-dimensional domain Ω can be obtained from the following minimum principle:

$$\min_{\underline{u}} \int_{\Omega} \left(\eta \underline{d} : \underline{d} + \tau_0 \sqrt{2\underline{d} : \underline{d}} \right) - \int_{\Omega} \underline{f} \cdot \underline{u} \quad (32a)$$

$$\text{s.t.} \quad \text{div } \underline{u} = 0 \quad (32b)$$

where $\underline{d} = \frac{\nabla^s \underline{u}}{2} = (\underline{\nabla} \underline{u} + \underline{\nabla} \underline{u}^T)/2$ is the strain rate tensor. Following the same procedure as in Section (3), this minimum principle gives rise to the following optimality conditions:

Find $(\underline{u}, p, t, \underline{\lambda})$ such that:

$$\int_{\Omega} (2\eta \underline{\nabla}^s \underline{u} + \tau_0 \underline{\lambda}) : \underline{\nabla}^s \underline{v} + p \text{div } \underline{v} = \int_{\Omega} \underline{f} \cdot \underline{v} \quad \forall \underline{v} \in \mathcal{V} \quad (33a)$$

$$\text{div } \underline{u} = 0 \quad (33b)$$

$$(t, \mathbf{d}) \circ (1, -\boldsymbol{\lambda}) = (0, \mathbf{0}) \quad (33c)$$

$$(t, \mathbf{d}) \in \mathcal{Q}^m, \quad (1, -\boldsymbol{\lambda}) \in \mathcal{Q}^m \quad (33d)$$

with $m = 4$ in 2D, $m = 7$ in 3D² and where we introduced a scaled vectorial representation of the strain rate tensor components d_{ij} as follows:

$$\left\{ \begin{array}{l} \mathbf{d} = \mathbf{B}\mathbf{u} = \begin{cases} \sqrt{2}d_{xx} \\ \sqrt{2}d_{yy} \\ 2d_{xy} \end{cases} \quad \text{in 2D} \\ \mathbf{d} = \mathbf{B}\mathbf{u} = \begin{cases} \sqrt{2}d_{xx} \\ \sqrt{2}d_{yy} \\ \sqrt{2}d_{zz} \\ 2d_{xy} \\ 2d_{xz} \\ 2d_{yz} \end{cases} \quad \text{in 3D} \end{array} \right. \quad (34)$$

and similarly for the tensor $\underline{\lambda}$. The discretized KKT conditions of the previous optimality conditions are given by:

Find $(\mathbf{u}, \mathbf{p}, t, \mathbf{d}, \boldsymbol{\lambda})$ such that:

$$\mathbf{K}\mathbf{u} + \mathbf{D}^T \mathbf{p} + \tau_0 \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{f} \quad (35a)$$

$$\mathbf{D}\mathbf{u} = \mathbf{0} \quad (35b)$$

$$\mathbf{B}\mathbf{u} - \mathbf{d} = \mathbf{0} \quad (35c)$$

$$(t_i, \mathbf{d}_i) \circ (1, -\boldsymbol{\lambda}_i) = \mathbf{0} \quad \forall i \quad (35d)$$

$$(t_i, \mathbf{d}_i) \in \mathcal{Q}^m, \quad (1, -\boldsymbol{\lambda}_i) \in \mathcal{Q}^m \quad (35e)$$

²It is possible to reduce the size of the conic constraint by taking the incompressibility condition into account e.g. in 2D, since $\text{tr } \underline{d} = 0$, $d_{xx} = -d_{yy}$ and $\|\mathbf{d}\| \leq t \iff \sqrt{2d_{xx}^2 + 2d_{yy}^2 + 4d_{xy}^2} = \sqrt{4d_{xx}^2 + 4d_{xy}^2} \leq t \iff (t, \mathbf{D}) \in \mathcal{Q}^3$ with $\mathbf{D} = (2d_{xx}, 2d_{xy})$.

where \mathbf{D} represents the divergence operator and \mathbf{p} is the pressure-like Lagrange multiplier associated with the incompressibility condition. We can then observe that the top-left block of the Newton matrix \mathbf{A} corresponding to the viscosity stiffness matrix \mathbf{K} is now replaced by a mixed displacement-pressure system of the form $\begin{bmatrix} \mathbf{K} & \mathbf{D}^T \\ \mathbf{D} & \mathbf{0} \end{bmatrix}$ with a new primal residual vector $\mathbf{r}_{p'}$ associated with the incompressibility constraint. Similarly, for the reduced linear system formulation described in section 4.2, the procedure is exactly the same except that the pressure variable \mathbf{p} cannot be eliminated so that the final system has the following symmetric mixed form:

$$\begin{bmatrix} \mathbf{K} + \tau_0 \mathbf{B}^T (\overline{\mathbf{F}}^{-2})^{-1} \mathbf{B} & \mathbf{D}^T \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}' \\ \mathbf{r}_{p'} \end{Bmatrix} \quad (36)$$

Finally, let us also mention that a traditional mixed Taylor-Hood $P_2 - P_1$ approximation for (\underline{u}, p) is chosen while all strain/stress-like variables $(t, \underline{d}, \underline{\lambda})$ are discretized using discontinuous linear elements. As a consequence each conic constraint is verified at the three vertices of each triangular element which is enough to satisfy it everywhere from convexity arguments.

5.2. Extension to other viscoplastic models

It was previously mentioned that second-order cone programs need to be formulated in a specific standard format to be solved by the interior-point solver MOSEK. In particular, the objective function has to be reformulated as a linear objective function by introducing additional variables and second-order cone constraints. If this reformulation is quite straightforward in the case of a quadratic viscous potential for a Bingham fluid, it is less obvious in the Herschel-Bulkley case. A specific procedure has been described in [12] to tackle the Herschel-Bulkley case with a rational power-law exponent. In addition to increasing the number of optimization variables, one drawback of such an approach is that the form of the reformulation is strongly dependent on the value of the exponent and has to be performed on a case-by-case basis.

In the proposed formulation, the quadratic viscous part of the potential has not been reformulated using cone constraints but kept as such in the objective function. It then appears through its constant Hessian matrix \mathbf{K} when forming the linearized KKT system. Therefore, we propose to deal with the Herschel-Bulkley case in the exact same way. The \mathbf{K} stiffness matrix will then be replaced, at each iteration, by the current tangent stiffness matrix related to the Herschel-Bulkley power-law behavior. The iterative process will then combine a traditional Newton method on the nonlinear viscous term as well as on the conic constraints. We do not expect a strong increase in the number of iterations to converge compared to the Bingham case since the power-law viscous behavior is smooth but this remains to be tested in the future.

More generally, the same splitting strategy could be used to any yield stress fluid constitutive models (Casson fluids for example) which can be split into a smooth and yield stress non-smooth part.

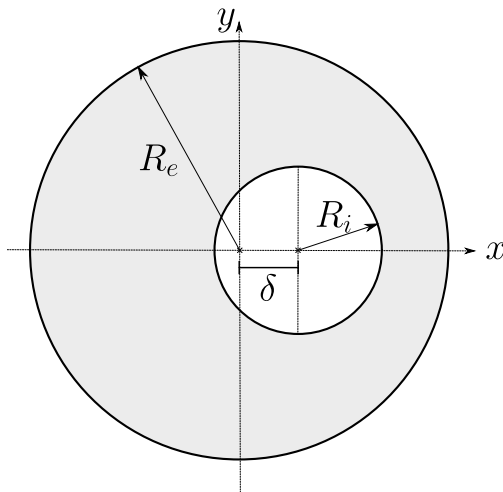


Figure 2: Geometry of the eccentric annulus channel section.

6. Illustrative applications

6.1. Channel flow in an eccentric annulus

As a first validation of the proposed approach, we consider the problem of the antiplane flow in a cylindrical channel, the section of which consists of an eccentric annulus [35, 36]. The outer circle has a normalized radius $R_e = 1$ while we fixed the inner circle radius to $R_i = 0.4$ (Figure 2). The center of the inner circle is located at a distance δ from that of the outer circle on the middle plane $y = 0$. For symmetry reasons, only the upper half of the section $y \geq 0$ has been modelled. The flow is driven by a uniform unit pressure gradient $f = 1$, the fluid viscosity is also taken as unity $\eta = 1$ and a no-slip velocity condition is assumed on both circle boundaries.

Except stated otherwise, we used the predictor-corrector interior-point algorithm with a reduced linear system (Algorithm 2). To assess its efficiency, its computational performance has been compared to both a standard AL approach and its accelerated version as described in section 2, the following results have been obtained using the ADMM algorithm, although the performance of AMA is very similar. Choosing the augmentation parameter r is a delicate point of AL algorithms and an optimal value is often problem-dependent. In [10], a link between such algorithms and accelerated proximal algorithms has been drawn provided that r is chosen as the Lipschitz constant of $G(x)$ in (3), which corresponds to $r = \eta$ in the present case. This choice also ensures convergence of the accelerated algorithm. For this reason, this choice has been retained in the subsequent simulations for both standard and accelerated variants.

