ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

A MULTIGRID OPTIMIZATION APPROACH FOR THE NUMERICAL SOLUTION OF A CLASS OF VARIATIONAL INEQUALITIES OF THE SECOND KIND

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN OPTIMIZACIÓN MATEMÁTICA

TESIS

SOFÍA ALEJANDRA LÓPEZ ORDÓÑEZ
sofia.lopezo@epn.edu.ec

Director: SERGIO GONZÁLEZ ANDRADE PHD
sergio.gonzalez@epn.edu.ec

QUITO, MAYO 2017

DECLARACIÓN

Yo, SOFÍA ALEJANDRA LÓPEZ ORDÓÑEZ declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Sofía Alejandra López Ordóñez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por SOFÍA ALEJANDRA LÓPEZ ORDÓÑEZ, bajo mi supervisión.

Sergio González Andrade PhD
Director del Proyecto

ACKNOWLEDGMENTS

I would like to thank my advisor Sergio González Andrade, for his support and patience, for the clarifying discussions, for his accurate corrections and comments, for the privilege to work with him. Also, I would like to thank all colleagues working at the Research Center on Mathematical Modeling-ModeMat and the director Juan Carlos De los Reyes for their support. In particular, I would like to thank the professors: Pedro Merino, Miguel Yangari and Diego Recalde who closely followed the process and the director of the Master program Luis Miguel Torres, for his commitment and help.

Finally, I thank my friends and classmates: Cristhian Núñez, Kateryn Herrera, Myrian Guanoluiza and David Villacís.

This thesis was developed under the financial support of the following projects: *Flujos de Materiales Viscoplásticos en la Industria Alimenticia, EPN-PIMI 14-12*; *Sparse Optimal Control of Differential Equations - SOCDE, MATH-AmSud* and the project *Sistema de Pronóstico del Tiempo para todo el Territorio Ecuatoriano: Modelización Numérica y Asimilación de Datos, INAMHI-MODEMAT-EPN*.

To Enrique and Bibiana, to whom I owe most.

Contents

Abstract

In this thesis 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. This approach follows from the fact that the solution of the variational inequality is given by the minimizer of a nonsmooth energy functional, Therefore, we proposed a Huber regularization of the functional and a finite element discretization for the problem. Further, we analyze the regularity of the discretized energy functional, and we are able to prove that its Jacobian is slantly differentiable. This regularity property is useful to analyze the convergence of the MG/OPT algorithm. In fact, we demostrate that the algorithm is global convergent by using the mean value theorem for slantly differentiable functions. Finally, we analyze the performance of the MG/OPT algorithm when used to simulate the visco-plastic flow of Bingham, Casson and Herschel-Bulkley fluids in a pipe. Several numerical experiments are carried out to show the main features of the proposed method.

Chapter 1

Introduction

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

$$\int_\Omega |\nabla u|^{p-2}(\nabla u, \nabla(v-u))\,dx + g\int_\Omega |\nabla v|\,dx - g\int_\Omega |\nabla u|\,dx \geq \int_\Omega f(v-u)\,dx,\ \forall v \in W_0^{1,p}(\Omega), \tag{1.1}$$

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 [2, 11]). 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 (1.1) corresponds to a first order necessary optimality condition for the following optimization problems.

$$\min_{u \in W_0^{1,p}(\Omega)} J(u) := \frac{1}{p}\int_\Omega |\nabla u|^p\,dx + \int_\Omega |\nabla u|\,dx - \int_\Omega fu\,dx. \tag{1.2}$$

In consequence, throughout this work we will focus in the resolution of this op-

timization 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 [25, 23]). In [23] 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 implies, 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 [12]) 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 (1.2).

The main idea of the MG/OPT method consists in generate, recursively, search directions through solutions of coarse problems to optimize the solution of fine problems. Regarding to this, the combination of a globally convergent underlying optimization algorithm with a line search technique, ensures the decay of the functional value at each cycle or iteration. Thus, the multigrid optimization algorithm is globally convergent. However, the convergence depends critically on whether the search directions found at coarse levels are descent directions. Taking this into account, we propose a convergence analysis of the method based on a general notion of differentiability (slant differentiability) of the functional J. Then, in order to prove global convergence of the MG/OPT algorithm, it is usual to require that the differential of the functional satisfy the mean value theorem [25]. Thus, we focus our analysis on the differentiability of functional J. However, the functional J is not twice Gâteaux differentiable. Hence, we introduce a general notion of differentiability and we propose to use the mean value theorem for slantly differentiable functions [7] in order to prove convergence of the algorithm for problem (2.6). Finally, we perform numerical experiments for solving the pipe flow of visco-plastic fluids

such as Herschel-Bulkley, Bingham and Casson, which are modeled by variational inequalities of the second kind.

Chapter 2

Preliminary Results and Problem Statement

2.1 Preliminary Results

2.1.1 Generalized Differentiability

In this section we introduce several concepts of differentiabilily: the concept of slanting function, slant differentiability and the mean value theorem for slantly differentiable funcions.

Definition 2.1.1. *Let X and Y be two normed spaces, D be a nonempty open set in X and $J : D \subset X \to Y$ be a given mapping. For $x \in D$ and $h \in X$, if the limit*

$$J'(x)(h) := \lim_{t \to 0} \frac{J(x+th) - J(x)}{t}$$

exists, the function is said to be directionally differentiable. Further, $J'(x)(h)$ is the directional derivative of J at x in the direction h. If the directional derivative $J'(x)(h)$ exists for all $h \in X$ and $J'(x)$ is a linear and continuous operator from X to Y, then J is said to be Gâteaux differentiable.

Definition 2.1.2. *Let X and Y be Banach spaces, and D be an open domain in X. A function $J : D \subset X \to Y$ is said to be slantly differentiable at $x \in D$ if there exists a mapping $J^\circ : D \to \mathcal{L}(X, Y)$ such that the family $\{J^\circ(x+h)\}$ of bounded linear operators is uniformly bounded in the operator norm, for h sufficiently small, and*

$$\lim_{h \to 0} \frac{J(x+h) - J(x) - J^\circ(x+h)h}{\|h\|} = 0.$$

The function J° is called a slanting function for J at x.

