A Multigrid Optimization Approach for the Numerical Solution of a Class of Variational Inequalities of the Second Kind

Sofía Alejandra López Ordóñez[†]; Sergio Alejandro González Andrade[‡]

[†]*Escuela Politécnica Nacional, Facultad de Ciencias, Quito, Ecuador*

e-mail: sofia.lopezo@epn.edu.ec

[‡]*Escuela Politécnica Nacional, Departamento de Matemática, Centro de Modelización Matemática, Quito, Ecuador*

e-mail: sergio.gonzalez@epn.edu.ec

Abstract: In this work we introduce a Multigrid Optimization Algorithm (MG/OPT) for the numerical solution of a class of quasilinear variational inequalities of the second kind, which involve the p-Laplacian operator and the L^1-norm of the gradient. The solution of the variational inequality is given by the solution of an optimization algorithm. We proposed a Huber regularization of the functional for the L^1-norm of the gradient and a finite element discretization for the problem. Finally, we analyze the performance of the MG/OPT algorithm when used to simulate the visco-plastic flows in a pipe. Several numerical experiments are carried out to show the main features of the proposed method.

Keywords: Variational inequalities of the second kind, multigrid algorithm, multigrid for optimization, finite element method, viscoplastic fluids.

1. Introducción

In this thesis we are concerned with the analysis, development and implementation of a multigrid algorithm for the numerical solution of a class of variational inequalities of the second kind: let Ω be an open and bounded set in $\mathbb{R}^n$ with Lipschitz boundary $\partial\Omega$, find $u \in W_0^{1,p}(\Omega)$ such that

$$\int_\Omega |\nabla u|^{p-2}(\nabla u, \nabla(v-u))\,dx + g\int_\Omega |\nabla v|\,dx \qquad (1)$$
$$-g\int_\Omega |\nabla u|\,dx \geq \int_\Omega f(v-u)\,dx,\ \forall v \in W_0^{1,p}(\Omega),$$

where $1 < p < \infty$, $g > 0$ and $f \in L^q(\Omega)$. Here, $q = \frac{p}{p-1}$ stands for the conjugate exponent of p.

The numerical resolution of variational inequalities involving the p-Laplacian constitutes an important research field. This operator is part of many mathematical models and has been widely studied due to its importance in modelling physical processes such as visco-plastic fluids flow, glaciology and diffusion and filtration processes (see [1, 8]). These problems are related to a wide range of industrial applications that can be studied inside the large scale optimization framework. Large scale problems involve a great amount of variables, thus, its numerical solution could take long periods of computation when executing an algorithm.

We also know that a variational inequality corresponds to the necessary condition of an optimization problem. Particularly, the solution of the variational inequality (1) corresponds to a first order necessary optimality condition for the following optimization problems.

$$\min_{u\in W_0^{1,p}(\Omega)} J(u) := \frac{1}{p}\int_\Omega |\nabla u|^p\,dx + \int_\Omega |\nabla u|\,dx - \int_\Omega fu\,dx. \qquad (2)$$

In consequence, throughout this work we will focus in the resolution of this optimization problem. Since our problem involves the L^1-norm, it corresponds to a non-differentiable minimization problem. Then, we propose a local Huber regularization technique to deal with the non-differentiable term. Further, we propose to solve the optimization problem by using a multigrid optimization (MG/OPT) algorithm. This algorithm was introduced as an efficient tool for large scale optimization problems (see [19, 17]). In [17] a multigrid optimization method is also presented for the optimization of systems governed by differential equations. The MG/OPT method focuses in optimization problems which are discretized in different levels of discretization generating a family of subproblems of different sizes. The idea of the algorithm is to take advantage of the solutions of problems discretized in coarse levels to optimize problems in fine meshes. The efficient resolution of coarse problems provide a way to calculate search directions for fine problems. Our purpose in this work is to propose, implement and analyze the MG/OPT algorithm for the resolution of nonsmooth problems with a finite element scheme. As the name im-

plies, the application of the MG/OPT method involves an underlying optimization algorithm at each level of discretization. Due to the limited regularity of the functional J and the p- Laplacian involved therein, we propose a class of descent algorithms such as the gradient method and a preconditioned descent algorithm (see [9]) as the underlying optimization algorithms. Particularly, the preconditioned descent algorithm was proposed to solve variational inequalities involving the p-Laplacian. Hence, our aim is to take advantage of the computational efficiency of the multigrid scheme and combine it with a suitable optimization algorithm for type problems (2).

1.1 Regularization and discretization

The minimization problem (2) involves a convex non-smooth functional. The norm $|\nabla y|$ in the second term leads us to a non-differentiable problem . We propose a local Huber regularization procedure in order to solve this issue. This regularization only changes locally the estructure of the functional, preserving the qualitative properties. This regularization has been used in several contributions (see [9]).

Let us introduce, for $\gamma > 0$, the function $\psi_\gamma : \mathbb{R}^n \to \mathbb{R}$ as follows:

$$\psi_\gamma : z \to \psi_\gamma(z) = \begin{cases} g|z| - \frac{g^2}{2\gamma} & if |z| > \frac{g}{\gamma} \\ \frac{\gamma}{2}|z|^2 & if |z| \leq \frac{g}{\gamma}. \end{cases}$$

The function ψ_γ corresponds to a local regularization of the Euclidean norm. Thanks to this procedure we obtain the following regularized optimization problem

$$\min_{u \in W_0^{1,p}(\Omega)} J_\gamma(u) := \frac{1}{p}\int_\Omega |\nabla u|^p \, dx + \int_\Omega \psi_\gamma(\nabla u)\, dx - \int_\Omega f u \, dx. \tag{3}$$

1.2 Finite element formulation

Let us introduce the finite element formulation of problem (3). Let Ω_h be a triangulation of the domain Ω , $n_e \in \mathbb{N}$ the number of triangles T_i such that $\bar{\Omega}_h = \cup_{i=1}^{n_e} T_i$ and n the number of nodes of the triangulation Ω_h. For any two triangles, their clousures are either disjoint or have a common vertex or a common edge. Finally, let $\{P_j\}_{j=1,\cdots,n}$ be the vertices (nodes) associated with Ω_h. Taking this into account, we define

$$V_h := \{v_h \in C(\bar{\Omega}_h) : v_h|_{T_i} \in \mathbb{P}_1, \ \ \forall T_i \in \Omega_h\},$$

where $\mathbb{P}_1$ is the space of continuous piecewise linear functions defined on Ω_h. Then the following space

$$V_h^0 = W_0^{1,p}(\Omega) \cap V_h \tag{4}$$

is the finite-dimensional space associated with the triangulation Ω_h.