Since the IPM is a second-order Newton-like approach, the number of iterations is expected to be orders of magnitude lower than first-order approaches like AL but also more expensive to compute. For this reason, the comparison between these approaches will be made in terms of CPU time. As regards both AL approaches, one attractive feature is that at each iteration, the same Stokes-like problem needs to be solved with different right-hand side

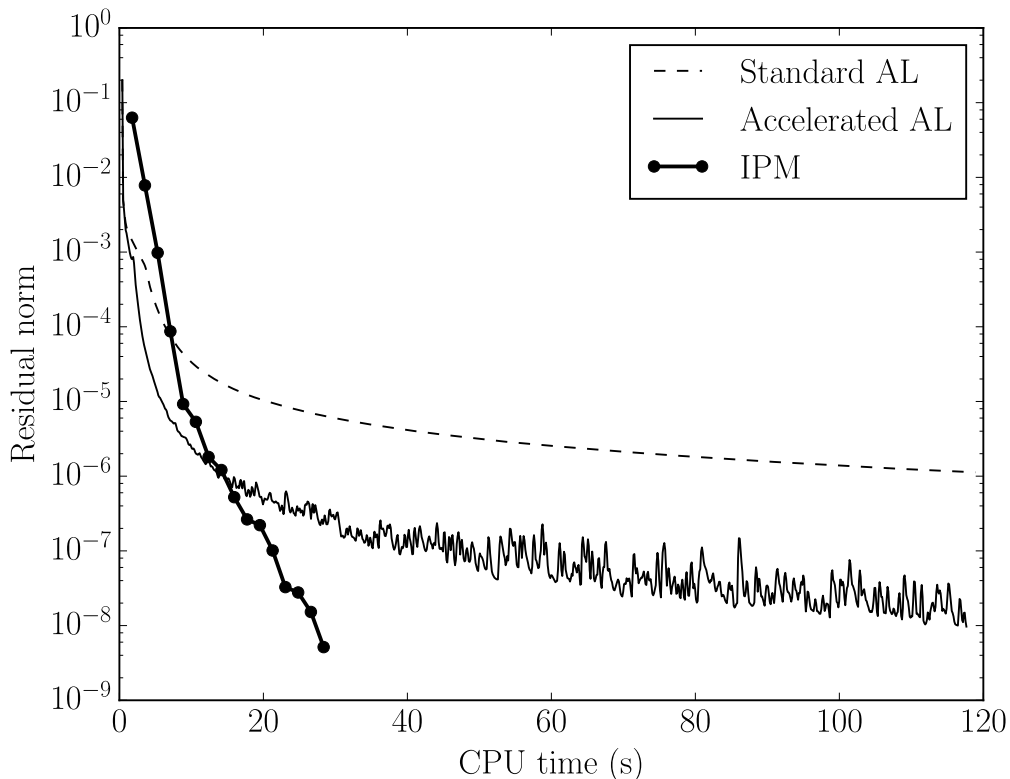


Figure 3: Comparison of CPU time for the annulus problem ($\delta = 0.04$) between the standard Augmented Lagrangian approach ($N_{iter} > 1000$: did not converge to the prescribed tolerance), its accelerated version ($N_{iter} = 986$) and the proposed interior-point method ($N_{iter} = 16$). The accelerated AL exhibits a faster convergence rate than the standard AL but is still outperformed in terms of computational cost by the proposed IPM approach.

members. In our FEniCS implementation, we fully exploit this feature by factorizing the corresponding matrix once and reusing it at each iteration, similarly to what is done for the predictor-corrector IP algorithm. The cost of one AL iteration is therefore much smaller. The residual measure for AL is given by $\|\nabla u^k - \underline{d}^k\|$ whereas for the IPM the evolution of the average complementarity gap \bar{g}^k is reported. The results are represented in Figure 3 for the annulus problem ($\delta = 0.04$) with a fine mesh ($N_e = 66,077$ elements) and for a yield stress $\tau_0 = 0.1$. It can clearly be observed that the accelerated AL algorithm exhibits a faster convergence than the standard AL algorithm. It reaches the tolerance of 10^{-8} in 986 iterations whereas the standard AL algorithm fails to reduce the residual below the prescribed tolerance in less than 1000 iterations. However, the IP algorithm converges to the optimal solution in less than 30s i.e. approximately 4 times faster than the accelerated AL. The average cost of one AL (either standard or accelerated) iteration is around 0.12s against 1.77s for the IPM. CPU times for other mesh sizes have also been reported in Table 1. It can be observed that the IPM is 4 to 6 times faster than the accelerated AL for all mesh sizes.

Mesh size	Accelerated AL (s)	IPM (s)	Speed-up factor
$N_e = 4\,092$	13.9	2.2	6.3
$N_e = 16\,492$	34.3	6.4	5.4
$N_e = 66\,077$	117.6	28.4	4.1
$N_e = 264\,230$	758.6	120.3	6.3

Table 1: CPU times for different meshes and speed-up factor of the IPM over accelerated AL. Standard AL never converged in less than 1000 iterations.

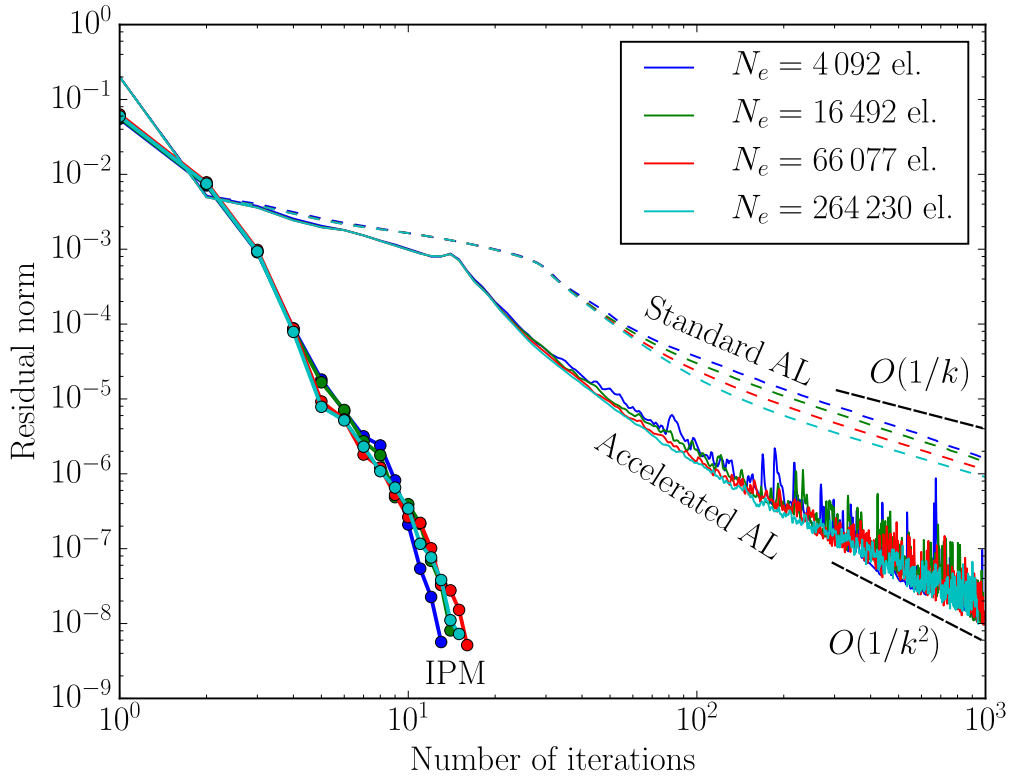


Figure 4: The mesh size N_e does not really change the number of iterations for convergence for IPM as well as AL. The accelerated AL version exhibits a $O(1/k^2)$ convergence rate compared to the standard $O(1/k)$ rate. The IPM requires, in practice, only a few tens of iterations to converge.

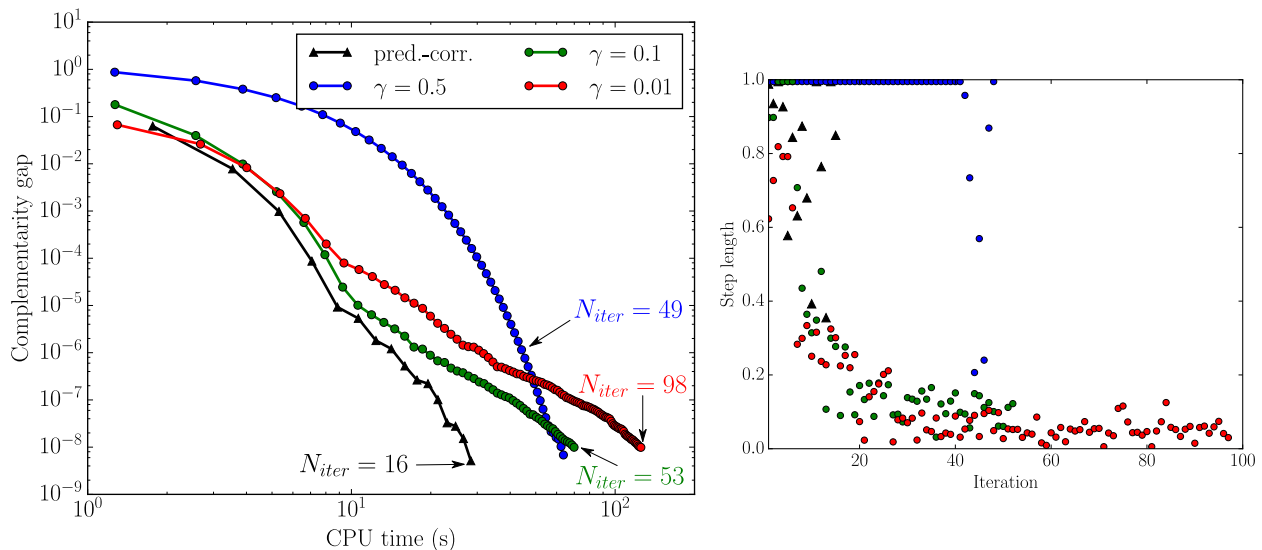


Figure 5: Left: Evolution of the complementarity gap \bar{g} for the annulus problem ($\delta = 0.04$) for the simple IPM algorithm with a constant centering parameter γ and for the adaptive predictor-corrector variant. The number of iterations and the total computing times are much lower with the adaptive variant than for the constant γ variant. Right: Step length of each iteration for the corresponding cases.

