J. Appl. Numer. Optim. 7 (2025), No. 1, pp. 113-140
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.7.2025.1.08

A PROXIMAL BUNDLE ALGORITHM FOR A CLASS OF QUASILINEAR VARIATIONAL INEQUALITIES OF THE SECOND KIND ARISING IN THE VISCOPLASTIC LAMINAR FLOW

SERGIO GONZÁLEZ-ANDRADE[1,2,*], DIEGO REYES[3]

[1] *Research Center in Mathematical Modeling and Optimization (MODEMAT), Quito, Ecuador*
[2] *Departamento de Matemática - Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito, Ecuador*
[3] *Facultad de Matemática, Astronomía, Física y Computación,*
Universidad de Córdoba, Av. Medina Allende, Córdoba, Argentina

Abstract. This paper focuses on the numerical solution of a variational inequality of the second kind, which arises as a model for the laminar flow of a Herschel-Bulkley fluid in the cross-section of a pipe. To tackle this problem, we develop a nonsmooth proximal bundle algorithm that bypasses the need for regularization techniques. We begin by formulating and analyzing an associated nonsmooth and convex optimization problem that characterizes the solution of the variational inequality. Following a discretize-then-optimize approach, we employ a first-order finite element discretization for the objective functional. The core of our method lies in the nonsmooth bundle algorithm, which leverages a Moreau-Yosida approximation combined with a quasi-Newton BFGS update. This approach approximates the function and gradient values through a finite inner bundle algorithm. We build and analyze the proposed algorithm, examining its convergence properties in the context of the flow model. Additionally, we demonstrate its efficiency through both theoretical analysis and numerical experiments.

Keywords. Bundle algorithms; BFGS algorithm; Herschel-Bulkley viscoplastic fluids; Moreau-Yosida regularization; p-Laplacian operator.

2020 Mathematics Subject Classification. 76M10 76M30, 90C30.

1. Introduction

Herschel-Bulkley fluids are a class of non-Newtonian fluids characterized by a yield stress and a non-linear relationship between the applied shear stress and the resulting shear rate. These fluids exhibit a combination of viscous, plastic, and sometimes elastic behavior, making their flow properties more complex than those of Newtonian fluids. They can be found in both natural systems—such as mud, quicksand, and blood—and in industrial products, including foodstuffs like tomato sauce and industrial materials like clay suspensions. Their applications are particularly prominent in the food, chemical, and oil industries, where understanding their unique flow properties is critical for optimizing production processes.

[*] Corresponding author.

E-mail address: sergio.gonzalez@epn.edu.ec (S. González-Andrade), diego.reyes.andrade@mi.unc.edu.ar (D. Reyes).

Received 16 September 2024; Accepted 10 February 2025; Published online 20 March 2025.

The Herschel-Bulkley model introduces two key parameters that define the fluid's behavior: the yield stress (or plasticity threshold) and the flow index. The yield stress represents the minimum stress required for the material to begin flowing, allowing it to transition from behaving as a rigid solid to a fluid. The flow index determines the degree of shear-thinning or shear-thickening behavior: fluids with a flow index less than 1 (shear-thinning) become less viscous under increased stress, as seen in materials like ketchup and paint, while those with an index greater than 1 (shear-thickening) become more viscous, such as the well-known cornstarch and water mixture, oobleck.

In this paper, we focus on the numerical resolution of the following variational inequality, whose solution characterizes the velocity of a stationary and laminar flow of a Herschel-Bulkley fluid in a pipe with cross-section given by Ω

$$\int_\Omega |\nabla u|^{n-1}(\nabla u\,,\nabla(v-u))\,dx + g\int_\Omega |\nabla v|\,dx - g\int_\Omega |\nabla u|\,dx \geq \int_\Omega f(v-u)\,dx, \tag{1.1}$$

where $0 < n < \infty$ stands for the flow index, $g > 0$ represents the yield stress and $f \in L^{n+1'}(\Omega)$.

Solving (1.1) is challenging due to the nonsmooth terms involved in the variational inequality. There are two major approaches to address this difficulty. The first involves nonsmooth methods, such as augmented Lagrangian schemes, as employed in [13]. In this work, the authors introduce two penalty terms into the functional in (1.1), and by solving a system of nonlinear equations, they iteratively tackle the minimization problem. The second approach is based on regularization techniques. For instance, in [10], the author applied a Huber-type regularization to smooth the nondifferentiable part of the functional, converting the problem into a differentiable one. A preconditioned descent algorithm was then designed for the numerical resolution of the smoothed problem. Similarly, in [12], the authors again used a Huber-type regularization and develop a multigrid optimization algorithm to solve the problem efficiently.

From our perspective, this problem has not been sufficiently explored within the framework of Non-Smooth Optimization (NSO) and its existing numerical methods for tackling such issues. NSO is a well-established field of study, applied across diverse areas like mechanics, computational chemistry, machine learning, and data mining (see [3, Chapter 7]). In the mid-1970s, NSO gained traction with the development of bundle methods, which remain among the most efficient and reliable techniques for solving non-smooth problems (see [3, 15, 16]). The primary goal of this paper is to construct an efficient and robust algorithm, grounded in NSO and bundle methods, that solves problem (1.1) without regularization, while maintaining efficiency comparable to other faster existing methods.

To construct the algorithm, we adopt a discretize-then-optimize approach, using a first-order finite element discretization. For the nonsmooth discrete problem, we apply the Moreau-Yosida penalization to obtain a smooth optimization problem that is equivalent to solving the original one. The primary challenge with this regularization is that we cannot compute the exact values of the objective function and its gradient at every point. This is where bundle methods prove crucial, as they allow us to approximate these values. Specifically, using available information from previous iterations, a cutting-plane model approximates the objective function. This enables us to formulate an inner optimization problem that returns approximated values of both the objective function and its gradient at any given point. With these approximations, we then propose a quasi-Newton algorithm, utilizing a BFGS-type update, for obtaining a descent direction in an efficient way.

The paper is organized as follows. In Section 2, we describe the subdifferential of the functional J in its functional form and discuss the discretization of the problem by means of the finite element method. In Section 3, we give a detailed construction of the nonsmooth algorithm and analyze its convergence properties, particularly, we prove global and superlinear convergence rate of the method. Section 4 is devoted to presenting the numerical results obtained. First we discuss the main issues regarding the implementation of our algorithms. Mainly, the discretization techniques and the implementation of an oracle which determines a subgradient of ∂J. Next, several numerical experiments, which illustrate the main features of the proposed approach, are carried out. Finally, in Section 5, we outline conclusions on this work and discuss some challenging issues that can be analyzed in future contributions.

2. Problem Statement

This paper is devoted to the numerical resolution of the following problem

$$\min_{u\in W_0^{1,p}(\Omega)} J(u)=\frac{1}{p}\int_\Omega |\nabla u|^p\,dx+g\int_\Omega |\nabla u|\,dx-\int_\Omega fu\,dx, \tag{2.1}$$

where $1<p<\infty$, $g>0$, and $f\in L^{p'}(\Omega)$. In this context, g represents the yield stress of the material, p is the flow index, and f accounts for the pressure decay in the pipe. It is well known that problem (1.1) expresses a necessary optimality condition for minimization problem (2.1). Therefore, the upcoming sections will focus on this optimization problem. Some clarifications are in order: We assume $p=n+1$, where n is the usual flow index, so that the problem aligns with the classical structure of the p-Laplacian. Under this consideration, shear-thinning flow occurs for $1<p<2$, while shear-thickening flow is observed for $p>2$. With these in mind, we can conclude that the solution of (2.1) describes the velocity field of the stationary laminar flow of a Herschel-Bulkley fluid in a pipe whose cross-section is represented by Ω.

The existence and uniqueness of solutions for (2.1) were thoroughly analyzed in, for instance, [13, 10].

In the remainder of this section, we discuss several important properties of J, with a particular focus on its regularity characteristics.

2.1. **Subdifferential of** $J(\cdot)$.

As noted in [15], bundle methods rely on the assumption that only the objective function value and a subgradient at each point are available. To incorporate the bundle concept into the NSO method for numerically solving (2.1), we first examine the subgradients of $J(\cdot)$ by analyzing its subdifferential at each point. In line with the approach of [8], we then reformulate the functional as follows

$$J(u)=J_1(u)+gJ_2(u),$$

where

$$J_1(u)=\frac{1}{p}\int_\Omega |\nabla u|^p\,dx-\int_\Omega fu\,dx \quad \text{and} \quad J_2(u)=\int_\Omega |\nabla u|\,dx.$$

Next, since $J_1(\cdot)$ and $J_2(\cdot)$ are weakly lower semicontinuous functionals, we can assert that the subdifferential of J at $u\in W_0^{1,p}(\Omega)$ is given by (see [8])

$$\partial J(u)=\partial J_1(u)+g\partial J_2(u).$$

This formulation of $J(\cdot)$ significantly simplifies the analysis of its subdifferential and, consequently, the characterization of the associated subgradients.

First, we focus on $J_1(\cdot)$, which corresponds to the energy functional associated with the p-Laplacian operator. It is well known that this functional is Fréchet differentiable on $W_0^{1,p}(\Omega)$ (see, for instance, [2, Section 2.6]). Therefore, we conclude that its subdifferential simplifies to

$$\partial J_1(u) = \left\{J_1'(u)\right\}, \text{ for } u \in W_0^{1,p}(\Omega), \tag{2.2}$$

where J_1' represents the Fréchet derivative of J_1 and is given by

$$J_1'(u)v = \int_\Omega |\nabla u|^{p-2} (\nabla u, \nabla v) \; dx - \int_\Omega f v \, dx, \; \forall v \in W_0^{1,p}(\Omega).$$

Now, we turn our attention to $J_2(\cdot)$. Clearly, this functional is not differentiable at 0. However, it is known that it is Féchet differentiable at $u \in W_0^{1,p}(\Omega) \setminus \{0\}$ with (see, for instance, [2, Sec. 2.6])

$$J_2'(u)v = \int_\Omega \frac{(\nabla u, \nabla v)}{|\nabla u|} \, dx, \; \forall v \in W_0^{1,p}(\Omega).$$

Returning to the subdifferential of $J_2(\cdot)$ at 0, we have by definition that

$$\partial J_2(0) = \left\{ \psi \in W^{-1,p'}(\Omega) : \; J_2(v) \geq \langle \psi, v \rangle, \; \forall \, v \in W_0^{1,p}(\Omega) \right\},$$

which induces the following characterization of the subgradients of $J_2(\cdot)$ at 0

$$\psi \in \partial J_2(0) \Leftrightarrow \langle \psi, v \rangle \leq \int_\Omega |\nabla v| \; dx, \quad \forall \, v \in W_0^{1,p}(\Omega). \tag{2.3}$$

Finally, (2.2), (2.3), and [5, Proposition 9.20] allow us to obtain the following characterization for the subdifferential of J

$$\partial J(u) = \begin{cases} \int_\Omega |\nabla u|^{p-2} (\nabla u, \nabla v) \; dx + g \int_\Omega \frac{(\nabla u, \nabla v)}{|\nabla u|} \, dx - \int_\Omega f v \, dx, \; \forall v \in W_0^{1,p}(\Omega), & \text{if} \quad u \neq 0, \\ g \int_\Omega (\Psi, \nabla v) \; dx - \int_\Omega f v \, dx, \; \forall v \in W_0^{1,p}(\Omega), & \text{if} \quad u = 0. \end{cases} \tag{2.4}$$

Here, $\Psi := (\Psi_1, \ldots, \Psi_n)$ with $\Psi_i \in L^{p'}(\Omega)$ for $i = 1, \ldots, n$.

Remark 2.1. The existence of the yield stress introduces an inherent discontinuity in the Herschel-Bulkley model, which is reflected in the nonsmooth term $\int_\Omega |\nabla u| \, dx$. As stated in the introduction, there are two main approaches for analyzing and numerically solving the resulting variational inequality of the second kind: smoothing-regularization techniques and nonsmooth optimization methods.