Considering the previous analysis, the finite element approximation of (3) is formulated as follows

$$\min_{u_h \in V_h^0} J_{\gamma,h}(u_h) := \frac{1}{p}\int_{\Omega_h} |\nabla u_h|^p \, dx + \int_{\Omega_h} \psi_\gamma(\nabla u_h)\, dx - \int_{\Omega_h} f u_h \, dx. \tag{5}$$

2. MULTIGRID METHODS

In this chapter we present the multgrid (MG) methods for solving discretized partial differential equations. This methods constitutes the basis of the multigrid optimization algorithm that will be applied in the development of this thesis. As the name implies, in the multigrid methods we work with problems discretized in different grids or meshes of several sizes. The method involves the application of an iterative method for solving the discrete partial differential equation. Then, the two main ideas of the algorithm is to take advantage of the smoothing effect that several iterative methods have on the error of an approximated solution and, using the size of the grids, to approximate this smooth error on a coarse grid in order to correct quantities. In what follows we explain these two considerations introducing the multigrid method for the discretized *Poisson* problem with Dirichlet boundary conditions.

$$\begin{aligned} -\Delta_h u_h(x,y) &= f_h(x,y) \text{ in } \Omega_h \\ u_h(x,y) &= g_h(x,y) \text{ on } \partial\Omega_h \end{aligned} \tag{6}$$

where $\Omega \subset \mathbb{R}^2$, $h = \frac{1}{n}$, with $n \in \mathbb{N}$.

There is a wide range of iterative methods to solve this particular problem, we can enumerate the Gauss-Seidel, Jacobi, Conjugate gradient and SOR methods among others. Let us denote by IM (Iteration Method) the iteration formula of any method and u_h^l the aproximation of $u_h(x_i, y_j)$ at iteration l. Then, we have

$$u_h^{l+1}(x_i, y_j) = IM(u_h^l(x_i, y_j), f_h)$$

where $(x_i, y_j) \in \Omega_h$. If we apply the previous equation to the Poisson problem a few times, the error of the aproximation $v_h^l(x_i, y_j) = u_h(x_i, y_j) - u_h^l(x_i, y_j)$ becomes smooth. Hence, the iteration formula can be interpreted as an error averaging process. Classical iterative methods have the property of smoothing the error for discrete elliptic problems [23, Sec 1.5]. The second main property of the multigrid approach consists in approximating the error on a coarse grid, this procedure is less expensive due to the fact that we have fewer grid points. It is called the *coarse grid correction* procedure.

Let us illustrate the smoothing process and the coarse grid principle by looking at the error $v_h(x, y)$. Since it is a function of x and y we can rewrite it as follows:

$$v_h(x,y) = \sum_{k,m=1}^{n-1} \alpha_{k,m} \sin k\pi x \sin m\pi y. \tag{7}$$

Here, the functions

$$\varphi_h^{k,m}(x,y) = \sin k\pi x \sin m\pi y \qquad (k, m = 1, \cdots, n-1) \tag{8}$$

are the discrete eigenfunctions of the discrete operator Δ_h. Then, the error has high frequency and low frequency components. We call high frequency components to the following functions:

$$\alpha_{k,m} \sin k\pi x \sin m\pi y \qquad \text{with } k \text{ or } m \text{ large} \tag{9}$$

and low frequency components to

$$\alpha_{k,m} \sin k\pi x \sin m\pi y \qquad \text{with } k \text{ or } m \text{ small} \tag{10}$$

The error becomes smooth because the high frequency components become small after some iterations steps of the iterative method. On the other hand, the low frequency components hardly change.

The coarse grid principle is explained as follows: let us consider the Poisson problem on a grid Ω_h with mesh size $h = \frac{1}{n}$. Since we have to approximate the error on a coarse grid, we consider the coarser grid Ω_H with mesh size $H = 2h$, which is always used in the multigrid framework. Then, we observe that the following eigenfunctions coincide on Ω_H in the following sense [23, Sec. 1.5.2]:

$$\varphi^{k,m}(x,y) = -\varphi^{n-k,m}(x,y) = -\varphi^{k,n-m}(x,y) = \varphi^{n-k,n-m}(x,y),$$

for $(x,y) \in \Omega_{2h}$.

Then, the previous eigenfunctions cannot be distinguished on Ω_H. Thus, we can redefine the components of the error as follows:

- low frequency if $\max(k,m) < \frac{n}{2}$,
- high frequency if $\frac{n}{2} \leq \max(k,m) < n$.

For k or $m = \frac{n}{2}$, the components $\varphi^{k,m}$ vanish on Ω_{2h}. Then, we can approximate the error on a coarser grid.

2.1 Two-grid scheme

In what follows we present the multigrid algorithm for solving the Poisson problem (6), we introduce the smoothing procedure and the coarse grid correction priciples as the fundamental ideas inside the multigrid approach. In order to illustrate the method we work with two grids and use the matrix notation A_h instead of the operator $-\Delta_h$ with the Dirichlet boundary coditions. For simplicity, we drop the dependence of the pair (x_i, y_j). Then we have the system

$$A_h u_h = f_h. \tag{11}$$

Using an iterative method with a smoothing property we have, after a few (ν_1) iterations of the method, an approximated solution $u_h^{\nu_1}$. The error is denoted by

$$v_h = u_h - u_h^{\nu_1}$$

and the residual is given by

$$r_h = f_h - A_h u_h^{\nu_1}. \tag{12}$$

Then, we have the residual equation

$$A_h v_h = r_h. \tag{13}$$

Since $u_h = u_h^{\nu_1} + v_h$, the residual equation (13) is equivalent to (11), however, v_h and r_h are smooth. Then, without any important loss of information, v_h can be approximated on a coarse level as the solution of a coarse problem defined by

$$A_H v_H = r_H, \tag{14}$$

where $H = 2h$ is the size of the coarse mesh. As we can see from the previous system, we need to redefine the residual in the coarser mesh. Thus, we introduce the *fine-to-coarse grid transfer operator* I_h^H, which is a restriction operator that transfers information from the fine grid to the coarser one. Then we have that $r_H = I_h^H r_h$. On the other hand, A_H corresponds to the $-\Delta_H$ operator discretized on the mesh with size H. If we solve system (14) we obtain v_H, which can be seen as an approximation of v_h on a coarse grid, i.e., an smooth approximation of the error of the solution in a coarse grid. Hence, we can interpolate this correction to the fine grid through the *coarse-to-fine grid transfer operator* I_H^h. Since we have that $u_h = u_h^{\nu_1} + v_h$, we can update the solution and compute a new approximation as follows

$$u_h^{new} = u_h^{\nu_1} + I_H^h v_h. \tag{15}$$

Taking into account the previous disscusion, we present the twogrid algorithm. Let us recall that the application of any iterative method (with smoothing properties) to solve problem (14) is denoted by $u_h^l = IM(u_h^{l-1}, f_h)$. In the literature, it is usual to denote this smothing procedure as $u_h^l = S(u_h^{l-1}, f_h)$. Finally, the two-grid algorithm reads as follows:

