Practical guidelines for fast, efficient and robust simulations of yield-stress flows without regularisation using accelerated proximal gradient or augmented Lagrangian methods

Timm Treskatis[a,*], Ali Roustaei[d,*], Ian Frigaard[a,b], Anthony Wachs[a,c]

[a] Department of Mathematics, The University of British Columbia, 1984 Mathematics Road, Vancouver, BC, Canada V6T 1Z2
[b] Department of Mechanical Engineering, The University of British Columbia, 2054 Applied Science Lane, Vancouver, BC, Canada V6T 1Z4
[c] Department of Chemical and Biological Engineering, The University of British Columbia, 2360 East Mall, Vancouver, BC, Canada V6T 1Z3
[d] Department of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

Abstract

The mathematically sound resolution of yield stress fluid flows involves nonsmooth convex optimisation problems. Traditionally, augmented Lagrangian methods developed in the 1980's have been used for this purpose. The main drawback of these algorithms is their frustratingly slow $O(1/\sqrt{k})$ worst-case convergence, where k is the iteration counter. Recently, an improved 'dual FISTA' algorithm (short: FISTA*) was introduced, which achieves the higher and provably optimal rate of $O(1/k)$. When implementing these algorithms in two finite-element packages (FreeFem++ by Frédéric Hecht, UPMC Paris and Rheolef by Pierre Saramito, UGA Grenoble), we observed that these theoretical convergence rates are not generally attained. In this article, we present four common numerical pitfalls that adversely impact the convergence of the optimisation algorithms. By means of constructive and practical guidelines we point out how a careful implementation can not only recover the full order of convergence, but also reduce the computational cost per iteration for further efficiency gains. Furthermore, we assess the performance and accuracy of FISTA* for the practical case of flow in wavy walled channel and demonstrate significant speed-up when FISTA* is employed instead of the classical augmented Lagrangian method.

Keywords: yield-stress flows, viscoplastic fluids, augmented Lagrangian method, FISTA*, finite elements
2010 MSC: 65K15, 76M10

1. Introduction

For several decades, the alternating direction method of multipliers (ADMM or ALG2) from the class of augmented Lagrangian methods has served as the workhorse for genuinely nonsmooth simulations of yield-stress fluid flows. With the attribute 'genuinely nonsmooth' we refer to algorithms that solve viscoplastic flow problems without artificial smoothing or other forms of regularisation.

As a minimal working example that captures all relevant effects without adding extra complexity, we

[*] Corresponding authors
Email addresses: timmt@math.ubc.ca (Timm Treskatis), aroustaei@ut.ac.ir (Ali Roustaei)

predominantly focus on the model problem of stationary creeping Bingham flow in two spatial dimensions

$$\boldsymbol{\tau} = 2\mu\mathcal{D}\boldsymbol{u} + \tau_0 \frac{\mathcal{D}\boldsymbol{u}}{|\mathcal{D}\boldsymbol{u}|} \qquad \text{if } |\mathcal{D}\boldsymbol{u}| \neq 0 \tag{1a}$$

$$|\boldsymbol{\tau}| \leq \tau_0 \qquad \text{if } |\mathcal{D}\boldsymbol{u}| = 0 \tag{1b}$$

$$-\operatorname{div}\boldsymbol{\tau} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega \tag{1c}$$

$$\operatorname{div}\boldsymbol{u} = 0 \qquad \text{in } \Omega \tag{1d}$$

$$\boldsymbol{u} = \boldsymbol{u}_\Gamma \qquad \text{on } \Gamma \tag{1e}$$

where $\boldsymbol{\tau} : \Omega \to \mathbb{R}^{2\times 2}_{\mathrm{sym}}$ is the deviatoric part of the Cauchy stress tensor, $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ the velocity field and $p : \Omega \to \mathbb{R}$ the pressure. The problem data consisting of a plastic viscosity parameter $\mu > 0$, a yield stress $\tau_0 \geq 0$, a body force $\boldsymbol{f} : \Omega \to \mathbb{R}^2$ and an admissible boundary velocity $\boldsymbol{u}_\Gamma : \Gamma \to \mathbb{R}^2$ with

$$\int\limits_\Gamma \boldsymbol{u}_\Gamma \cdot \boldsymbol{n} \,\mathrm{d}s = 0$$

are given. We use the notation $\mathcal{D} := (\nabla + \nabla^\top)/2$ for the symmetric part of the gradient operator and for a tensor $\boldsymbol{\tau}$ we define a norm by setting

$$|\boldsymbol{\tau}| := \sqrt{\sum_{i,j} \frac{\tau_{ij}^2}{2}}.$$

Γ is the boundary of the Lipschitz domain $\Omega \subset \mathbb{R}^2$ with outward pointing unit normal vectors $\boldsymbol{n}$.

A convenient alternative formulation of the Bingham model from Equations (1a) and (1b) reads

$$\mathcal{D}\boldsymbol{u} = \boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}) := \frac{1}{2\mu}\left(|\boldsymbol{\tau}| - \tau_0\right)_+ \frac{\boldsymbol{\tau}}{|\boldsymbol{\tau}|}. \tag{2}$$

Whenever the thresholding function $(\cdot)_+ = \max\{0, \cdot\}$ returns zero, we define the entire expression on the right hand side to be zero.

The main difficulty of genuinely nonsmooth formulations of viscoplastic flow problems consists in their lack of differentiability, due to the presence of a yield stress. Simultaneously, the stress $\boldsymbol{\tau}$ is generally not uniquely determined whenever $\tau_0 > 0$.

1.1. Genuinely nonsmooth optimisation algorithms

The traditional solution approach of Duvaut and Lions [1] identifies Problem (1) with the nonsmooth convex minimisation problem

$$\min_{\substack{\boldsymbol{u}\in U\\ \boldsymbol{d}\in S}} \left\{ I(\boldsymbol{u},\boldsymbol{d}) := 2\mu \int\limits_\Omega |\boldsymbol{d}|^2 \,\mathrm{d}x + 2\tau_0 \int\limits_\Omega |\boldsymbol{d}| \,\mathrm{d}x - \int\limits_\Omega \boldsymbol{f}\cdot\boldsymbol{u} \,\mathrm{d}x \right\} \tag{3a}$$

$$\text{subject to } \mathcal{D}\boldsymbol{u} = \boldsymbol{d}, \tag{3b}$$

where the extra strain-rate field $\boldsymbol{d}$ is introduced for algorithmic reasons. The set of admissible flow velocities

$$U = \left\{ \boldsymbol{u} \in H^1(\Omega)^2 \;\middle|\; \operatorname{div}\boldsymbol{u} = 0 \text{ and } \boldsymbol{u} = \boldsymbol{u}_\Gamma \text{ on } \Gamma \right\}$$

incorporates incompressibility and the boundary condition into this formulation. Strain rates belong to the space

$$S = \left\{ \boldsymbol{d} \in L^2(\Omega)^{2\times 2} \;\middle|\; \boldsymbol{d} = \boldsymbol{d}^\top \right\}$$

of symmetric tensor-valued functions, as does the stress. In the sequel, we will use boldface symbols $\boldsymbol{H}^1(\Omega) = H^1(\Omega)^2$ or $\boldsymbol{L}^2(\Omega) = L^2(\Omega)^{2\times 2}$ for product spaces, where their multiplicity shall be clear from the context. We will also suppress the domain Ω.

Note that the stress $\boldsymbol{\tau}$ does not explicitly appear in the *primal problem* (3). It does, however, act as a Lagrange multiplier for the constraint (3b). Conversely, in the equivalent *dual problem* [2, 3]

$$\min_{\substack{\boldsymbol{\tau}\in S\\ p\in P}} \left\{ J(\boldsymbol{\tau}) := \frac{1}{2\mu}\int_\Omega \left(|\boldsymbol{\tau}| - \tau_0\right)_+^2 \,\mathrm{d}x \right\} \tag{4a}$$

$$\text{subject to } -\operatorname{div}\boldsymbol{\tau} + \nabla p = \boldsymbol{f}. \tag{4b}$$

with the pressure space

$$P = \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p \,\mathrm{d}x = 0 \right\},$$

the velocity $\boldsymbol{u}$ has been eliminated, but it assumes the role of a Lagrange multiplier for the constraint (4b).

It is an important observation that the primal problem (3) is strongly convex but, if $\tau_0 > 0$, not differentiable. In contrast, the dual problem (4) is once (Lipschitz) continuously differentiable but, if $\tau_0 > 0$, not strongly convex [3].

Both traditional and more recently devised optimisation algorithms exploit this structural information: starting from an arbitrarily chosen initial guess, numerical methods for the solution of Problem (3) compute updates iteratively by taking steps along the negative gradient of the dual problem (4).

The classical augmented Lagrangian method applied to Problem (3) has appeared e.g. in [4, 5, 6] and reads as follows:

Algorithm 1 (Alternating Direction Method of Multipliers (ADMM / ALG2) for Bingham flow)**.**

Input *Initial guesses* $\boldsymbol{d}^0, \boldsymbol{\tau}^0$, *a penalty parameter* $r > 0$

Initialisation $k = 0$

Step 1 *Solve the Stokes problem*

$$\begin{aligned} -\operatorname{div}\left(r\mathcal{D}\boldsymbol{u}^{k+1}\right) + \nabla p^{k+1} &= \boldsymbol{f} - \operatorname{div}\left(r\boldsymbol{d}^k - \boldsymbol{\tau}^k\right) && \text{in } \Omega\\ \operatorname{div}\boldsymbol{u}^{k+1} &= 0 && \text{in } \Omega\\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}_\Gamma && \text{on } \Gamma. \end{aligned}$$

Step 2 *Evaluate the modified Bingham model*

$$\boldsymbol{d}^{k+1} = \frac{1}{2\mu + r}\left(|\boldsymbol{\tau}^k + r\mathcal{D}\boldsymbol{u}^{k+1}| - \tau_0\right)_+ \frac{\boldsymbol{\tau}^k + r\mathcal{D}\boldsymbol{u}^{k+1}}{|\boldsymbol{\tau}^k + r\mathcal{D}\boldsymbol{u}^{k+1}|}.$$

Step 3 If *the algorithm has converged* then *stop* else *update the stress field*

$$\boldsymbol{\tau}^{k+1} = \boldsymbol{\tau}^k + r\left(\mathcal{D}\boldsymbol{u}^{k+1} - \boldsymbol{d}^{k+1}\right).$$

set $k \leftarrow k + 1$ *and* go to *Step 1.*

The Bermúdez-Moreno method [7] is a more general algorithm for solving variational inequalities. When applied to the Bingham flow problem, it does however result in a very similar algorithm, possibly with different free parameters [8].

A few remarks on the convergence of Algorithm 1 are in order. In the literature on convex optimisation, convergence results are usually available for the optimality gap in function values, i.e. how fast the sequence $I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) - I(\bar{\boldsymbol{u}}, \mathcal{D}\bar{\boldsymbol{u}})$ or $J(\boldsymbol{\tau}^k) - J(\bar{\boldsymbol{\tau}})$ approaches zero. Here, $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{\tau}}$ denote the unique velocity solution and an admissible shear stress, respectively. Since Problem (3) is strongly convex, convergence in function values automatically enforces convergence of the velocity iterates $\boldsymbol{u}^k \to \bar{\boldsymbol{u}}$ in the energy norm of the problem, by means of the inequality [2, p 23]

$$I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) - I(\bar{\boldsymbol{u}}, \mathcal{D}\bar{\boldsymbol{u}}) \geq 2\mu \|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|^2_{\boldsymbol{L}^2}. \tag{5a}$$

In a similar fashion, convergence in the dual objective forces strain-rate iterates to converge as well [2, p 53]:

$$J(\boldsymbol{\tau}^k) - J(\bar{\boldsymbol{\tau}}) \geq 2\mu \|\boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k) - \mathcal{D}\bar{\boldsymbol{u}}\|^2_{\boldsymbol{L}^2} \tag{5b}$$

under the proviso that the stress iterate $\boldsymbol{\tau}^k$ satisfies the constraint (4b) of the dual problem. Recall that $\boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k)$ defines the strain-rate tensor corresponding to the stress $\boldsymbol{\tau}^k$ according to the Bingham model (2).

The convergence analysis of the alternating direction method of multipliers is and remains challenging, but a number of theoretical results are available which prove that

$$I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) - I(\bar{\boldsymbol{u}}, \mathcal{D}\bar{\boldsymbol{u}}) \leq O(1/k)$$

at least in special cases [9, 10]. In such a case, we obtain from (5) that the iterates of velocity fields $\boldsymbol{u}^k$ generated by Algorithm 1 converge to the analytical solution $\bar{\boldsymbol{u}}$ like

$$\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2} \leq O(1/\sqrt{k}).$$

Indeed, this low sublinear convergence rate has been observed in numerical simulations of viscoplastic flow problems [3].

However, from the theory of complexity we know [11] that by exploiting only function values and gradients—just like Algorithm 1—optimisation problems of the form (4) can be solved at the rate

$$J(\boldsymbol{\tau}^k) - J(\bar{\boldsymbol{\tau}}) \leq O(1/k^2).$$

According to (5), this yields a still sublinear, but compared to ADMM / ALG2 higher rate of order $O(1/k)$ for the iterates. Very recently, Attouch and Peypouquet [12] refined this result further to even faster $o(1/k^2)$ convergence in function values or $o(1/k)$ convergence of the iterates. They also re-confirmed that there is no $\epsilon > 0$ for which a rate of order $O(1/k^{1+\epsilon})$ could generally be guaranteed. In that sense, the sublinear $o(1/k)$ rate is optimal for the genuinely nonsmooth solution of yield-stress flow problems.