The influence of the mesh size has been investigated in Figure 4 and shows that its influence with respect to the number of iterations required to reach the prescribed tolerance is quite weak for the IPM. In particular, it remains of the order of a few tens of iterations as generally reported in practice for state-of-the-art interior-point solvers. As regards AL, the mesh size has some influence on the standard version whereas it is more difficult to assess on the accelerated version since it is not a descent algorithm. We also verify that both AL algorithms reached their theoretical convergence rate.

In Figure 5, the simple variant of the IPM with a constant centering parameter γ (Algorithm 1) has been compared to Algorithm 2. For the comparison to be fair, steps 6 and 7 in Algorithm 1 are modified to solve the reduced linear system $\mathbf{A}'\Delta\mathbf{u} = \mathbf{r}'$ as described in section 4.2 instead of the global KKT system, we then only investigate the influence of the adaptive choice of the centering parameter through the predictor-corrector scheme. It can clearly be observed that the predictor-corrector variant converges with much fewer iterations (16 against 50 to 100 depending on γ) but also in much less computation time than Algorithm 1. The average computing cost of one iteration for Algorithm 1 is around 1.3 s for this specific example whereas the cost of one iteration of Algorithm 2 is around 1.77 s. Hence, the cost of the corrector step in Algorithm 2 induces an overhead of only 36% per iteration but with an important saving in the number of iterations. Besides, Figure 5 also shows that an optimal value of γ may be difficult to find since the step length of each iteration can be very small for small values of γ despite a fast reduction of the complementarity gap. Conversely for larger values of γ , the complementarity gap is slow to decrease despite large

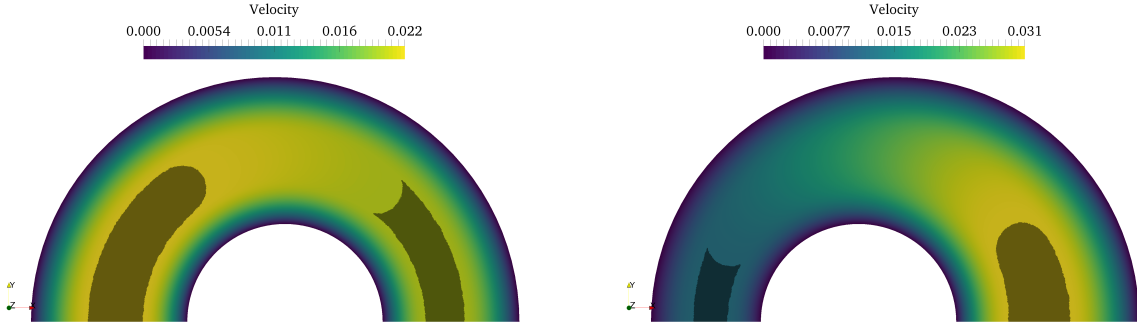


Figure 6: Optimal velocity field and unyielded regions (in gray) for the annulus problem with $\delta = 0.04$ (left) and $\delta = -0.15$ (right). The unyielded region shape and location are consistent with those obtained with AL approaches and results from [35, 36].

step lengths. On the contrary, the adaptive heuristic (25) works very well in general with a fast reduction of the gap and sufficiently large step lengths so that Algorithm 2 does not require fine tuning of the algorithm parameters.

In [37], it was mentioned that *”replicating the yield surface positions found in the computations of [35] is a worthy challenge for the hardened regulariser!”*. We therefore checked that the IPM is able to replicate such yield surfaces as shown in Figure 6 (see the results reported in [35, 36]), notice in particular the sharp re-entrant corner which are difficult to obtain with regularized models. These yield surfaces are identified as the regions for which $\|\underline{\sigma}\| \leq \tau_0$ where we recall that $\underline{\sigma} = \eta \nabla u + \tau_0 \underline{\lambda}$, thus combining information from both the primal variable u and the dual variable $\underline{\lambda}$. At convergence, the strain rate norm inside the yield surface is below 10^{-8} which is the order of the prescribed convergence tolerance. One could then argue that the strain rate is never zero in unyielded zones but it could easily be post-processed by projecting the strain rate to zero in unyielded regions. The difference between these two strain rate states would induce only a minor difference in the residual norms and would still satisfy the convergence tolerance. Hence, from a user perspective, the IPM does not seem to exhibit any of the typical difficulties of regularized approaches as the complementarity gap (and hence the barrier parameter μ) has been driven to zero, within the prescribed numerical tolerance.

For the $\delta = -0.15$ case, we computed the flow rate $Q = \int_{\Omega} u dS$ of the optimal velocity field. The results are reported in Figure 7 in terms of a non-dimensional flow rate $Q^* = Q/Q_0$ where Q_0 corresponds to the flow rate of a Newtonian fluid in a concentric annulus $\delta = 0$ (see [35]) and for varying pressure gradient f and yield stress τ_0 . In the Newtonian case, we obtain a linear relationship as expected whereas for non-zero yield stress, no flow occurs below a critical pressure gradient, the value of which increases with the yield stress. The computed flow rate Q^* in these situations is not strictly zero but is also around the order of the convergence tolerance. For high pressure gradients, the response is again linear with

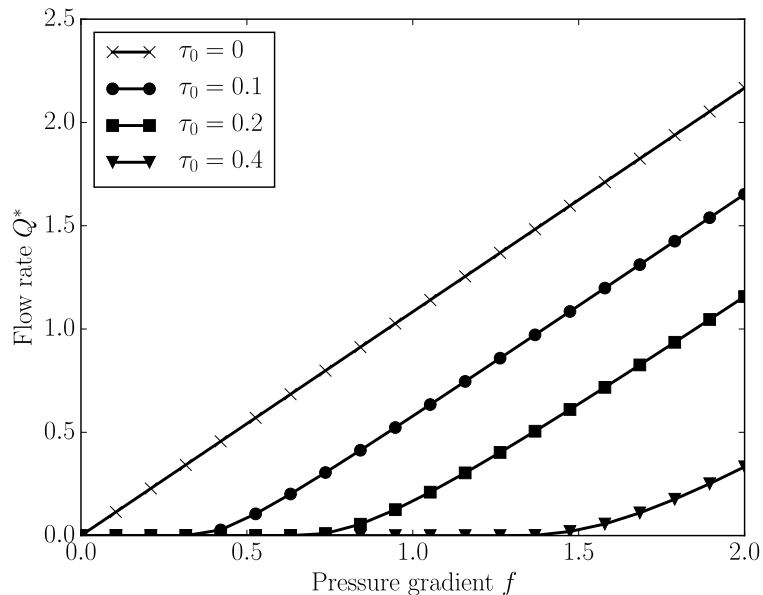


Figure 7: Non-dimensional flow rate $Q^* = Q/Q_0$ as a function of the applied pressure gradient f for different yield stress τ_0 . No flow occurs below a critical pressure gradient whereas the response becomes linear at high pressure gradients with the same slope as the Newtonian case $\tau_0 = 0$.

the same slope as for the Newtonian case.

6.2. Two-dimensional flow: the lid-driven cavity

We consider here the typical benchmark of the lid-driven cavity problem: a unit-square cavity with an applied velocity $\underline{u} = \underline{e}_x$ on the top surface $y = 1$, the other boundaries being subjected to a no-slip velocity condition. We define the Bingham number as $Bi = \tau_0 L / \eta U$, with here $U = L = 1$, as the non-dimensional number measuring the influence of yield stress effects over viscous effects. The initial pressure field for the starting point of the interior-point algorithm has been chosen to be zero whereas the velocity field is initialized to zero for all degrees of freedom except those located on the top surface on which the imposed velocity is prescribed. During the iterative process, the Newton directions $\Delta \mathbf{u}$ are then sought as corrections kinematically admissible to zero i.e. $\Delta \mathbf{u}_i = 0$ everywhere on the cavity boundary³. For this initialization, the strain rate primal residual \mathbf{r}_p is non zero since the initial zero strain rate is not compatible with this initial velocity field. In fact, all residuals are here non zero at the beginning.

We first report the convergence history of the different residuals with a structured discretization of element size $h = 1/50$ (Figure 8). It can be observed that all residuals, either

³Note that it would be possible instead to add the imposed velocity condition $\underline{u} = \underline{e}_x$ as an additional linear constraint to the optimization problem, the residual of which would also converge to zero during the IPM iterative process.