DEFINITION 2.1.3. *A function $J : D \subset X \to Y$ is said to be slantly differentiable in an open domain $D_0 \subset D$ if there exists a mapping $J^\circ : D \to L(X, Y)$ such that J° is a slanting function for J at every point $x \in D_0$. In this case, J° is called a slanting function for J in D_0.*

DEFINITION 2.1.4. *Suppose that $J^\circ : D \to L(X, Y)$ is a slanting function for J at $x \in D$. We call the set*

$$\partial_S J(x) := \{\lim_{x_k \to x} J^\circ(x_k)\}$$

the slant derivative of J associated with J° at $x \in D$. Here, $\lim_{x_k \to x} J^\circ(x_k)$ exists for any sequence $\{x_k\} \subset D$ such that $x_k \to x$.

PROPOSITION 2.1.5.

1. *If J is continuosly differentiable in D, we can set $J^\circ(u) := J'(u) \ \ \forall u \in D$. Thus, J' is a slanting function for J at every point of D.*
2. *A slantly differentiable function J at u can have infinitely many slanting functions at u. If J° and H° are both slanting functions for J at $u \in D$, then*

$$P^\circ := \lambda J^\circ + (1 - \lambda) H^\circ$$

is also a slanting function for J at $u \in D$, where $\lambda \in [0, 1]$. Moreover,

$$\lim_{h \to 0} \|J^\circ(u + h)h - H^\circ(u + h)h\| = 0.$$

Proof. See [7, Sec.2] □

Next, we present an important example of a slanting differentiable function that will be useful in the subsequent sections.

EXAMPLE 2.1.6. *[14, Sec.3, Lemma 3.1] Let $g > 0$ be a constant. The mapping*

$$\vec{z} \to \max(g, \gamma|\vec{z}|)$$

from $\mathbb{R}^n$ to $\mathbb{R}$ is slantly differentiable on $\mathbb{R}^n$. Further, the slant derivative of this function is the characteristic function $\chi_{A_\gamma}(\vec{z})$ defined by

$$\chi_{A_\gamma}(\vec{z}) = \begin{cases} 1, & \text{if } \vec{z} \in A_\gamma, \\ 0, & \text{if } \vec{z} \in X \setminus A_\gamma, \end{cases}$$

where $A_\gamma := \{\vec{z} : \gamma|\vec{z}| \geq g\}$

We now present the mean value theorem for slantly differentiable functions (see [7, Sec. 2, Cor. 2.7, p. 1207]). This is a key result when proving convergence of the multigrid optimization method. The mean value theorem can be seen as a corollary of the following theorem that states the necessary and sufficient condition for slant differentiability.

THEOREM 2.1.7. *An operator $J : X \to Y$ is slantly differentiable at x if and only if J is Lipschitz continuous at x.*

Proof. See [7, Sec. 2]. □

Before presenting the mean value theorem we need to introduce the following lemma.

LEMMA 2.1.8. *Let X be a normed space and h be a fixed element of X, $h \neq 0$. Then there exists an element $g \in X^*$, where X^* is the dual of X, such that*

$$g(h) = \|h\| \text{ and } \|g\| = 1.$$

THEOREM 2.1.9. *(Mean value theorem for slantly differentiable funcions.) Let X and Y be Banach spaces, and D be an open domain in X. Suppose that $J : D \subset X \to Y$ is slantly differentiable at x. Then for any $h \neq 0$ such that $x + h \in D$, there exists a slanting function for J at x such that*

$$J(x+h) - J(x) = J^\circ(x+h)h.$$

Proof. Since J is slantly differentiable, we can construct a slanting function for J at x. From Lemma 2.1.8, we know that for each $h \neq 0$ there exists a continuous linear functional $g_h \in X^*$ such that $g_h(h) = \|h\|$ and $\|g_h\| = 1$. Then, we can define the following slanting function

$$J^\circ(x+h) := \frac{(J(x+h) - J(x))}{\|h\|} g_h, \tag{2.1}$$

thus, for each $h \neq 0$ we have that

$$\begin{aligned} J^\circ(x+h)h &= \frac{(J(x+h) - J(x))}{\|h\|} g_h(h) \\ &= J(x+h) - J(x). \end{aligned} \tag{2.2}$$

Now, it remains to prove that the family $\{J^\circ(x+h)\}$ is uniformly bounded for h sufficiently small.

From (2.1) and for any $z \in X$ we have that

$$J^\circ(x+h)z = \frac{J(x+h) - J(x)}{\|h\|} g_h(z).$$

Hence, since $g_h \in X^*$,

$$\begin{aligned} \|J^\circ(x+h)z\| &= \frac{\|J(x+h) - J(x)\|}{\|h\|} |g_h(z)| \\ &\leq \frac{\|J(x+h) - J(x)\|}{\|h\|} \|g_h\| \|z\| \\ &= \frac{\|J(x+h) - J(x)\|}{\|h\|} \|z\|. \end{aligned}$$

Since we defined J° as a bounded linear operator on X into Y, we have that

$$\sup_{z \neq 0} \frac{\|J^\circ(x+h)z\|}{\|z\|} \leq \frac{\|J(x+h) - J(x)\|}{\|h\|},$$

therefore

$$\|J^\circ(x+h)\| \leq \frac{\|J(x+h) - J(x)\|}{\|h\|}.$$

From Theorem 2.1.7 we know that J is Lipschitz continuous at x, then

$$\|J^\circ(x+h)\| \leq L.$$

□

2.1.2 Important Theorems in existence and uniqueness of the solution

In what follows we present some important results needed when proving existence and uniqueness of the solution of problem (1.1).

THEOREM 2.1.10. *Let S ve a nonempty, convex, closed and bounded subset of a reflexive real Banach space, and let $J : S \to \mathbb{R}$ be a continuous quasiconvex functional. Then J has at least one minimal point on S.*

Proof. See [Th. 2.12, Ch 2.][18] □

LEMMA 2.1.11. *Let S ve a nonempty, convex and closed subset of a real normed space. If the functional $J : S \to \mathbb{R}$ is continuous and quasiconvex, then J is weakly lower*

semicontinuous.