The augmented Lagrangian method from Algorithm 1 requires only minor modifications to turn it into a method with such optimal convergence. The following class of algorithms was recently introduced in [3]:

Algorithm 2 (Dual-based Proximal Gradient Methods for Bingham flow)**.**

Input $\boldsymbol{\tau}^0$

Initialisation $k = 0$, $\boldsymbol{\tau}^0_{aux} = \boldsymbol{\tau}^0$

Step 1 *Evaluate the Bingham model* (2)

$$\boldsymbol{d}^k = \frac{1}{2\mu} \left(|\boldsymbol{\tau}^k| - \tau_0\right)_+ \frac{\boldsymbol{\tau}^k}{|\boldsymbol{\tau}^k|}.$$

Step 2 *Solve the Stokes problem*

$$\begin{aligned} -\operatorname{div}\left(2\mu\mathcal{D}\boldsymbol{u}^k\right) + \nabla p^k &= \boldsymbol{f} - \operatorname{div}\left(2\mu\boldsymbol{d}^k - \boldsymbol{\tau}^k\right) && \textit{in } \Omega \\ \operatorname{div}\boldsymbol{u}^k &= 0 && \textit{in } \Omega \\ \boldsymbol{u}^k &= \boldsymbol{u}_\Gamma && \textit{on } \Gamma. \end{aligned}$$

Step 3 If *the algorithm has converged* then *stop* else *update the auxiliary stress field*

$$\boldsymbol{\tau}^{k+1}_{aux} = \boldsymbol{\tau}^k + 2\mu\left(\mathcal{D}\boldsymbol{u}^k - \boldsymbol{d}^k\right).$$

Step 4 *Extrapolate*

$$\boldsymbol{\tau}^{k+1} = \boldsymbol{\tau}_{aux}^{k+1} + \xi^{k+1} \left(\boldsymbol{\tau}_{aux}^{k+1} - \boldsymbol{\tau}_{aux}^{k} \right),$$

set $k \leftarrow k+1$ *and* go to *Step 1.*

There are several common choices for the step-size sequence $(\xi^k)_k$:

- The constant sequence $\xi^k \equiv 0$ gives the suboptimal dual iterative shrinkage-thresholding algorithm (ISTA*) [13], which is almost equivalent to the augmented Lagrangian method from Algorithm 1 (ADMM / ALG2) [3].
- The traditional FISTA sequence is defined through the recursion [13]

$$t^1 = 1, \qquad t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}, \qquad \xi^k = \frac{t^k - 1}{t^{k+1}}. \tag{6}$$

  With this dual-based fast iterative shrinkage-thresholding algorithm (FISTA*), iterates converge at a rate of the order $O(1/k)$. [3]

- Another classical choice stems from Nesterov's acceleration scheme [11]

$$\xi^k = \frac{k-1}{k-1+\alpha} \tag{7}$$

  with a parameter $\alpha > 0$, and typically $\alpha = 3$. $o(1/k)$ convergence of the iterates is obtainable with $\alpha > 3$ [12]. Surely, the improvement from 'big O' to 'little o' is of no practical relevance, but setting $\alpha > 3$ also yields additional and stronger convergence results e.g. for the sequence of stress iterates $\boldsymbol{\tau}^k$ [12, 14]. We therefore use Algorithm 2 with the Nesterov sequence and $\alpha = 4$.

Both Algorithms 1 and 2 define an 'outer' optimisation loop, which converges at a sublinear rate. Every single iteration only demands for simple function evaluations and the solution of a Stokes problem. This Stokes problem in Step 1 or 2, respecitively is conventionally solved with an 'inner' iterative method, such as the conjugate-gradient-Uzawa algorithm [15] which is robust, which converges linearly and for which efficient preconditioners [16, Ch 2] are readily available.

1.2. Other genuinely nonsmooth methods

In recent years, several attempts to solve Steps 1-3 in Algorithms 1 and 2 in a more coupled fashion have appeared in the literature. For instance, Aposporidis and co-authors [17] replaced the methodology based on optimisation techniques with a Picard iteration for Problem (1). A Newton algorithm for viscoplastic pipe flow problems appeared in [18], and Saramito replaced the trust-region strategy of the latter work with line searches in [19].

Compared to the algorithms presented above, the alternative methods from [17] and [19] transfer the main difficulty of yield-stress flow simulations—non-differentiability in $\boldsymbol{u}$ and / or $\boldsymbol{d}$ and non-uniqueness of $\boldsymbol{\tau}$—from the 'outer' to the 'inner' loop. More specifically, the Picard and Newton methods lead to singular linear systems, where the linear operator possesses a large nontrivial kernel. This affects the convergence of these methods as follows:

Convergence of 'outer' iterations Since the objective I of the primal optimisation problem (3) is both strongly convex and some of its terms are Lipschitz continuously differentiable, complexity theory guarantees that 'outer' iterations of a suitable algorithm may converge at a linear rate (see e.g. [20]). Whether the algorithm in [17] actually attains this rate has not yet been fully investigated, as far as we are aware. Aposporidis et al. [17] showed that their proof of linear convergence breaks down in the genuinely nonsmooth case. The system matrix in the Newton iterations in [19] is generally singular. Therefore [21], the convergence rate of Newton's method deteriorates from locally quadratic e.g. for generalised Newtonian flows, to linear when applied to the problem of yield-stress flows.

Convergence of 'inner' iterations The singular linear systems that arise in each 'outer' Picard or Newton iteration may be identified with a quadratic and hence smooth, but not strongly convex optimisation problem. Hence, the optimal convergence rate of 'inner' iterates for the solution of these linear systems is the sublinear rate of $o(1/k)$. We are not currently aware of works that would have addressed the question whether common solvers such as MINRES or GMRES actually attain this optimal rate. Due to the infinite condition number of the linear operators involved, we certainly do not expect linear convergence.

In summary, while the classical optimisation approach leads to algorithms with slow, sublinearly convergent 'outer' iterations and fast, linearly convergent 'inner' iterations, the situation is reversed for the algorithms in [17, 19]: here the 'outer' iterations are (at best) linearly convergent and the 'inner' iterations converge at a sublinear rate.

It should also be noted that GMRES has a significantly larger memory footprint than Algorithms 1 and 2 insofar as it accumulates the *full* convergence history up until the method is restarted. In contrast, FISTA* only requires that *two* auxiliary stress iterates be kept in memory.

Another open question, as pointed out in [17], concerns LBB-stable discretisation schemes for these formulations.

Our objective is to obtain solvers that are designed for a robust and efficient solution of viscoplastic flow problems. Considering the more mature state of the optimisation algorithms ADMM / ALG2 and particularly FISTA* in terms of convergence analysis, preconditioning and efficient implementations, we focus on these latter two algorithms in this work. However, many of our results are equally valid for implementations of other genuinely nonsmooth algorithms, such as Picard iterations or Newton's method.

1.3. Outline of this paper

This paper is mostly based on our experience with implementing Algorithm 2 in the free and open source finite-element packages FreeFem++ [22] and Rheolef [23]. We present and analyse four numerical challenges in Section 2, which negatively impact the convergence of the optimisation algorithms, along with strategies to overcome them. Section 3 is a short case study to asses FISTA* for solving more practical flows. We solve the Stokes flow of a Bingham fluid in a wavy walled channel and compare it to the previously published results obtained using augmented Lagrangian. Finally, we close with some concluding remarks in Section 4.

2. Implementation of ADMM / ALG2 and FISTA*

This part is devoted to practical implementation guidelines for Algorithms 1 and 2, backed by convex and numerical analysis. In the following four subsections, we shall address these computational aspects:

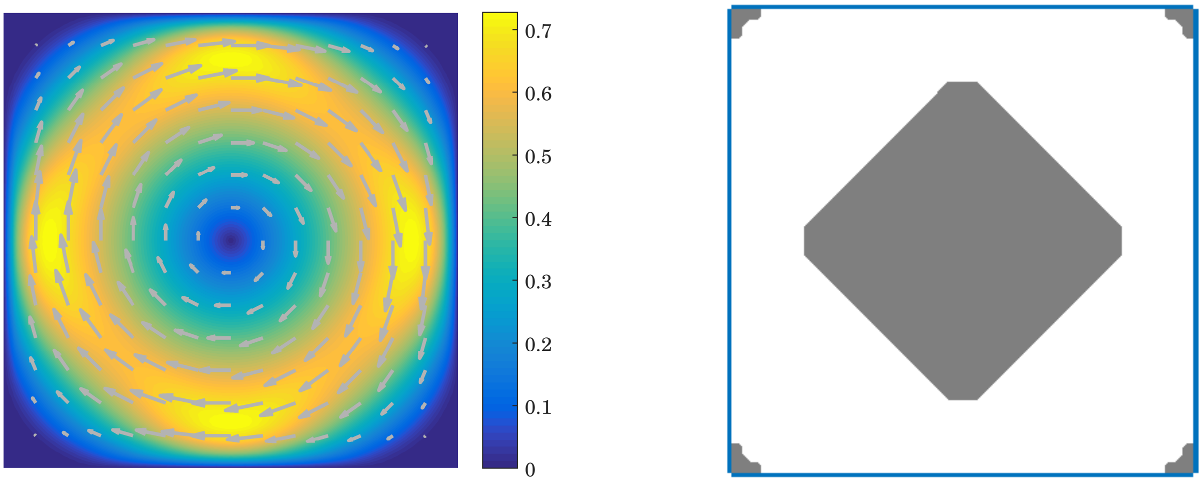
1. What is a feasible way of tracking the convergence of algorithms for viscoplastic flow problems?
2. What accuracy is required for the solution of the Stokes problem in each iteration?
3. What finite elements should be used to discretise flow problems of yield-stress fluids?
4. What triangulations of the flow domain are admissible?

To illustrate the results of our numerical studies in this section, we select the prototypical benchmark problem of force-driven flow enclosed in a square cavity. More precisely, in Problem (1) we define the flow domain as the unit square $\Omega =]0, 1[^2$, on the wall we impose the no-slip condition $\boldsymbol{u}_\Gamma \equiv \boldsymbol{0}$ and we set the right-hand side to

$$\boldsymbol{f}(\boldsymbol{x}) = 300 \begin{pmatrix} x_2 - 0.5 \\ 0.5 - x_1 \end{pmatrix}.$$

For the rheological parameters we choose $\mu = 1$ and $\tau_0 = 10$ (all units are SI units). Reference solutions are available in the literature [2, 3, 24].

An analytical solution for this problem appears to be unknown. For its numerical solution, we start from a Cartesian grid of 32 × 32 squares and subdivide each cell diagonally into four congruent triangles.



(a) Flow velocity $\bar{\boldsymbol{u}}$ and its norm.

(b) Yielded (white) and unyielded (grey) zones.

Figure 1: Numerical reference solution for the force-driven Bingham flow problem in a square cavity, computed with the MATLAB / Octave package TOOTHPASTE [26] on a 32×32 grid. There is no flow in the corners of the domain. In the central plug region that is shaped like a truncated square, the flow is constrained to rigid-body rotation.

On this triangulation, we discretise the flow equations with the conforming and inf-sup stable Bercovier-Pironneau elements [25], i.e. P_1-iso-P_2/P_1 elements for the velocity-pressure complex. Since the velocity is a continuous, piecewise linear function, the discrete shear stress and strain rate are naturally chosen to be (discontinuous) piecewise constant. After $k = 1{,}000{,}000$ iterations of FISTA*, initialised with $\boldsymbol{\tau}^0 \equiv \mathbf{0}$ and always solving the Stokes problems to machine accuracy, we obtain our numerical reference solution $\bar{\boldsymbol{u}} \approx \boldsymbol{u}^{1{,}000{,}000}$. This 'rotating plug' is shown in Figure 1.

2.1. Measuring convergence

The objective of a numerical flow simulation is to find a velocity field $\boldsymbol{u}$, which approximates the true solution $\bar{\boldsymbol{u}}$ as closely as possible. With the error measured in the energy norm, we shall accept an approximation $\boldsymbol{u}$ as soon as

$$\|\mathcal{D}\boldsymbol{u} - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2} \leq \text{tol} \tag{8}$$

for a prescribed tolerance tol > 0. In practice, a sensible numeric value for tol could be derived from the characteristic length and velocity scales of the problem, or one could use a relative criterion of the form

$$\text{tol} = \text{reltol}\|\mathcal{D}\boldsymbol{u}\|_{\boldsymbol{L}^2}$$

with reltol set to e.g. 1% or 0.1%.

Of course, the stopping criterion (8) is not practically applicable, since $\bar{\boldsymbol{u}}$ is unknown. Instead of (8), the common stopping criterion used in simulations of viscoplastic fluids [27, 28, 29, 30, 31] is of the form

$$\|\mathcal{D}\boldsymbol{u} - \boldsymbol{d}\|_{\boldsymbol{L}^2} \leq \epsilon. \tag{9}$$

This mismatch between the two strain-rate fields $\mathcal{D}\boldsymbol{u}$ and $\boldsymbol{d}$ is commonly referred to as *residual* in the literature. The field $\mathcal{D}\boldsymbol{u} - \boldsymbol{d}$ is the negative gradient corresponding to the dual problem (4), and this gradient surely vanishes at the optimal solution.