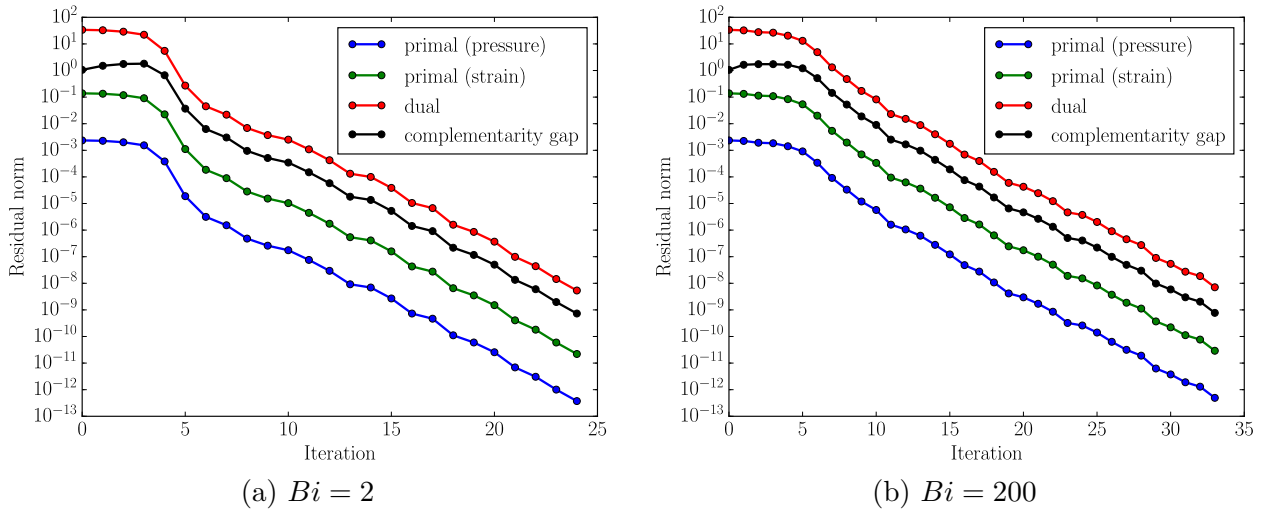


Figure 8: Convergence of the different IPM residuals for the lid-driven cavity problem ($h = 1/50$) and two different Bingham numbers. Each type of residual clearly converges at the same rate. The number of iterations to convergence slightly increases for high Bingham numbers.

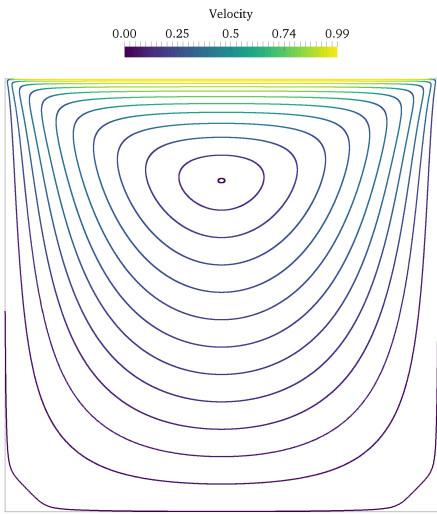
the primal (\mathbf{r}_p for the strain rate and $\mathbf{r}_{p'}$ for the pressure), the dual \mathbf{r}_d or the average complementarity gap \bar{g} , converge to zero at the same rate as previously remarked. When increasing the Bingham number to a high value, we observe a smaller overall convergence rate, thus needing a larger number of iterations to reach the prescribed tolerance. However, this increase remains moderate, the IPM needing only a few tens of iterations to converge. The same trend is observed when refining the mesh. On the contrary, the number of iterations for AL algorithms seems more dependent on the increase of the Bingham number.

During the iteration process, we observe that the iso-stress contour lines converge to a well-defined location. Therefore, no problem regarding the lack of point-wise convergence for stresses seem to appear with the IPM contrary to what has been observed for regularized models [37, 38]. The location of the yielded regions are reported in Figure 9 for different Bingham numbers and are consistent with results from the literature (see for instance [10, 39]).

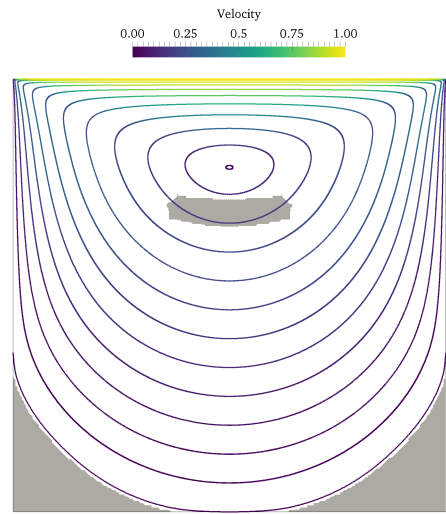
6.3. Three-dimensional flow in a porous medium

As a final illustration, we investigate a three-dimensional flow around a fixed body-centered cubic stacking of rigid spheres representing a porous medium. The unit cell of this periodic structure is represented in Figure 10. Taking into account the various symmetries, only one eighth of the model is considered with appropriate boundary conditions. The flow is driven by a uniform downwards pressure gradient $\underline{f} = -f\mathbf{e}_z$ and no-slip velocity conditions are assumed on the sphere boundaries. The yield stress is here $\tau_0 = 10$, the viscosity is still $\eta = 1$ and the sphere radius has been taken as $R = 0.4$.

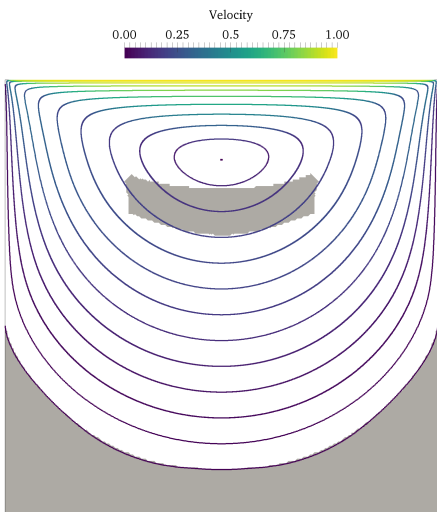
The computed velocity field along with unyielded points have been represented in Figure 11 for $f = 140$ and in Figure 12 for $f = 300$. The first case corresponds to a situation for



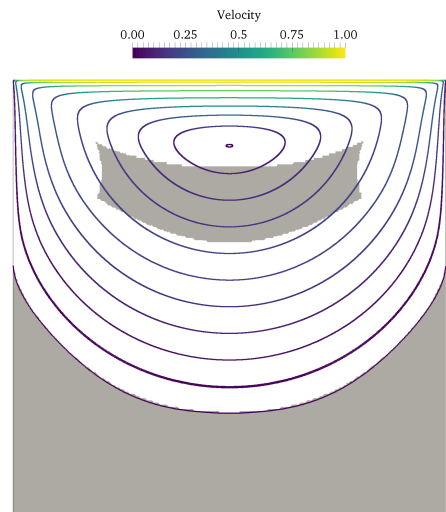
(a) $Bi = 0$



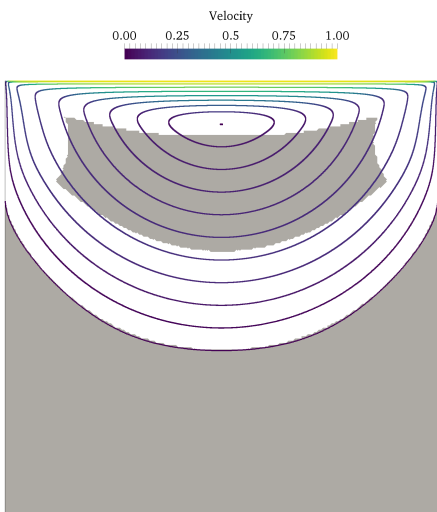
(b) $Bi = 1$



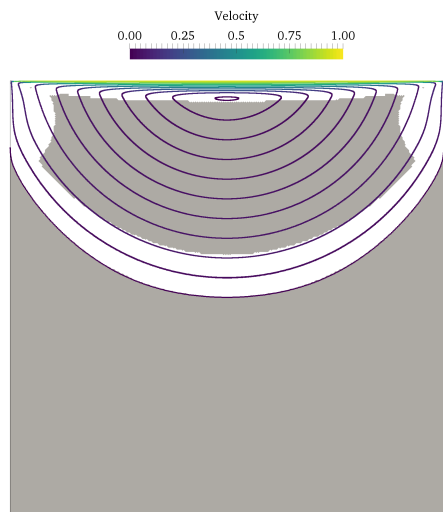
(c) $Bi = 2$



(d) $Bi = 5$



(e) $Bi = 20$



(f) $Bi = 200$

Figure 9: Velocity streamlines and unyielded elements (in gray) for the lid-driven cavity problem ($h = 1/200$) with varying Bingham numbers.