Proof. See [Lemma. 2.11, Ch 2.][18] □

THEOREM 2.1.12. *Consider the function*

$$J(u) = J_1(u) + J_2(u) \tag{2.3}$$

where we assume that the functions $J_i(u), i = 1, 2$, are continuous, convex, and lower semi-continuous in the weak topology. Further let

$$J(u) \to +\infty \text{ as } \|u\| \to +\infty, \text{ for all } u \in W_0^{1,p}(\Omega). \tag{2.4}$$

We assume that the function $u \to J_1(u)$ is differentiable, but J_2 is not necessarily differentiable. Finally assume that J is strictly convex. Then the unique element $v \in W_0^{1,p}(\Omega)$ such that $J(v) = \inf_{u \in W_0^{1,p}(\Omega)} J(u)$ is characterized by

$$J_1'(v) \cdot (u - v) + J_2(u) - J_2(v) \geq 0 \text{ for all } v \in W_0^{1,p}(\Omega). \tag{2.5}$$

Proof. See [Th. 1.6, Ch.1][24] □

2.2 Problem Statement

In this work, we are concerned with the numerical solution of the following class of nonsmooth optimization problems:

$$\min_{u \in W_p^{1,p}(\Omega)} J(u) := \frac{1}{p} \int_\Omega |\nabla u|^p \, dx + g \int_\Omega |\nabla u| \, dx - \int_\Omega f u \, dx. \tag{2.6}$$

where $1 < p < \infty$, $g > 0$ and $f \in L^q(\Omega)$.

The minimizer of this nonsmooth energy functional is given by the solution of a variational inequality of the second kind. In order to show this statement let us introduce the following notation: $\mathcal{F} = \frac{1}{p} \int_\Omega |\nabla u|^p \, dx - \int_\Omega f u \, dx$ and $G = g \int_\Omega |\nabla u| \, dx$. It is clear that both functions are continuous and convex. Further, from Lemma 2.1.11, and since every convex function is also quasiconvex, $\mathcal{F}$ and G are weakly lower semicontinuous.

Next, problem (2.6) can be written as

$$J(u) = \frac{1}{p} \|u\|^p_{W_0^{1,p}} + g \int_\Omega |\nabla u| \, dx - \int_\Omega f u \, dx.$$

Therefore, $J(\cdot)$ satisfies that

$$\lim_{\|u\|_{W_0^{1,p}} \to \infty} J(u) = +\infty. \tag{2.7}$$

Note that, following Theorem 2.1.10, and because J is strictly convex, the minimizer of (2.6) is characterized by the following variational inequality of the second kind: find $u \in W_0^{1,p}(\Omega)$ such that

$$\langle \mathcal{A}u, v - u \rangle + gj(v) - gj(u) \geq 0, \; \forall v \in W_0^{1,p}(\Omega)$$

where

$$\langle \mathcal{A}u, v \rangle = \int_\Omega |\nabla u|^{p-2}(\nabla u \cdot \nabla v) \, \mathrm{d}\Omega - \int_\Omega fv \, \mathrm{dx},$$

and

$$j(u) = \int_\Omega |\nabla u| \, dx.$$

Here, $\langle Au, v \rangle$ stands for the derivative of $\mathcal{F}(u)$. In particular, the minimizer of the functional $\mathcal{F}$ satisfies the Dirichlet problem for the p-Laplace operator. Following [11], we know that $\mathcal{F}(u)$ is a strictly convex funcional and its gradient is given by the following operator

$$\mathcal{A} : W_0^{1,p}(\Omega) \to W^{-1,q}(\Omega),$$

defined by

$$\langle \mathcal{A}u, v \rangle = \int_\Omega |\nabla u|^{p-2}(\nabla u \cdot \nabla v) - \int_\Omega fv \, dx \, dx, \;\; \forall v \in W_0^{1,p}.$$

Here, $W_0^{1,p}(\Omega)$ is the Sobolev space and $W^{-1,q}(\Omega)$ is its dual space with $q = \frac{p}{p-1}$. Further, $\mathcal{F}(u)$ is twice Gâteaux differentiable [2] , with

$$\begin{aligned} \mathcal{F}'(u)(v) &= \int_\Omega |\nabla u|^{p-2}(\nabla u \cdot \nabla v) \, dx - \int_\Omega fv \, dx \\ &= \langle Au, v \rangle, \;\; \forall v \in W_0^{1,p}, \end{aligned} \tag{2.8}$$

and

$$\begin{aligned} \mathcal{F}''(u)(v, w) &= \int_\Omega |\nabla u|^{p-2}(\nabla v, \nabla w) \, dx \\ &\quad + (p-2) \int_\Omega |\nabla u|^{p-4}(\nabla u, \nabla v)(\nabla u, \nabla w) \, dx, \;\; \forall v, w \in W_0^{1,p}. \end{aligned} \tag{2.9}$$

Once we have analyzed the variational inequality, we will focus on the resolution of (2.6) because our goal is to solve the minimization problem. In what follows we

summarize known results about the existence of a unique solution for this problem.

THEOREM 2.2.1. *Let $1 < p < \infty$. Then, problem* (2.6) *has a unique solution $\bar{u} \in W_0^{1,p}(\Omega)$.*

Proof. The space $W_0^{1,p}(\Omega)$ is a reflexive Banach space if $1 < p < \infty$. It is easy to verify that $J(\cdot)$ is a continuous strictly convex functional. Then, Theorem 2.1.10 implies that J has at least one minimal point. Further, from (2.7) and applying Theorem 2.1.12 we know that the minimal point found is the unique solution $\bar{u} \in W_0^{1,p}(\Omega)$ for problem (2.6). □

2.2.1 Regularization and discretization

The minimization problem (2.6) 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 [12]).