Regarding regularization techniques for VIs, there are two general approaches: global and local regularization schemes. In the first group, methods such as the Bercovier-Engelman and Papanastasiou schemes generate a smooth approximation across the entire domain, which means that the approximation to the exact solution is only achieved in the limit (see [4, 21, 22]). On the other hand, local regularization techniques, such as the Huber or bi-viscosity schemes (see [10, 12]), apply smoothing only within small neighborhoods around the discontinuity points while preserving the original model in the rest of the domain. The main advantage of using regularization schemes is that the resulting smoothed systems can be solved using fast-converging algorithms, typically based on Newton or generalized Newton-type schemes. However, the introduction of smoothing may have physical drawbacks, as it modifies the original model and can introduce perturbations in the computed solutions.

Nonsmooth optimization methods, by contrast, allow the analysis of the problem without altering the original model, preserving its intrinsic structure (see [9, 13]). However, these methods are generally slower and less efficient compared to regularization-based approaches. To address this limitation, in this paper, we explore the use of bundle methods in combination with a preconditioned Moreau-Yosida scheme, where the BFGS update is employed as a preconditioner in the associated metric (see (3.2)). This approach results in a nonsmooth algorithm that achieves, at least locally, superlinear convergence, providing an efficient scheme while maintaining the original nature of the model.

2.2. **Finite element discretization.** As mentioned in the introduction, we adopt a discretize-then-optimize approach for addressing problem (2.1). In this context, before detailing the NSO method, we first explore a first-order finite element discretization of $J(\cdot)$ and its subdifferential $\partial J(\cdot)$. Additionally, by means of the Galerkin method, we reformulate the problem in $\mathbb{R}^n$, setting the stage for the subsequent numerical analysis.

Let Ω be a nonempty, open, bounded, and sufficiently regular domain. Let $\mathscr{T}^h$ be a regular triangulation of Ω in the sense of Ciarlet (see [6, p. 38]). Next, let Ω^h be a polygonal approximation of Ω, given by

$$\Omega^h = \bigcup_{T \in \mathscr{T}^h} \overline{T},$$

where all the triangles T are disjoint one to one and whose diameter is bounded by h. Further, for any pair of triangles, their closures are either disjoint or have a common vertex or a common edge. Finally, let $\{P_j\}_{j=1}^N$ be the vertices (nodes) associated to the triangulation $\mathscr{T}^h$. Hereafter, we assume that $P_j \in \partial\Omega^h$ implies that $P_j \in \partial\Omega$ and that $\Omega^h \subset \Omega$.

Taking all this into account, we introduce the following finite-dimensional space associated to the triangulation $\mathscr{T}^h$

$$W_0^h := W_0^{1,p}(\Omega) \cap \{v \in C(\overline{\Omega^h}) : v|_\tau \in \mathbb{P}_1, \forall T \in \mathscr{T}^h \text{ and } v = 0 \text{ on } \partial\Omega^h\},$$

where $\mathbb{P}_1$ is the space of polynomials with degree less than or equal to 1.

Together with these considerations, it is possible to define the following finite element approximation of problem (2.1):

$$\min_{u^h \in W_0^h} J_h(u^h) := \frac{1}{p}\int_{\Omega^h} |\nabla u^h|^p dx + g\int_{\Omega^h} |\nabla u^h| dx - \int_{\Omega^h} f u^h dx. \tag{2.5}$$

The existence of a unique solution for (2.5) directly follows from the fact that W_0^h is a closed subspace of $W_0^{1,p}(\Omega)$ (see [9, Sec. 3.2]).

In the same manner, we introduce the following finite element approximation of the subdifferential $\partial J(\cdot)$:

$$\partial J_h(u^h) = \begin{cases} \int_\Omega |\nabla u^h|^{p-2}(\nabla u^h, \nabla v^h)\, dx + g\int_\Omega \frac{(\nabla u^h, \nabla v^h)}{|\nabla u^h|}\, dx - \int_\Omega f v^h\, dx & \text{if } u^h \neq 0, \\ g\int_\Omega (\psi, \nabla v^h)\, dx - \int_\Omega f v^h\, dx & \text{if } u^h = 0, \end{cases} \tag{2.6}$$

where ψ is a function that satisfies (2.4).

In order to have a discrete version of equations (2.5) and (2.6), we follow [10, Sec. 4.1]. In this aim, let $\{\varphi_j\}$, $j = 1, \ldots, n$, be the basis functions of W_0^h, and let us denote by $\vec{u}$ the vector that stores the associated coefficients of the approximated functions u^h. Assuming that $card(T^h)=m$, we use the following discrete version of the gradient

$$\nabla^h := \begin{pmatrix} \partial_1^h \\ \partial_2^h \end{pmatrix} \in \mathbb{R}^{2m\times n}, \tag{2.7}$$

where $(\partial_1^h)_{k,j} := \frac{\partial \varphi_j(x)}{\partial x_1}|_{T_k}$ and $(\partial_2^h)_{k,j} := \frac{\partial \varphi_j(x)}{\partial x_2}|_{T_k}$, for $j=1,\dots,n$, $k=1,\dots,m$ and $T_k \in \mathscr{T}^h$. Note that $\frac{\partial \varphi_j(x)}{\partial x_1}|_{T_k}$ and $\frac{\partial \varphi_j(x)}{\partial x_2}|_{T_k}$ are the constant values of $\frac{\partial \varphi_j}{\partial x_1}$ and $\frac{\partial \varphi_j}{\partial x_2}$ over each triangle T_k, respectively. Therefore, $\nabla^h \vec{u}$ is the discrete approximation of $\nabla u^h(x)$.

Now, let us define the function $\eta : \mathbb{R}^{2m} \to \mathbb{R}^m$ by

$$\eta(w)_k = |(w_k, w_{k+m})|^T, \quad k=1,\dots,m.$$

This function allows us to approximate $|\nabla u^h(x)|$ by $\eta(\nabla^h \vec{u})$, considering that $\eta(\nabla^h \vec{u})_k$ represents the value of $|\nabla u^h(x)|$ in each triangle $T_k \in \mathscr{T}^h$.

Next, by applying Galerkin's method, the approximation of the first term in J_h is given by

$$\frac{1}{p}\int_\Omega |\nabla u^h|^{p-2}(\nabla u^h, \nabla u^h)dx \approx \frac{1}{p}\sum_{i,j=1}^n u^i u^j \sum_{T_k\in T^h}\int_{T_k}\Big(\eta(\nabla^h \vec{u})_k\Big)^{p-2}(\nabla\varphi_i, \nabla\varphi_j)\,dx,$$

where $\big(\eta(\nabla^h\vec{u})_k\big)^{p-2}$ is constant in every triangle T_k. Thus, if we introduce the matrix $A^h(\vec{u}) \in \mathbb{R}^{n\times n}$ with entries given by

$$(a(\vec{u}))_{i,j} = \sum_{T_k\in T^h}\Big(\eta(\nabla^h\vec{u})_k\Big)^{p-2}\int_{T_k}(\nabla\varphi_i,\nabla\varphi_j)dx,$$

we obtain that

$$\frac{1}{p}\int_\Omega |\nabla u^h|^{p-2}(\nabla u^h,\nabla u^h)dx \approx \frac{1}{p}\vec{u}^\top A(\vec{u})\vec{u}. \tag{2.8}$$

On the other hand, we approximate the second term as follows

$$g\int_\Omega |\nabla u^h|\,dx \approx g\sum_{T_k\in T^h}\int_{T_k}\eta(\nabla^h\vec{u})_k\,dx =: I(\vec{u}),$$

where $I(\cdot)$ stands for a suitable quadrature rule. This quadrature rule also will be applied to $\int_\Omega f^h\varphi_j\,dx$ to generate the term $\vec{f}$.

Summarizing, the discrete approximation of equation (2.5) is given as

$$\mathscr{J}_h(\vec{u}) = \frac{1}{p}\vec{u}^\top A(\vec{u})\,\vec{u} + I(\vec{u}) + (\vec{f})^\top\vec{u}. \tag{2.9}$$

This will be our objective functional in the following sections.

In order to approximate the subdifferential of $\mathscr{J}_h(\cdot)$, we proceed in the same way as before to obtain that

$$\partial\mathscr{J}_h(\vec{u}) = \begin{cases} \vec{u}^\top A(\vec{u})\vec{v} + \vec{u}^\top\tilde{A}(\vec{u})\vec{v} - (\vec{f}^h)^\top\vec{v},\ \forall\vec{v}\in\mathbb{R}^n, & \text{if } u^h \neq 0,\\ g\vec{\Psi}^\top\vec{v} - (\vec{f}^h)^\top\vec{v},\ \forall\vec{v}\in\mathbb{R}^n, & \text{if } u^h = 0.\end{cases}$$

Here, $\tilde{A}(\vec{u}) \in \mathbb{R}^{n\times n}$ is a matrix with entries given by

$$(\tilde{a}(\vec{u}))_{i,j} = \sum_{T_k\in T^h}(\eta(\nabla^h\vec{u})_k)^{-1}\int_{T_k}(\nabla\varphi_i,\nabla\varphi_j)\,dx,$$

while $\vec{\Psi}$ is obtained by applying a suitable quadrature formula to the term $\int_\Omega (\psi, \nabla v^h)\, dx$.

Remark 2.2. Matrices $A(\vec{u})$ and $\tilde{A}(\vec{u})$ are weighted stiffness matrices, and they have the same construction just like in [10, 11].

Equations (2.9) and (2.2) are the discrete approximations of J_h and $\partial J_h(\cdot)$ in its functional form, respectively. From now on, we assume that J_h and ∂J_h are in their discrete form. Consequently, it is possible to describe the NSO bundle algorithm for the numerical resolution of problem (2.1).

Remark 2.3. Without causing any ambiguity, we can remove the vector notation "→" for simplicity of writing.

3. Nonsmooth Proximal Bundle Algorithm

In this section, we incorporate the core ideas of the quasi-Newton bundle-type algorithm introduced in [18] with the aim of developing the NSO optimization method to solve the following optimization problem

$$\min_{u\in\mathbb{R}^n} \mathscr{J}_h(u), \tag{3.1}$$

where $\mathscr{J}_h(\cdot)$ is defined as in (2.9). It is clear that $\mathscr{J}_h(\cdot)$ is a strictly convex and continuos functional. Also, we know explicitly its subdifferential at every point (see (2.2)).

3.1. Moreau-Yosida regularization.

One way to apply a quasi-Newton algorithm to a nonsmooth problem is by means of the Morea-Yosida regularization (see, e.g., [14, 17, 18]). In this regard, let M be a positive definite symmetric matrix. The Moreau-Yosida regularization of $\mathscr{J}_h$ is given by

$$F_M(u) := \min_{v\in\mathbb{R}^n} \left\{ \mathscr{J}_h(v) + \frac{1}{2}\|v-u\|_M^2 \right\}, \tag{3.2}$$

where $\|x\|_M := \sqrt{x^T M x}$. F_M is a continuously differentiable convex function. Furthermore, the derivative of F_M at u is defined by

$$G_M(u) := \nabla F_M(u) = M(u - p_M(u)) \in \partial \mathscr{J}_h(p_M(u)),$$

where $p_M(u)$ is the unique solution of (3.2) and is called the proximal point of u. In addition, G_M is globally Lipschitz continuous with modulus $\|M\|$, and minimizing F_M is equivalent to minimizing $\mathscr{J}_h$, that is,

$$\bar{u} = \arg\min_{u\in\mathbb{R}^n} \mathscr{J}_h(u) \Leftrightarrow \bar{u} = \arg\min_{u\in\mathbb{R}^n} F_M(u).$$

Therefore, $\bar{u}$ minimizes $\mathscr{J}_h$ if and only if $G_M(\bar{u}) = 0$ and $p_M(\bar{u}) = \bar{u}$ (see [18] and the references therein).

The computation of $F_M(u)$ requires solving the optimization problem on the right-hand side of (3.2). Instead of solving it directly, we employ the bundle method (see [3, Chap. 12]) to implement a routine that provides approximate values for both F_M and G_M (see [17, 18]).