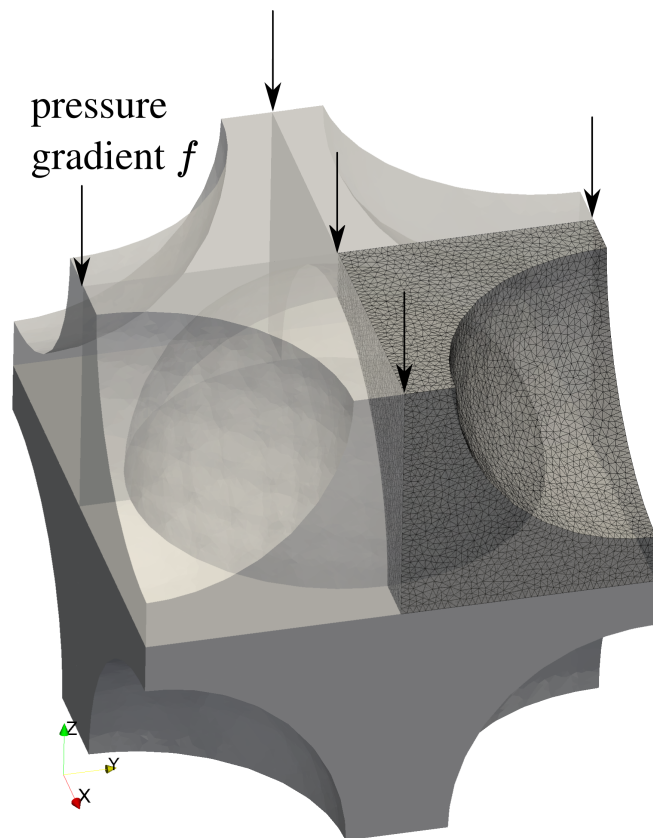


Figure 10: Three-dimensional unit cell of the porous medium ($R = 0.4$) along with the computational domain taking into account the various symmetries.

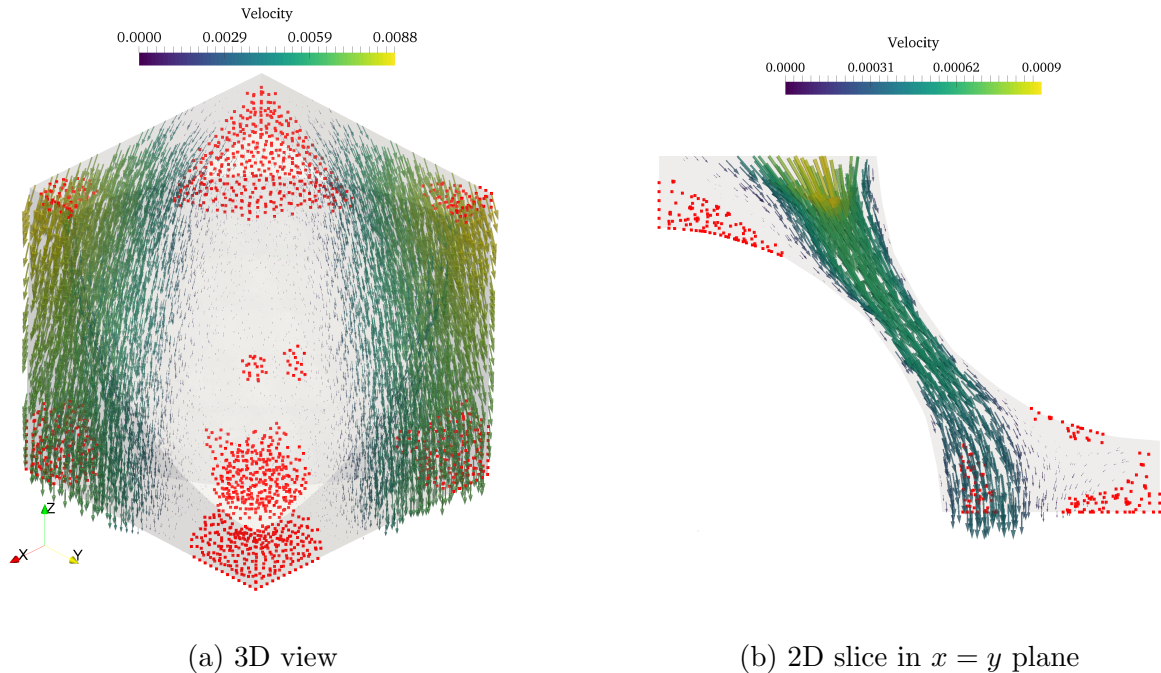


Figure 11: Velocity field for a pressure gradient $f = 140$ slightly above the critical value (red dots denote unyielded regions).

which the pressure gradient is slightly above the critical value for which flow occurs whereas in the second case, the pressure gradient is more than twice its critical value. It can be noted that, as expected, the flow is mainly concentrated along paths which are far from the spheres and that many points near the obstacles are still unyielded. For the larger pressure gradient, the flow spreads in the central part of the computational domain around the obstacles and less unyielded points are observed. When looking at the flow along a 2D slice of equation $x = y$, in the most tortuous part of the unit cell, we also observe a secondary flow at the bottom of the computational domain for the larger pressure gradient. The evolution of the downwards flow rate $Q_z = \int_{\Omega} -u_z dV$ over the computational domain has been represented in Figure 13 as well as the horizontal flow rate $Q_x = \int_{\Omega} u_x dV$ (also equal to Q_y because of symmetry). We obtain the same overall behavior as in Figure 7 for the annulus channel, that is a non-flowing part below a critical pressure gradient and an almost linear response at high velocity gradients.

As regards computing times, for a mesh made of approximately 50 000 tetrahedra, the complete optimization required about 15-20 minutes. For a mesh made of 110 000 tetrahedra, the computing time increased to slightly more than 2 hours.

7. Conclusions and perspectives

In this work, we tackled the problem of computing viscoplastic fluid flows by solving the associated velocity minimum principle, reformulated using second-order cone constraints.

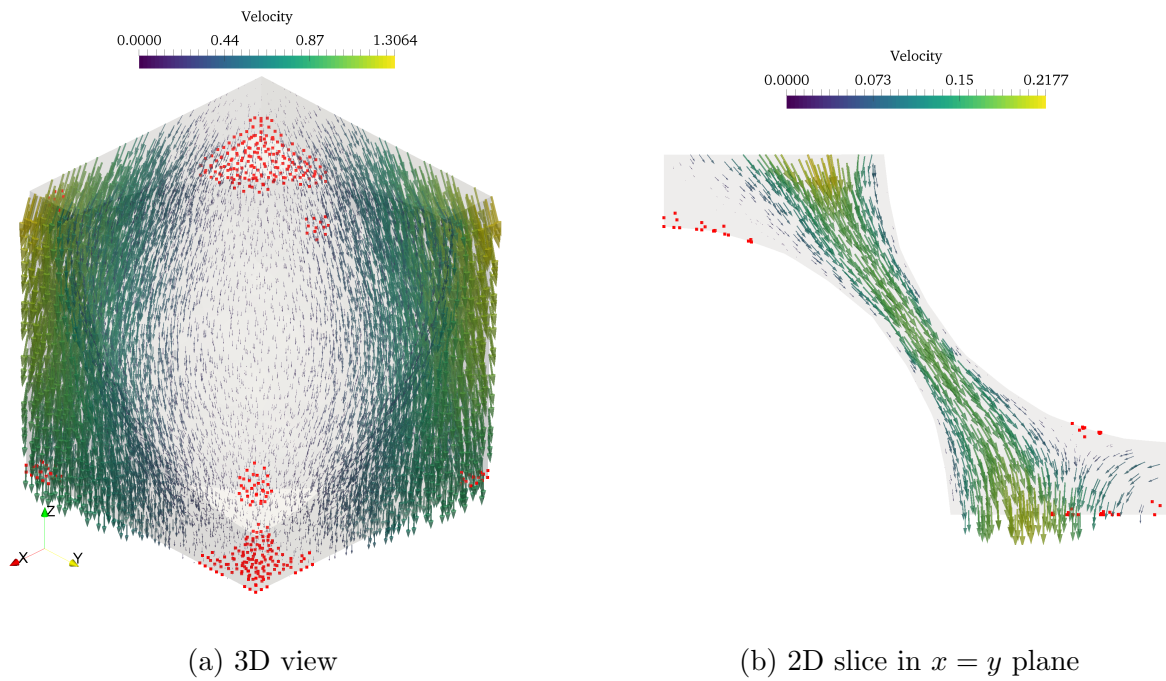


Figure 12: Velocity field for a pressure gradient $f = 300$ much higher than the critical value (red dots denote unyielded regions).

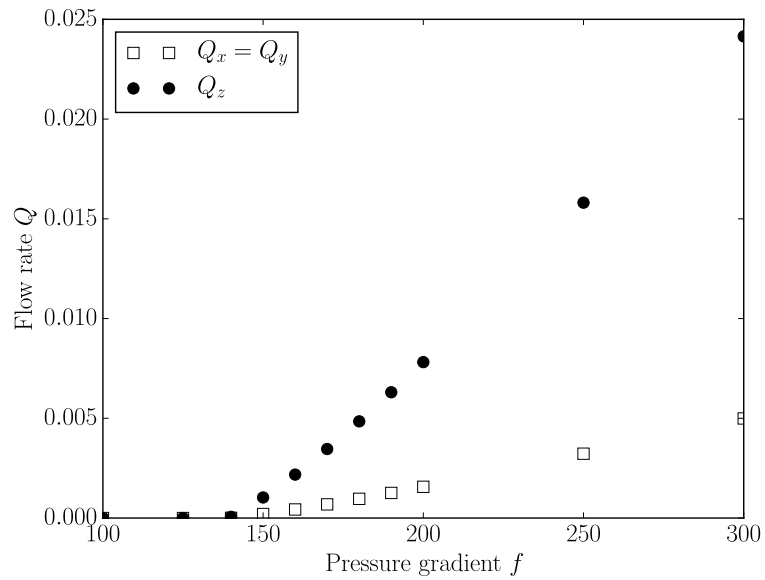


Figure 13: Vertical and horizontal flow rates over the computational domain as a function of the pressure gradient.