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

$$\min_{u \in W_0^{1,p}(\Omega)} J_\gamma(u) := \frac{1}{p} \int_\Omega |\nabla u|^p \, dx + \int_\Omega \psi_\gamma(\nabla u) \, dx - \int_\Omega f u \, dx. \tag{2.10}$$

THEOREM 2.2.2. *Let $1 < p < \infty$ and $\gamma > 0$. Then, problem* (2.10) *has a unique solution $u_\gamma \in W_0^{1,p}(\Omega)$. Also, the sequence $\{u_\gamma\} \subset W_0^{1,p}(\Omega)$ converges strongly in $W_0^{1,p}(\Omega)$ to the solution $\bar{u}$ of problem* (2.6) *as $\gamma \to \infty$.*

Proof. See [12, Sec. 2]. □

2.2.2 Finite element formulation

Let us introduce the finite element formulation of problem (2.10). 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

$$V_h^0 = W_0^{1,p}(\Omega) \cap V_h \tag{2.11}$$

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

Considering the previous analysis, the finite element approximation of (2.10) is formulated as follows

$$\min_{u_h \in V_h^0} J_{\gamma,h}(u_h) := \frac{1}{p} \int_{\Omega_h} |\nabla u_h|^p \, dx + \int_{\Omega_h} \psi_\gamma(\nabla u_h) \, dx - \int_{\Omega_h} f u_h \, dx. \tag{2.12}$$

In the following proposition we analyze the derivative of the discrete functional $J_{\gamma,h}(u_h)$. This analysis will be crucial in the convergence discussion of the multigrid algorithm presented in Chapter 4.

Hereafter, since we are working in finite dimensional spaces, the derivative $J_h'(u_h)(v_h)$ can be rewritten, using the canonical basis of $\mathbb{R}^n$, as the following product $J_h'(u_h)(v_h) = \sum_{i=1}^{n} J_h'(u_h)(e_i) v_h^i = \sum_{i=1}^{n} \frac{\partial J(u_h)}{\partial e_i} v_h^i = \nabla J_h(u_h)^\top v_h$ (see [10, Sec. 1.1.2 b,c]). Also, we adopt the notation: $\overset{\circ}{\nabla} J_{\gamma,h}(u_h)$ for a slanting function of $\nabla J_h(u_h)$.

PROPOSITION 2.2.3. *Let $1 < p < \infty$. The functional $J_{\gamma,h}(u_h)$ is Gâteaux differentiable with*

$$\nabla J_{\gamma,h}(u_h)^\top v_h := \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla u_h \cdot \nabla v_h \, dx + g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)} \, dx - \int_{\Omega_h} f v_h \, dx \ \ \forall v_h \in V_h^0. \tag{2.13}$$

Furthermore $\nabla J_{\gamma,h}(u_h)$ is a slantly differentiable function with

$$\begin{aligned}\overset{\circ}{\nabla} J_{\gamma,h}(u_h)(v_h, w_h) \;&:= \int_{\Omega_h} |\nabla u_h|^{p-2}\nabla v_h \cdot \nabla w_h \, dx \\ &+(p-2)\int_{\Omega_h} |\nabla u_h|^{p-4}(\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h)\, dx \\ &+ \int_{A_\gamma} g\frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|}\, dx - \int_{A_\gamma} g\frac{(\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h)}{|\nabla u_h|^3}\, dx \\ &+ \int_{\Omega_{k-1}\setminus A_\gamma} \gamma(\nabla v_h \cdot \nabla w_h)\, dx, \;\; \forall v_h, w_h \in V_h^0.\end{aligned} \tag{2.14}$$

Proof. Let us decompose the funcional $J_{\gamma,h}(u_h)$ as follows.

$$J_{\gamma,h}(u_h) := \mathcal{F}_h(u_h) + \mathcal{G}_{\gamma,h}(\nabla u_h), \tag{2.15}$$

where

$$\mathcal{F}_h(u_h) := \frac{1}{p}\int_{\Omega_h} |\nabla u_h|^p \, dx - \int_{\Omega_h} f u_h \, dx,$$

and

$$\mathcal{G}_{\gamma,h}(\nabla u_h) := \int_{\Omega_h} \psi_\gamma(\nabla u_h)\, dx.$$

As V_h^0 is a closed subspace of $W_0^{1,p}(\Omega)$ (see [11, Sec. 3]), the properties of the functional J_γ are inherited by the discrete functional $J_{\gamma,h}$. Hence, in what follows we present the derivative of the functional $J_{\gamma,h}(u_h)$. From (2.8), we have that the derivative of the discrete functional $\mathcal{F}_h(u_h)$ is given by

$$\nabla \mathcal{F}_h(u_h)^\top v_h = \int_{\Omega_h} |\nabla u_h|^{p-2}\nabla u_h \cdot \nabla v_h \, dx - \int_{\Omega_h} f v_h \, dx, \;\; \forall v_h \in V_h^0. \tag{2.16}$$

Next, let us analyse the functional $\mathcal{G}_{\gamma,h}(\nabla u_h)$. It is known that $\mathcal{G}_{\gamma,h}$ is once Gâteaux differentiable (see [12, Sec. 2.2]), and moreover, we know that

$$\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^\top v_h = g\int_{A_{\gamma,h}} \frac{\nabla u_h \cdot \nabla v_h}{|\nabla u_h|}\, dx + g\int_{\Omega_h \setminus A_{\gamma,h}} \gamma(\nabla u_h \cdot \nabla v_h)\, dx, \;\; \forall v_h \in V_h^0,$$

where

$$A_{\gamma,h} = \{x \in \Omega_h \,:\, \gamma|\nabla u_h(x)| \geq g\}.$$

By using the max function, we can rewrite $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)v_h$ in the following way.

$$\nabla \mathcal{G}_{\gamma,h}^{\top}(\nabla u_h)v_h := g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)}\, dx, \;\; \forall v_h \in V_h^0. \tag{2.17}$$

Next, from (2.15), it follows that

$$\nabla J_{\gamma,h}(u_h)^{\top} v_h := \nabla \mathcal{F}_h(u_h)^{\top} v_h + \nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^{\top} v_h, \tag{2.18}$$

which, thanks to (2.17) and (2.16), implies that

$$\begin{aligned} \nabla J_{\gamma,h}(u_h)^{\top} v_h \;\; &:= \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla u_h \cdot \nabla v_h \, dx + g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)}\, dx \\ &\quad - \int_{\Omega_h} f v_h \, dx \;\; \forall v_h \in V_h^0. \end{aligned}$$

The last expression corresponds to the Gâteaux derivative of the discretized functional $J_{\gamma,h}(u_h)$. Next, from Proposition 2.1.5 we have that $\nabla J_{\gamma,h}(u_h)v_h := J'^{\circ}_{\gamma,h}(u_h)v_h$.