Sometimes, the criterion (9) is also complemented with a restriction on the velocity increment $\boldsymbol{u}^k - \boldsymbol{u}^{k-1}$ between two subsequent iterations [29, 31]. If measured in the energy norm, such a criterion assumes the form

$$\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\boldsymbol{u}^{k-1}\|_{\boldsymbol{L}^2} \leq \epsilon. \tag{10}$$

Quantity	Definition	Measure for
error	$\boldsymbol{u} - \bar{\boldsymbol{u}}$	proximity to the true solution
residual	$\mathcal{D}\boldsymbol{u} - \boldsymbol{d}$	proximity to the affine subspace where $\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$
increment	$\boldsymbol{u}^k - \boldsymbol{u}^{k-1}$	progress of the algorithm
primal-dual gap	$I(\boldsymbol{u}, \mathcal{D}\boldsymbol{u}) + J(\boldsymbol{\tau})$	difference between the primal functional I and the dual functional $-J$

Table 1: Common nomenclature for the convergence analysis of algorithms in viscoplasticity.

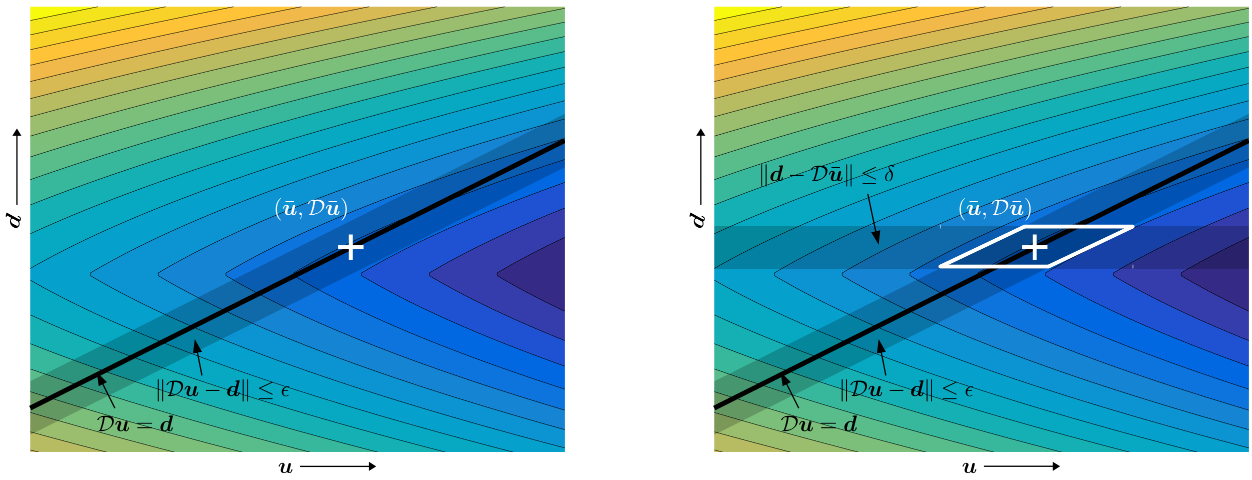

$(\bar{u}, \mathcal{D}\bar{u})$
$\|\mathcal{D}\boldsymbol{u} - \boldsymbol{d}\| \leq \epsilon$
$\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$
$\boldsymbol{d}$
$\boldsymbol{u}$
$\|\boldsymbol{d} - \mathcal{D}\bar{\boldsymbol{u}}\| \leq \delta$
$(\bar{u}, \mathcal{D}\bar{u})$
$\|\mathcal{D}\boldsymbol{u} - \boldsymbol{d}\| \leq \epsilon$
$\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$
$\boldsymbol{d}$
$\boldsymbol{u}$

Figure 2: Sketch of Problem (3): contours of the nonsmooth convex objective $I(\boldsymbol{u}, \boldsymbol{d})$, with its minimiser subject to the constraint $\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$ marked by a $+$. Sets of points that satisfy a stopping criterion are shaded in grey. Note that the usual criterion $\|\mathcal{D}\boldsymbol{u} - \boldsymbol{d}\| \leq \epsilon$ alone does not imply proximity to the solution (left); a second contribution to the error has to be controlled as well (right).

In Table 1 we offer a summary of various technical expressions that are used to describe the convergence of algorithms for simulating yield-stress flows, along with some less abstract interpretations. In other works, the notion of convergence has sometimes been used to describe at what rate the residuals approach zero. To avoid any confusion, we consistently refer to the sequence of errors whenever we speak of the convergence of an optimisation algorithm.

Our analysis shall lead us towards two main conclusions, a negative result and a positive one:

- The commonly used stopping criteria (9) and (10) for Algorithms 1 and 2 do not reliably control the error.
- A both reliable and sharp error bound is computable at relatively low cost.

We will try to illustrate why the residuals are not an adequate measure of convergence. Unfortunately, the plausible implication

$$\text{small residual} \implies \text{small error}$$

is false. In Figure 2, we provide a sketch of the primal problem (3) that we obtain after replacing the unknowns of the vector-valued velocity field $\boldsymbol{u}$ and the tensor-valued strain-rate field $\boldsymbol{d}$ with real numbers only. In this simplified setting, the constraint $\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$ defines a line. The set of all points that satisfy the residual-based stopping criterion (9) forms an infinitely long band around this line. Consequently, even if an iterate $(\boldsymbol{u}, \boldsymbol{d})$ gives a tiny residual, $\boldsymbol{u}$ could still lie arbitrarily far away from the true solution $\bar{\boldsymbol{u}}$. Some numerical experiments shall reveal shortly that this does indeed happen for our test problem of force-driven Bingham flow.

Note that we may decompose the error of the k-th velocity iterate as follows:

$$\underbrace{\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}}_{\text{velocity error}} \leq \underbrace{\|\mathcal{D}\boldsymbol{u}^k - \boldsymbol{d}^k\|_{\boldsymbol{L}^2}}_{\text{residual}} + \underbrace{\|\boldsymbol{d}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}}_{\text{strain-rate error}} \tag{11}$$

Firstly, this inequality shows once once more that a small residual is not generally a sufficient criterion for a small error. Secondly, one may hypothesise that there is a certain risk of using a large penalty parameter r in the augmented Lagrange formulation of the primal problem (3) that Algorithm 1 is based on: in that case, ADMM / ALG2 converges towards the affine subspace where $\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$ with high priority (residuals decay rapidly), whereas convergence towards the true solution plays a subordinate role (errors remain large). Observing only residuals, one might get the impression of seemingly accelerated convergence.

Fortunately, convex optimisation offers a solution to these issues. The strong duality of the primal and dual problems (3) and (4) signifies that for arbitrary velocity fields $\boldsymbol{u}$ and for arbitrary stress fields $\boldsymbol{\tau}$ that satisfy the dual constraint (4b), we generally have

$$I(\boldsymbol{u}, \mathcal{D}\boldsymbol{u}) \geq -J(\boldsymbol{\tau}) \tag{12}$$

and if equality holds, then $\boldsymbol{u}$ and $\boldsymbol{\tau}$ already solve the viscoplastic flow problem [2, p 45]. Combining the inequalities (5) and (12) yields

$$2\mu\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}^2 \leq I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) - I(\bar{\boldsymbol{u}}, \mathcal{D}\bar{\boldsymbol{u}}) \leq I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) + J(\boldsymbol{\tau})$$

for an arbitrary shear stress $\boldsymbol{\tau}$ subject to the constraint (4b). It makes sense to evaluate this so-called *primal-dual gap* on the right-hand side with the best available guess for a solution $\bar{\boldsymbol{\tau}}$ that also satisfies (4b), namely the auxiliary iterate $\boldsymbol{\tau}_{\text{aux}}^{k+1}$. For $k \geq 1$ but not for $k = 0$, the extrapolated stress $\boldsymbol{\tau}^{k+1}$ would be another admissible choice. This way, we obtain

$$\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2} \leq \frac{1}{\sqrt{2\mu}}\sqrt{I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) + J(\boldsymbol{\tau}_{\text{aux}}^{k+1})} =: \eta^k. \tag{13}$$

Even though the error on the left-hand side is unknown, the right-hand side is computable at every iteration. Explicitly, we have

$$\eta^k = \frac{1}{\sqrt{2\mu}}\sqrt{2\mu\int\limits_\Omega |\mathcal{D}\boldsymbol{u}^k|^2\,\mathrm{d}x + 2\tau_0\int\limits_\Omega |\mathcal{D}\boldsymbol{u}^k|\,\mathrm{d}x - \int\limits_\Omega \boldsymbol{f}\cdot\boldsymbol{u}^k\,\mathrm{d}x + \frac{1}{2\mu}\int\limits_\Omega \left(|\boldsymbol{\tau}_{\text{aux}}^{k+1}| - \tau_0\right)_+^2\,\mathrm{d}x} \tag{14}$$

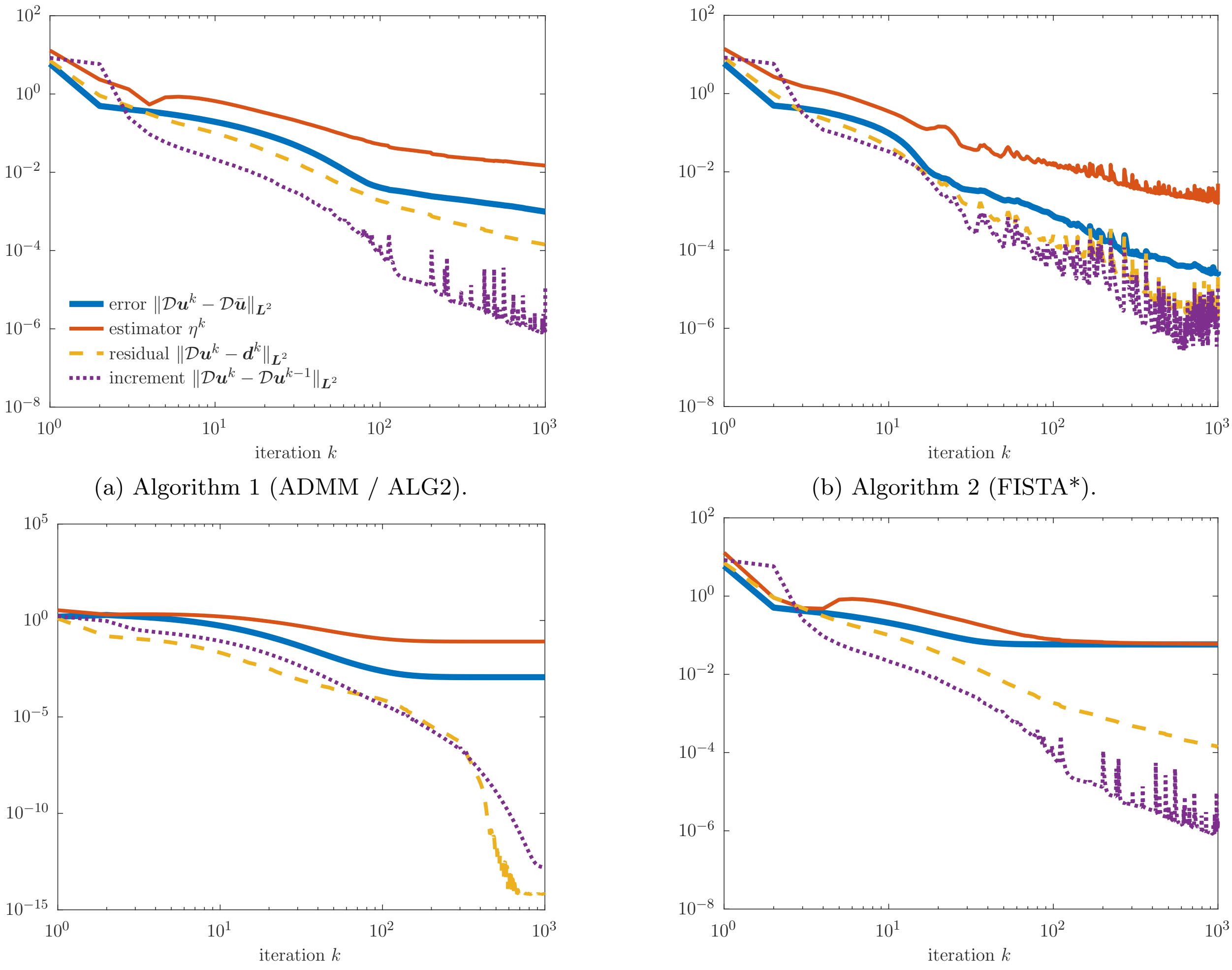
Since this quantity η^k is guaranteed to be a reliable upper bound for the actual error, we refer to it as an *error estimate.*

Numerical experiments

After these theoretical considerations, we now support our conclusions with numerical results. We shall show that residuals and velocity increments are not generally suitable measures of convergence for simulations of yield-stress flows. Furthermore, while the theory already ensures that η^k is a *reliable* upper bound for the error, we also want to find out whether it is *sharp*, i.e. whether it decays at the same and not a lower rate than the actual error.

To that end, we conduct four different experiments, each time solving our test problem of force-driven Bingham flow. In each case, we compute for the first 1,000 iterations

- the error measured in the energy norm, $\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$, with the numerical reference solution $\bar{\boldsymbol{u}}$ from Figure 1,
- the estimator $\eta^k = \sqrt{I(\boldsymbol{u}^k, \mathcal{D}\boldsymbol{u}^k) + J(\boldsymbol{\tau}_{\text{aux}}^{k+1})}/\sqrt{2\mu}$,
- the residual $\|\mathcal{D}\boldsymbol{u}^k - \boldsymbol{d}^k\|_{\boldsymbol{L}^2}$,
- the velocity increment, also measured in the energy norm, $\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\boldsymbol{u}^{k-1}\|_{\boldsymbol{L}^2}$.



(a) Algorithm 1 (ADMM / ALG2).

(b) Algorithm 2 (FISTA*).

(c) Algorithm 1 as penalty method.

(d) Algorithm 1 with incorrect strain rate.

Figure 3: Inadequacy of customary stopping criteria: neither the residuals nor the increments provide reliable error bounds. The error estimator η^k, however, is both reliable and sharp.