We presented an interior-point algorithm to efficiently solve the associated convex optimization problem, without relying on black-box commercial solvers as proposed in [12]. In particular, it enables to reduce the number of auxiliary variables needed to formulate the problem in a standard form. It also paves the way to tackling more complex viscoplastic models by taking directly into account the smooth non-linear viscous part in the objective problem, which was not possible in the commercial software used previously. The detailed steps of the primal-dual IPM have been described and more advanced aspects such as the predictor-corrector scheme and the reduced linear system form ensuring a good efficiency have been discussed. The illustration of the algorithm on various numerical examples ranging from channel flows to 2D and 3D flows enabled to show that:

- the proposed IPM algorithm is very efficient in terms of total computing time compared to standard as well as accelerated AL algorithms. The typical number of iterations is moderate as usually reported for such methods. It seems also less sensitive to mesh size and Bingham number than AL approaches;
- although it is difficult to compare it precisely, the total computing time of our home-made IPM algorithm seems also quite comparable to what we experienced using the commercial IPM solver MOSEK on similar problems;
- the yield surfaces obtained using this method are in agreement with those obtained using AL approaches and results from the literature;
- although the IPM can be interpreted as a Newton method on a regularized problem with a continuation of the regularization parameter towards zero, we did not observe any of the reported problems of regularization approaches with the IPM. From a user perspective, the obtained solution is close to the one corresponding to the original unperturbed problem within the prescribed numerical tolerance.

Although the efficiency of IPM in the context of viscoplastic flows is very encouraging, some work is still needed to reach large-scale three-dimensional simulations. The main bottleneck that needs to be tackled first to achieve this objective concerns the linear solver for the Newton system. Although MUMPS is an efficient direct solver, which enabled to solve moderate 3D problems on a standard computer, and can be used in parallel, the memory cost of the LU decomposition will be prohibitive for very large systems. The use of iterative solvers instead of direct solvers for the Newton system will however need the development of good preconditioners, the main problem being that the condition number of \mathbf{A}' deteriorates when approaching the solution (\mathbf{x} and \mathbf{s} are approaching the boundary of the second-order cone). Recent works regarding preconditioning techniques for viscoplastic flows can be found in [40, 41].

In the present work, only steady-state flows have been considered. The computation of time-dependent problems is, in theory, quite immediate as the time-discretization adds a quadratic "mass" term in the objective function and changes the loading vector by taking

into account inertial terms from the last iteration. Time-dependent problems therefore require solving successively similar problems. However, it is known that IPM cannot easily take advantage of a previously known solution as a good initial guess ("warm-start") as it is too close from the feasible region boundary and far from the new central path. Finding good warm-starting strategies for IPM is still an active research topic [42–44].

Finally, adaptive mesh refinement strategies have been demonstrated to increase the accuracy of the computed solutions by resolving the transition between flowing and non-flowing regions [7, 45]. The coupling of IPM with auto-adaptive meshes should not pose any difficulty.

Appendix A. Quadratic second-order cone programming

Let us consider here the following quadratic second-order cone program:

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{K} \end{aligned} \tag{A.1}$$

where \mathbf{P} is symmetric positive definite and \mathcal{K} is a tensorial product of cones $\mathbf{x} \in \mathcal{K} \Leftrightarrow \mathbf{x}_i \in \mathcal{K}_i$ with \mathcal{K}_i being either:

- the full space of dimension n_i : \mathbb{R}^{n_i} (i.e. \mathbf{x}_i is a vector of free variables)
- the positive orthant: $\mathbb{R}_+^{n_i}$
- the Lorentz second-order cone: $\mathcal{Q}^{n_i} = \{\mathbf{z} = (z^0, \bar{\mathbf{z}}) \in \mathbb{R} \times \mathbb{R}^{n_i-1} \text{ s.t. } z^0 \geq \|\bar{\mathbf{z}}\|\}$

This problem is not a standard SOCP program because of the quadratic objective term, it can however be reformulated as a problem involving a linear objective function only by introducing auxiliary variables and additional conic constraints [22, 29].

The Lagrangian \mathcal{L} of the previous program is given by:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{c}^T \mathbf{x} - \mathbf{y}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) - \mathbf{s}^T \mathbf{x} \tag{A.2}$$

and the corresponding optimality conditions by [46]:

$$\partial_{\mathbf{x}} \mathcal{L} = \mathbf{P} \mathbf{x} + \mathbf{c} - \mathbf{A}^T \mathbf{y} - \mathbf{s} = \mathbf{0} \tag{A.3a}$$

$$\partial_{\mathbf{y}} \mathcal{L} = \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0} \tag{A.3b}$$

$$\mathbf{x} \in \mathcal{K}, \mathbf{s} \in \mathcal{K}^* \tag{A.3c}$$

$$\mathbf{x}^T \mathbf{s} = 0 \tag{A.3d}$$

where \mathcal{K}^* is the tensorial product of the dual cones \mathcal{K}_i^* . Note that $\mathcal{K}_i^* = \{\mathbf{0}\}$ if $\mathcal{K}_i = \mathbb{R}^{n_i}$, for the two other cases, the cones are self-dual $\mathcal{K}_i^* = \mathcal{K}_i$.

If \mathcal{K}_i are Lorentz cones, the complementary slackness condition $\mathbf{x}^T \mathbf{s} = 0$ is equivalent to $\mathbf{x}_i \circ \mathbf{s}_i = \mathbf{0} \forall i$ where :

$$\mathbf{x} \circ \mathbf{s} = \begin{cases} x^0 s^0 + \bar{\mathbf{x}}^T \bar{\mathbf{s}} \\ x^0 \bar{\mathbf{s}} + s^0 \bar{\mathbf{x}} \end{cases} \quad (\text{A.4})$$

with $\mathbf{x} = (x^0, \bar{\mathbf{x}})$, the $\bar{\mathbf{x}}$ notation representing the tail of a vector \star i.e. the vector obtained by removing the first element (here noted x^0).

Appendix B. Results on the solutions to the Newton equations

We consider the solutions to the following Newton equations with a residual vector parametrized by $\gamma, \kappa \in [0; 1]$:

$$\begin{bmatrix} \mathbf{K} & \mathbf{0} & \mathbf{0} & \tau_0 \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{C}_\lambda^k & -\mathbf{C}_d^k \\ \mathbf{0} & -\text{diag}(\boldsymbol{\lambda}^k) & \mathbf{I} & -\mathbf{C}_t^k \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{t} \\ \Delta \mathbf{d} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \kappa(\mathbf{f} - \mathbf{K}\mathbf{u}^k - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^k) \\ \kappa(\mathbf{d}^k - \mathbf{B}\mathbf{u}^k) \\ \gamma \bar{g}^k - \mathbf{t}^k + \mathbf{C}_\lambda^k \mathbf{d}^k \\ \mathbf{C}_t^k \boldsymbol{\lambda}^k - \mathbf{d}^k \end{bmatrix} \quad (\text{B.1})$$

where we recall that \bar{g}^k is the average complementarity gap given by $\bar{g}^k = \frac{1}{N} \sum t_i^k - \boldsymbol{\lambda}_i^k \cdot \mathbf{d}_i^k$. We then have the following result:

Lemma Appendix B.1. *The directions $\Delta \mathbf{z}$ solutions to (B.1) are such that:*

$$\mathbf{f} - \mathbf{K}\mathbf{u}^+ - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^+ = (1 - \alpha \kappa)(\mathbf{f} - \mathbf{K}\mathbf{u}^k - \tau_0 \mathbf{B}^T \boldsymbol{\lambda}^k) \quad (\text{B.2a})$$

$$\mathbf{d}^+ - \mathbf{B}\mathbf{u}^+ = (1 - \alpha \kappa)(\mathbf{d}^k - \mathbf{B}\mathbf{u}^k) \quad (\text{B.2b})$$

$$\Delta \mathbf{d}_i^T \Delta \boldsymbol{\lambda}_i = \kappa(1 - \gamma - \kappa) N \bar{g}^k \quad (\text{B.2c})$$

$$\bar{g}^+ = (1 - \alpha(1 - \gamma) + \alpha^2 \kappa(1 - \gamma - \kappa)) \bar{g}^k \quad (\text{B.2d})$$

where $\mathbf{z}^+ = \mathbf{z}^k + \alpha \Delta \mathbf{z}$ and \bar{g}^+ is the new average complementarity gap associated with \mathbf{z}^+ .

Proof. The first two relations are easily obtained due to the linearity of the associated residuals. Besides, (B.2d) follows directly from (B.2c) and the third line of (B.1). Finally the proof to (B.2c) can be easily adapted from the one found in [47]. \square

Let us remark that if $\kappa = 1$, (B.1) reduces to (21). From the results of the previous lemma, we observe that the primal and dual residuals decrease by $1 - \alpha$ whereas the average complementarity gap decreases by $1 - \alpha(1 - \gamma(1 - \alpha))$. This justifies the fact that the complementarity gap is not reduced a lot if γ is close to 1 (see the discussion in section 3.4). Hence, a primal-dual IPM aims at using a small value of γ but, at the same time, small values of γ also usually lead to small step lengths α (see Figure 5).