Algoritmo 2.1 Two-grid method for solving $A_h u_h = f_h$.

Pre-smoothing

- Compute $u_h^{\nu_1}$ applying ν_1 iterations of an iterative method:

$$u_h^\ell = IM(u_k^{\ell-1}, f_h), \;\; \ell = 1, \ldots, \nu_1.$$

Coarse-grid correction

- Computation of the residual: $r_h = f_h - A_h u_h^{\nu_1}$
- Restrict: $r_H = I_h^H r_h$.
- Solve: $A_H v_H = r_H$ on Ω_H.
- Interpolate the correction: $\hat{v}_h = I_H^h v_H$.
- Compute the corrected approximation: $u_h^{\nu_1+1} = u_h^{\nu_1} + \hat{v}_h$.

Post-optimization

- Apply ν_2 iterations of an iterative algorithm:

$$u_h^l = IM(u_h^{l-1}, f_h), \;\; \ell = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1.$$

In the coarse-grid correction procedure, when interpolating the correction $\hat{v}_h = I_H^h v_H$, the procedure may introduce some errors. Then, it is necessary to apply ν_2 iterations of the smoothing process.

2.2 Multigrid scheme

The two-grid method corresponds to the basis of the multigrid, MG, algorithm. However, the two-grid scheme is not widespread used in practice due to the fact that the coarse problem is still very large. Even more, there is no need to solve the coarse system (14) exactly. Since the residual equation in the coarse space has the same form as system (11), we can use an approximation to v_H. The idea of the multigrid algorithm is to apply the two-grid scheme in order to determine an approximation to v_H. This means that we have to introduce an even coarser grid and coarser problem. In the multigrid algorithm, the previous idea is applied recursively until an specific coarsest grid, where the residual equation is solved by any method (even a direct method) because it is inexpensive to solve in the few points of the coarsest grid. In order to present the multigrid algorithm, we first need to introduce a sequence of partitions $\{\Omega_k\}_{k=0,\ldots,m}$ of Ω_h such that the mesh size of the grids satisfies that $h_k = \frac{1}{2}h_{k-1}$. Then, Ω_0 corresponds to the coarsest grid and Ω_m is the finest one. Also, the multigrid approach involves several auxiliary operators. As we are working with a set of meshes and the algorithm runs at each level of discretization, we need to transfer information among the different grids. Hence, we introduce the *fine-to-coarse grid transfer operator*, I_k^{k-1}, and the *coarse-to-fine grid transfer operator*, I_{k-1}^k. Given a coarse mesh denoted by Ω_{k-1}, we can obtain a finer mesh Ω_k by regular subdivision. As the name implies, the *coarse-to-fine grid transfer operator* transfers information from the coarse mesh Ω_k to the finer mesh Ω_{k-1}. It is also called the prolongation operator. The *fine-to-coarse grid transfer operator* or restriction operator transfers information from the fine grid to the coarser one. Once we have introduced the auxiliary operators we are ready to introduce the multigrid algorithm. In the multigrid scheme we introduce the cycle index γ, which corresponds to the number of times the multigrid scheme is applied to obtain a good aproximation to the solution of $A_{k-1}v_{k-1} = r_{k-1}$. When $\gamma = 1$ the multigrid scheme is called *V-cycle*, in the case $\gamma = 2$ we refer it as the *W-cycle* (see Figure 1).

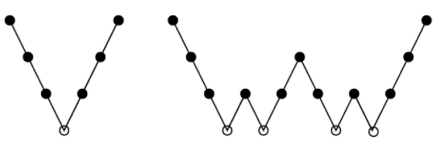

Figura 1. MG scheme with four grids. Left: $\gamma = 1$, right: $\gamma = 2$ [23, Sec. 2.4].

Algoritmo 2.2 Multigrid algorithm for solving $A_k u_k = f_k$.

If $k = 0$, solve $A_k u_k = f_k$ directly.
Pre-smoothing

- Compute $u_k^{\nu_1}$ applying ν_1 iterations of an iterative method:

$$u_k^{\ell} = IM(u_k^{\ell-1}, f_h), \;\; \ell = 1, \ldots, \nu_1.$$

Coarse-grid correction

- Computation of the residual: $r_k = f_k - A_k u_k^{\nu_1}$
- Restrict: $r_{k-1} = I_k^{k-1} r_k$.

Compute an approximated solution v_{k-1} of the residual equation

$$A_{k-1} v_{k-1} = r_{k-1}$$

on Ω_{k-1} by performing k-grid cycles as follows:

- Set $v_{k-1} = 0$
- Call γ times the MG scheme to solve $A_{k-1}v_{k-1} = r_{k-1}$.
- Interpolate the correction: $v_k = I_{k-1}^k v_{k-1}$.
- Compute the corrected approximation: $u_k^{\nu_1+1} = u_k^{\nu_1} + v_k$.

Post-optimization

- Apply ν_2 iterations of an iterative algorithm:

$$u_k^{l} = IM(u_k^{l-1}, f_h), \;\; \ell = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1.$$

2.3 Full multigrid scheme

The main idea of the full multigrid, FMG, scheme is to provide a good initial approximation at each level k of discretization, i.e., at each grid Ω_k, we start with a good approximation. Given a discrete problem, the FMG scheme uses nested iterations starting from a coarse grid, at this level, since we have few points in the grid, we can inexpensively solve the discrete problem. Then, the algorithm interpolates this solution to the next finer grid. The interpolated solution corresponds to the first guess at this new level. The algorithm repeats this process up to a certain finest grid. It is important to recall that, at each level of discretization, we perform a few, r, cycles of the MG scheme. The structure of the FMG is presented in Figure 2.