The second Gâteaux derivative of $J_{\gamma,h}(u_h)$ does not exist. In fact, the functional $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)$ is not Gâteaux differentiable since this functional involves the max function. However, the max function is slantly differentiable when defined in finite dimensional spaces (see Example 2.1.6). Thus, we can calculate the slant derivative of $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)$, denoted by $\overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)$, as follows.

$|\nabla u| \geq \frac{g}{\gamma}$: Here, we have that

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)(v_h, w_h) = \;& g \int_{A_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{\max(g, \gamma|\nabla u_h|)}\, dx \\ - \;& g \int_{A_{\gamma,h}} \frac{\chi_{A_{\gamma,h}}(x) \cdot \gamma(\nabla u_h \cdot \nabla w_h)}{(\max(g, \gamma|\nabla u_h|))^2 |\nabla u_h|} \gamma(\nabla u_h \cdot \nabla v_h)\, dx \\ = \;& g \int_{A_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{\gamma|\nabla u_h|}\, dx \\ - \;& g \int_{A_{\gamma,h}} \frac{\gamma^2(\nabla u_h \cdot \nabla w_h)(\nabla u_h \cdot \nabla v_h)}{(\gamma|\nabla u_h|))^2 |\nabla u_h|}\, dx \\ = \;& g \int_{A_{\gamma,h}} \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|}\, dx \\ - \;& g \int_{A_{\gamma,h}} \frac{(\nabla u_h \cdot \nabla w_h)(\nabla u_h \cdot \nabla v_h)}{|\nabla u_h|^3}\, dx, \end{aligned}$$

where $\chi_{A_{\gamma,h}}$ is the slant derivative of function $\max(g, \gamma|\nabla u_h|)$.

$|\nabla u| < \frac{g}{\gamma}$: Here, we have that

$$\begin{aligned}
\overset{\circ}{\nabla}\mathcal{G}_{\gamma,h}(\nabla u_h)(v_h, w_h) = \; & g\int_{\Omega_h\setminus A_{\gamma,h}} \frac{\gamma(\nabla v_h\cdot\nabla w_h)}{\max(g,\gamma|\nabla u_h|)}\,dx \\
& - g\int_{\Omega_h\setminus A_{\gamma,h}} \frac{\chi_{A_{\gamma,h}}(x)\cdot\gamma(\nabla u_h\cdot\nabla w_h)}{(\max(g,\gamma|\nabla u_h|))^2|\nabla u_h|}\gamma(\nabla u_h\cdot\nabla v_h)\,dx \\
= \; & g\int_{\Omega_h\setminus A_{\gamma,h}} \frac{\gamma(\nabla v_h\cdot\nabla w_h)}{g}\,dx \\
= \; & \int_{\Omega_h\setminus A_{\gamma,h}} \gamma(\nabla v_h\cdot\nabla w_h)\,dx.
\end{aligned}$$

Then, the slant derivative of $\nabla\mathcal{G}_{\gamma,h}(\nabla u_h)$ reads as follows

$$\begin{aligned}
\overset{\circ}{\nabla}\mathcal{G}_{\gamma,h}(u_h)(v_h, w_h) = \; & \int_{A_{\gamma,h}} g\frac{(\nabla v_h\cdot\nabla w_h)}{|\nabla u_h|} - \int_{A_{\gamma,h}} g\frac{(\nabla u_h\cdot\nabla v_h)(\nabla u_h\cdot\nabla w_h)}{|\nabla u_h|^3} \\
& + \int_{\Omega_h\setminus A_{\gamma,h}} \gamma(\nabla v_h\cdot\nabla w_h),\;\; \forall v_h, w_h\in V_h^0.
\end{aligned} \tag{2.19}$$

On the other hand, from (2.9) we have that

$$\begin{aligned}
\nabla^2\mathcal{F}_h(u_h)(v_h, w_h) \; = & \int_{\Omega_h} |\nabla u_h|^{p-2}\nabla v_h\cdot\nabla w_h \\
& + (p-2)\int_{\Omega_h} |\nabla u_2|^{p-4}(\nabla u_h\cdot\nabla v_h)(\nabla u_h\cdot\nabla w_h),\;\; \forall v_h, w_h\in V_h^0.
\end{aligned} \tag{2.20}$$

Hence, from (2.18), we obtain the slant derivative of $\nabla J_{\gamma,h}(u_h)^\top v_h$ as follows

$$\begin{aligned}
\overset{\circ}{\nabla}J(u_h)(v_h, w_h) \; & = \nabla^2\mathcal{F}_h(u_h)(v_h\cdot w_h) \\
& \quad + \overset{\circ}{\nabla}\mathcal{G}_{\gamma,h}(\nabla u_h)(v_h\cdot w_h)\;\; \forall v_h, w_h\in V_h^0,
\end{aligned} \tag{2.21}$$

which, thanks to (2.19), (2.20) and (2.21), yields that

$$\begin{aligned}\overset{\circ}{\nabla} J(u_h)(v_h, w_h) = & \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla v_h \nabla w_h \, dx \\ & + (p-2) \int_{\Omega_h} |\nabla u_h|^{p-4} (\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h) \, dx \\ & + \int_{A_\gamma} g \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|} \, dx - \int_{A_\gamma} g \frac{\nabla u_h \cdot \nabla v_h (\nabla u_h \cdot \nabla w_h)}{|\nabla u_h|^3} \, dx \\ & + \int_{\Omega_{k-1} \setminus A_\gamma} \gamma (\nabla v_h \cdot \nabla w_h) \, dx, \quad \forall v_h, w_h \in V_h^0.\end{aligned} \tag{2.22}$$

□

Chapter 3

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.

$$\begin{aligned} -\Delta_h u_h(x,y) &= f_h(x,y) \text{ in } \Omega_h \\ u_h(x,y) &= g_h(x,y) \text{ on } \partial\Omega_h \end{aligned} \tag{3.1}$$