The matrix M plays a pivotal role in the following sections. As shown in (3.2), it defines the Moreau-Yosida penalty metric, which "controls" the approximation and the determination of the descent direction in the algorithm. A key idea in [18] is to combine this approximation with a BFGS-type update. This not only preserves the positive definiteness of matrix M at each iteration but also ensures superlinear local convergence rates for the algorithm.

3.2. **Appending the bundle concept.** Let u be given in $\mathbb{R}^n$. By taking $w := v - u$ in (3.2) and minimizing over w instead of v, we have

$$F_M(u) = \min_{w \in \mathbb{R}^n} \left\{ \mathscr{J}_h(u+w) + \frac{1}{2} w^\top M w \right\},$$

and $p_M(u) = u + \bar{w}$, where $\bar{w} = \arg\min_{w \in \mathbb{R}^n} \left\{ \mathscr{J}_h(u+w) + \frac{1}{2} w^T M w \right\}$ (see [18]).

Since $\mathscr{J}_h(\cdot)$ is convex, we approximate $\mathscr{J}_h(u+w)$ by the following cutting plane model (see [17, 18])

$$\check{\mathscr{J}}_h(u+w) = \max_{j=1,\ldots,l} \left\{ \mathscr{J}_h(v^j) + (\xi^j)^\top (u+w-v^j) \right\},$$

where the triplets $\left(v^j, J_h(v^j), \xi^j = \xi(v^j) \in \partial \mathscr{J}_h(v^j)\right)$ constitute an array $\mathscr{B}$ that has been generated sequentially starting from u and $\xi(u) \in \partial \mathscr{J}_h(u)$, or possibly, a certain subset of past iterations used to generate u.

Now, let us define the following linearization error

$$e(u, v^j) = \mathscr{J}_h(u) - \mathscr{J}_h(v^j) - (\xi^j)^T (u - v^j).$$

Consequently, $\check{\mathscr{J}}_h(u+w)$ can be written as

$$\check{\mathscr{J}}_h(u+w) = \mathscr{J}_h(u) + \max_{j=1,\ldots,l} \left\{ (\xi^j)^T w - e(u, v^j) \right\}. \tag{3.3}$$

Thus, we define $\check{F}_M$ as follows

$$\begin{aligned} \check{F}_M(u) &:= \min_{w \in \mathbb{R}^n} \left\{ \check{\mathscr{J}}_h(u+w) + \frac{1}{2} w^\top M w \right\} \\ &= \mathscr{J}_h(u) + \min_{w \in \mathbb{R}^n} \left\{ \max_{j=1,\ldots,l} \left\{ (\xi^j)^\top w - e(u, v^j) \right\} + \frac{1}{2} w^\top M w \right\}. \end{aligned} \tag{3.4}$$

Since $\mathscr{J}_h(u+w) \geq \check{\mathscr{J}}_h(u+w)$, it follows from the definition of F_M that $\check{F}_M(u) \leq F_M(u)$.

On the other hand, the auxiliary minimization problem in (3.4) can be reformulated as a quadratic optimization problem, similar to the approach in [17]. Letting

$$\nu = \max_{j=1,\ldots,l} \left\{ (\xi^j)^\top w - e(u, v^j) \right\},$$

the following optimization problem becomes equivalent to the inner optimization problem in (3.4)

$$\begin{aligned} \min_{(w,\nu)} \quad & \nu + \frac{1}{2} w^\top M w \\ \text{s.t.} \quad & -e(u, v^j) + (\xi^j)^\top w \leq \nu, \quad j = 1, \ldots, l. \end{aligned} \tag{3.5}$$

The optimality conditions of problem (3.5) imply the existence of non-negative multipliers $\lambda_j(u)$ that sum one and verify

$$Mw(u) = -\sum_{j=1}^{l} \lambda_j(u) \xi^j,$$

$$\nu(u) = (\xi^j)^{\mathrm{T}} w(u) - e\left(u, v^j\right), \text{ if } \lambda_j(u) > 0,$$

with $(w(u), \nu(u))$ the unique solution of (3.5) (see [15, 17]). Therefore, is follows that

$$\check{F}_M(u) = \mathscr{J}_h(u) + \nu(u) + \frac{1}{2}(w(u))^\top M w(u).$$

Further, let $a(u) = u + w(u)$ be an approximation of $p_M(u)$ and let

$$\hat{F}_M(u) = \mathscr{J}_h(a(u)) + \frac{1}{2}(w(u))^\top M w(u).$$

Since $p_M(u)$ is the unique minimizer of (3.2), it follows that $F_M(u) \leq \hat{F}_M(u)$, equality holds if and only if $a(u) = p_M(u)$. Thus we have the similar statements of Lemma 2.1 in [18].

(1) $F_M(u) \leq F_M(u) \leq \hat{F}_M(u)$,
(2) $F_M(u) = \hat{F}_M(u)$ if and only if $a(u) = p_M(u)$.

Let

$$\varepsilon(u) = \hat{F}_M(u) - \check{F}_M(u). \tag{3.6}$$

We accept that $a(u)$ is an approximation of $p_M(u)$ if the next inequality is satisfied

$$\varepsilon(u) \leq \delta(u) \min\left\{(w(u))^\top M w(u), N\right\}, \tag{3.7}$$

where $\delta(u)$ and N are given positive numbers (see [18]), and $\delta(u)$ remains fixed during the bundle subprocess. If (3.7) is not verified, we let $v^{j+1} = u + w(u)$ and $\xi^{j+1} \in \partial \mathscr{J}_h(v^{j+1})$, add $(\xi^{j+1})^T d - e(u, v^{j+1})$ to (3.3). Replace $j = j+1$ and update $\mathscr{B} = \mathscr{B} \cup \left\{\left(v^{j+1}, J_h(v^{j+1}), \xi^{j+1}\right)\right\}$. Then, solve the new subproblem in (3.5) which gives a new $w(u)$ and $\varepsilon(u)$ to be tested in (3.7).

In the subsequent algorithm, we describe the bundle sub-process that returns an approximation of $p_M(u)$.

Algorithm 1 Bundle sub-process

```
Step 0. Call the oracle to obtain $\mathscr{J}_h(u)$ and $\xi \in \partial \mathscr{J}_h(u)$, initialize $\mathscr{B} = \{(u, \mathscr{J}_h(u), \xi)\}$, or use a previous $\mathscr{B}$ that saves information from past iterations and has been used to generate $u$.
Step 1. Find $w(u)$ by solving problem (3.5), then fix $a(u) = u + w(u)$, and call the oracle to obtain $(\mathscr{J}_h(a(u)), \xi^a)$.
Step 2. Determine $\check{F}_M(u)$ and $\hat{F}_M(u)$, and $\varepsilon(u)$ as in (3.6).
Step 3. Check if (3.7) is satisfied, if so STOP. Otherwise, let $v^{j+1} = a(u)$ and $\xi^{j+1} = \xi^a$, append $(\xi^{j+1})^T d - e(u, v^{j+1})$ to (3.3). Update $j = j+1$ and $\mathscr{B} = \mathscr{B} \cup \left\{\left(v^{j+1}, f(v^{j+1}), \xi^{j+1}\right)\right\}$; return to step 1.
Result: $a(u)$ approximation of $p_M(u)$.
```

Remark 3.1. Let $\tilde{G}_M(u) := M(u - a(u)) = -Md(u)$. Thus $\tilde{G}_M(u)$ is accepted as an approximation of $G_M(u)$, which depends on $\varepsilon(u)$. Therefore, it is possible to calculate the gradient of F_M approximately (see [18, Lemma 2.3]). In addition, if u does not minimize J_h, then Algorithm 1 ends in a finite number of steps (see [18, Lemma 2.4]).

As noted, it is possible to calculate the values of F_M and G_M approximately. Consequently, we can use these values in order to build up a quasi-Newton type method.

3.3. **BFGS update.** In this section, we discuss a BFGS update for the matrix M. The main idea is to modify the Moreau-Yosida penalty metric, originally defined by M, to improve the descent direction. Specifically, if certain conditions—outlined above —are satisfied, we apply a BFGS update to replace M with a quasi-Newton approximation. This allows for more efficient optimization by adjusting the metric to better capture the problem's local curvature. If the conditions are not met, we continue using the matrix M without modification. This adaptive strategy ensures we either refine the descent direction with the BFGS update or retain the robustness of the original Moreau-Yosida metric.

Given the vectors Δu and Δy, the BFGS update of a symmetric matrix $B \in \mathbb{R}^{n\times n}$ is

$$BFGS(B,\Delta u,\Delta y) := B + \frac{\Delta u \Delta y^T}{\Delta u^T \Delta y} + \frac{B\Delta u \Delta u^T B}{\Delta u^T B \Delta u}.$$

If B is positive definite and $\Delta u^T \Delta y > 0$, then the symmetric matrix $B^+ := BFGS(B,\Delta u,\Delta y)$ is also positive definite (see [19, p. 31]).

In our method, we assume that $B_0 = M$ and additionally $\sum_{k=1}^{\infty} \delta_k^{1/3} < \infty$ (see [18]). During each iteration, if the following two conditions are verified, B_k is updated to $B_{k+1} = BFGS(B_k,\Delta u^k,\Delta y^k)$; otherwise, $B_{k+1} := M$ is taken. Given $\alpha_1 \in (0,+\infty)$ and $\alpha_2 \in (0,1)$, these two conditions are

$$||\Delta u^k||_M(\sqrt{2\varepsilon_k} + \sqrt{2\varepsilon_{k+1}}) \leq \alpha_1 (\Delta u^k)^T \Delta y^k, \tag{3.8}$$

$$2||\Delta y^k||_M(\sqrt{2\varepsilon_k} + \sqrt{2\varepsilon_{k+1}}) \leq \min\{\alpha_2, \delta_k^{1/3} + \delta_{k+1}^{1/3}\}|\Delta y^k|^2, \tag{3.9}$$

where $\varepsilon_k = \varepsilon(u^k)$, $\Delta u^k = u^{k+1} - u^k$ and $\Delta y^k = \tilde{G}_M\left(u^{k+1}\right) - \tilde{G}_M\left(u^k\right)$.

3.4. **Line search algorithm.** In this section, we discuss the line search technique which is used in the implementation of our method. The following algorithm relies in the use of polynomial models of the objective function in order to perform a backtracking method, and it was originally proposed in [7, Sec. 6.3.2].

Algorithm 2 Backtracking line search algorithm

```
Data: Let $\sigma \in (0, 1/2)$ and $\lambda_0 = 1$.
Step 1. Decide wether $F_M(u^k + \lambda_k s^k) \leq F_M(u^k) + \sigma\lambda_k (s^k)^\top G_M(u^k)$ holds. If so, STOP and
  let $\lambda_k = \lambda_0$. Otherwise:
Step 2. : Decide whether the step length is too small. If so, STOP and terminate algorithm:
  routine failed to locate $u^{k+1}$ sufficiently distinct from $u^k$. Otherwise:
Step 3. Decrease $\lambda$ by a factor between 0.1 and 0.5 as follows:
  (1) On the first backtrack: let $\lambda_k := \tilde{\lambda}_2 = \arg\min m_2(\lambda)$, but constraint the new $\lambda_k$ to be
      greater than 0.1.
  (2) On all the subsequent backtracks: set $\lambda_k := \tilde{\lambda}_3 = \arg\min m_3(\lambda)$, but constraint the new
      $\lambda_k$ to be in $[0.1\lambda_{\mathrm{p}}, 0.5\lambda_{\mathrm{p}}]$.
Step 4. Return to step 1.
```

Here, m_2 and m_3 represent quadratic and cubic polynomial models, respectively, used to approximate the value of $\mathscr{J}_h$. We provide a brief description of these models for the sake of clarity and completeness of the paper.