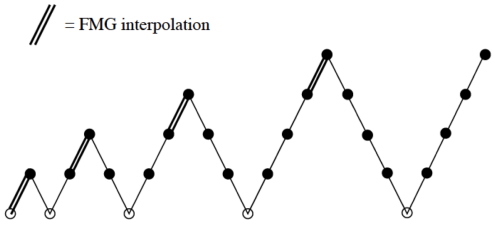

Figura 2. FMG scheme with $r = 1$ and $\gamma = 1$ [23, Sec. 2.6].

As we can see from Figure 2, the FMG scheme involves an interpolator operator that will be denoted by Π_{k-1}^{k}, generally speaking, this operator is of a higher accuracy than the operator I_{k-1}^{k}. However, if no such accuracy is needed, one can use the *coarse-to-fine grid transfer operator*, I_{k-1}^{k}. In what follows we present the FMG algorithm.

Algoritmo 2.3 Full multigrid algorithm for solving $A_k u_k = f_k$.

For $k = 0$, solve $A_k u_k = f_k$, providing $u_k^{FMG} = u_0$.

For $k = 1, 2, \cdots, m$:

- $u_k^0 := \Pi_{k-1}^{k} u_{k-1}^{FMG}$
- Apply r times MG γ-cycles to solve $A_k u_k = f_k$ initialized with u_k^0.
- $u_k = u_k^{FMG}$.

2.4 Multigrid methods for nonlinear problems, FAS scheme

The multigrid algorithm can also be applied to nonlinear problems. The most common multigrid algorithm in the nonlinear framework is the *full approximation scheme*, FAS [2]. The structure of this algorithm constitutes the basis for the multigrid optimization method and other advanced multigrid techniques [23, Sec. 5.3.7]. In what follows we briefly present the FAS idea for solving the nonlinear differential equation:

$$N_k(u_k) = f_k, \tag{16}$$

where $N_h(\cdot)$ is a discrete nonlinear differential operator. At the starting point of the FAS algorithm we apply a few times a nonlinear iterative method (or a relaxation type method [23, Sec 5.3.2]), for solving problem (16). As we know from the previous sections, this procedure corresponds to the smoothing process of the error and it is denoted by $u_k = IM(u_k, f)$. Since we apply this process only a few times we obtain an approximated solution $\tilde{u}_k$. Thus, the desired solution u_k is given by $u_k = \tilde{u}_k + v_k$, where v_k is the error at level k. Hence, we rewrite the problem as follows.

$$N_k(\tilde{u}_k + v_k) = f_k.$$

If we define the residual as $r_k = f_k - N_k(\tilde{u}_k)$, we can write the *correction* equation in the following way.

$$N_k(\tilde{u}_k + v_k) - N_k(\tilde{u}_k) = r_k. \tag{17}$$

Now, let's represent $\tilde{u}_k + v_k$ on the coarse grid in terms of the coarse-grid variable

$$\hat{u}_{k-1} := \hat{I}_k^{k-1}\tilde{u}_k + v_{k-1}.$$

Here, in contrast to the multigrid scheme, we perform a restriction procedure of the approximated solution $\tilde{u}_k$ through the operator $\hat{I}_k^{k-1}$. This operator may be different from I_k^{k-1} [23, Sec. 5.3.4]. In the same way, we formulate equation (17) on the coarse level by replacing $N_k(\cdot)$ by $N_{k-1}(\cdot)$, $\tilde{u}_k$ by $I_k^{k-1}\tilde{u}_k$, and r_k by $I_k^{k-1}r_k = I_k^{k-1}(f_k - N_k(\tilde{u}_k))$. Finally, we get the FAS equation:

$$N_{k-1}(\hat{u}_{k-1}) = I_k^{k-1}(f_k - N_k(\tilde{u}_k)) + N_{k-1}(I_k^{k-1}\tilde{u}_k). \tag{18}$$

This equation can be rewritten as follows

$$N_{k-1}(\hat{u}_{k-1}) = I_k^{k-1}f_k + \tau_k^{k-1},$$

where

$$\tau_k^{k-1} = N_{k-1}(I_k^{k-1}\tilde{u}_k) - I_k^{k-1}N_k(\tilde{u}_k),$$

this term is called *fine-to-coarse residual correction*. Then, a simple but important fact is that the fine grid is now used as a mechanism for calculating the correction τ_k^{k-1} to the FAS equation. The next step consists in the *coarse-grid* correction: the interpolation $I_{k-1}^{k}\hat{u}_{k-1}$ introduces errors of the full solution $\hat{u}_{k-1}$ instead of only the error v_{k-1} as in the multigrid scheme. For this reason, the following coarse-grid correction is used

$$u_k = \tilde{u}_k + I_{k-1}^{k}(\hat{u}_{k-1} - I_k^{k-1}\tilde{u}_k).$$

Algoritmo 2.4 FAS scheme for solving $N_k(u_k) = f_k$

if $k = 0$ **then**
  solve $N_k(u_k) = f_k$ directly.
end if
Pre-smoothing steps:
$u_k^l = S(u_k^{l-1}, f_k)$, for $l =, 1, \cdots, \nu_1$.
Coarse grid correction:

- Computation of the residual: $r_k = f_k - N_k u_k^{\nu_1}$.
- Restriction of the residual: $r_{k-1} = I_k^{k-1}r_k$.
- Set $u_{k-1} = I_k^{k-1}u_k^{(\nu_1)}$.
- Set $f_{k-1} = r_{k-1} + N_{k-1}(u_{k-1})$.
- Call $\hat{\gamma}$ times the FAS scheme to solve $N_{k-1}(u_{k-1}) = f_{k-1}$.
- Compute the corrected approximation: $u_k^{\nu_1+1} = u_k^{(\nu_1)} + I_{k-1}^{k}(u_{k-1} - I_k^{k-1}u_k^{(\nu_1)})$.

Post-smoothing steps on the fine grid:
$u_k^{(l)} = S(u_k^{(l-1)}, f_k)$, for $l = \nu_1 + 2, \cdots, \nu_1 + \nu_2 + 1$.