Algorithm 1 (ADMM / ALG2). In a first experiment, we use the classical augmented Lagrangian method with a fixed penalty parameter of $r = 2\mu = 2$, initialised with $\boldsymbol{d}^0 = \boldsymbol{\tau}^0 = \mathbf{0}$. The results are shown in Figure 3a.

As predicted by the convergence analysis of this method, the velocity iterates approach the true solution at a low sublinear rate (namely $O(1/\sqrt{k})$). Meanwhile, both the residuals and the increments converge at a higher rate (namely $O(1/k)$). Consequently, if one only measures these two quantities as it is common practice, one might be misled to believe that the algorithm converges faster than it actually does. In other words, these two quantities cannot measure the actual error reliably, which limits their applicability for a stopping criterion.

In contrast, the error estimate exhibits the same convergence rate of order $O(1/\sqrt{k})$ as the errors. As expected, it is also consistently an upper bound for the error. Hence, the estimator does indeed provide both a reliable and sharp error bound in this case.

Algorithm 2 (FISTA).* Repeating the same procedure for Algorithm 2, we obtain a very similar picture, as shown in Figure 3b. FISTA* reduces the error more efficiently than ADMM / ALG2: the velocity iterates now approach the solution at a rate of order $O(1/k)$ (or $o(1/k)$). Due to the non-monotone nature of accelerated methods, it is difficult to infer a quantitative convergence rate of residuals and increments from the graph. The error estimator, however, also captures the improved convergence rate compared to Algorithm 1 accurately.

Algorithm 1 as penalty method. In a third experiment, we study the influence of the penalty parameter r on the convergence of the augmented Lagrangian method. Bernabeu observed that 'the convergence [of the residuals] attains a linear regime' [32, p 59] if the penalty parameter grows exponentially with the number of iterations. For a flow problem different from the one considered here, the author initialised the sequence of penalty parameters with $r = 10$ and then increased their value by 1% from one iteration to the next. We use the same methodology for the test problem at hand, but increasing the parameter r by 2.5% in every iteration. We do this solely for practical purposes, to ensure that the interesting dynamics are still observable within the first 1,000 iterations. The qualitative features remain the same as with the slower growth studied by Bernabeu.

As can be seen from Figure 3c, residuals and velocity increments converge linearly, starting at about 300 iterations and until machine accuracy is reached. Since the growing penalty parameter increasingly prioritises satisfaction of the constraint $\mathcal{D}\boldsymbol{u} = \boldsymbol{d}$ over the minimisation of the objective, such behaviour is expected. Meanwhile, it also becomes clear that large penalty parameters do not actually improve the convergence of the algorithm, but they rather lead to early stagnation.

In line with our aforementioned hypothesis, these findings should serve as a word of warning to practitioners who may feel tempted to tune an implementation of Algorithm 1 for faster convergence by using a large (fixed or variable) penalty parameter. Even though the method will likely satisfy the two common stopping criteria after fewer iterations already, the approximate solution may well be further away from the exact solution as with a smaller parameter.

Algorithm 1 with incorrect strain rate. Our fourth and final convergence experiment is a particularly extreme case. This time, we use Algorithm 1 set up as for the first experiment. However, we deliberately implement a mistake in Step 2 and add a 1% error to the strain-rate tensor by re-scaling $\boldsymbol{d}^{k+1} \leftarrow 1.01\boldsymbol{d}^{k+1}$. Of course, this algorithm does not converge to the solution of the test problem.

Indeed, from Figure 3d we observe that after approximately 50 iterations, the error sequence $\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$ stagnates at a value between 10^{-2} and 10^{-1}. The error estimator η^k predicts this value correctly even quantitatively. Residuals and increments, however, converge in the same manner as in the first experiment, giving the false appearance of convergence to the solution of the problem! From the residuals and increments alone, it is impossible to distinguish a good approximation of $\bar{\boldsymbol{u}}$ from a completely nonsensical velocity field.

Practical consequences

We return to our initial question regarding the issue of how the convergence of optimisation algorithms for viscoplastic flow simulations can be measured.

Residuals and velocity increments paint an overly optimistic picture of the convergence, risking that an algorithm is stopped too early while iterates $\boldsymbol{u}^k$ are still far away from the solution $\bar{\boldsymbol{u}}$. The conclusion that 'the algorithm has converged' because these commonly observed quantities have become small is not generally valid.

The expression (14) for the error estimator η^k requires about the same computational cost as the evaluation of residuals and velocity increments combined. However, it does not share the latters' inadequacy for measuring convergence. The estimator is provably reliable, it is sharp in all four of the above experiments and it hence qualifies as an ideal candidate for a rigourous stopping criterion. To test for convergence in Step 3 of Algorithm 1 or 2, we suggest to check whether $\eta^k \leq \mathrm{tol}$, cf (8). Once this criterion is met, the actual error $\|\mathcal{D}\boldsymbol{u}^k - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$—even though usually unknown—is automatically bounded by tol as well.

Finally, large penalty parameters in the augmented Lagrangian method may be harmful. Besides, to obtain actual and not just apparent convergence acceleration, we recommend to replace ADMM / ALG2 with FISTA*.

2.2. Solving the Stokes problem

We now turn towards the second question regarding the required accuracy of solutions to the Stokes problems which appear in Step 1 of Algorithm 1 and Step 2 of Algorithm 2.

We now turn towards the second question regarding the required accuracy of solutions to the Stokes problems

$$\begin{aligned} -\operatorname{div}(c\mathcal{D}\boldsymbol{u}) + \nabla p &= \boldsymbol{f} - \operatorname{div}\left(c\boldsymbol{d}^k - \boldsymbol{\tau}^k\right) && \text{in } \Omega \\ \operatorname{div}\boldsymbol{u} &= 0 && \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}_\Gamma && \text{on } \Gamma. \end{aligned}$$

which appear in Step 1 of Algorithm 1 with $c = r$, $\boldsymbol{u} = \boldsymbol{u}^{k+1}$, $p = p^{k+1}$ and in Step 2 of Algorithm 2 with $c = 2\mu$, $\boldsymbol{u} = \boldsymbol{u}^k$, $p = p^k$.

In our analysis so far we have assumed that these subproblems are solved exactly. In our numerical tests we have respected this assumption as far as practically possible, by solving the velocity problems in the conjugate-gradient Uzawa algorithm directly through matrix factorisation and iterating until

$$\|\operatorname{div}\boldsymbol{u}\|_{\boldsymbol{L}^2} \leq \mathrm{abstol} \tag{15}$$

where the absolute tolerance is set to machine accuracy $\epsilon \approx 2 \times 10^{-16}$.

The solution of Stokes' problem is by far the most expensive component of Algorithms 1 and 2; in fact it is the only step that contributes a significant computational cost. Consequently, it appears wasteful to solve these subproblems with excessive accuracy while iterates are still far from convergence.

On another hand, it has already been observed in the past that large errors in the evaluation of iterates may impact the convergence rate of fast methods to an extent at which their advantage of convergence acceleration is lost [33]. Thus, while it would be desirable to solve the Stokes problems only inexactly, a certain level of accuracy is still required to preserve the fast and robust convergence of the numerical simulation. Fortunately, recent results from the convergence analysis of convex optimisation algorithms provide a simple and practical criterion that achieves just that.

In [20], Schmidt and co-authors analyse how the convergence rates of proximal gradient methods, such as the basic ISTA and the accelerated FISTA, are affected by inexact evaluations of the subproblems. With their results put in the context of our problem, they show that the convergence rates remain unchanged, provided that the Stokes problem at iteration k of Algorithm 2 is solved with

$$\mathrm{abstol} \leq O(1/k^{2+\delta}), \tag{16}$$

where $\delta > 0$ is an arbitrarily small constant. For the more slowly converging method without acceleration, the rate of convergence remains unaffected if

$$\mathrm{abstol} \leq O(1/k^{1+\delta}). \tag{17}$$

These results motivate the following strategy for an inexact algorithm:

1. For the initial iteration, solve the Stokes problem for some absolute tolerance abstol $= \alpha > 0$.
2. At iteration k, solve the Stokes problem with abstol $= \alpha/k^{1+\delta}$ (for a basic method like Algorithm 1) or abstol $= \alpha/k^{2+\delta}$ (for an accelerated method like Algorithm 2).

Numerical experiments

We proceed with a few simulations of the force-driven Bingham flow problem in order to validate this strategy of inexact augmented Lagrangian or dual FISTA methods.

Inexact variants of Algorithm 1. In Figure 4a we show how the convergence rate of ADMM / ALG2 is affected by inexact evaluations of the Stokes problem. We initially compute an error bound η^0 for the data the algorithm is initialised with, then we define the Stokes tolerance for the first iteration to be $\alpha = 10^{-2}\eta^0$, i.e. we accept a 1% error in the 'inner' iterations relative to the 'outer' iterations. We set the safety factor δ to 10^{-3}.

If the absolute tolerance for the Stokes solver decays at a rate of only $O(1/k^{0.5+\delta})$, then the convergence rate of the augmented Lagrangian method deteriorates from $O(1/\sqrt{k})$ to what appears to be $O(1/k^{3/8})$. This observation indicates that the theoretically derived sufficient condition of [20] cannot be relaxed.

With the Stokes tolerance adjusted more rapidly at a rate of $O(1/k^{1+\delta})$, no visual difference are perceivable between the convergence of the inexact and exact versions of the augmented Lagranian algorithm. Despite virtually identical convergence behaviour in terms of iteration numbers, the inexact method is actually three times faster in terms of computing time. These results are shown in Figure 4c.

Inexact variants of Algorithm 2. In an analogous fashion, we first solve the Stokes problems in FISTA* with abstol decaying like $1/k^{1+\delta}$ or $1/k^{2+\delta}$. As can be seen from Figure 4b, the former rate is insufficient for optimal convergence while the latter strategy yields errors that are indistinguishable from the exact algorithm.

Looking at the performance of the different algorithms over time, we first observe that FISTA* with exact subproblems is more efficient than either version of ADMM / ALG2. With inexactly solved Stokes problems, the computing time of FISTA* is improved further by an additional factor of approximately two.

Algorithm 2 (FISTA) with default settings of finite-element programs.* The FreeFem++ software package [22] implements two types of stopping criteria for Stokes problems: an absolute stopping criterion of the form (15) or, alternatively, a relative one,

$$\|\operatorname{div} \boldsymbol{u}\|_{\boldsymbol{L}^2} \leq \text{reltol}\|\operatorname{div} \tilde{\boldsymbol{u}}\|_{\boldsymbol{L}^2}, \tag{18}$$

where $\tilde{\boldsymbol{u}} = \boldsymbol{u}_0$ refers to the initial velocity iterate of the Stokes solver.

Rheolef [23] by default measures convergence of the Stokes solver with a relative criterion of the form (18), however with $\tilde{\boldsymbol{u}}$ defined as the solution to the elliptic problem

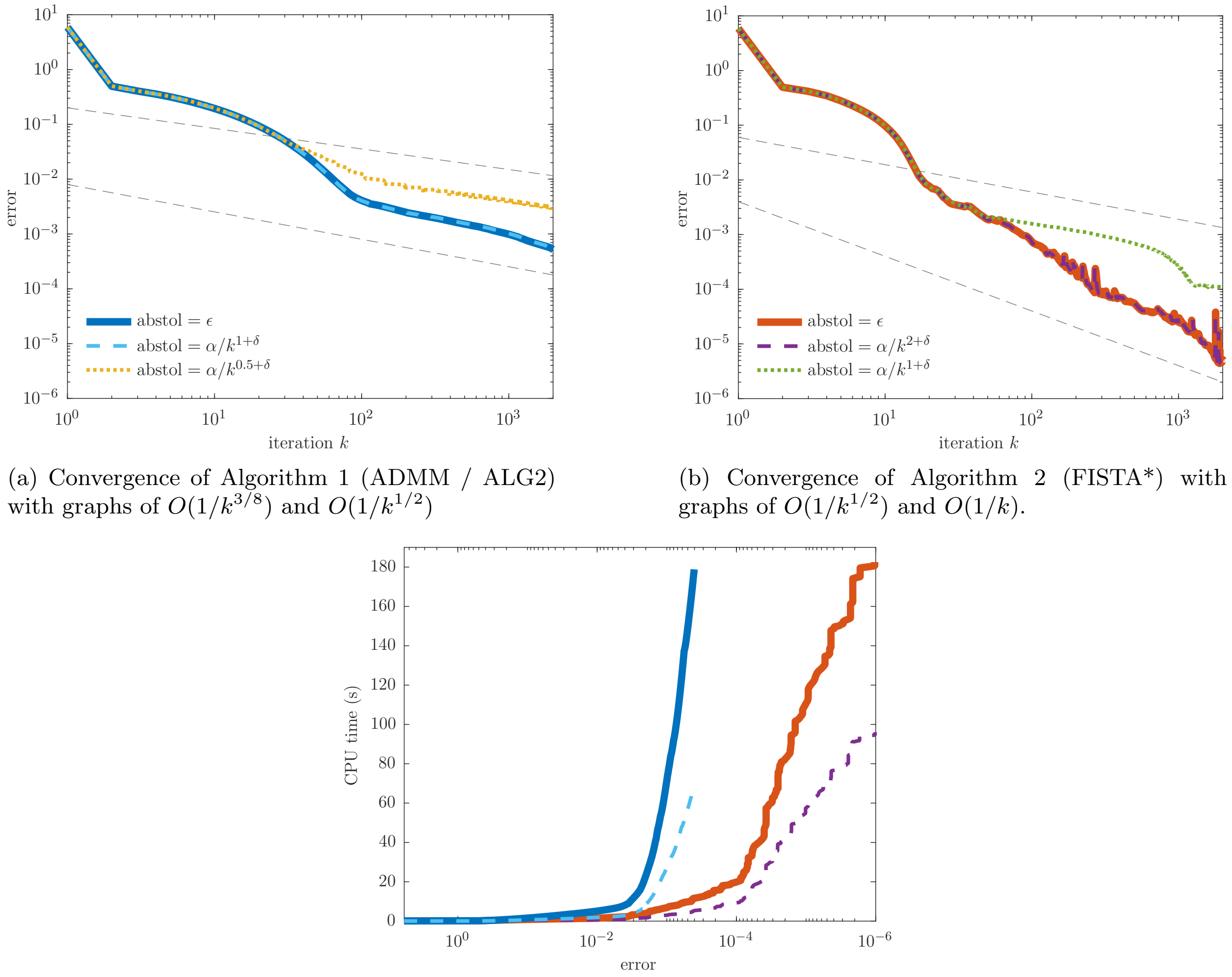
$$-\operatorname{div}(c\mathcal{D}\tilde{\boldsymbol{u}}) = \boldsymbol{f} - \operatorname{div}\left(c\boldsymbol{d}^k - \boldsymbol{\tau}^k\right)$$

which originates from the Schur-complement formulation of the Stokes problem.