Let $\varphi_k(\lambda) := F_M(u^k + \lambda s^k)$. Then the quadratic model m_2 is given by

$$m_2(\lambda) := \left(\varphi_k(1) - \varphi_k(0) - \varphi_k'(0)\right)\lambda^2 + \varphi_k'(0)\lambda + \varphi_k(0),$$

while the cubic model m_3 is given by

$$m_3(\lambda) = c\lambda^3 + d\lambda^2 + \varphi_k'(0)\lambda + \varphi_k(0),$$

where

$$\begin{pmatrix} c \\ d \end{pmatrix} = \frac{1}{\lambda_{\mathrm{p}} - \lambda_{2\mathrm{p}}} \begin{pmatrix} \frac{1}{\lambda_{\mathrm{p}}^2} & -\frac{1}{\lambda_{2\mathrm{p}}^2} \\ -\frac{\lambda_{2\mathrm{p}}}{\lambda_{\mathrm{p}}^2} & \frac{\lambda_{\mathrm{p}}}{\lambda_{2\mathrm{p}}^2} \end{pmatrix} \begin{pmatrix} \varphi_k(\lambda_{\mathrm{p}}) - \varphi_k(0) - \varphi_k'(0)\lambda_{\mathrm{p}} \\ \varphi_k(\lambda_{2\mathrm{p}}) - \varphi_k(0) - \varphi_k'(0)\lambda_{2\mathrm{p}} \end{pmatrix}.$$

and λ_p and λ_{2p} are the two previous values of λ_k. For further details, we refer the reader to [7, Sec. 6.3.2], [10, 12] and the references therein.

Finally, our main algorithm is described as follows

Algorithm 3 Proximal bundle algorithm applied to the viscoplastic laminar flow

Data: Let $\sigma \in (0, 1/2)$, $N \in \mathbb{N}$ and $tol \lll 1$. Also, let $\{\delta_k\}$ be a sequence of positive numbers such that $\sum_{k=0}^{\infty} \delta_k^{1/3} < \infty$. Let $u^0 \in \mathbb{R}^n$ be an initial guess, $M \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix, and $\mathscr{B}$ be a bundle.

Step 0. **(Initialization)** Fix $k := 0$ and $B_0 := M$, find (w^0, v^0) and ε_0 by means of Algorithm 1 such that

$$\varepsilon_0 \leq \delta^0 \left\{(w^0)^T M w^0, N\right\}.$$

For example, initializing the bundle with $\left\{(u^0, J_h^0, \xi^0)\right\}$.

Step 1. **(Descent direction)** If $|v^k| < \mathrm{tol}$, STOP with u^k being optimal. Otherwise, calculate

$$s^k = -B_k^{-1}\tilde{G}_M(u^k).$$

Step 2. **(Line search)** Starting with $\lambda_k = 1$, determine if the following condition is satisfied

$$\check{F}_M(u^k + \lambda_k s^k) \leq \check{F}_M(u^k) + \sigma\lambda_k (s^k)^\top \tilde{G}_M(x^k), \tag{3.10}$$

where $\check{F}_M(u^k + \lambda_k s^k)$ is the approximation of F_M at $u^k + \lambda_k s^k$ and verifies

$$\begin{aligned} &\hat{F}_M\left(u^k + \lambda_k s^k\right) - \check{F}_M\left(u^k + \lambda_k s^k\right) \\ &\leq \delta_{k+1} \min\left\{\left(w\left(u^k + \lambda_k s^k\right)\right)^\top M w\left(u^k + \lambda_k s^k\right), N\right\}. \end{aligned}$$

If not, backtrack λ_k by means of Algorithm 2 until (3.10) be fulfilled. Fix $u^{k+1} = u^k + \lambda_k s^k$.

Step 3. **(Update of the Quasi-Newton matrix)** Let $\Delta u^k = u^{k+1} - u^k$ and $\Delta y^k = \tilde{G}_M(u^{k+1}) - \tilde{G}_M(u^k)$. If (3.8) and (3.9) are verified, update B_k to B_{k+1} by using the BFGS update formula

$$B_{k+1} = BFGS(B_k, \Delta u^k, \Delta y^k).$$

Otherwise, choose $B_{k+1} = M$.

Step 4. Update $k = k+1$ and return to step 1.

The stopping criteria for the algorithm follows [17, Theorem 1]. Indeed, the method stops if the subbundle algorithm returns $v(u)$ such that $|v(u)| \leq tol$, where tol is a small enough positive parameter.

3.5. **Analysis of the algorithm.** In this section, we study convergence properties of Algorithm 3. We assume that the solution of problem (3.1) is different from 0. This a reasonable assumption since a zero velocity of deformation would not provide much information about the behavior of the fluid in the pipe.

From now on, we assume that Algorithm 3 does not finish. Thus the sequence generated by the algorithm $\{u^k\}$ is infinite.

Consider the set D which is defined by

$$D := \left\{ u \in \mathbb{R}^N : F_M(u) \leq F_M(u^0) + NC \right\}.$$

We present the following result that turns out to be very useful in order to prove global convergence of the method.

Lemma 3.1. *For all $k \geq 0$, $u^k \in D$, and*

$$F_M(u^{k+1}) \leq F_M(u^k) + N\delta_{k+1}. \tag{3.11}$$

Proof. By the algorithm rules, it follows that

$$\begin{aligned} F_M(u^{k+1}) &\leq \check{F}_M(u^k) + \sigma\lambda_k (s^k)^\top \tilde{G}_M(u^k) + N\delta_{k+1}, \\ &= \check{F}_M(u^k) - \sigma\lambda_k \tilde{G}_M(u^k)^\top B_k^{-1} \tilde{G}_M(u^k) + N\delta_{k+1}. \end{aligned}$$

Because B_k is definite positive, we see from the fact that $\check{F}_M \leq F_M$ that

$$F_M(u^{k+1}) \leq F_M(u^k) + N\delta_{k+1}.$$

Consequently, for all $k \geq 0$, (3.11) holds and $u^{k+1} \in D$. Finally, $u^0 \in D$. □

The next result guarantees global converge of Algorithm 3.

Theorem 3.1. *Suppose that there exist two positive constants c_1 and c_2 such that $\|B_k\| \leq c_1$ and $\|B_k^{-1}\| \leq c_2$. Then any accumulation point of $\{u^k\}$ minimizes $\mathscr{J}_h$.*

Proof. By Lemma 3.1, it follows that $F_M(u^k)$ is bounded from above. On the other hand, since $\mathscr{J}_h$ is bounded from below, F_M must also be bounded from below. Set $F_M^* := \liminf_{k\to\infty} F_M(x^k)$. From the fact that $\sum_k \delta_k < \infty$ and (3.11), we have $\lim_{k\to\infty} F_M(u^k) = F_M^*$. Since $\delta_k \to 0$ and the algorithm rules, we have that

$$\lim_{k\to\infty} \check{F}_M(u^k) = \lim_{k\to\infty} \hat{F}_M(u^k) = F_M^*.$$

On the other hand, it once again follows by the algorithm rules that $\lim_{k\to\infty} \lambda_k (s^k)^T \tilde{G}_M(u^k) = 0$. By using the fact that B_k y B_k^{-1} are bounded, we conclude that

$$\lim_{k\to\infty} \lambda_k |\tilde{G}_M(u^k)|^2 = 0. \tag{3.12}$$

Now, let $\bar{u}$ be any accumulation point of $\{u^k\}$. We consider a subsequence $\{u^k\}$ (denoted as the original one) such that $u^k \to \bar{u}$. Thanks to Lemma 2.3 in [18], it follows that

$$0 \leq |G_M(u^k) - \tilde{G}_M(u^k)|^2 \leq 2\|M\|\varepsilon_k.$$

Note that $\varepsilon_k \to 0$. Thus $\lim_{k\to\infty} \tilde{G}_M(u^k) = \lim_{k\to\infty} G_M(u^k)$. From the continuity of G_M, we have

$$\lim_{k\to\infty} \tilde{G}_M(u^k) = G_M(\bar{u}). \tag{3.13}$$

By the line search algorithm, it follows that $\liminf_{k\to\infty} \lambda_k > 0$. Therefore, from (3.12) and (3.13) we conclude that $G_M(\bar{u}) = 0$. Consequently $\bar{u}$ minimizes $\mathscr{J}_h$. □

In [18], in order to obtain a superlinear convergence rate of their method, the authors assumed that F_M is strongly convex and that G_M is Fréchet differentiable. From our point of view, the role of strong convexity has to do with granting the uniqueness of the solution and that the Hessian of F_M being positive definite at the minimum. This last property represents a challenging task, since, the second derivative of F_M only exists under certain conditions (see [14]). Moreover, the strong convexity of the regularized functional is directly related to strong convexity of the objective functional (see [14, Theorem 2.2]). However, we already know that, for any $p \in (1,+\infty)$, problem (3.1) has a unique solution. Now, for the last property, we proceed as follows: Note that J_h is twice differentiable for any $u \neq 0$, and its second derivative is given by

$$J_h''(u^h)(v^h,w^h) = (p-2)\int_\Omega |\nabla u^h|^{p-4}(\nabla u^h,\nabla v^h)(\nabla u^h,\nabla w^h)\,dx$$
$$+\int_\Omega |\nabla u^h|^{p-2}(\nabla v^h,\nabla w^h)\,dx + g\int_\Omega \frac{(\nabla v^h,\nabla w^h)}{|\nabla u^h|}\,dx - g\int_\Omega \frac{(\nabla u^h,\nabla v^h)(\nabla u^h,\nabla w^h)}{|\nabla u^h|^3}\,dx,$$

for all $v^h \in W_0^h$. Further, since we have assumed that $\bar{u} \neq 0$, then $J_h''(\bar{u})$ exists. This fact allows us to state that the Hessian of $\mathscr{J}_h$ is also well defined in $\mathbb{R}^n$. We will use the notation $\mathscr{J}_h''(\vec{u})$ to represent this Hessian. Thus, by [14, Th. 3.1], the Hessian of F_M at $\bar{u}$ exists, and is given by

$$\nabla^2 F_M(\bar{u}) = M - M[\mathscr{J}''(\bar{u}) + M]^{-1}M. \tag{3.14}$$

Moreover, considering $V := B_r(\bar{u})$, where $r := \frac{|\bar{u}|}{\max\{2,L_p\}}$, and L_p is the Lipschitz continuity constant associated to the operator p_M (see [14, Prop. 3.2]), we have the following lemma.

Lemma 3.2. *For each $u \in V$, $p_M(u) \neq 0$.*

Proof. Note that $0 \notin B_{L_p r}(\bar{u})$, since $|\bar{u}| \geq L_p r$. Now, we claim that, for any $u \in V$, $p_M(u) \in B_{L_p r}(\bar{u})$. Indeed,

$$|p_M(u) - \bar{u}| = |p_M(u) - p_M(\bar{u})| \leq L_p|u-\bar{u}| < L_p r.$$

Thus the result follows. □

Remark 3.2. As a consequence of Lemma 3.2, for $u \in V$, the second derivative of $\mathscr{J}_h$ is well defined. Thus F_M is twice differentiable in V (see [14, Theorem 3.1]). In addition, since $\mathscr{J}_h$ is strictly convex, we see from [1, Theorem 2.2] that F_M is also strictly convex. Consequently, F_M has a positive definite Hessian in V.