For further details of the FAS scheme we refer the reader see [2]. Considering the scheme of the FAS method, in the next chapter we introduce the multigrid optimization, MG/OPT algorithm. Multigrid for optimization problems, MG/OPT In this chapter, we present the multigrid optimization (MG/OPT) algorithm for solving the discretized and regularized optimization problem (5). The MG/OPT method corresponds to a nonlinear programming adaptation of the *full approximation scheme*, FAS (see [2, 23]). The multigrid subproblems arising

from the different discretization levels are nonlinear optimization problems [17]. The MG/OPT method was introduced as an efficient tool for large scale optmization problems (see [19, 17]). In fact, the idea of the algorithm is to take advantage of the solutions of problems discretized in coarse meshes to compute search directions for finer problems. The efficient resolution of coarse problems provide a way to calculate search directions for large optimization problems.

3. MG/OPT Algorithm

Now we are ready to discuss the MG/OPT method for problem (5). In order to understand the link between the FAS structure and the MG/OPT scheme, we present one iteration loop for the two-grid optimization algorithm ([3, Sec.6]). As we are working only with two grids, the finest grid is Ω_1 and the coarsest one is denoted by Ω_0. Therefore the solutions u_1 and u_0 correspond to the solutions in the grids Ω_1 and Ω_0 respectively.

In the two-grid algorithm, at the coarsest grid, $k = 0$, we solve

$$\min_{u_k} J_{\gamma,k}(u_k),$$

otherwise, we apply ν_1 iterations of an optimization algorithm to the problem and obtain an approximated solution $u_1^{\nu_1}$. Consequently the desired solution u_1 is given by $u_1 = u_1^{\nu_1} + e_1$, for some error e_1, and the problem

$$\min_{u_1} J_{\gamma,1}(u_1)$$

is equivalent to solving $\nabla J_{\gamma,1}(u_1) = 0$. Thus, we can write the problem as follows

$$\nabla J_{\gamma,1}(u_1^{\nu_1} + e_1) = 0$$

or

$$\nabla J_{\gamma,1}(u_1^{\nu_1} + e_1) - \nabla J_{\gamma,1}(u_1^{\nu_1}) = -\nabla J_{\gamma,1}(u_1^{\nu_1}). \tag{19}$$

Now, following the idea of the FAS scheme, our aim is to present this problem on the coarsest grid. Therefore, we restrict $u_1^{\nu_1} + e_1$ to the grid Ω_0 as follows

$$u_0 = I_1^0 u_1^{\nu_1} + e_0.$$

Then, we can represent (19) on Ω_0. The main idea is to obtain the structure of the FAS equation (18) in order to take advantage of the FAS algorithm's scheme. Hence, we formulate equation (19) on the coarse level by applying the operator I_1^0 to the right-hand side. The left-hand side is represented by $\nabla J_{\gamma,0}(\cdot)$ and applying I_1^0 to $u_1^{\nu_1}$. Then, the equation reads as

$$\nabla J_{\gamma,0}(u_0) - \nabla J_{\gamma,0}(I_1^0 u_1^{\nu_1}) = -I_1^0 \nabla J_{\gamma,1}(u_1^{\nu_1}). \tag{20}$$

If we denote

$$\tau_0 = \nabla J_{\gamma,0}(I_1^0 u_1^{\nu_1}) - I_1^0 \nabla J_{\gamma,1}(u_1^{\nu_1}),$$

then we have the following equation:

$$\nabla J_{\gamma,0}(u_0) = \tau_0. \tag{21}$$

At this point, solving the previous equation is equivalent to solving the optimization problem

$$\min_{u_0}(J_{\gamma,0}(u_0) - f_0^\top u_0),$$

where $f_0 = \tau_0$ in the grid Ω_0. Suppose that the solution to the problem in the coarsest grid is u_0. Then, the next step is called the coarse-to-fine minimization step and consists in performing a line search procedure

$$u_1 = u_1^{\nu_1} + \alpha I_0^1(u_0 - I_1^0 u_1^{\nu_1}). \tag{22}$$

Where α is the step size and the descent direction is $I_0^1(u_0 - I_1^0 u_1^{\nu_1})$. In the last step we apply ν_2 iterations of an optimization algorithm to the original problem inicialized with u_1. Finally, we obtain $u_1^{\nu_1+\nu_2+1}$. At this point, we can compare the scheme presented with the FAS algorithm. An important but simple fact is that MG/OPT algorithm is a programming adaptation of the FAS scheme.

Once we have introduced the algorithm for two grids we first make some comments before introducing the multigrid algorithm. MG/OPT is related to different optimization techniques ranging from the gradient method to quasi Newton methods to solve the problems at each level. The multigrid for optimization approach makes minimal requests about the underlying optimization algorithm. However, it is important to highlight that at each level of discretization we need to find an estimated solution for the minimization subproblem. Then, the election of the underlying optimization algorithm is not trivial and depends on the inner characteristics of the optimization problem. As our goal is to solve problem (5), we use a class of descent algorithms as the underlying algorithm. This choice was made based on the structure of problem (5). As the p - Laplacian is involved in the functional, we have to consider that its finite element approximation derives in a highly nonlinear and degenerate finite dimensional problem [13]. Also, the functional $J_{\gamma,h}$ involves a semismooth regular function. Then, the class of descent algorithms chosen is suitable to deal with this issue. As we mentioned before, the main idea of the MG/OPT algorithm is to use coarse problems to generate, recursively, search directions for finer problems. Then, a line search procedure, along with the underlying opimization algorithm is used to improve the solution on each level of discretization.

In what follows we present the MG/OPT algorithm. The underlying optimization algorithm will be denoted by S_{opt} inside the multigrid approach. The initial discretized problem is given on the finest grid. To facilitate the implementation of the algorithm, the MG/OPT scheme is presented in a recursive formulation. Hence, we introduce the following slightly different notation for the optimization problem

$$\min_{u_k} \left(\hat{J}_{\gamma,k}(u_k) - f_k^\top u_k\right).$$

We set $f_k = 0$ at the finest level $k = m$. $\hat{J}_{\gamma,k}$ corresponds to the functional J_γ discretized at each level $k = 1, \ldots, m$. Summarizing, the algorithm is presented as follows.

Algoritmo 3.1 MG/OPT recursive(ν_1, ν_2).

if $k = 0$ **then**, solve $\min_{u_k} \left(\hat{J}_{\gamma,k}(u_k) - f_k^\top u_k\right)$ and return.
end if
Otherwise, $k > 0$.
Pre-optimization: Apply ν_1 iterations of the optimization algorithm to the problem at level k.

$$u_k^\ell = S_{opt}(u_k^{\ell-1}), \quad \ell = 1, \ldots, \nu_1.$$

Coarse-grid correction.