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 [29, Sec 1.5]. The second main property of the multigrid ap-

proach 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:

$$v_h(x, y) = \sum_{k,m=1}^{n-1} \alpha_{k,m} \sin k\pi x \sin m\pi y. \tag{3.2}$$

Here, the functions

$$\varphi_h^{k,m}(x, y) = \sin k\pi x \sin m\pi y \quad (k, m = 1, \cdots, n-1) \tag{3.3}$$

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:

$$\alpha_{k,m} \sin k\pi x \sin m\pi y \quad \text{with } k \text{ or } m \text{ large} \tag{3.4}$$

and low frequency components to

$$\alpha_{k,m} \sin k\pi x \sin m\pi y \quad \text{with } k \text{ or } m \text{ small} \tag{3.5}$$

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 [29, Sec. 1.5.2]:

$$\varphi^{k,m}(x, y) = -\varphi^{n-k,m}(x, y) = -\varphi^{k,n-m}(x, y) = \varphi^{n-k,n-m}(x, y) \quad \text{for } (x, y) \in \Omega_{2h}.$$

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.

3.1 Multigrid for differential problems

3.1.1 Two-grid scheme

In what follows we present the multigrid algorithm for solving the Poisson problem (3.1), 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

$$A_h u_h = f_h. \tag{3.6}$$

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

$$r_h = f_h - A_h u_h^{\nu_1}. \tag{3.7}$$

Then, we have the residual equation

$$A_h v_h = r_h. \tag{3.8}$$

Since $u_h = u_h^{\nu_1} + v_h$, the residual equation (3.8) is equivalent to (3.6), 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

$$A_H v_H = r_H, \tag{3.9}$$

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 (3.9) 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

$$u_h^{new} = u_h^{\nu_1} + I_H^h v_h. \tag{3.10}$$

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 (3.9) 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:

Algorithm 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.

3.1.2 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 (3.9) exactly. Since the residual equation in the coarse space has

the same form as system (3.6), 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.

Algorithm 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:

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

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 3.1).

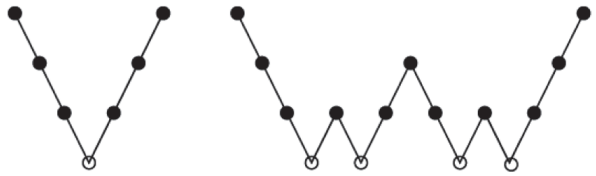

Figure 3.1: MG scheme with four grids. Left: $\gamma = 1$, right: $\gamma = 2$ [29, Sec. 2.4].

3.1.3 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 3.2.

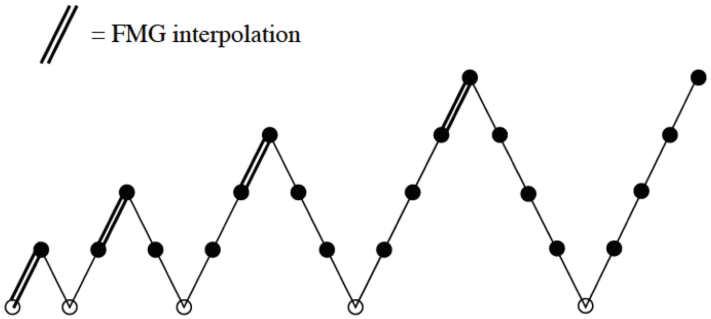

Figure 3.2: FMG scheme with $r = 1$ and $\gamma = 1$ [29, Sec. 2.6].

As we can see from Figure 3.2, 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.

Algorithm 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.
- $u_k = u_k^{FMG}$.

3.1.4 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 [3]. The structure of this algorithm constitutes the basis for the multigrid optimization method and other advanced multigrid techniques [29, Sec. 5.3.7]. In what follows we briefly present the FAS idea for solving the nonlinear differential equation:

$$N_k(u_k) = f_k, \tag{3.11}$$

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 [29, Sec 5.3.2]), for solving problem (3.11). 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.

$$N_k(\tilde{u}_k + v_k) - N_k(\tilde{u}_k) = r_k. \tag{3.12}$$

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} [29, Sec. 5.3.4]. In the same way, we formulate equation (3.12) 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:

$$N_{k-1}(\hat{u}_{k-1}) = I_k^{k-1}(f_k - N_k(\tilde{u}_k)) + N_{k-1}(I_k^{k-1}\tilde{u}_k). \tag{3.13}$$

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

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

Algorithm 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 [3]. Considering the scheme of the FAS method, in the next chapter we introduce the multigrid optimization, MG/OPT algorithm.

Chapter 4

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 (2.12). The MG/OPT method corresponds to a nonlinear programming adaptation of the *full approximation scheme*, FAS (see [3, 29]). The multigrid subproblems arising from the different discretization levels are nonlinear optimization problems [23]. The MG/OPT method was introduced as an efficient tool for large scale optmization problems (see [25, 23]). 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.

4.1 Transfer operators

In order to present the algorithm, we shall introduce the following preliminaries. Let $\{\Omega_k\}_{k=0,\ldots,m}$ be a sequence of partitions of Ω_h. As usual, Ω_k is obtained from Ω_{k-1} by a regular subdivision: the procedure joins the edge midpoints of any triangle in mesh Ω_{k-1} by edges, and forms the new triangles of Ω_k (see Figure 4.1). This procedure guarantees that the discretization parameters h_k and h_{k-1} satisfy that $h_k = \frac{1}{2}h_{k-1}$. Then we have that $\Omega_{k-1} \subset \Omega_k$. Furthermore, let V_k^0 be the FEM space associated to Ω_k, we have that $V_{k-1}^0 \subset V_k^0$, for $k = 1, \ldots, m$ (see, for instance, [2, 13, 22]).