It is important to note that the convergence analysis of convex optimisation and the previous two numerical experiments require the absolute tolerance, i.e. the entire right-hand side of (18) to be sufficiently small. This may be difficult to achieve with a relative stopping criterion, since the factor $\|\operatorname{div} \tilde{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$ is not immediately controlled. Indeed, as presented in Figure 5, even with a very small relative tolerance imposed on the Stokes solver, the optimisation algorithm may fail to converge.

Practical consequences

In conclusion, careful attention should be given to the seemingly small detail of a stopping criterion for the Stokes problems in Algorithms 1 and 2. While it is crucial to control the absolute magnitude of the Stokes residuals, it is not necessary to impose a small tolerance right from the start. With a sufficient

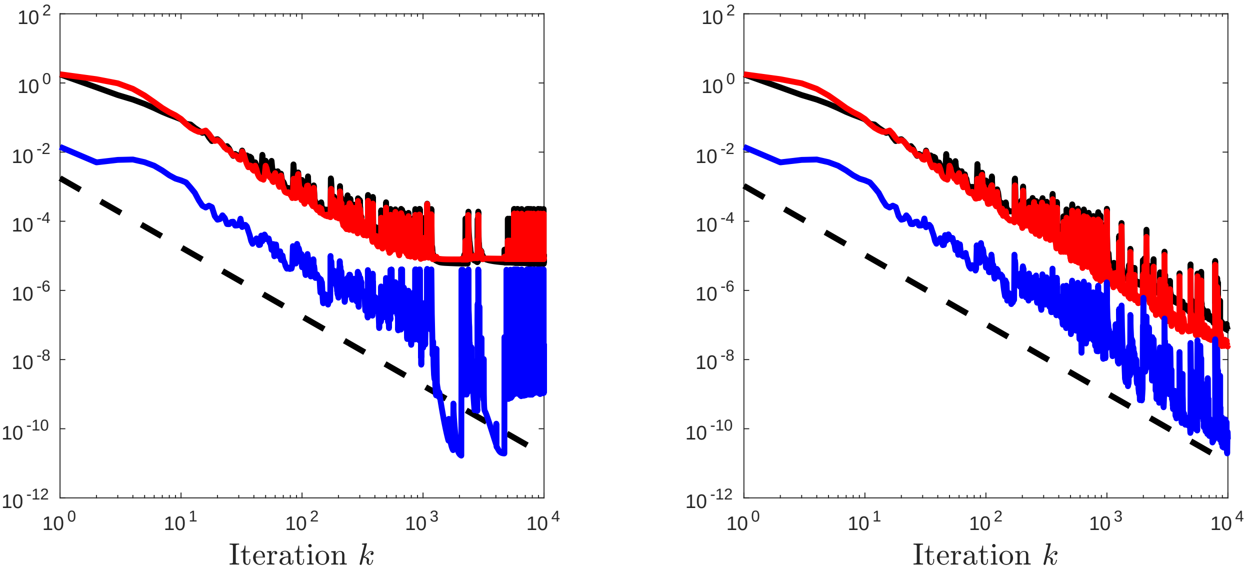


(a) Convergence of Algorithm 1 (ADMM / ALG2) with graphs of $O(1/k^{3/8})$ and $O(1/k^{1/2})$

(b) Convergence of Algorithm 2 (FISTA*) with graphs of $O(1/k^{1/2})$ and $O(1/k)$.

(c) Computing time of exact vs inexact algorithms depending on the imposed level of accuracy. Please refer to the legends in (a) and (b).

Figure 4: Algorithms 1 and 2 with exact or inexact solutions of the Stokes problems. If the absolute tolerance for the Stokes solver decays at a sufficiently high rate, then the inexact algorithm converges at the same rate as the original algorithm. With respect to computing time, the inexact version of each algorithm converges significantly faster than the original method. Data: $\delta = 10^{-3}$, $\alpha \approx 6 \times 10^{-2}$.



(a) Relative stopping criteria

(b) Absolute stopping criteria

Figure 5: Convergence plots of FISTA* for the rotating plug flow. a) Using relative stopping criterion reltol $= 10^{-11}$ for solving Stokes subproblems b) Using absolute stopping criterion for solving Stokes subproblems. The blue, black and red curves show velocity increments in the $\boldsymbol{L}^2$ norm, residuals and the primal-dual gap respectively; the dashed line shows the theoretical $1/k^2$ decay rate. Implementation in FreeFem++.

decay rate, viscoplastic flow problems may be solved several times faster in terms of CPU time, while the algorithms still maintain their favourable robust convergence behaviour.

We point out that for inexactly solved Stokes problems, the error estimator η^k may no longer be a valid error bound since stress iterates may now violate the constraint (4b). We leave a generalised error estimate for inexact algorithms for future research. For now, we recommend to solve only the first and the last iteration with a sharp Stokes tolerance:

- As in our numerical experiments, the error estimator for the initial iterate can provide a sensible value for the absolute Stokes tolerance that an inexact method is initialised with.
- Once it appears like the inexact algorithm has converged, another evaluation of the error estimator with a strict Stokes tolerance confirms whether convergence has indeed been attained.

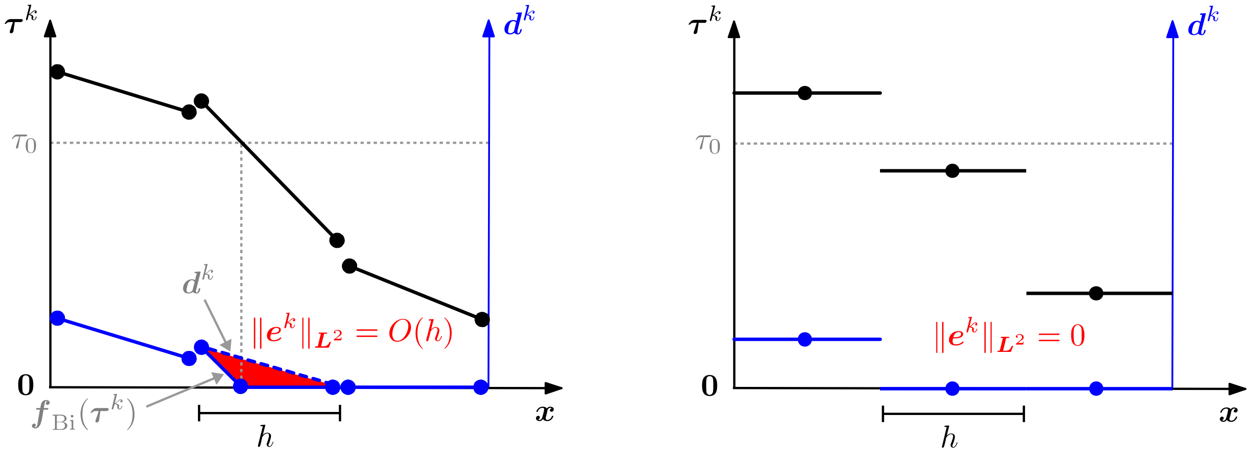
2.3. Evaluating the constitutive model

It is rather obvious that an ‘inner’ iterative method for Stokes’ problem stopped after a finite number of iterations introduces a possibly significant error. A much more subtle, but at least equally onerous error source lurks in the constitutive equation. Step 2 of Algorithm 1 and Step 1 of Algorithm 2 demand for an evaluation of the constitutive model defined by $\boldsymbol{f}_{\mathrm{Bi}}$ in (2). Generally, this cannot be done exactly.

We shall motivate this issue with an example: a common, inf-sup stable element pair for velocity and pressure is the Taylor-Hood or P_2/P_1 element. Due to the continuous, piecewise quadratic velocity approximation, it is natural to discretise the strain-rate and shear-stress tensors with P_1^{dc} elements of discontinuous, piecewise linear functions. However, the nonlinear and nonsmooth function $\boldsymbol{f}_{\mathrm{Bi}}$ turns a piecewise linear stress input $\boldsymbol{\tau}^k$ into an output $\boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k)$ that is usually no longer piecewise linear. Hence, an additional interpolation step is needed to obtain a piecewise linear strain-rate field $\boldsymbol{d}^k$. Overall, we obtain

$$\boldsymbol{d}^k = \boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k) + \boldsymbol{e}^k \tag{19}$$

where $\boldsymbol{e}^k$ assumes the role of an interpolation error. This error renders the Bingham flow problem, discretised with Taylor-Hood elements, inconsistent.



(a) Discontinuous P_1^{dc} elements for shear stress and strain rate, e.g. due to a Taylor-Hood discretisation of the velocity-pressure system.

(b) Discontinuous P_0 elements for shear stress and strain rate, e.g. due to a Bercovier-Pironneau discretisation of the velocity-pressure system.

Figure 6: Evaluation of the Bingham model in 1D: piecewise linear and higher-order elements for the shear stress and strain rate give rise to a mesh-dependent interpolation error; piecewise constant elements allow for an exact evaluation. In 2D and 3D, piecewise linear and higher-order elements additionally introduce an interpolation error of order $O(h)$ or higher on cells where the fluid is fully yielded.

We propose two solution strategies:

1. To *avoid* this inconsistent discretisation, one may use low-order finite elements that lead to a piecewise constant approximation of the strain-rate and shear-stress variables.
2. To *alleviate* the inconsistency issue of higher-order elements, we suggest adaptive enrichment of the discretisation to reduce the magnitude of the interpolation error $\boldsymbol{e}^k$ as the optimisation algorithm progresses.

The avoidance strategy is remarkably simple: if the input $\boldsymbol{\tau}^k$ is piecewise constant, then the output $\boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k)$ is automatically piecewise constant as well and both at the continuous and the discrete level it holds

$$\boldsymbol{d}^k = \boldsymbol{f}_{\mathrm{Bi}}(\boldsymbol{\tau}^k)$$

with $\boldsymbol{e}^k \equiv \boldsymbol{0}$. In Figure 6 we illustrate this concept in one spatial dimension.

Since $\mathcal{D}\boldsymbol{u}$ is a piecewise constant function if and only if $\boldsymbol{u}$ is piecewise linear, we suggest the following two ‘special’ choices of finite elements that are both inf-sup stable and that automatically lead to a fully consistent discretisation of yield-stress fluid flows:

Bercovier-Pironneau elements P_1-iso-P_2 elements for the velocity, which is continuous and piecewise linear on a refined grid; P_0^{dc} elements for the shear stress and strain rate on the same grid; P_1 elements for the pressure, which is continuous and piecewise linear on the given, coarser grid. This approximation is conforming with respect to the function spaces that these variables live in, but not generally discretely mass-conservative.

1$^{\mathrm{st}}$ order Brezzi-Douglas-Marini elements BDM_1 elements for the velocity, which is piecewise linear and continuous in the normal direction across cell interfaces; P_0^{dc} elements for the shear stress, strain rate and pressure. This approximation is a nonconforming discontinuous Galerkin method and discretely mass-conservative.

A sketch of these elements is shown in Figure 7.