- Restrict: $u_{k-1}^{\nu_1} = I_k^{k-1} u_k^{\nu_1}$.
- Compute the fine-to-coarse gradient correction:

$$\tau_{k-1} := \nabla \hat{J}_{\gamma,k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla \hat{J}_{\gamma,k}(u_k^{\nu_1}).$$

- Define $f_{k-1} := I_k^{k-1} f_k + \tau_{k-1}$ and apply one cycle of MGOPT(ν_1, ν_2) to

$$\min_{u_{k-1}} \left(\hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^\top u_{k-1}\right)$$

to obtain $\widetilde{u}_{k-1}$.

Coarse-to-fine minimization.

- Prolongate error: $e := I_{k-1}^k(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1})$.
- Line search in e direction to obtain a step size α_k.
- Calculate the coarse-to-fine minimization step: $u_k^{\nu_1+1} = u_k^{\nu_1} + \alpha_k e$.

Post-optimization: Apply ν_2 iterations of the optimization algorithm to the problem at level k.

$$u_k^\ell = S_{opt}(u_k^{\ell-1}), \quad \ell = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1.$$

The algorithm presented above contemplates one iteration of a V-cycle initialized with a rough estimate of the solution on the finest grid.

4. Numerical Experiments

In this section we present the application of the MG/OPT method to numerical simulation of the steady flow of viscoplastic fluids. These materials are characterized by the existence of a yield stress [6, 9, 14]. This implies that the viscoplastic material exhibits no deformation if the shear stress imposed does no exceed the yield stress, i.e., it behaves as an ideal rigid solid. On the other hand, if the shear stress overpasses the yield stress, the material will deform as a nonlinear viscous fluid in most of the cases. In fact, Herschel - Bulkley and Casson fluids present a nonlinear stress-shear rate relationship, while Bingham fluids behave as a viscous fluid with linear stress-shear rate relationship (see Figure 3). Summarizing, the existence of the yield stress makes the flow of these materials to present rigid zones, known as the plug flow, and yielded zones.

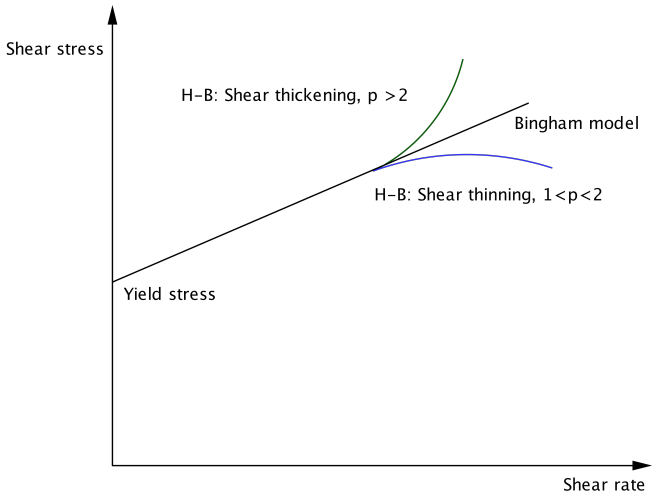

Figura 3. Viscoplastic models

In this work we simulate the pipe flow for the classic viscoplastic model Herschel-Bulkley. Other models like Bingham and Casson fit the kind of nonsmooth optimization problems under study. In fact, it is well known that the velocity field of the flow across the cross-section of a pipe can be approximated by the solution of the following discretized minimization problem:

$$\min_{u_h \in V_h^0} J_h(u_h) := \phi(\nabla u_h) + \int_{\Omega_h} \psi_\gamma(\nabla u_h)\, dx - \int_{\Omega_h} f u_h\, dx, \tag{23}$$

where

$$\phi(\nabla u_h) = \begin{cases} \dfrac{1}{p}\displaystyle\int_{\Omega_h} |\nabla u_h|^p\, dx, & \text{H-B.} \\ \dfrac{1}{2}\displaystyle\int_{\Omega_h} |\nabla u_h|^2\, dx, & \text{Bing.} \\ \dfrac{1}{2}\displaystyle\int_{\Omega_h} |\nabla u_h|^2\, dx + \dfrac{4}{3}\sqrt{g}\displaystyle\int_{\Omega_h} |\nabla u_h|^{\frac{3}{2}}\, dx, & \text{Casson.} \end{cases}$$

In the coming numerical experiments, we present the results of the MG/OPT algorithm applied to the numerical solution of (23) for Herschel-Bulkley. For the computations in the MG/OPT algorithm we implement a V-cycle scheme with $\nu_1 = \nu_2 = 2$ as the pre and post optimization iterations. In all algorithms the stopping criteria is fixed at a tolerance of 10^{-7}. We perform the numerical experiments in two types of domains: unit square and unit circle. Also, we compare the performance of the MG/OPT algorithm with the performance of the underlying optimization algorithm when solving the same problem in the finest grid.

In the following tables we summarize the information of the different grids used at each level k of the V-cycle implemented for the MG/OPT algorithm.

Mesh (Ω_k)	Nodes (n_k)	Elements (n_e)
Ω_6	8321	16384
Ω_5	2113	4096
Ω_4	545	1024
Ω_3	145	256
Ω_2	41	64
Ω_1	13	16

Tabla 1. Unit circle mesh information.

Mesh (Ω_k)	Nodes (n_k)	Elements (n_e)
Ω_5	4225	8192
Ω_4	1089	2048
Ω_3	289	512
Ω_2	81	128
Ω_1	25	32

Tabla 2. Unit square mesh information.

4.1 Herschel-Bulkley

Herschel-Bulkley fluids are power-law materials with plasticity. The behaviour of these fluids depends on the value of p, which plays the role of the flow index. If $1 < p < 2$ the material exhibits a pseudoplastic or shear-thinning behaviour. On the other hand, if $p > 2$ the behaviour is shear-thickening (see Figure 3). Thanks to this index, the power-law model has been widely used to characterize several materials that include liquid foams, whipped cream, fluid foods, silly putty and some polymers.