Theorem 3.2 (Superlinear convergence rate of Algorithm 3)**.** *The sequence $\{u^k\}$ generated by Algorithm 3 converges locally Q-superlinearly to $\bar{u}$. Moreover, the sequences of matrices $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded.*

Proof. We use the same procedure as in [18, Theorem 5.12]. From the fact that F_M has a positive definite Hessian in V, it remains to show that there exists a constant $L > 0$ such that

$$\sup_{|d|=1} |G_M'(u;d) - G_M'(\bar{u};d)| \leq L|u-\bar{u}|, \tag{3.15}$$

for any $u \in V$. Note that

$$\begin{aligned}
&|G'_M(u;d) - G'_M(\bar{u};d)| \\
&\quad \leq |M|^2 \left| \mathscr{J}''_h(p_M(u)) + M]^{-1} \right| \left| \mathscr{J}''_h(p_M(\bar{u})) + M]^{-1} \right| \left| \mathscr{J}''_h(p_M(\bar{u})) - \mathscr{J}''_h(p_M(u)) \right| |d| \\
&\quad \leq L_1 \left(\sup_{u \in V} \left| [\mathscr{J}''_h(p_M(u)) + M]^{-1} \right| \right) \left| \mathscr{J}''_h(p_M(u)) - \mathscr{J}''_h(p_M(\bar{u})) \right| |d| \\
&\quad \leq L_2 \left| \mathscr{J}''_h(p_M(u)) - \mathscr{J}''_h(p_M(\bar{u})) \right| |d|.
\end{aligned}$$

Observe that $\sup_{u \in V} \left| [\mathscr{J}''_h(p_M(u)) + M]^{-1} \right|$ is finite, since $\mathscr{J}''_h$ is locally Lipschitz continuous and p_M is Lipschitz continuos. Thus, we conclude that

$$\sup_{|d|=1} |G'_M(u;d) - G'_M(\bar{u};d)| \leq L|u - \bar{u}|.$$

Therefore, (3.15) is verified in V.

Now, thanks to Lemma 2.2 in [20], (3.15) implies that

$$\frac{|\Delta \bar{y}^k - G'_M(\bar{u})\Delta u^k|}{|\Delta u^k|} \leq L \max \left\{ |u^{k+1} - \bar{u}|, |u^k - \bar{u}| \right\}, \tag{3.16}$$

for k large enough (u^k belonging to V), where $\Delta \bar{y}^k = G_M(u^{k+1}) - G_M(u^k)$. Thus, in virtue of Theorem 5.9 in [18], equation (3.16) implies the boundedness of the sequences of the matrices $\{B_k\}$ y $\{B_k^{-1}\}$. In addition, this also implies that

$$\lim_{k \to \infty} \frac{|(B_k - G'_M(\bar{u}))\Delta u^k|}{|\Delta u^k|} = 0.$$

From this equation, one has a Q-superlinear rate of convergence of Algorithm 3 (see [18, Theorem 5.12]). □

4. Numerical Results

In this section, we carry out several experiments to demonstrate the behaviour of Algorithm 3. We focus on the numerical simulation of the laminar flow of Herschel-Bulkley fluid in a pipe by considering a constant right hand side f, which represents the linear decay of pressure in the pipe.

One key feature in our algorithm is the implementation of the oracle, which is nothing but an evaluation algorithm for both $\mathscr{J}_h$ and $\partial \mathscr{J}_h$. The algorithm needs to determine if the actual u^k, $k \in \mathbb{N}$, equals to zero or not. If $u^k \neq 0$, thanks to equation (2.2), the oracle just returns the values of $\mathscr{J}_h(u^k)$ and $\mathscr{J}'_h(u^k)$ by direct calculation. Otherwise, if $u^k = 0$, then the oracle has to determine a function $\psi \in \partial \mathscr{J}_h(0)$. Note that if $\psi \in C_c^\infty(\Omega)$ is such that $|\nabla \psi| \leq 1$ in Ω, then $\psi \in \partial J_h(0)$. Thus, we define the following function

$$\psi(x,y) = \begin{cases} e^{1/(|(x,y)|^2 - 1)} & \text{if} \quad |(x,y)| < 1, \\ 0 & \text{if} \quad |(x,y)| \geq 1 \end{cases}$$

whose gradient is given by

$$|\nabla \psi(x,y)| = \begin{cases} 2e^{1/(|(x,y)|^2 - 1)} \frac{|(x,y)|}{(|(x,y)|^2 - 1)^2} & \text{if} \quad |(x,y)| < 1, \\ 0 & \text{if} \quad |(x,y)| \geq 1. \end{cases}$$

In Figure 1, the graph of $|\nabla\psi(x,y)|$ is presented. Therefore, the oracle will return this function and the value $\mathscr{J}_h(0)$.

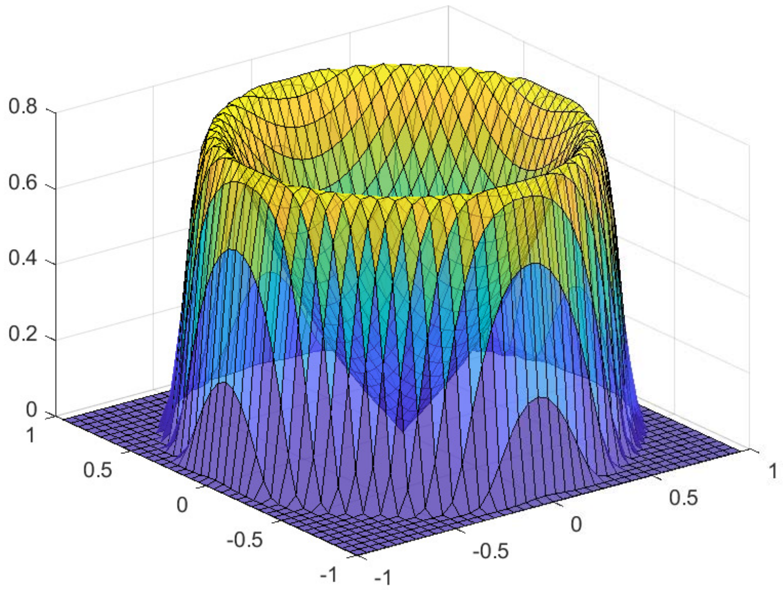


FIGURE 1. Graph of the function $|\nabla\psi(x,y)|$.

We utilized MATLAB's quadprog function to address the quadratic optimization problem (3.5). This function, built upon efficient interior-point and active-set algorithms, is particularly well-suited for solving constrained quadratic programming problems. One of its main strengths is its ability to robustly manage both equality and inequality constraints while efficiently handling large-scale problems, all while ensuring numerical stability.

4.1. **Shear-thinning flow:**$1 < p < 2$**.** In the next experiments, we observe the performance of Algorithm 3 when modelling the laminar flow of a shear-thinning fluid ($1 < p < 2$). We consider a constant linear decay of pressure given by $f = 1$. Also, we take g as the number of Oldroy which models the plasticity threshold of the fluid. For further details in the mechanichs of these problems, we refer the reader to [10, 13] and the references therein.

From now on, we use uniform triangulations described by h, the radius of the inscribed circumferences of the triangles in the mesh. In the next experiments, we initialize Algorithm 3 with the solution of the Poisson problem $-\Delta u_0^h = f^h$. Unless otherwise stated, we consider $\delta_0 = 0.0001$ and $\delta_k = 1/2^k$ for $k = 1,\dots$. In addtion, the parameter for the stopping criteria is $tol = 1e-05$ or $tol = 1e-06$.

4.1.1. *Experiment 1.* In this experiment, we consider Ω as the unit ball in $\mathbb{R}^2$. We model the laminar flow of a Herschel-Bulkley material with $p = 1.75$. We analyze the behaviour of the algorithm with $g = 0.2$ and a mesh of size $h \approx 0.0086$. For the Moreau-Yosida regularization, we consider $M = A$, the stiffness matrix of the finite element method.

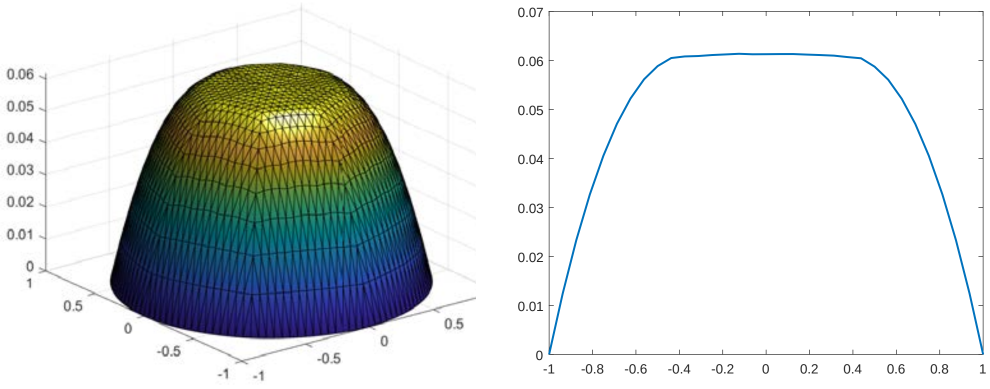


FIGURE 2. Calculated velocity u (left) and velocity profile along the diameter of the pipe (right). Parameters: $p = 1.75$, $g = 0.2$ and $M = A$.

In Figure 2, it is shown the resulting velocity function and its profile along the diameter of the pipe. These graphics illustrate the expected mechanical properties of the material, i.e., since the shear stress transmitted by a fluid layer decreases toward the center of the pipe, the Herschel–Bulkley fluid moves like a solid in that sector. This effect explains the flattening of the velocity in the center of the pipe .

In Table 1, we summarize the behaviour of the method in each iteration. For each one, we also show the value of $\mathscr{J}_h$, the norm of u^k in W_0^h, the update type of the B_k matrix, the number of iterations of the inner bundle process (Algorithm 1), the value of $|v|$, the size of the step lenght and the number of iterations of Algorithm 2.

it.	$J_h(u^k)$	$\|u^k\|$	B_k	Its. Bundle	$\|v\|$	λ_k	it. ls.
1	0.0812	0.6545	M	389	0.0914	1	0
2	-0.0102	0.3934	M	1	0.0189	1	0
3	-0.0252	0.2830	M	1	0.0045	1	0
4	-0.0293	0.2301	M	1	$3.3207e-04$	1	0
5	-0.0296	0.2123	M	1	$8.2777e-07$	1	0

TABLE 1. Convergence behaviour of Algorithm 3. Parameters: $p = 1.75$, $g = 0.2$ and $M = A$.

Note that the value of the functional $\mathscr{J}_h$ is reduced in each iteration. Regarding the number of iterations of Algorithm 1, we observe that at least at the beginning it requires a higher quantity of iterations to converge, later it only requires one iteration. This effect happens because of the choice of $\{\delta_k\}$, since at the beginning δ_0 is much smaller than the firsts values of δ_k, making the computational cost higher in the first iteration than the others. On the other hand, the stopping criterion $|v|$ has a high decay rate in the last iterations, this allows us to verify the superlinear convergence of Algorithm 3. It is also remarkable that, in this experiment, the step length remained constant throughout the entire execution of the method.

Next, in Figure 3, we observe the behaviour of ν. As expected, there is evidence of a fast decrease, which is characteristic of a superlinear convergence rate.

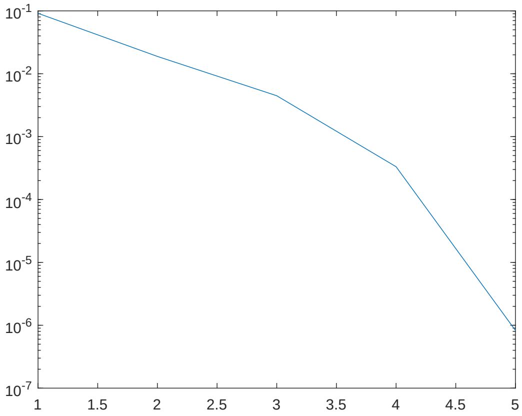


FIGURE 3. Calculated values of ν in each outer iteration of Algorithm 3. Parameters: $p = 1.75$, $g = 0.2$ and $M = A$.

4.1.2. *Experiment 2.* We set $\mathbb{R}^2 \supset \Omega = [-1,1] \times [-1,1]$. Again, we model the laminar flow of a Herschel-Bulkley material with $p = 1.75$, and analyze the behaviour of Algorithm 3 with $g = 0.2$ and a mesh given by $h \approx 0.0047$. Regarding the matrix for the Moreau-Yosida regularization we set $M = I$, the identity matrix.

The resulting velocity function and its profile along the diagonal of the pipe are shown in Figure 4. As in the previous experiment, these illustrate the expected mechanical properties of the material (see Subsection 4.1.1).