Similarly, choosing $\kappa = 1 - \gamma$ yields that both residuals and complementarity gap are reduced at the same $1 - \alpha(1 - \gamma)$ rate so that the algorithm is globally convergent if a strictly positive step α can be taken at each iteration.

In the predictor-corrector framework described in section 4, the first affine step corresponds to (B.1) with $\gamma = 0$ and $\kappa = 1$. Hence, the associated directions satisfy $(\Delta \mathbf{d}_i^a)^\top \Delta \boldsymbol{\lambda}_i^a = 0$ for all i and a purely affine step of length α_{max}^a would reduce residuals and complementarity gap by $1 - \alpha_{max}^a$. This justifies the choice of γ close to 1 if α_{max}^a is small for the corrector step in order to improve centering. Finally, because $(\Delta \mathbf{d}_i^a)^\top \Delta \boldsymbol{\lambda}_i^a = 0$, adding the quadratic term correction to the residuals of (26) preserves the results of the previous lemma for the corrector step with $\kappa = 1 - \gamma$.

Appendix C. Nesterov-Todd scaling for the reduced linear system

This section gives general results on Jordan algebra and the Nesterov-Todd scaling for second-order cones. Further details can be found in [29, 31, 48, 49].

For a vector $\mathbf{v} = (v^0, \bar{\mathbf{v}}) \in \mathcal{Q}^{m+1}$, we define:

$$\mathbf{mat}(\mathbf{v}) = \begin{bmatrix} v^0 & \bar{\mathbf{v}}^\top \\ \bar{\mathbf{v}} & v^0 \mathbf{I}_m \end{bmatrix} \in \mathbb{R}^{m+1 \times m+1} \quad (\text{C.1})$$

$$\det(\mathbf{v}) = \det(\mathbf{mat}(\mathbf{v})) = (v^0)^2 - \|\bar{\mathbf{v}}\|^2 \quad (\text{C.2})$$

$$\hat{\mathbf{v}} = \mathbf{Q}\mathbf{v} = (v^0, -\bar{\mathbf{v}}) \quad (\text{C.3})$$

$$(\mathbf{mat}(\mathbf{v}))^{-1} = \frac{1}{\det(\mathbf{v})} \begin{bmatrix} v^0 & -\bar{\mathbf{v}}^\top \\ -\bar{\mathbf{v}} & \frac{\det(\mathbf{v})\mathbf{I}_m + \bar{\mathbf{v}}\bar{\mathbf{v}}^\top}{v^0} \end{bmatrix} \quad (\text{C.4})$$

with \mathbf{I}_m being the identity matrix of size m and $\mathbf{Q} = \mathbf{diag}(1, -\mathbf{I}_m)$. Note that the last relation is valid only if \mathbf{v} is located strictly inside \mathcal{Q}^{m+1} , i.e. $\mathbf{v} \in \text{int}(\mathcal{Q}^{m+1})$, which is equivalent to $\det(\mathbf{v}) > 0$.

Let us consider here the complementary slackness condition:

$$\mathbf{x} \circ \mathbf{s} = \begin{Bmatrix} x^0 s^0 + \bar{\mathbf{x}}^\top \bar{\mathbf{s}} \\ x^0 \bar{\mathbf{s}} + s^0 \bar{\mathbf{x}} \end{Bmatrix} = \mathbf{X}\mathbf{s} = \mathbf{S}\mathbf{x} = \mathbf{X}\mathbf{S}\mathbf{e} \quad (\text{C.5})$$

where $\mathbf{X} = \mathbf{mat}(\mathbf{x})$, $\mathbf{S} = \mathbf{mat}(\mathbf{s})$ and $\mathbf{e} = (1, \mathbf{0})$. We further assume that $\mathbf{x}, \mathbf{s} \in \text{int}(\mathcal{Q}^{m+1})$. It can then be shown that there exists a unique symmetric matrix \mathbf{F} (depending on \mathbf{x} and \mathbf{s}) such that:

$$\mathbf{F}\mathbf{x} = \tilde{\mathbf{x}} = \tilde{\mathbf{s}} = \mathbf{F}^{-1}\mathbf{s} \text{ and } \mathbf{x}^\top \mathbf{s} = \tilde{\mathbf{x}}^\top \tilde{\mathbf{s}} \quad (\text{C.6})$$

The scaled point $\mathbf{v} = \tilde{\mathbf{x}} = \tilde{\mathbf{s}}$ satisfies the following properties:

$$\mathbf{x} \in \mathcal{Q}^m \Leftrightarrow \mathbf{v} \in \mathcal{Q}^m \quad (\text{C.7})$$

$$\mathbf{x} \in \text{int}(\mathcal{Q}^m) \Leftrightarrow \mathbf{v} \in \text{int}(\mathcal{Q}^m) \quad (\text{C.8})$$

The linearization of the complementarity equation is thus rewritten around point \mathbf{v} as follows:

$$(\mathbf{v} + \Delta\mathbf{v}) \circ (\mathbf{v} + \Delta\mathbf{v}) = (\mathbf{F}(\mathbf{x} + \Delta\mathbf{x})) \circ (\mathbf{F}^{-1}(\mathbf{s} + \Delta\mathbf{s})) \quad (\text{C.9})$$

$$\approx \mathbf{v} \circ \mathbf{v} + \mathbf{v} \circ \mathbf{F}\Delta\mathbf{x} + \mathbf{v} \circ \mathbf{F}^{-1}\Delta\mathbf{s} \quad (\text{C.10})$$

$$\approx \mathbf{v} \circ \mathbf{v} + \mathbf{V}\mathbf{F}\Delta\mathbf{x} + \mathbf{V}\mathbf{F}^{-1}\Delta\mathbf{s} \quad (\text{C.11})$$

In the case of second-order cones, this transformation is explicitly given by the following symmetric matrix:

$$\mathbf{F} = \theta \begin{bmatrix} w^0 & \bar{\mathbf{w}}^T \\ \bar{\mathbf{w}} & \mathbf{I}_m + \frac{\bar{\mathbf{w}}\bar{\mathbf{w}}^T}{1+w^0} \end{bmatrix} = \theta \left(-\mathbf{Q} + \frac{(\mathbf{e} + \mathbf{w})(\mathbf{e} + \mathbf{w})^T}{1+w_0} \right) \quad (\text{C.12})$$

$$\mathbf{w} = \frac{\theta^{-1}\mathbf{s} + \theta\hat{\mathbf{x}}}{\sqrt{2}\sqrt{\mathbf{x}^T\mathbf{s}} + \sqrt{\det(\mathbf{x})\det(\mathbf{s})}} \quad (\text{C.13})$$

$$\theta = \left(\frac{\det(\mathbf{s})}{\det(\mathbf{x})} \right)^{1/4} \quad (\text{C.14})$$

We then have the following results:

$$\mathbf{F}^2 = \theta^2(-\mathbf{Q} + 2\mathbf{w}\mathbf{w}^T) \quad (\text{C.15})$$

$$\mathbf{F}^{-1} = \theta^{-2}\mathbf{Q}\mathbf{F}\mathbf{Q} \quad (\text{C.16})$$

$$\mathbf{F}^{-2} = \theta^{-2}(-\mathbf{Q} + 2\hat{\mathbf{w}}\hat{\mathbf{w}}^T) \quad (\text{C.17})$$

From the last relation, we deduce that $\overline{\mathbf{F}^{-2}} = \theta^{-2}(\mathbf{I}_m + 2\bar{\mathbf{w}}\bar{\mathbf{w}}^T)$ where $\bar{\star}$ is the tail i.e. the bottom-right $m \times m$ block from matrix \star . Applying the Sherman-Morrison formula, we obtain that:

$$(\overline{\mathbf{F}^{-2}})^{-1} = \theta^2 \left(\mathbf{I}_m - 2\frac{\bar{\mathbf{w}}\bar{\mathbf{w}}^T}{1 + 2\|\bar{\mathbf{w}}\|^2} \right) \quad (\text{C.18})$$

which is easily checked to be symmetric positive definite.