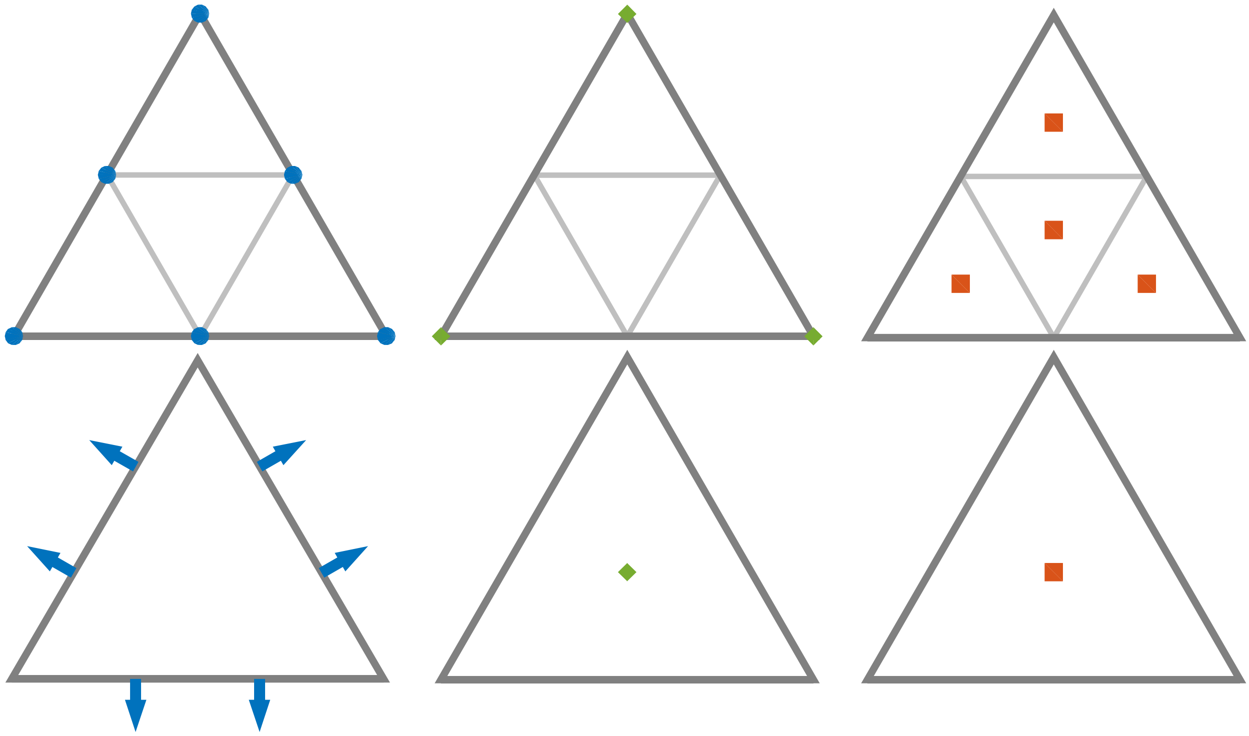


Figure 7: Degrees of freedom of the Bercovier-Pironneau elements (top) and the first-order Brezzi-Douglas-Marini elements (bottom) with the corresponding elements for the velocity (left), pressure (centre) and shear stress / strain rate (right). Nodal values are represented by solid markers, normal components by arrows. Since the velocity is piecewise linear in both cases, the strain rate is a piecewise constant function.

In order to improve upon the approximation quality of these low-order elements, adaptive mesh refinement may be applied if so desired. However, the optimisation algorithm will also converge to a solution on arbitrarily coarse meshes.

This is no longer true if the problem is discretised with higher-order elements, such as Taylor-Hood elements or a discontinuous Galerkin scheme with shape functions of higher polynomial degree. Applying once again the convergence analysis of inexact algorithms of Schmidt and co-workers [20], the interpolation error in the evaluation of the strain rate is required to decay like

$$\|\boldsymbol{e}^k\|_{\boldsymbol{L}^2} \leq O(1/k^{1+\delta}) \tag{20}$$

for the basic ISTA* method without acceleration and like

$$\|\boldsymbol{e}^k\|_{\boldsymbol{L}^2} \leq O(1/k^{2+\delta}) \tag{21}$$

for FISTA* in order to guarantee that the convergence rate does not deteriorate. As for the inexact solution of the Stokes problems, $\delta > 0$ is an arbitrary parameter.

The interpolation error $\boldsymbol{e}^k$ depends on the chosen elements, in particular their polynomial degree, the smoothness of the approximated function and the mesh parameter h, which we define as the maximum diameter over all cells of the grid. These interpolation estimates are available in the literature on finite-element methods, e.g. [34].

On cells where $|\boldsymbol{\tau}^k| > \tau_0$, i.e. where the fluid is fully yielded, $\boldsymbol{f}(\boldsymbol{\tau}^k)$ is a fully smooth function. An interpolation estimate then assumes the form

$$\|\boldsymbol{e}^k\|_{\boldsymbol{L}^2} \leq O(h^p) \tag{22}$$

for the error restricted to those cells of the mesh that are fully contained in the yielded subdomain. With p we denote the degree of the largest possible polynomial space P_p^{dc} that is a subset of the discrete stress and strain-rate space. Consequently, the error may be reduced by h-adaptivity, p-adaptivity or a combination thereof, i.e. by refining the mesh cells locally or by increasing the polynomial degree of the shape and test functions.

On cells that are cut by the yield surface, however, the approximation order is limited by the lack of smoothness of $\boldsymbol{f}(\boldsymbol{\tau}^k)$. In the direction perpendicular to the interface between yielded and unyielded regions, this function possesses a 'kink'. Such weak singularities are only poorly approximated by polynomial functions, resulting in a lower rate of convergence than in the smooth case (22) (cf [35, Ch 6]) and a possibly very large leading-order coefficient.

Thus, the total approximation error will usually be dominated by the contributions from cells on the yield surface. Local mesh refinement at least in the direction perpendicular to the boundary of any plugs is one theoretically admissible strategy to obtain a convergent algorithm—an approach implemented by Roquet and Saramito in [36]. Note that it is crucial to refine the mesh at a sufficiently high rate: for an optimal convergence rate of FISTA*, (21) imposes that the mesh be refined locally around the yield surface at a rate of

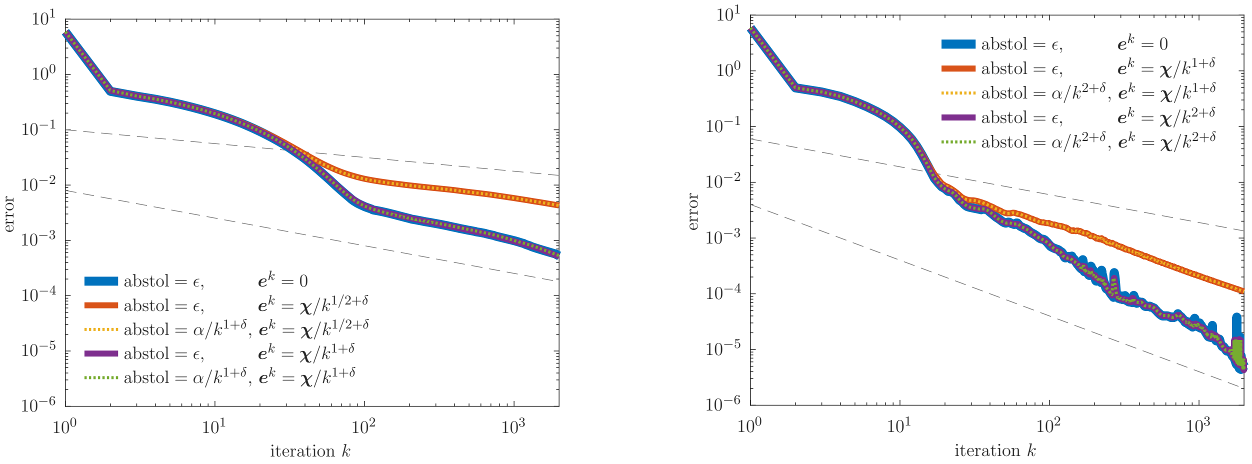
$$h \leq O(1/k^{2+\delta}).$$

Considering that due to the algorithms' sublinear convergence a relatively large number of iterations $k \sim 10^3$ is often necessary for a reasonable accuracy, it may be challenging to achieve the required mesh resolution.

Extended finite element methods (XFEM) have been designed as a generalisation of p-adaptive enrichment for similar problems that exhibit singularities across interfaces, see e.g. [37, 38]. In the context of viscoplastic fluid flow problems, the following approach may be viable:

1. Compute the level set $|\boldsymbol{\tau}^k| = \tau_0$ which defines the yielded-unyielded interface, and approximate it with a piecewise polynomial curve.
2. Enrich or replace polynomial shape functions of elements cut by these curves with more tailor-made shape functions, which also possess a weak singularity ('kink') in the direction perpendicular to this curve.

Further research will be required to assess the feasibility of such an approach in the context of numerical simulations of viscoplastic fluid flow.



(a) Convergence of Algorithm 1 (ADMM / ALG2) with graphs of $O(1/k^{1/4})$ and $O(1/k^{1/2})$

(b) Convergence of Algorithm 2 (FISTA*) with graphs of $O(1/k^{1/2})$ and $O(1/k)$.

Figure 8: Convergence of Algorithms 1 and 2 in the presence of errors in the strain rate and with exact or inexact solutions of the Stokes problems. Strain-rate errors may adversely affect the convergence of the algorithm, unless they decay sufficiently fast. In the latter case, the algorithms remain robust even with inexactly solved Stokes problems. Data: $\delta = 10^{-3}$, $\alpha \approx 6 \times 10^{-2}$. The entries of the tensor field $\boldsymbol{\chi}$ are randomly drawn from a uniform distribution on $[-0.1, 0.1]$.

Numerical experiments

We first investigate how general errors in the strain-rate affect the convergence of ADMM / ALG2 and FISTA*, then we compare Bercovier-Pironneau and Taylor-Hood elements for viscoplastic flow simulations.

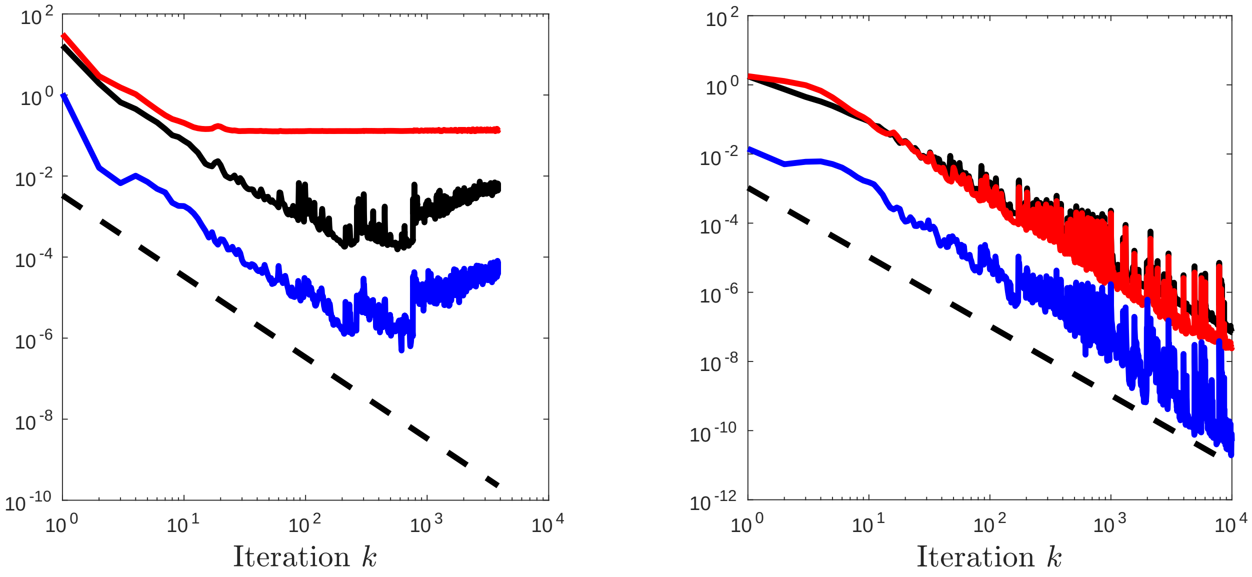
ADMM with strain-rate errors. We solve the rotating-plug problem once more, at this stage still with Bercovier-Pironneau elements for which no interpolation error occurs. Instead, we generate a strain-rate perturbation field $\boldsymbol{\chi}$, a piecewise constant function on the finer velocity grid whose function values are randomly drawn from a uniform distribution on the interval $[-0.1, 0.1]$. After evaluating the exact strain-rate iterate $\boldsymbol{d}^k$ as prescribed by Step 2 of Algorithm 1, we add an artificial error $\boldsymbol{e}^k = \boldsymbol{\chi}/k^{1+\delta}$ that decays at a rate sufficient for unobstructed convergence, or $\boldsymbol{e}^k = \boldsymbol{\chi}/k^{1/2+\delta}$, which decays slower. We perturb both the exact algorithm and the algorithm with inexact solutions of the Stokes problems.

In Figure 8a we show that strain-rate errors of order $O(1/k^{1/2+\delta})$ have a detrimental effect on convergence. No negative impact is observable when the accuracy of the strain-rate iterates improves sufficiently fast. Remarkably, even if both the velocity iterates and the strain-rate iterates are computed inexactly, the algorithm remains robust.

FISTA with strain-rate errors.* Figure 8b admits analogous conclusions. Provided that the sufficient decay rates for errors in the Stokes problems and the constitutive model are obeyed, FISTA* converges at the expected higher rate with no perceivable differences to the exact algorithm. In the presence of strain-rate errors that only decay at a rate proportional to $1/k^{1+\delta}$, convergence like $1/k^2$ still appears to be preserved, at least for the first 2,000 iterations. Local superconvergence effects have vanished, though. The accelerated algorithm, too, always converges in a robust manner even in the presence of velocity and /or strain-rate errors.

Bercovier-Pironneau vs Taylor-Hood elements. We now analyse more specifically the effect of the interpolation error that results from a Taylor-Hood discretisation with discontinuous, piecewise linear stress and strain-rate approximations.

In Figure 9 we compare how Algorithm 2 converges with a discretely consistent approximation using Bercovier-Pironneau elements and with an inconsistent approximation using Taylor-Hood elements. We do not employ any adaptation techniques such as mesh refinement. Hence, the Taylor-Hood interpolation error



(a) Discretisation with Taylor-Hood elements. Implementation in Rheolef.

(b) Discretisation with Bercovier-Pironneau elements. Implementation in FreeFem++.

Figure 9: Convergence of Algorithm 2 for the benchmark problem of force-driven Bingham flow; $\boldsymbol{L}^2$ norm of the velocity increments (blue), residuals (black) and the primal-dual gap (red). On a fixed grid, the interpolation error introduced by Taylor-Hood elements does not decay, leading to stagnation as soon as this error dominates.