In the following experiments, we compute the velocity field for a Herschel-Bulkley material for $1 < p < 2$ in a pipe, considering circle and square cross sections. In these experiments we stablished the preconditioned descent algorithm (see [9]) as the underlying optimization algorithm S_{opt}. We set $\epsilon = 10^{-6}$.

Experiment 1

In this experiment we set the following parameters, $p = 1{,}75$ and $f = 1$. In Table 3, each row represents one experiment. For each experiment, we present the finest and coarsest mesh, the number of V-cycles of the MG/OPT algorithm until the stopping tolerance is achieved, the tolerance reached and the execution time of the algorithm i.e., the CPU time when the stopping criteria is achieved.

This experiment was initialized with the solution of the Poisson problem, $-\Delta u_h = f$. From Table 3 we can notice that the number of V-cycles is similar when solving the problem at the different levels of discretization. In Figure 5 we can see the decay of the norm $|\nabla J_\gamma^\top e|$. It behaves typically as in a steepest descent algorithm. However, it decays faster in the last iterations. This behaviour is inherited from the preconditioned descent algorithm (see [9, Sec. 4.3.1]). The resulting velocity field is displayed in Figure 4. The flattening of the velocity in the center of the pipe corresponds to the plug flow velocity, where the material presents rigid zones.

g	F. mesh.	C. mesh.	V. C	$\lvert\nabla J_\gamma^\top e\rvert$	Time
0.2	Ω_6	Ω_1	11	7.38e-07	965.49
	Ω_5	Ω_1	15	3.75e-07	258.9
	Ω_4	Ω_1	13	4.70e-07	51.2
	Ω_3	Ω_1	10	3.75e-07	10.04
0.4	Ω_6	Ω_1	26	1.08e-07	3914.61
	Ω_5	Ω_1	17	3.79e-07	329.07
	Ω_4	Ω_1	13	7.23e-07	51.33
	Ω_3	Ω_1	23	5.00 e-07	24.07

Tabla 3. Results of the resolution of problem (23) with $p = 1{,}75$, $\gamma = 10^3$ and $f = 1$.

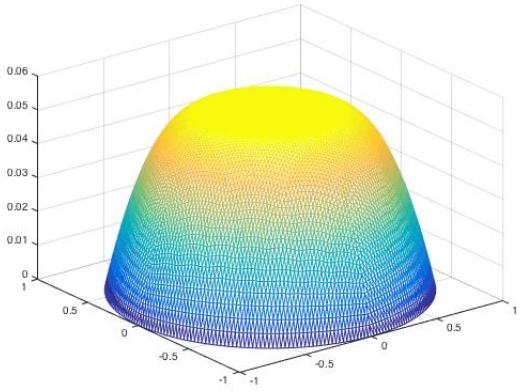

Figura 4. Calculated velocity u for mesh Ω_6. Parameters: $p = 1{,}75, g = 0{,}2, \gamma = 10^3$ and $\epsilon = 10^{-6}$

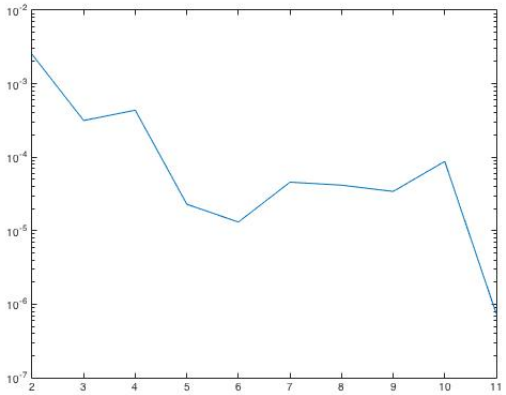

Figura 5. Calculated $|\nabla J_\gamma^\top e|$ for mesh Ω_6 and $g = 0{,}2$.

In Tables 4, 5, 6 and 7, we present the performance of the line search globalization technique, in 4 V-cycles randomly chosen for Experiment 1. We consider $g = 0{,}2$, Ω_6 as the finest mesh and Ω_1 as the coarsest one. The experiment finished after 11 V-cycles, and we perform 5 line

search procedures at each cycle. Also, the second column of each table let us observe that e is, indeed, a descent direction at each level.

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	α_k	l.s it
Ω_1 -Ω_2	-1.36e-11	1	0
Ω_2 -Ω_3	-8.75e-09	1	0
Ω_3 -Ω_4	-1.20e-05	1	0
Ω_4 -Ω_5	-0.0013	0.2040	1
Ω_5 -Ω_6	-0.0108	0.2673	1

Tabla 4. Line search for V-cycle 1

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	α_k	l.s it
Ω_1 -Ω_2	-1.95e-08	1	0
Ω_2 -Ω_3	-2.50e-05	0.2228	1
Ω_3 -Ω_4	-0.0012	0.0121	3
Ω_4 -Ω_5	-2.29e-04	0.0135	3
Ω_5 -Ω_6	-6.99e-05	0.0232	3

Tabla 5. Line search for V-cycle 4

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	α_k	l.s it
Ω_1 -Ω_2	-1.04e-08	1	0
Ω_2 -Ω_3	-5.38e-06	0.2758	1
Ω_3 -Ω_4	-1.41e-04	0.1493	1
Ω_4 -Ω_5	-1.49e-04	0.0067	4
Ω_5 -Ω_6	-1.84e-05	0.0722	2

Tabla 6. Line search for V-cycle 7

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	α_k	l.s it
Ω_1 -Ω_2	-4.47e-09	1	0
Ω_2 -Ω_3	-1.70e-06	1	0
Ω_3 -Ω_4	-8.52e-05	0.0194	2
Ω_4 -Ω_5	-2.44e-05	0.0119	3
Ω_5 -Ω_6	-7.29e-06	0.0619	2

Tabla 7. Line search for V-cycle 10

Experiment 2: Comparison between MG/OPT and descent algorithms

In this experiment, we compare the behaviour of the MG/OPT approach versus an optimization algorithm for solving the same problem in the finest grid. In the MG/OPT method these optimization algorithms were used as the underlying optimization algorithms as well. In the following tables we compare the CPU time and the stopping criteria registered for solving the problem described in the previous experiment. Once again, in Tables 8 and 9 we compare one experiment at each row based on the election of the finest grid.

Time (s)		
Mesh	MG/OPT	Gradient method
Ω_4	196.76	-
Ω_3	73.57	2883.55

Tabla 8. Time comparison, Experiment 1

$\lvert\nabla J_\gamma^\top e\rvert$		
Mesh	MG/OPT	Gradient method
Ω_4	3.76e-07	-
Ω_3	2.09e-07	9.92e-07

Tabla 9. Norm $|\nabla J_\gamma^\top e|$comparison, Experiment 1.