Appendix D. Condensation of the KKT system

From system (30), elimination of the last row gives:

$$\begin{bmatrix} \mathbf{K} & \tau_0\mathbf{B}^T\mathbf{J} \\ \mathbf{B} & -\mathbf{J}\mathbf{F}^{-2} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{s} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_d \\ \mathbf{r}_p - \mathbf{J}\mathbf{F}^{-1}\mathbf{V}^{-1}\mathbf{R}_c \end{Bmatrix} \quad (\text{D.1a})$$

$$\Delta\mathbf{x} = -\mathbf{F}^{-2}\Delta\mathbf{s} + \mathbf{F}^{-1}\mathbf{V}^{-1}\mathbf{R}_c \quad (\text{D.1b})$$

Now, since $\Delta\mathbf{s}^i = \begin{Bmatrix} 0 \\ -\Delta\lambda^i \end{Bmatrix}$ for all i , the previous system is equivalent to:

$$\begin{bmatrix} \mathbf{K} & \tau_0\mathbf{B}^T \\ \mathbf{B} & -\overline{\mathbf{F}^{-2}} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u} \\ \Delta\lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_d \\ \mathbf{r}_p + \overline{\mathbf{F}^{-1}\mathbf{V}^{-1}\mathbf{R}_c} \end{Bmatrix} \quad (\text{D.2})$$

where $\bar{\mathbf{x}}$ denotes the block matrix/vector obtained by keeping only the tail of each corresponding block (e.g. $\bar{\Delta \mathbf{s}} = -\Delta \boldsymbol{\lambda}$). This last system can be further condensed into:

$$\underbrace{\left(\mathbf{K} + \tau_0 \mathbf{B}^T (\bar{\mathbf{F}}^{-2})^{-1} \mathbf{B} \right)}_{\mathbf{A}'} \Delta \mathbf{u} = \mathbf{r}_d + \tau_0 \mathbf{B}^T (\bar{\mathbf{F}}^{-2})^{-1} (\mathbf{r}_p + \overline{\mathbf{F}^{-1} \mathbf{V}^{-1} \mathbf{R}_c}) = \mathbf{r}' \quad (\text{D.3a})$$

$$\Delta \boldsymbol{\lambda} = (\bar{\mathbf{F}}^{-2})^{-1} \left(\mathbf{B} \Delta \mathbf{u} - \mathbf{r}_p - \overline{\mathbf{F}^{-1} \mathbf{V}^{-1} \mathbf{R}_c} \right) \quad (\text{D.3b})$$

- [1] N. J. Balmforth, I. A. Frigaard, G. Ovarlez, Yielding to stress: recent developments in viscoplastic fluid mechanics, *Annual Review of Fluid Mechanics* 46 (2014) 121–146.
- [2] P. Coussot, Bingham’s heritage, *Rheologica Acta* 6 (2016) 163–176.
- [3] M. Bercovier, M. Engelman, A finite-element method for incompressible non-Newtonian flows, *Journal of Computational Physics* 36 (1980) 313–326.
- [4] T. C. Papanastasiou, Flows of materials with yield, *Journal of Rheology* 31 (1987) 385.
- [5] M. Fortin, R. Glowinski, *Méthodes de lagrangien augmenté: applications à la résolution numérique de problèmes aux limites*, Dunod, 1982.
- [6] R. Glowinski, P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM, 1989.
- [7] P. Saramito, N. Roquet, An adaptive finite element method for viscoplastic fluid flows in pipes, *Computer methods in applied mechanics and engineering* 190 (2001) 5391–5412.
- [8] E. J. Dean, R. Glowinski, G. Guidoboni, On the numerical simulation of Bingham visco-plastic flow: old and new results, *Journal of non-newtonian fluid mechanics* 142 (2007) 36–62.
- [9] R. Glowinski, A. Wachs, On the numerical simulation of viscoplastic fluid flow, *Handbook of numerical analysis* 16 (2011) 483–718.
- [10] T. Treskatis, M. A. Moyers-González, C. J. Price, An accelerated dual proximal gradient method for applications in viscoplasticity, *Journal of Non-Newtonian Fluid Mechanics* 238 (2016) 115–130.
- [11] P. Saramito, A. Wachs, Progress in numerical simulation of yield stress fluid flows, *Rheologica Acta* (2017) 1–20.
- [12] 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.
- [13] A. Friaâ, Le matériau de Norton-Hoff généralisé et ses applications en analyse limite, *Comptes Rendus de l’Académie des Sciences, Paris Série AB* 286 (1978) A953–A956.
- [14] T. Guennouni, P. Le Tallec, Calcul à la rupture, régularisation de Norton-Hoff et Lagrangien augmenté, *Journal de Mécanique Théorique et Appliquée* 2 (1982) 75–99.
- [15] M. Vicente da Silva, A. Antao, A non-linear programming method approach for upper bound limit analysis, *International Journal for Numerical Methods in Engineering* 72 (2007) 1192–1218.
- [16] G. Carlier, M. Comte, I. Ionescu, G. Peyré, A projection approach to the numerical analysis of limit load problems, *Mathematical Models and Methods in Applied Sciences* 21 (2011) 1291–1316.
- [17] S. J. Wright, *Primal-dual interior-point methods*, Siam, 1997.
- [18] K. D. Andersen, E. Christiansen, M. L. Overton, Computing limit loads by minimizing a sum of norms, *SIAM Journal on Scientific Computing* 19 (1998) 1046–1062.
- [19] A. V. Lyamin, S. W. Sloan, Upper bound limit analysis using linear finite elements and non-linear programming, *International Journal for Numerical and Analytical Methods in Geomechanics* 26 (2002) 181–216.
- [20] K. Krabbenhoft, L. Damkilde, A general non-linear optimization algorithm for lower bound limit analysis, *International Journal for Numerical Methods in Engineering* 56 (2003) 165–184.
- [21] A. Makrodimopoulos, C. Martin, Upper bound limit analysis using simplex strain elements and second-order cone programming, *International journal for numerical and analytical methods in geomechanics* 31 (2007) 835–865.

- [22] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, *Linear algebra and its applications* 284 (1998) 193–228.
- [23] Mosek, The Mosek optimization software., Available from: <http://www.mosek.com/>, December 2014. URL: Available from: <http://www.mosek.com/>.
- [24] J. Bleyer, Viscoplastic flows : supplementary code for "advances in the simulation of viscoplastic fluid flows using interior-point methods", Zenodo, 2017. doi:10.5281/zenodo.1038519.
- [25] T. Goldstein, B. O'Donoghue, S. Setzer, R. Baraniuk, Fast alternating direction optimization methods, *SIAM Journal on Imaging Sciences* 7 (2014) 1588–1623.
- [26] Y. Nesterov, A method of solving a convex programming problem with convergence rate $o(1/k^2)$, in: *Soviet Mathematics Doklady*, volume 27, 1983, pp. 372–376.
- [27] R. Glowinski, J.-L. Lions, R. Tremolieres, *Numerical analysis of variational inequalities*, volume 8, Elsevier, 2011.
- [28] N. Megiddo, Pathways to the optimal set in linear programming, in: *Progress in mathematical programming*, Springer, 1989, pp. 131–158.
- [29] F. Alizadeh, D. Goldfarb, Second-order cone programming, *Mathematical programming* 95 (2003) 3–51.
- [30] S. Mehrotra, On the implementation of a primal-dual interior point method, *SIAM Journal on optimization* 2 (1992) 575–601.
- [31] E. D. Andersen, C. Roos, T. Terlaky, On implementing a primal-dual interior-point method for conic quadratic optimization, *Mathematical Programming* 95 (2003) 249–277.
- [32] A. Logg, K.-A. Mardal, G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84, Springer Science & Business Media, 2012.
- [33] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, *Parallel computing* 32 (2006) 136–156.
- [34] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal on Matrix Analysis and Applications* 23 (2001) 15–41.
- [35] P. Szabo, O. Hassager, Flow of viscoplastic fluids in eccentric annular geometries, *Journal of Non-Newtonian Fluid Mechanics* 45 (1992) 149–169.
- [36] A. Wachs, Numerical simulation of steady Bingham flow through an eccentric annular cross-section by distributed Lagrange multiplier/fictitious domain and augmented Lagrangian methods, *Journal of Non-Newtonian Fluid Mechanics* 142 (2007) 183–198.
- [37] I. Frigaard, C. Nouar, On the usage of viscosity regularisation methods for visco-plastic fluid flow computation, *Journal of Non-Newtonian Fluid Mechanics* 127 (2005) 1–26.
- [38] A. Putz, I. Frigaard, D. Martinez, On the lubrication paradox and the use of regularisation methods for lubrication flows, *Journal of Non-Newtonian Fluid Mechanics* 163 (2009) 62–77.
- [39] A. Syrakos, G. C. Georgiou, A. N. Alexandrou, Solution of the square lid-driven cavity flow of a Bingham plastic using the finite volume method, *Journal of Non-Newtonian Fluid Mechanics* 195 (2013) 19–31.
- [40] A. Aposporidis, P. S. Vassilevski, A. Veneziani, Multigrid preconditioning of the non-regularized augmented Bingham fluid problem, *Electronic Transactions on Numerical Analysis* 41 (2014) 42–61.
- [41] P. Saramito, A damped Newton algorithm for computing viscoplastic fluid flows, *Journal of Non-Newtonian Fluid Mechanics* 238 (2016) 6–15.
- [42] E. A. Yildirim, S. J. Wright, Warm-start strategies in interior-point methods for linear programming, *SIAM Journal on Optimization* 12 (2002) 782–810.
- [43] J. Gondzio, A. Grothey, Reoptimization with the primal-dual interior point method, *SIAM Journal on Optimization* 13 (2002) 842–864.
- [44] E. John, E. A. Yildirim, Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension, *Computational Optimization and Applications* 41 (2008) 151–183.
- [45] N. Roquet, P. Saramito, An adaptive finite element method for Bingham fluid flows around a cylinder, *Computer methods in applied mechanics and engineering* 192 (2003) 3317–3341.
- [46] S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

- [47] X. Xu, P.-F. Hung, Y. Ye, A simplified homogeneous and self-dual linear programming algorithm and its implementation, *Annals of Operations Research* 62 (1996) 151–171.
- [48] Y. E. Nesterov, M. J. Todd, Self-scaled barriers and interior-point methods for convex programming, *Mathematics of Operations research* 22 (1997) 1–42.
- [49] H. Yamashita, H. Yabe, A primal–dual interior point method for nonlinear optimization over second-order cones, *Optimization Methods & Software* 24 (2009) 407–426.