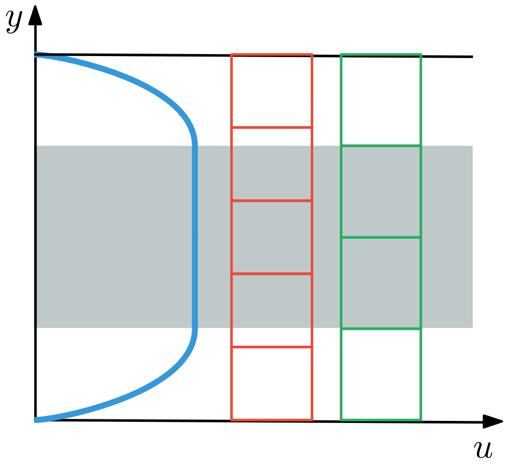
is of order $O(1)$ with respect to the iteration counter k. This systematic error $\boldsymbol{e}^k$ thus becomes increasingly notable as the optimisation algorithm progresses towards the solution, leading to increasingly perturbed updates until the method stagnates. Indeed, convergence of FISTA* is neither expected, nor is it observed in our numerical results. In constrast, no such issue occurs for the discretisation with Bercovier-Pironneau elements.

A viscoplastic flow problem in only one spatial dimension allows us to pinpoint that the dominant contribution to the interpolation error arises from the poor approximation quality in cells that are cut by the yield surface. We consider a simple Bingham-Poiseuille flow problem, for which the exact Algorithm 1 or 2 returns the solution after the first iteration. This it not generally true, however, when this problem is discretised with Taylor-Hood elements. In Figure 10a, we show that even after ten iterations the method fails to converge, unless the boundary of the central plug coincides exactly with the boundary of two mesh cells. Whenever the yielded-unyielded interface lies in the interior of an element, the results are of poor accuracy, even on a very fine mesh.

Practical consequences

There are two kinds of ‘special’ finite-element methods that are particularly interesting for simulations of yield-stress fluid flows: the Bercovier-Pironneau and the Brezzi-Douglas-Marini elements of first order. These schemes lead to a piecewise constant approximation of the strain-rate and shear-stress fields and are thus fully consistent at the discrete level. The convergence of the optimisation method is independent of the mesh resolution. Adaptive mesh refinement, e.g. to improve the spatial resolution of stagnant flow regions, is *optional.*

Higher order elements generally introduce a mesh-dependent consistency error. The optimisation algorithm does not converge, unless this error is decreased at a sufficiently high rate. Adaptive mesh refinement (or other model enrichment) is *mandatory.*



(a) Velocity profile of a Bingham-Poiseuille flow with the plug shaded in grey. If discretised with an even number of intervals across the channel width, the yielded-unyielded interface lies exactly on an element border.

n_y	$\|\boldsymbol{u} - \bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$	$\|\boldsymbol{d} - \mathcal{D}\bar{\boldsymbol{u}}\|_{\boldsymbol{L}^2}$
14	1.6e-14	4.5e-14
15	2.6e-03	1.8e-02
16	3.3e-15	2.2e-14
29	7.1e-04	6.7e-03
30	6.2e-13	1.1e-13
119	4.2e-05	8.1e-04
120	1.8e-12	3.3e-12

(b) Velocity / strain-rate errors of FISTA* after 10 iterations (using the P_2/P_1^{dc} element pair for velocity / strain rate) compared to the analytical solution $\bar{\boldsymbol{u}}$ of the Poiseuille flow problem. n_y is the number of elements across the channel width.

Figure 10: Poiseuille flow of a Bingham fluid. The yield stress is specifically chosen such that the central plug fills exactly half of the channel.

2.4. Meshing

Finally, we address one seemingly minor detail, which may nevertheless have a notable negative effect on convergence:

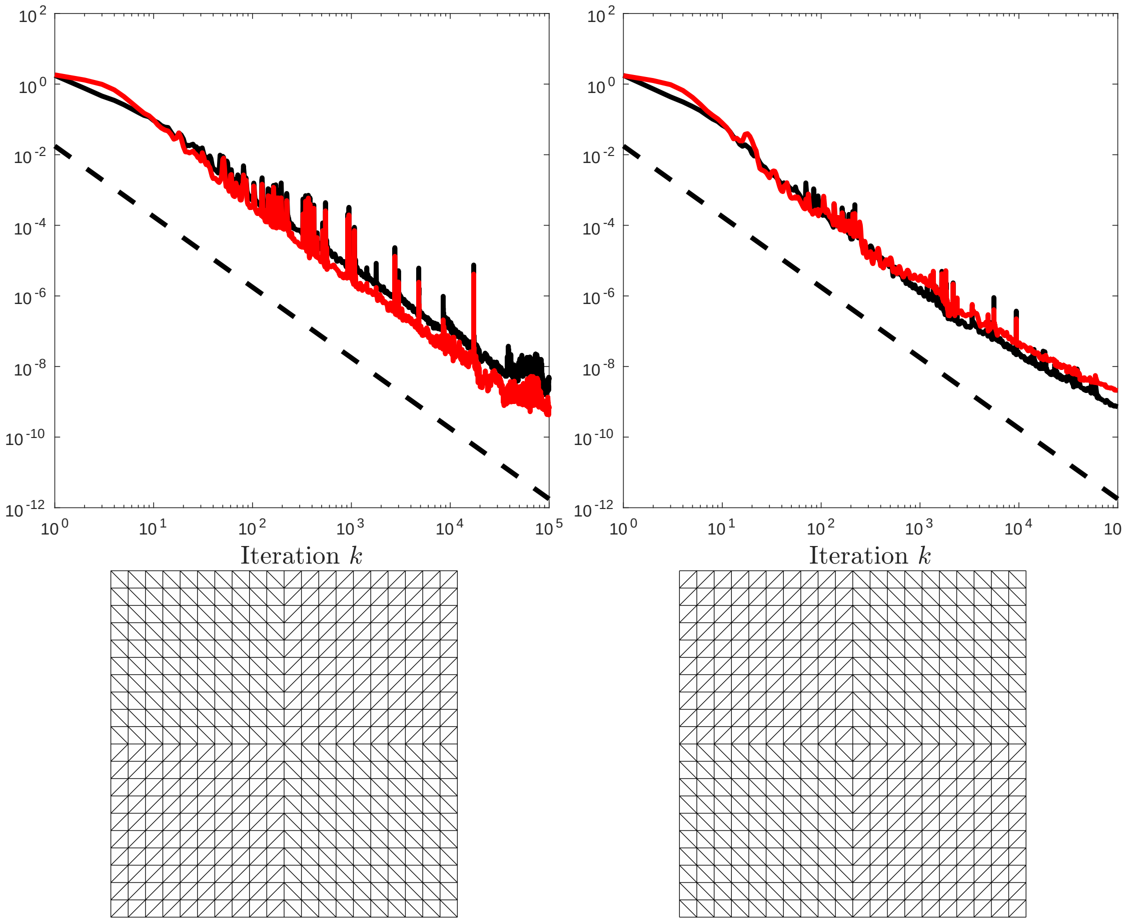
Some mesh generators, including the meshing module of FreeFem++, may produce corner triangles with two edges on the boundary of the domain, unless specifically configured otherwise. Certain elements, such as P_1 elements, then have no remaining degrees of freedom on such cells. The Stokes problems are hence overdetermined and solved in a least-squares sense. This case may be interpreted as an inconsistent discretisation as well, with analogous consequences as discussed in the previous subsection. Fortunately, any inconsistencies resulting from singular corner triangles are easy to eliminate.

In Figure 11 we show how the mesh topology may affect the convergence of FISTA*.

3. Case Study: wavy channel

In this section we demonstrate the capabilities of FISTA* for the resolution of complex and more practical flows. One such example is the Stokes flow of a Bingham fluid through a wavy channel. This flow has rich features that a good numerical algorithm should be able to capture accurately. In [39], the case of small wave amplitude (h) for long channels ($\delta = w/l \ll 1$, w: width, l: length of channel) is studied using asymptotic methods. It is shown that for $h < O(\delta)$ the plug at the centre of the channel is intact, additionally is wider at narrower sections of the channel which is counter-intuitive. Larger amplitudes $h \sim O(\delta)$ are considered in [40]. The central plug breaks into two islands at the widest and narrowest sections of the channel in this case and a pseudo-plug forms between the two. A pseudo-plug is a region with small strain-rates where the stress only exceeds the yield stress slightly. For large wave amplitudes $h = O(1)$, as examined in [41], a static zone forms in the widest section of the channel.

In Figure 12 we have used FISTA* to reproduce Figure 3 of [41] which was done using an augmented Lagrangian method and which shows the subsequent variations in the flow starting from asymptotically small wave amplitude to bigger $O(1)$ value at fixed Bingham number $B = 10$. We can see that FISTA* resolves all the features: the intact central plug at small wave amplitude, the broken central plug and pseudo-plug



(a)

(b)

Figure 11: The convergence plots and corresponding mesh (shown below) for two cases. a) Mesh with corner points divided between two elements b) Mesh with corner points owned by one element leading to overdetermined system. The black and red curves show strain rate residual and duality gap respectively and the dashed line shows the theoretical $1/k^2$ decay rate.

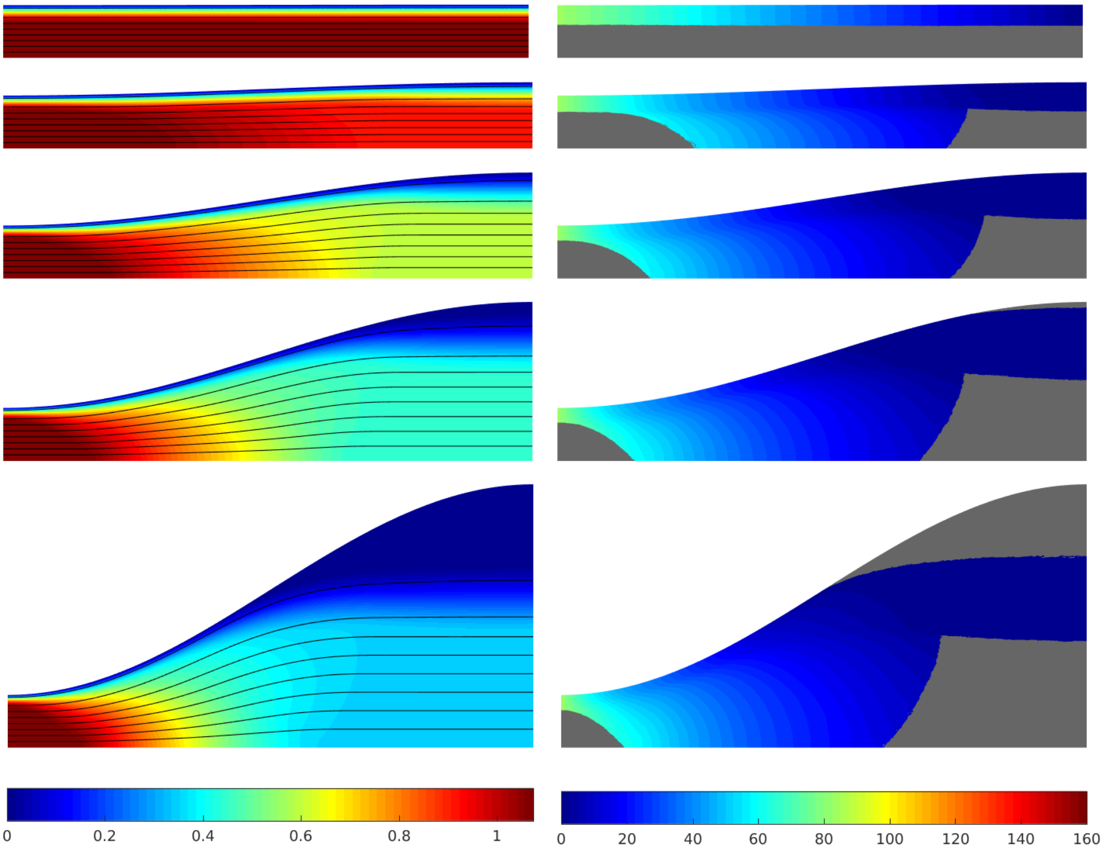


Figure 12: Figure 3 of [41] reproduced by FISTA*. Speed and streamline (left panel) and pressure (right panel) for $B = 10$ and wave amplitude $h = .01, .25, 1, 2, 4$. The color keys at bottom are valid for all h, gray regions on the right show unyielded regions.

at slightly bigger amplitude, the thin fouling layer at the widest section of the channel for $h = 2$ and finally, the large static zone for $h = 4$, all as shown in Figure 3 of [41].

As in [41], we used three cycles of mesh adaptation to increase the accuracy of computations. Note that adapting mesh elements to the yield surface can have significant effect on the accuracy of solutions as mentioned in Section 2.3. Fig 13 shows the final adapted mesh for the channel of $h = 2$ in Fig 12.

Fig 14 shows the comparison of convergence plots of FISTA* and the canonical augmented Lagrangian ALG2 for the wavy channel of $h = 2$ in Fig 12. Both implementations were done in FreeFem++ using exactly the same mesh, element type (velocity: P_1-iso-P_2, pressure: P_1, stress/strain rate: P_0-iso-P_2) and Stokes solver to make the comparison as fair as possible. We see that in ~8000 iterations FISTA* has narrowed the duality gap down to $\sim 10^{-6}$ while the augmented Lagrangian method reaches $\sim 10^{-4}$ which is a significant difference. Note that the main computational cost per iteration of both algorithms is the Stokes solver and thus for an equal number of iterations both take approximately equal time.