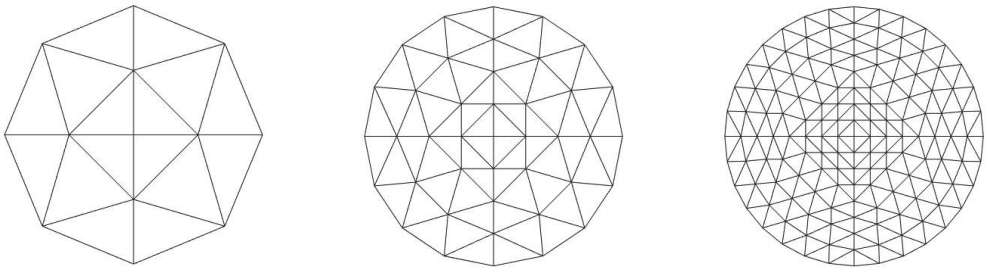

Figure 4.1: Regular subdivision of meshes (from left to right): Ω_0, Ω_1 and Ω_2.

In this work we use the mesh data structure and the transfer operators presented in [13, Ch. 6, Ch. 13] to implement the operators I_k^{k-1} and I_{k-1}^k. This data structure describes the linear Lagrange triangles of a mesh and allows us to recover the index of a given node and identify its node parents in any mesh created by regular subdivision. Taking this into account, we are ready to introduce the transfer operators. For the *coarse-to-fine grid transfer operador*, I_{k-1}^k, let denote $\vec{v}_{k-1}$ as the vector containing the nodal values of a piecewise linear function on Ω_{k-1}. The idea of this operator is to obtain the vector $\vec{v}_k$ that contains the nodal values of the piecewise linear function on Ω_k. Let us illustrate the procedure in a single triangle. Suppose that we have the nodal values of its three nodes. If we refine it we obtain four triangles and six nodes as is shown in Figure 4.2. In order to obtain the nodal values of the midpoints with indexes 4, 5 and 6, we compute the average of the extreme nodes values. For instance, the value of the node with index 4 is the average of the nodal values with indexes 1 and 2. This procedure is repeated for all the midpoints on the finer mesh Ω_k

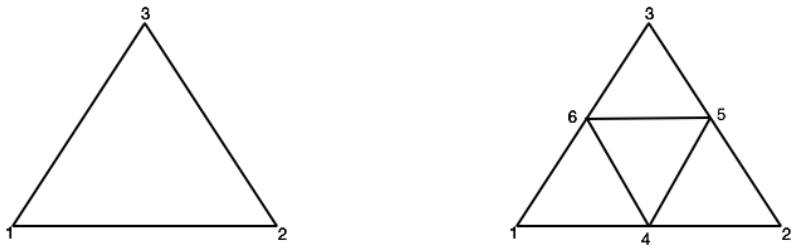

Figure 4.2: Refinement of a single triangle

Due to the fact that each node in Ω_k either belongs to Ω_{k-1} or is the midpoint of an edge in Ω_{k-1}, the transfer algorithm computes the vector of nodal values for inner and boundary nodes denoted by $\check{v}_k$ on Ω_k. At the end, the algorithm extracts the components corresponding to the boundary nodes to obtain $\vec{v}_k$.

Next, we present the algorithm to compute $\vec{v}_k$. Let N_k be the vector that contains the indices of nodes in Ω_k that are not in Ω_{k-1}.

Algorithm 5 Coarse-to-fine grid transfer operador I_{k-1}^{k}

1. Copy the components of $\vec{v}_{k-1}$ that corresponds to the components of $\check{v}_k$.
2. For $i \in N_k$ do
$$(\check{v}_k)_i = \frac{1}{2}\left((\check{v}_k)_{i_{end_1}} + (\check{v}_k)_{i_{end_2}}\right).$$
3. Extract the components of $\check{v}_k$ corresponding to the boundary nodes and finally obtain $\vec{v}_k$, (the inner nodes of $\check{v}_k$).

Here i_{end_1} and i_{end_2} denote the indices of the endpoints of the edge in Ω_{k-1} of which $(\check{v}_k)_i$ is the midpoint.

For the *fine-to-coarse grid transfer operador*, I_k^{k-1}, let us recall that the vector $\vec{v}_k$ contains the nodal values of a piecewise linear function on Ω_k. Then, the vector with the nodal values of a piecewise linear function associated to the indices in N_k will be denoted as $\check{v}_k$. In this case, we perform the inverse procedure than the stated before. We present the following algorithm in orer to illustrate it.

Algorithm 6 Fine-to-coarse grid transfer operador I_k^{k-1}

1. Copy the components of $\vec{v}_k$ that corresponds to the components of $\check{v}_k$.
2. Initialize to zero the other components of $\check{v}_k$.
3. For $i \in N_k$ do
$$(\check{v}_k)_{i_{end_1}} = \frac{1}{2}(\check{v}_k)_i,$$
$$(\check{v}_k)_{i_{end_2}} = \frac{1}{2}(\check{v}_k)_i,$$
4. Extract the components of $\check{v}_k$ corresponding to the nodes in Ω_{k-1}, except the boundary nodes, to obtain $\vec{v}_{k-1}$.

The mesh data structure implemented allows us to copy and extract nodes from any specific discretization. We refer the reader to see [13, Ch. 13, sec. 13.2.1] for a detailed explanation of the transfer operators.

In multigrid schemes is standard to assume that

$$I_k^{k-1} = c\left(I_{k-1}^k\right)^T,$$

where c is a constant. In our case, the restriction and prolongation operators satisfy the condition (see [13, Sec. 13.2.1, p. 294])

$$I_k^{k-1} = \left(I_{k-1}^k\right)^T. \tag{4.1}$$

4.2 MG/OPT Algorithm

Once we have introduced the interpolation operators, we are ready to discuss the MG/OPT method for problem (2.12). 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 ([4, 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

$$\nabla J_{\gamma,1}(u_1^{\nu_1} + e_1) - \nabla J_{\gamma,1}(u_1^{\nu_1}) = -\nabla J_{\gamma,1}(u_1^{\nu_1}). \tag{4.2}$$

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 (4.2) on Ω_0. The main idea is to obtain the structure of

the FAS equation (3.13) in order to take advantage of the FAS algorithm's scheme. Hence, we formulate equation (4.2) 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

$$\nabla J_{\gamma,0}(u_0) - \nabla J_{\gamma,0}(I_1^0 u_1^{\nu_1}) = -I_1^0 \nabla J_{\gamma,1}(u_1^{\nu_1}). \tag{4.3}$$

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:

$$\nabla J_{\gamma,0}(u_0) = \tau_0. \tag{4.4}$$

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

$$u_1 = u_1^{\nu_1} + \alpha I_0^1(u_0 - I_1^0 u_1^{\nu_1}). \tag{4.5}$$

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 (2.12), we use a class of descent algorithms as the underlying algorithm. This choice was made based on the structure of problem (2.12). 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 [16]. 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.