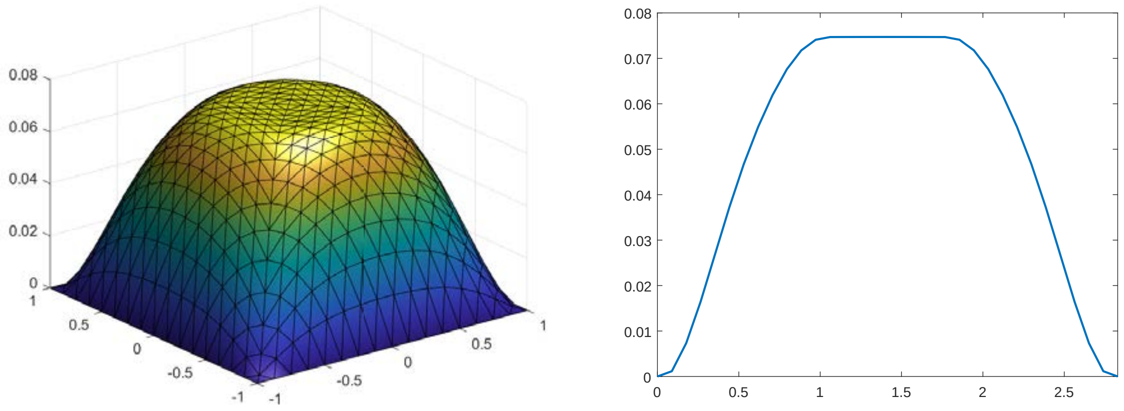


FIGURE 4. Calculated velocity u (left) and velocity profile along the diagonal of the pipe (right). Parameters: $p = 1.75$, $g = 0.2$ and $M = I$.

it.	$J_h(u^k)$	$\|u^k\|$	B_k	Its. Bundle	$\|\nu\|$	λ_k	it. ls.
1	0.1163	0.8133	M	57	0.0156	1	0
10	0.0300	0.6382	M	1	0.0058	1	0
19	-0.0104	0.5047	M	1	0.0034	1	0
38	-0.0421	0.2858	M	1	$6.0815e-05$	1	0
39	-0.0422	0.2800	M	1	$2.9216e-05$	1	0
40	-0.0422	0.2773	M	1	$1.1508e-05$	0.4973	1
41	-0.0422	0.2745	M	1	$3.2250e-06$	0.4973	0

TABLE 2. Convergence behaviour of Algorithm 3. Parameters: $p = 1.75$, $g = 0.2$ and $M = I$.

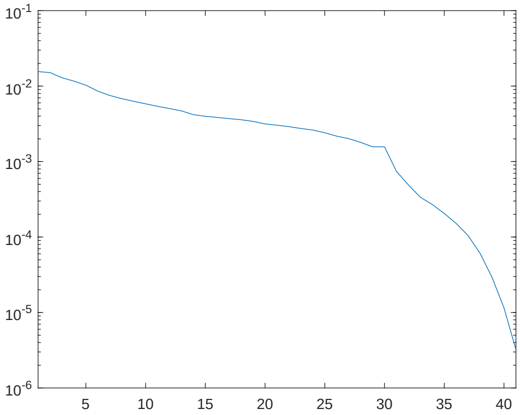


FIGURE 5. Calculated values of ν in each outer iteration of Algorithm 3. Parameters: $p = 1.75$, $g = 0.2$ and $M = I$.

In Table 2, we summarize the behaviour of the method at each iteration. Compared with the previous experiment, there is a large increase in the number of iterations performed until convergence. Another aspect that stands out in this experiment is the decrease of the functional $\mathscr{J}_h$, during the last iterations its value does not show any variation. From the second iteration of the method, the inner bundle process performed exactly one iteration, again, this is due to the choice of $\{\delta_k\}$. It is also observed that the values of ν show a monotonic decrease. As for the step size λ_k it is corroborated that it was constant until the penultimate iteration, where Algorithm 2 performed one iteration to converge, after that λ_k was reduced more or less by half.

Figure 5 shows the behaviour of the calculated values of ν in each iteration. It can be noticed that, from iteration 31 onwards, the decrease is more pronounced, this is an evidence of the good convergence properties of the algorithm, i.e., it has a superlinear convergence rate.

4.1.3. *Experiment 3.* In this experiment, we analyze the behavior of Algorithm 3 by varying the initialization parameter δ_0. We set Ω as the unit ball of $\mathbb{R}^2$. We model the flow of a material with $p = 1.75$, $g = 0.2$ and a mesh given by $h \approx 0.0386$. For the matrix associated with the Moreau-Yosida regularization we consider $M = I$. We perform Algorithm 3 with δ_0 equal to 1, 0.75, 0.50 and 0.25.

The resulting velocity function and its profile along the diameter of the pipe are presented in Figure 6. As in the first experiment, these graphs illustrate the same expected mechanical properties of the material.

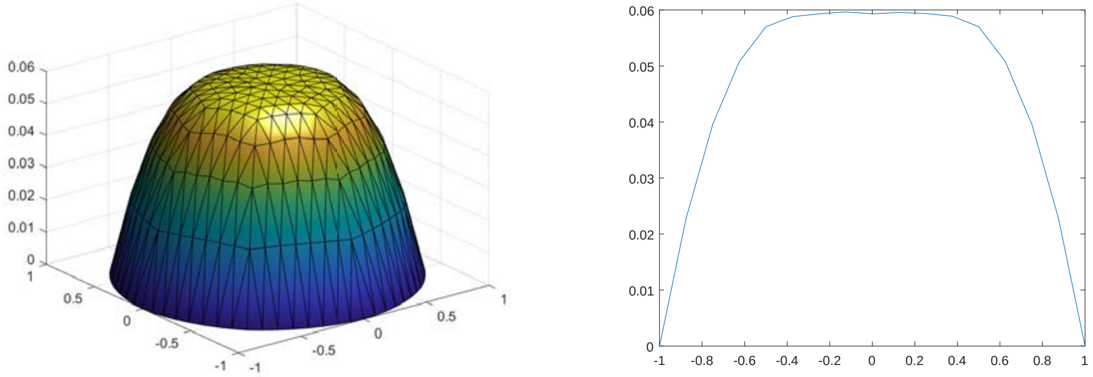


FIGURE 6. Calculated velocity u (left) and velocity profile along the diameter of the pipe (right). Parameters: $p = 1.75$, $g = 0.2$, $M = I$ and $\delta_0 = 0.25$.

In Table 3 we summarize the general behavior of the method for each value of δ_0. As can be observed, for a greater δ_0 the method presents less accuracy in computing the approximate solution. Also, note that the number of iterations performed in the case when $\delta_0 = 1$ is very much higher compared to the other cases. Regarding the behavior of the inner bundle process, it is worth mentioning that, for each iteration of the general algorithm, this method performed an average of 5 iterations to converge.

In this experiment it could be observed that the Algorithm 3 performs BFGS type updates, this is a good characteristic of this experiment compared to previous ones. It can be noticed that, for smaller values of δ_0, the method made fewer iterations using BFGS type updates. This behavior is also observed in previous experiments. In those cases we initialized with $\delta_0 = 0.0001$, causing that Algorithm 3 did not perform BFGS type updates in its total execution.

δ_0	$J_h(\bar{u})$	$\|\bar{u}\|$	its.	Its. Bundle	Its. BFGS	Total time
1	-0.0283	0.2074	108	436	24	8145.27 s
0.75	-0.0286	0.2096	38	161	4	787.35 s
0.50	-0.0288	0.2109	18	105	3	495.73 s
0.25	-0.0290	0.2115	26	253	3	624.65 s

TABLE 3. Convergence behaviour of Algorithm 3 with different choices of δ_0. Parameters: $p = 1.75$, $g = 0.2$ and $M = I$.

On the other hand, Table 4 collects the information of the Algorithm's performance in the case of $\delta = 0.75$. It is observed that in the iterations when BFGS type update occurs the value of the functional decays faster than the other iterations where the update matrix $B_k = M$ is taken. Note that the bundle inner process in general does not have a high computational cost, in the last iterations of the method it is observed that this process requires to perform more iterations to converge, however, it does not exceed in large number as seen in the previous experiments. In addition, the step length values are presented as same as the number of iterations performed by Algorithm 2.

it.	$J_h(u^k)$	$\|u^k\|$	B_k	Its. Bundle	$\|\nu\|$	λ_k	it. ls.
1	0.0862	0.6516	$BFGS$	2	0.0576	1	0
2	0.0398	0.5522	$BFGS$	2	0.0316	1	0
3	0.0278	0.3400	$BFGS$	6	0.0210	1	0
5	0.0256	0.3361	$BFGS$	5	0.0538	0.0046	4
15	0.0246	0.3314	M	1	0.0501	0.0046	0
25	0.0223	0.3280	M	1	0.0478	0.0046	0
30	-0.0265	0.2361	M	3	$4.0416e-04$	0.5000	1
35	-0.0271	0.2117	M	14	$2.1106e-05$	0.5000	0
38	-0.0286	0.2065	M	9	$2.7607e-07$	0.2500	1

TABLE 4. Convergence behaviour of Algorithm 3. Parameters: $p = 1.75$, $g = 0.2$, $M = I$ and $\delta_0 = 0.75$.

The convergence behaviour of the method for every value of δ_0 is presented in Figure 7. For $\delta_0 = 1$, a monotonic decay can be observed until the tenth iteration, subsequently, the parameter ν oscillates until the end of the iterations, requiring the total execution of the method to take a larger amount of time.

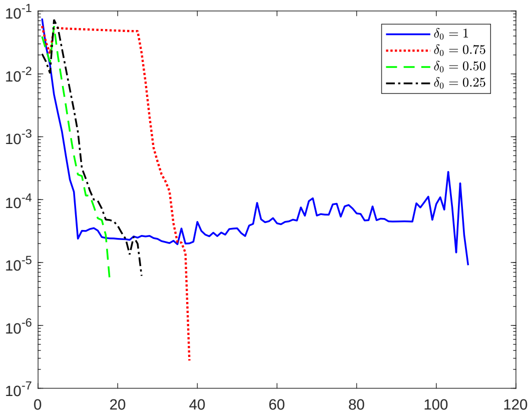


FIGURE 7. Calculated values of ν by Algorithm 3 for each choise of δ_0. Parameters: $p = 1.75$, $g = 0.2$ and $M = I$.

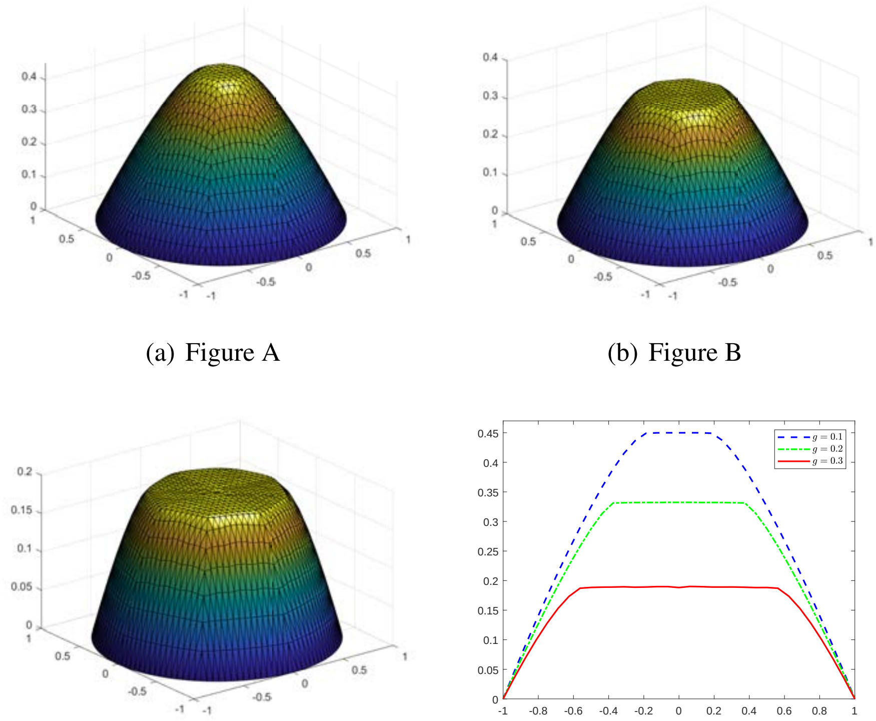
For the other values of δ_0, it can be observed that the method presents a similar convergence behavior (superlinear rate). In general, a monotonic decay of the values of ν is noticed, in the case when $\delta_0 = 0.75$ the decay is deeper. In these cases, the execution time of the method is much faster than the time required for $\delta_0 = 1$.