As it was expected, it is shown that the MG/OPT performance is more efficient than the descent gradient method. Even if the gradient algorithm is stablished as the underlying optimization algorithm of the MG/OPT. In what follows we present the same comparison criteria for the preconditioned descent algorithm and the multigrid optimization scheme.

Time (s)		
Mesh	MG/OPT	preconditioned descent alg.
Ω_6	**965.49**	2146.37
Ω_5	258.95	71.67
Ω_4	**51.23**	63.11
Ω_3	10.04	8.30

Tabla 10. Time comparison, Experiment 1

$\lvert\nabla J_\gamma^\top e\rvert$		
Mesh	MG/OPT	preconditioned descent alg.
Ω_6	7.38e-07	9.35e-07
Ω_5	3.75e-07	3.26e-07
Ω_4	4.70e-07	9.52e-07
Ω_3	3.75e-07	7.84e-07

Tabla 11. Norm $|\nabla J_\gamma^\top e|$comparison, Experiment 1

From Tables 10 and 11 we see that the MG/OPT performs better when reaching the stopping criteria in all cases. The CPU time registered and the tolerance reached are better when working with the MG/OPT algorithm than with the preconditioned descent algorithm at the finest grid Ω_6. In this case, CPU time decreases almost in half. However, we can not achieve CPU time savings in all the resolution levels. An interesting case is shown in Table 10, in this case, the preconditioned descent algorithm achieves the convergence tolerance in less time than the MG/OPT algorithm when working with the finest level Ω_5. However, in the bigger mesh Ω_6, which its

size satisfy that $h_6 = \frac{1}{2}h_5$, we reduced the time convergence to half.

5. Conclusions and Outlook

We proposed and analyzed a multigrid for optimization (MG/OPT) algorithm for the numerical solution of a class of quasilinear variational inequalities of the second kind. We analyzed the variational inequality via the minimization of the associated energy functional. First, we regularized the non-differentiable part of the energy by using a Huber regularization approach. Next, we proposed a finite element discretization for the problem. The MG/OPT algorithm was presented and we showed that several classical models for viscoplastic flow correspond to the class of variational inequalities under study. Therefore, we focussed the numerical experiments on this kind of problems.

5. Referencias

[1] R. Bermejo and J-. A. Infante, *A Multigrid Algorithm for the p-Laplacian*, SIAM J. Sci. Comput., 21 (2000) 1774-1789.

[2] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of computation, 21 (1977) 333-390.

[3] A. Borzì and M. Vallejos, *Multigrid Optimization Methods for Linear and Bilinear Elliptic Optimal Control Problems*, Computing, 82 (2008) 31-52.

[4] X. Chen, Z. Nashed and L. Qi, *Smoothing Methods and Semismooth Methods for Nondifferentiable Operator Equations*, SIAM J. Numer. Anal., 38 (2000) 1200-1216.

[5] J.E. Dennis Jr and R.B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations* , SIAM, 16 (1996)

[6] G. Duvant and J.L Lions, *Inequalities in mechanics and physics*, Springer-Verlag, (1976).

[7] M. Giaquinta and G. Modica, *Mathematical analysis: an introduction to functions of several variables*, Springer Science & Business Media, (2010).

[8] R. Glowinski and J-. A. Marroco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, 9 (1975) 41-76.

[9] S. González-Andrade, *A Preconditioned Descent Algorithm for Variational Inequalities of the Second Kind Involving the p-Laplacian Operator*, Computational Optimization and Applications. 66 (2017) 123-162.

[10] M. S. Gockenbach, *Understanding and Implementing the Finite Element Method*, SIAM, U.S.A, 2006.

[11] M. Hintermüller, K. Ito and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002) 865-888.

[12] M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich, *Optimization with PDE constraints* , Springer Science & Business Media , 23 (2008).

[13] Y.Q. Huang, R. Li and W. Liu, *Preconditioned descent algorithms for p-Laplacian* , Springer, Journal of Scientific Computing, 32 (2007) 343-371.

[14] R.R. Huilgol and Z. You, *Application of the augmented Lagrangian method to steady pipe flows of Bingham, Casson and Herschel-Bulkley fluids*, Journal of non-newtonian fluid mechanics, 128 (2005) 126-143.

[15] J. Jahn, *Introduction to the theory of nonlinear optimization*, Springer Science & Business Media, (2007).

[16] O. Lass, M. Vallejos, A. Borzì and C. C. Douglas, *Implementation and Analysis of Multigrid Schemes with Finite Elements for Elliptic Optimal Control Problems*, Computing, 84 (2009) 27-48.

[17] R. M. Lewis and S. G. Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM J. Scientific Computing, 26 (2005) 1811-1837.

[18] J.L Lions, *Optimal control of systems governed by partial differential equations*, Springer Verlag, 170 (1971).

[19] S. G. Nash, *A Multigrid Approach to Discretized Optimization Problems*, Optimization Methods and Software, 14 (2000) 99-116.

[20] J. Nocedal and S. Wright, *Numerical optimization* , Springer Science & Business Media, (2006).

[21] A. Quarteroni, M. Tuveri and J-. A. Veneziani, *Computational vascular fluid dynamics: problems, models and methods*, Springer J. Computing and Visualization in Science, 2 (2000) 163-197.

[22] A. Rao, *Rheology of Fluid, Semisolid, and Solid Foods* , Springer, (2014) 27-61.

[23] U. Trottenberg, C. W. Oosterlee and A. Schuller, *Multigrid*, Academic press, 2000.

[24] M. Ulbrich, *Semismooth Newton methods for variational inequalities and constrained optimization problems in function spaces* , SIAM. (2011)

[25] D. Sun and J. Han, *Newton and Quasi-Newton Methods for a Class of Nonsmooth Equations and Related Problems*, SIAM J. Optim., 7 (1997) 463-480.

[26] WP. Walawender, TY. Chen and DF. Cala, *An approximate Casson fluid model for tube flow of blood*, Biorheology, 12 (1975) 111-119.

[27] Z. Wen and D. Goldfarb, *A Line Search Multigrid Method for Large-Scale Nonlinear Optimization*, SIAM J. Optim., 20 (2009) 1478-1503.