4. Conclusions

A fast optimisation algorithm, a carefully designed discretisation scheme and fine-tuned algebraic solvers are essential for convergent and efficient numerical simulations of yield-stress fluid flows:

1. The residual, i.e. the mismatch between the two numerical strain-rate fields, is neither an adequate stopping criterion nor a valid measure for convergence. Instead, we recommend to gauge convergence through the error estimator that is based on the primal-dual gap.

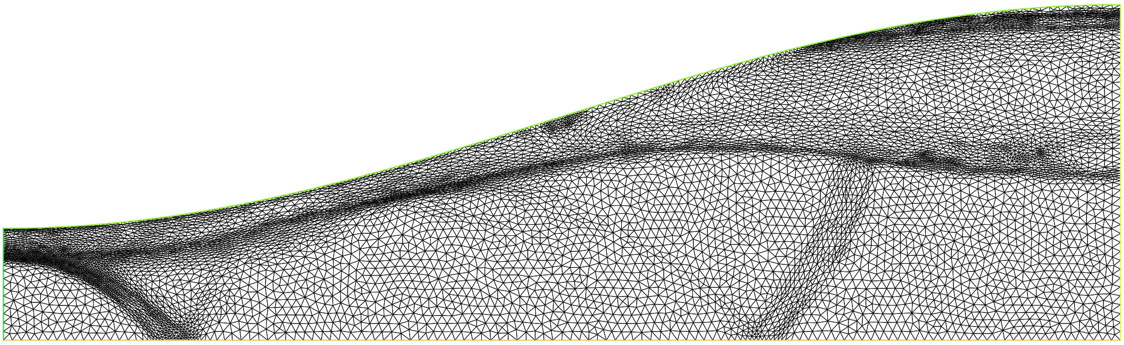
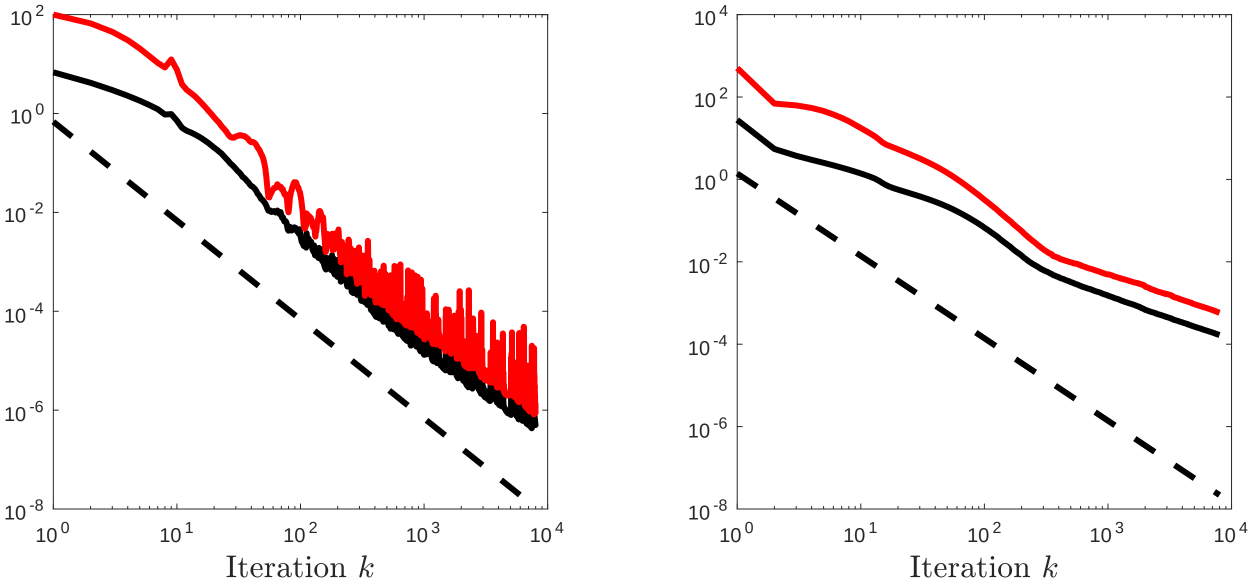


Figure 13: The mesh of wavy channel flow of $h = 2$ in Fig 12 after three cycles of adaptation.



(a) FISTA*

(b) Augmented Lagrangian (ALG2)

Figure 14: Convergence plots of FISTA* and augmented Lagrangian for the wavy channel flow. The black and red curves show the residuals and the duality gap respectively and the dashed line shows the theoretical $1/k^2$ decay rate of FISTA*.

2. The Stokes problems that arise in every iteration of the optimisation algorithm may be solved inexactly without affecting the order of convergence, provided that the tolerance decays at a sufficiently high rate. Inexact algorithms may reduce the computing time by a factor of two or even more.
3. Since all viscoplastic constitutive relations are inherently nonlinear and nonsmooth, an interpolation error is automatically introduced when the numerical strain-rate and stress fields are discretised with piecewise linear or higher-order elements. This interpolation error imposes a barrier for the optimisation algorithm, sooner or later resulting in stagnation. h, p or other forms of adaptive enrichment are mandatory for convergence. Only for piecewise constant stress and strain-rate approximations, which appear naturally in a discretisation with Bercovier-Pironneau or first-order Brezzi-Douglas-Marini elements, the interpolation error vanishes and convergence is generally guaranteed, independently of the underlying mesh.
4. Finally, mesh generators should be configured to compute triangulations with no singular corner vertices in order to avoid elements with no effective degrees of freedom. Else, the discrete Stokes problems will be solved in a least-squares sense, leading to unnecessary errors and inhibited convergence.

All of these conclusions are valid both for classical optimisation algorithms such as the augmented Lagrangian method ADMM / ALG2 and accelerated methods such as FISTA*. The simulation results of flow through the wavy channel demonstrated that FISTA* can achieve its fast theoretical convergence rate, provided that the necessary details are met. This order of magnitude speed-up is definitely a step forward to practical 3D computations of yield-stress flows and even for unsteady 2D flow. Finally, it is important to mention the value of open source software. It would have been very difficult or impossible to pinpoint and resolve some details (e.g. relative/absolute stopping criteria in Section 2.2) without inspecting the source code of the software.

In future work, we shall investigate the issue of error estimation for inexact algorithms further. Even though our practical guidelines apply analogously to yield-stress fluids governed by constitutive models other than the Bingham model, the extra smoothness of the Herschel-Bulkley or Casson models can likely be exploited for even more efficient algorithms. To generalise this work further, we also consider the full Navier-Stokes equations supplemented with a viscoplastic model an interesting computational challenge.

Acknowledgement

We would like to thank P. Saramito and F. Hecht for their advice on Rheolef and FreeFem++. Compute Canada is acknowledged for providing computational infrastructure. Ali Roustaei and Ian Frigaard acknowledge the support of NSERC and Schlumberger through CRD Project 444985-12.

References

[1] G. Duvaut, J. L. Lions, Inequalities in mechanics and physics, Grundlehren der mathematischen Wissenschaften, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

[2] T. Treskatis, Fast proximal algorithms for applications in viscoplasticity, Ph.D. thesis, University of Canterbury, Christchurch (2016).

[3] 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.

[4] M. Fortin, R. Glowinski, Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems, North-Holland, 1983.

[5] R. Glowinski, P. Le Tallec, Augmented Lagrangian and operator-splitting methods in nonlinear mechanics, Vol. 9, SIAM, 1989.

[6] R. Glowinski, A. Wachs, On the numerical simulation of viscoplastic fluid flow, in: P. G. Ciarlet (Ed.), Numerical Methods for Non-Newtonian Fluids, Vol. 16 of Handbook of Numerical Analysis, North-Holland, Amsterdam, 2011.

[7] A. Bermúdez, C. Moreno, Duality methods for solving variational inequalities, Computers & Mathematics with Applications 7 (1) (1981) 43–58.

[8] E. D. Fernández-Nieto, J. M. Gallardo, P. Vigneaux, Efficient numerical schemes for viscoplastic avalanches. Part 1: the 1D case, Journal of Computational Physics 264 (2014) 55–90.

[9] B. He, X. Yuan, On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers, Numerische Mathematik (2012) 1–11.

[10] B. He, X. Yuan, On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method, SIAM Journal on Numerical Analysis 50 (2) (2012) 700–709.

[11] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, in: Soviet Mathematics Doklady, Vol. 27, 1983, pp. 372–376.

[12] H. Attouch, J. Peypouquet, The rate of convergence of Nesterov's accelerated forward-backward method is actually faster than $1/k^2$, SIAM Journal on Optimization 26 (3) (2016) 1824–1834.

[13] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[14] A. Chambolle, C. Dossal, On the convergence of the iterates of the fast iterative shrinkage/thresholding algorithm, Journal of Optimization Theory and Applications 166 (3) (2015) 968–982.

[15] J. Cahouet, J.-P. Chabard, Some fast 3D finite element solvers for the generalized Stokes problem, International Journal for Numerical Methods in Fluids 8 (8) (1988) 869–895.

[16] S. Turek, Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, Vol. 6, Springer Science & Business Media, 1999.

[17] A. Aposporidis, E. Haber, M. A. Olshanskii, A. Veneziani, A mixed formulation of the Bingham fluid flow problem: analysis and numerical solution, Comput. Methods Appl. Mech. Engrg. 200 (29-32) (2011) 2434–2446. doi:10.1016/j.cma.2011.04.004.

[18] T. Treskatis, M. A. Moyers-González, C. J. Price, A trust-region SQP method for the numerical approximation of viscoplastic fluid flow, arXiv preprint.
URL http://arxiv.org/pdf/1504.08057

[19] P. Saramito, A damped Newton algorithm for computing viscoplastic fluid flows, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 6–15.

[20] M. Schmidt, N. L. Roux, F. R. Bach, Convergence rates of inexact proximal-gradient methods for convex optimization, in: Advances in Neural Information Processing Systems, 2011, pp. 1458–1466.

[21] D. Decker, H. Keller, C. Kelley, Convergence rates for Newtons method at singular points, SIAM Journal on Numerical Analysis 20 (2) (1983) 296–314.

[22] F. Hecht, New development in FreeFem++, Journal of Numerical Mathematics 20 (3-4) (2012) 251–265.

[23] P. Saramito, Efficient C++ finite element computing with Rheolef, CNRS-CCSD ed., 2015, http://cel.archives-ouvertes.fr/cel-00573970.

[24] J. C. De los Reyes, S. González Andrade, A combined BDF-semismooth Newton approach for time-dependent Bingham flow, Numerical Methods for Partial Differential Equations 28 (3) (2012) 834–860.

[25] M. Bercovier, O. Pironneau, Error estimates for finite element method solution of the Stokes problem in the primitive variables, Numerische Mathematik 33 (2) (1979) 211–224.

[26] T. Treskatis, TOOTHPASTE: The viscoplastic CFD app for MATLAB, version 1.1, https://www.mathworks.com/matlabcentral/fileexchange/62924-toothpaste (May 2017).

[27] P. Saramito, N. Roquet, An adaptive finite element method for viscoplastic fluid flows in pipes, Computer methods in applied mechanics and engineering 190 (40) (2001) 5391–5412.

[28] E. J. Dean, R. Glowinski, G. Guidoboni, On the numerical simulation of Bingham visco-plastic flow: old and new results, J. Non-Newtonian Fluid Mech. 142 (1) (2007) 36–62.

[29] Z. Yu, A. Wachs, A fictitious domain method for dynamic simulation of particle sedimentation in Bingham fluids, Journal of Non-Newtonian Fluid Mechanics 145 (2) (2007) 78–91.

[30] L. Muravleva, Uzawa-like methods for numerical modeling of unsteady viscoplastic Bingham medium flows, Applied Numerical Mathematics 93 (2015) 140–149.

[31] A. Wachs, I. A. Frigaard, Particle settling in yield stress fluids: limiting time, distance and applications, Journal of Non-Newtonian Fluid Mechanics 238 (2016) 189–204.

[32] N. Bernabeu, Modélisation multi-physique des écoulements viscoplastiques: application aux coulées de lave volcanique, Ph.D. thesis, Université de Grenoble, Grenoble (2015).

[33] O. Devolder, F. Glineur, Y. Nesterov, First-order methods of smooth convex optimization with inexact oracle, Mathematical Programming 146 (1-2) (2014) 37–75.

[34] P. G. Ciarlet, The finite element method for elliptic problems, Elsevier, 1978.

[35] D. Funaro, Polynomial approximation of differential equations, Vol. 8, Springer Science & Business Media, 2008.

[36] N. Roquet, P. Saramito, An adaptive finite element method for Bingham fluid flows around a cylinder, Comput. Methods Appl. Mech. Engrg. 193 (2003) 3317–3341.

[37] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: an overview of the method and its applications, International Journal for Numerical Methods in Engineering 84 (3) (2010) 253–304.

[38] T. Carraro, S. Wetterauer, On the implementation of the eXtended Finite Element Method (XFEM) for interface problems, arXiv preprint.
URL http://arxiv.org/pdf/1507.04238

[39] I. Frigaard, D. Ryan, Flow of a visco-plastic fluid in a channel of slowly varying width, Journal of Non-Newtonian Fluid Mechanics 123 (1) (2004) 6783. doi:10.1016/j.jnnfm.2004.06.011.
URL http://dx.doi.org/10.1016/j.jnnfm.2004.06.011

[40] 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 (1-3) (2009) 6277. doi:10.1016/j.jnnfm.2009.06.006.
URL http://dx.doi.org/10.1016/j.jnnfm.2009.06.006

[41] A. Roustaei, I. Frigaard, The occurrence of fouling layers in the flow of a yield stress fluid along a wavy-walled channel,

Journal of Non-Newtonian Fluid Mechanics 198 (2013) 109124. doi:10.1016/j.jnnfm.2013.03.005. URL http://dx.doi.org/10.1016/j.jnnfm.2013.03.005