4.2. **Shear-thickening flow:** $p > 2$**.** In the next experiments, we observe the performance of Algorithm 3 when modelling the laminar flow of a shear-thickening fluid ($p > 2$). As in the previous section, we set $f = 1$, the parameter that represents a linear decay of pressure in the pipe. Also, we take g as the number of Oldroy which models the plasticity threshold of the fluid. For further details in the in the mechanichs of these problems, we refer the reader to [13] and the references therein.

Similarly, we use uniform triangulations described by h, the radius of the inscribed circumferences of the triangles in the mesh. We initialize Algorithm 3 with the solution of the Poisson problem $-\Delta u_0^h = f^h$. Unless otherwise stated, we consider $\delta_0 = 0.0001$ and $\delta_k = 1/2^k$ for $k = 1, \ldots$. In addition, the parameter for the stopping criteria is $tol = 1e-05$ or $tol = 1e-06$.

4.2.1. *Experiment 1.* In this experiment, we consider Ω as the unit ball in $\mathbb{R}^2$. We model the laminar flow of a Herschel-Bulkley material with $p = 4$. We analyze the behaviour of the algorithm with $g = 0.1$, $g = 0.2$ and $g = 0.3$, and a mesh give by $h \approx 0.0086$. For the Moreau-Yosida regularization we consider $M = A$, the stiffness matrix of the finite element method.

In Figure 8, the resulting velocity function and its profile along the diameter of the pipe are displayed. These graphics illustrate the expected mechanical properties of the material, i.e., the viscosity of shear-thickening materials increases with the rate of shear strain. In this case, since the shear stress transmitted by a fluid layer decreases toward the center of the pipe, the velocity takes a conical form with a flat part in the exact center of the geometry. Moreover, note that for a greater value of g then this flat zones are bigger.



(a) Figure A

(b) Figure B

(c) Figure C

(d) Figure D

FIGURE 8. Calculated velocity u for $p = 4$ and $g = 0.1$ (Figure A), $g = 0.2$ (Figure B) and $g = 0.3$ (Figure C). Velocity profile for the calculated velocities along the diameter of the pipe (Figure D).

Table 5 describes the behavior of Algorithm 3 for each case of g. We exhibit the value of the functional $\mathscr{J}_h$ at the minimum $\bar{u}$ and the norm of this point. As mentioned in [10, 13], for large values of g the problem is less regular and therefore harder to be approximated, however, it can be observed that the algorithm requires fewer iterations for its convergence for $g = 0.3$ than $g = 0.2$, which turns out to be a good quality of our method. On the other hand, note that the number of iterations of Algorithm 1 increases when g is larger, which in terms of the execution time of the method represents a higher computational cost.

g	$J_h(\bar{u})$	$\|\bar{u}\|$	its.	Its. Bundle	Its. BFGS	Total time
0.1	-0.3780	0.8255	7	220	0	479.62 s
0.2	-0.2174	0.7282	12	367	0	2560.78 s
0.3	-0.1872	0.5656	9	532	0	32430.54 s

TABLE 5. Convergence behaviour of Algorithm 3. Parameters: $p = 4$ and $M = A$.

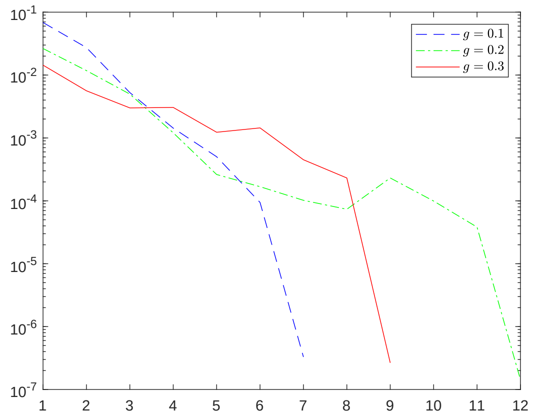
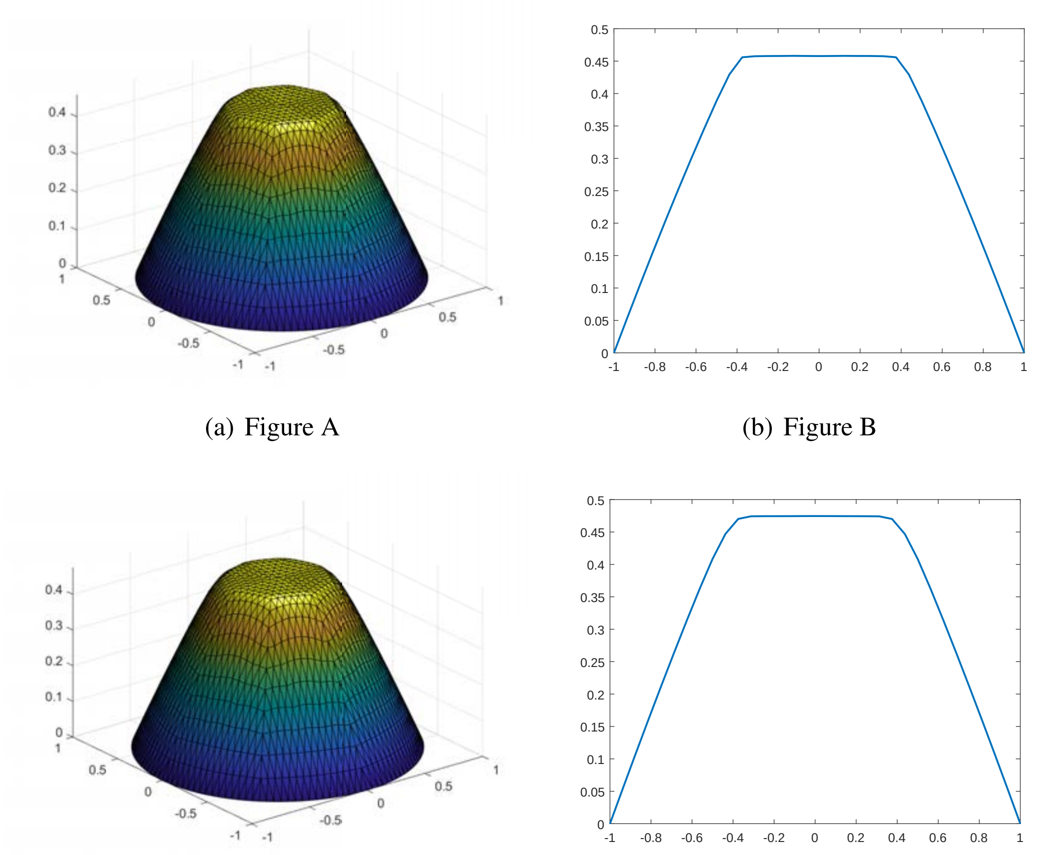


FIGURE 9. Calculated values of ν in each outer iteration of Algorithm 3 for every value of g. Parameters: $p = 4$ and $M = A$.

Finally, Figure 9 shows the convergence behavior of the Algorithm 3 for each value of g. It is possible to corroborate the superlinear convergence of the method for this case, oscillations are noticed during the first iterations, later an abrupt decrease is evidenced, typical of this type of rate of convergence.

4.2.2. *Experiment 2.* In this experiment, we set Ω as the unit ball of $\mathbb{R}^2$. As mentioned in [10], when the value of p is large, the numerical resolution of problem (3.1) becomes a challenging task. Consequently, we test Algorithm 3 by modeling the flow of a material with $p = 8$ and $p = 10$. We analyze the behavior of the method with $g = 0.2$ and a mesh given by $h \approx 0.0086$. For the Moreau-Yosida regularization we consider $M = A$, the stiffness matrix obtained by the finite element method.

The resulting velocities functions and their profile along the diameter of the pipe are shown in Figure 10. Again, these graphs illustrate the expected mechanical properties of the material, i.e., the viscosity of shear-thickening fluids increases with shear stress rate. Since the shear stress transmitted by a fluid layer decreases toward the center of the pipe, the velocity takes a conical form with a flat part in the exact center of the geometry.



(a) Figure A

(b) Figure B

(c) Figure C

(d) Figure D

FIGURE 10. Calculated velocity u and profile of the velocity along the diameter for $p = 8$ (Figures A and B) and for $p = 10$ (Figures C and D). Parameters: $g = 0.2$ and $M = A$.

Behavior of Algorithm 3 for both cases is reported in Table 6. The value of the functional $\mathscr{J}_h$ at its minimum points so as the norm of the points are presented. Regarding the number of iterations it is interesting to note that for $p = 10$ the method needs 109 iterations to converge, evidencing the difficulty of solving the problem. On the other hand, it is observed that the iterations performed by the bundle inner process are numerous, which directly influences the total time of the execution of the method. It can be noted that the convergence of the algorithm took a large amount of time.

p	$J_h(\bar{u})$	$\|\bar{u}\|$	its.	Its. Bundlle	Its. BFGS	Total time
8	-0.3376	0.8743	11	311	0	49325.69 s
10	-0.3639	0.8998	109	1045	0	92880.14 s

TABLE 6. Behavior of Algorithm 3 for higher values of p. Parameters: $g = 0.2$ and $M = A$.

Finally, Figure 11 shows the convergence behavior of Algorithm 3 for each value of p. It is possible to corroborate the superlinear convergence of the method for both. For $p = 8$ a monotonic decay is observed followed by an abrupt decrease, typical of the superlinear behavior. On the other hand for $p = 10$ it is possible to notice that the values of ν remain constant by sections, this phenomenon is attributed to the bundle inner process, since the new information added to the bundle was not good enough to obtain a better value of ν. However, it is possible to notice a steep decrease starting at iteration 97.

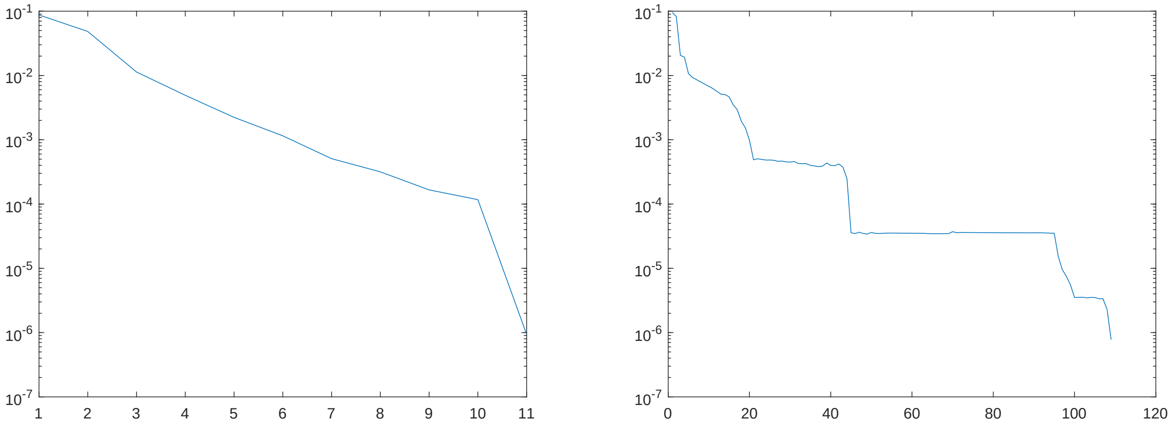


FIGURE 11. Calculated values of ν in each outer iteration of Algorithm 3 for $p = 8$ (left) y $p = 10$ (right). Parameters: $g = 0.2$ and $M = A$.

4.2.3. *Experiment 3.* In this experiment, we analyze the behavior of Algorithm 3 by varying the initialization parameter δ_0. We set Ω as the unit ball of $\mathbb{R}^2$. We model the flow of a material with $p = 4$, $g = 0.2$ and a mesh given by $h \approx 0.0386$. For the matrix associated with the Moreau-Yosida regularization we consider $M = I$. We perform Algorithm 3 with δ_0 equal to 1, 0.75, 0.50 and 0.25.

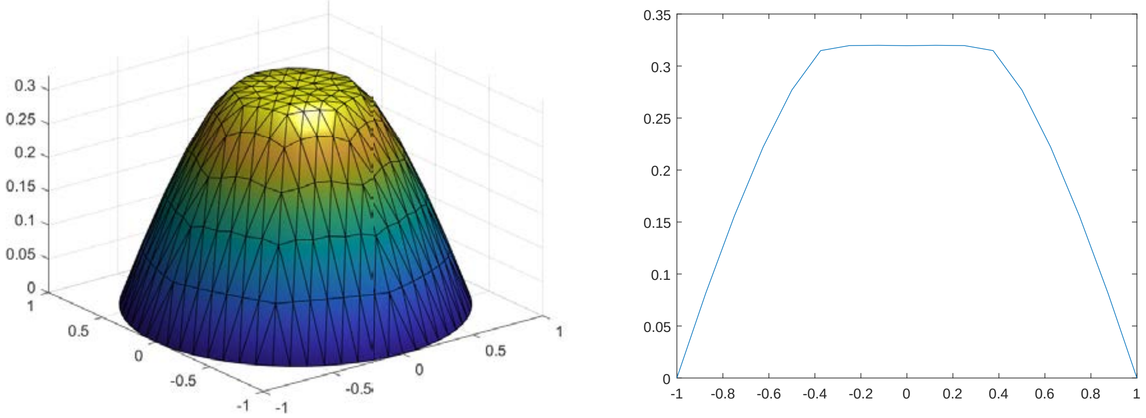


FIGURE 12. Calculated velocity u (left) and its profile along the diameter of the pipe (right). Parameters: $p = 4$, $g = 0.2$, $M = I$ and $\delta_0 = 0.25$.

Figure 12 shows the velocity function and its profile along the diameter of the pipe. As in the first experiment of this section, these graphs illustrate the same expected mechanical properties of the material. For each value of δ_0, the behavior of Algorithm 3 is presented in Table 7. In this case, it can be observed that for larger δ_0 the solution is more accurate, since the value of $\mathscr{J}_h$ functional at $\bar{u}$ is smaller, as well as the value of $\bar{u}$, which increases. With respect to the number of iterations, for $\delta_0 = 1$ the method has a better behavior than the cases $\delta_0 = 0.75$ and $\delta_0 = 0.50$. However, it is evident that the Algorithm 3 has its best performance when $\delta_0 = 0.25$ than in the others.

It is important to mention that, for $\delta_0 = 0.75$ and $\delta_0 = 0.50$, the Algorithm 3 showed the same behavior, it can be observed that in both cases the same solution was obtained, so as the same number of iterations, same number of iterations of the bundle inner process and the same number of BFGS type updates. As well as the same amount of execution time of the method.

The behavior of the Algorithm 1 is quite good for each initialization parameter. It is possible to observe that in the last case it performs almost the same number of iterations as the general method has performed. For the intermediate cases, it is noted that the number of iterations performed are approximately twice the number of outer iterations. In the the first case, it is observed that the number of iterations performed by the bundle inner process exceeds by more than three times the number of outer iterations.

δ	$J_h(\bar{u})$	$\|\bar{u}\|$	its.	Its. Bundle	Its. BFGS	Total time
1	-0.2087	0.7012	30	103	9	930.87 s
0.75	-0.2086	0.6969	37	71	6	1086.27 s
0.50	-0.2086	0.6969	37	71	6	1086.27 s
0.25	-0.2080	0.6798	19	21	1	365.21 s

TABLE 7. Convergence behaviour of Algorithm 3 with different choices of δ_0. Parameters: $p = 4$, $g = 0.2$ and $M = I$.

As in the previous section, the variation of the initialization parameter δ_0 allows the method to perform iterations with the BFGS type update. Again it is observed that, when $delta_0$ is small, then the method performs fewer BFGS type updates. This is also supported by the previous experiments in this section, since when taking $\delta_0 = 0.0001$, the method does not perform any BFGS iteration.

The most noticeable advantage of the BFGS type update is that the method converges more quickly to the solution. This can be noticed in the total execution time of the Algorithm 3. In the first case, the method requires less time than for the second and third cases. Even when the number of iterations of Algorithm 1 is excessively higher than the iterations performed by the general method.

it.	$J_h(u^k)$	$\|u^k\|$	B_k	Its. Bundle	$\|\nu\|$	λ_k	it. ls.
1	-0.1695	0.4893	$BFGS$	1	0.0266	1	0
2	-0.1697	0.5703	$BFGS$	3	0.0140	1	0
3	-0.1759	0.6227	M	2	0.0243	0.3994	1
7	-0.2011	0.6540	$BFGS$	1	0.0043	0.3994	0
9	-0.2043	0.6597	$BFGS$	1	0.0021	0.3994	0
13	-0.2068	0.6716	M	1	$6.5891e-04$	0.3994	0
15	-0.2075	0.6782	$BFGS$	1	$3.6662e-04$	0.3994	0
18	-0.2080	0.6870	$BFGS$	1	$2.1609e-04$	0.3994	0
23	-0.2082	0.6980	$BFGS$	1	$1.2821e-04$	0.3994	0
25	-0.2084	0.7005	$BFGS$	1	$1.0648e-04$	0.3994	0
26	-0.2085	0.7008	$BFGS$	7	$8.7622e-05$	0.3994	0
29	-0.2086	0.7010	M	28	$1.2858e-05$	0.3994	0
30	-0.2087	0.7012	M	15	$5.8191e-06$	0.2500	1

TABLE 8. Convergence behaviour of Algorithm 3. Parameters: $p = 4$, $g = 0.2$, $M = I$ and $\delta_0 = 1$.

Table 8 compiles the information of the development of Algorithm 3 with $\delta_0 = 1$ as initialization parameter. It is observed that there is a monotonic decrease in the value of the functional $\mathscr{J}_h$ as an increase in the norm of u^k at each iteration. In addition, the type of update of the B_k matrix is reported for each iteration. With respect to Algorithm 1, it is noted that it does not require a very high computational cost to converge; it is observed that it performs few iterations in its majority, it presents an increase of these in the last iterations of the general method, however, they are not excessively large as observed in the previous experiments. In addition, it is observed that the linear search algorithm performs few iterations along the method, in most iterations of the method the step length is constant and changes in the last iteration.

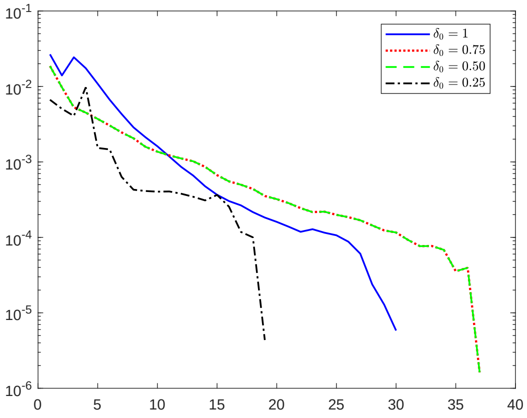


FIGURE 13. Calculated values of ν in each outer iteration of Algorithm 3 for each δ_0. Parameters: $p = 4$, $g = 0.2$ y $M = I$.

The convergence of the method through the values of ν obtained in each iteration is observed in Figure 13. Note that for all cases, these values decay monotonically in their majority. In addition, it is observed that in the final iterations the decrease of v is more pronounced, and this corroborates the superlinear convergence order towards the minimum for this type of materials.

5. Conclusions

In this paper, we focused on the numerical resolution of a class of variational inequalities of the second kind involving the p-Laplacian operator and the L^1-norm of the gradient. We applied the Moreau-Yosida regularization in order to get rid of non differentiability and then proposed a method which has an inner bundle algorithm, we proved that the outer algorithm is globally and superlinearly convergent. Several numerical experiments were carried out to show the main features of the numerical approach. These numerical examples were constructed focusing on the applications to the flow of Herschel–Bulkley materials. Due to the structure of the algorithm, it is observed that the inner bundle process has a higher computational cost that in some experiments made the overall execution of the method very slow. Nevertheless, in most of the experiments, we observe that the outer algorithm does not need of a higher number iterations to achieve convergence, thanks to the rate of convergence. In order to continue this research, we propose to study this problem by considering new techniques developed for bundle methods that are more efficient and have a lower computational cost.

References

[1] A. Bacho, A generalization of the Moreau-Yosida regularization. arXiv:2201.04175.

[2] M. Badiale and E. Serra, Semilinear Elliptic Equations for Beginners, Springer-London, 2011.

[3] A. Bagirov, N. Karmitsa, M. M. Mäkelä, Introduction to Nonsmooth Optimization, Springer, 2014.

[4] M. Bercovier, M. Engelman, A finite-element method for incompressible non-Newtonian flows, J. Comput. Phys. 36 (1980) 313–326..

[5] H. Brezis, Functional Analysis, Sobolev Spaces and Partial Differential Equations, Springer, 2011.

[6] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, SIAM, 2002.

[7] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, 1996.

[8] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1976), 17-40.

[9] R. Glowinski, A. Marroco, Sur L'Approximation par Elements Finis d'Ordre Un, et la Resolution, par Penalisation-Dualite, d'une Classe de Problemes de Dirichlet non Lineaires, RAIRO, 9 (1975) 41-76.

[10] S. González-Andrade, A preconditioned descent algorithm for variational inequalities of the second kind involving the p-Laplacian operator, Comput. Optim. Appl. 66 (2017), 123-162.

[11] S. González-Andrade, A BDF2-semismooth Newton algorithm for the numerical solution of the Bingham flow with temperature dependent parameters, J. Non-Newtonian Fluid Mech. 284 (2020), 104380.

[12] S. González-Andrade, S. López-Ordóñez, A multigrid optimization algorithm for the numerical solution of quasilinear variational inequalities involving the p-Laplacian, Comput. Math. Appl. 75 (2018), 1107-1127.

[13] R.R. Huilgol, Z. You, Application of the augmented Lagrangian method to steady pipe flows of Bingham, Casson and Herschel–Bulkley fluids, J. Non-Newtonian Fluid Mech. 128 (2005), 126-143.

[14] C. Lemaréchal, C. Sagastizábal, Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries, SIAM J. Optim. 7 (1997), 367-385.

[15] M. M. Mäkelä, Survey of bundle methods for nonsmooth optimization, Optim. Meth. Softw. 17 (2002), 1-29.

[16] M. M. Mäkelä, P. Neittaanmäki, Nonsmooth Optimization, World Scientific, 1992.

[17] R. Mifflin, A quasi-second-order proximal bundle algorithm, Math. Program. 73 (1996), 51-72.

[18] R. Mifflin, D. Sun, L. Qi, Quasi-Newton bundle-type methods for nondifferentiable convex optimization, SIAM J. Optim. 8 (1998), 583-603.

[19] J. Nocedal, S. Wright, Numerical Optimization, Springer, 2006.

[20] J.-S. Pang, Newton's method for B-differentiable equations, Math. Oper. Res. 15 (1990), 311-341.

[21] T.C. Papanastasiou, Flows of materials with yield. J. Rheol. 31 (1987), 385–404.

[22] W. Han, A regularization procedure for a simplified friction problem, Math. Comput. Modelling 15 (1991), 65-70.