Algorithm 7 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.

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

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.

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

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

4.3 Convergence Analysis

In this section, we discuss the convergence properties of Algorithm 7. Following [23, 25], we can state that the global convergence of the underlying optimization algorithm ensures global convergence of the MG/OPT method. This comes from the fact that if we have an approximated solution (given by the underlying optimization algorithm) at each discretization level, the algorithm generates search directions for

problems discretized in finer meshes. Once we have the descent direction, a line search procedure is used to improve the solution at each finer problem.

In the classic convergence analysis of the MG/OPT methods [23, 25, 4], the three following conditions are critical.

1. The optimization problem keeps the convexity property at each level of discretization.

2. The subproblems
$$\min_{u_{k-1}} \left(\hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^{\top} u_{k-1} \right) \tag{4.6}$$
are solved accurately enough.

3. The transfer operators satisfy the standard condition $I_k^{k-1} = c \left(I_{k-1}^k \right)^T$.

These last conditions are helpful to prove that the search direction provided by the MG/OPT algorithm is indeed a descent direction, *i.e.*, e satisfies that $\nabla J_{\gamma,k}(u_k)^{\top} e < 0, \quad \forall k = 0, \cdots m$. Further, the convexity condition is key to prove that the Hessian is positive definite. However, in our case, the usual Hessian does not exists. We will use the slantly differentiability of the functional $J_{\gamma,h}$ to perform our convergence analysis.

Next, we state some comments about the previous conditions in our problem. The coarse grids subproblems correspond to the discrete optimization problem over the nested spaces $V_{k-1}^0 \subset V_k^0$. Hence, the inclusion preserves the convexity of the subproblems.

Since we perform a few iterations of a suitable globally convergent optimization algorithm (S_{opt}), we ensure that the subproblems $\min_{u_{k-1}} \left(\hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^{\top} u_{k-1} \right)$ are solved accurately enough. Finally, (4.24) yields that $I_k^{k-1} = \left(I_{k-1}^k \right)^T$.

Let us recall that the descent direction for the MG/OPT algorithm is denoted by e and search directions of the underlying optimization algorithm, (inside the MG/OPT loop) are denoted by w_k.

Once we have discussed the convergence conditions, we introduce the following theorem of convergence for the MG/OPT Algorithm 7.

THEOREM 4.3.1. *Suppose that the following hypothesis are satisfied:*

- *The optimization algorithm, S_{opt}, applied to an optimization problem of any resolution, is globally convergent, i.e.,*
$$\lim_{k \to \infty} \|\nabla J_{\gamma,h}(u_k)\| = 0. \tag{4.7}$$

- *At least one of the parameters ν_1 or ν_2 is positive.*
- *The search direction $e = I_{k-1}^{k}(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1})$ is a descent direction.*

Then, MG/OPT algorithm is globally convergent in the same sense than (4.7).

Proof. The global convergence of the underlying optimization algorithm ensures the global convergence of MG/OPT since the approximated solution given at each level improves at every cycle of the multigrid optimization algorithm. Hence, we need to guarantee that the direction e, provided by the MG/OPT algorithm, is a descent direction. *i.e.*, we have to prove that

$$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e < 0, \quad \forall\, k = 0, \cdots m. \tag{4.8}$$

From this point, for the sake of readiness of the proof, we drop the subscript γ. First note that, if we solve

$$\min_{u_{k-1}} \left(J_{k-1}(u_{k-1}) - \tau_{k-1}^\top u_{k-1} \right)$$

exactly, then

$$\nabla J_{k-1}(\widetilde{u}_{k-1}) - \tau_{k-1} = 0.$$

Since we are solving the problem approximately (or accurately enough), then we have that

$$\nabla J_{k-1}(\widetilde{u}_{k-1}) - \tau_{k-1} = z, \tag{4.9}$$

for some z as small as the algorithm accuracy allows us. From Algorithm 3 we have that

$$\tau_{k-1} := \nabla J_{k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla J_k(u_k^{\nu_1}).$$

Hence, we can rewrite (4.9) as follows

$$\nabla J_{k-1}(\widetilde{u}_{k-1}) = \nabla J_{k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla J_k(u_k^{\nu_1}) + z. \tag{4.10}$$

Thus, from (4.10) we have that

$$\begin{aligned}
\nabla J_k(u_k^{\nu_1})^\top e &= \nabla J_k(u_k^{\nu_1})^\top I_{k-1}^{k}(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\
&= \nabla J_k(u_k^{\nu_1})^\top (I_k^{k-1})^\top (\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\
&= (I_k^{k-1} \nabla J_k(u_k^{\nu_1}))^\top (\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\
&= (\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\widetilde{u}_{k-1}) + z)^\top (w_{k-1}) \\
&= (\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\widetilde{u}_{k-1}))^\top (w_{k-1}) + z^\top w_{k-1},
\end{aligned} \tag{4.11}$$

where

$$w_{k-1} = \widetilde{u}_{k-1} - u_{k-1}^{\nu_1}.$$

Next, let us focus on the two first terms in the right-hand side of (4.11).

$$(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\widetilde{u}_{k-1}))^\top w_{k-1} = \nabla J_{k-1}(u_{k-1}^{\nu_1})^\top w_{k-1} - \nabla J_{k-1}(\widetilde{u}_{k-1})^\top w_{k-1}. \tag{4.12}$$

We know, from Proposition 2.2.3, that ∇J_{k-1} is slantly differentiable. Thus, Theorem 2.1.9 and Proposition 2.1.5 allow us to state that

$$\begin{aligned} -(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\widetilde{u}_{k-1}))^\top w_{k-1} &= (\nabla J_{k-1}(\widetilde{u}_{k-1}) - \nabla J_{k-1}(u_{k-1}^{\nu_1}))^\top w_{k-1} \\ &= \overset{\circ}{\nabla} J_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}), \end{aligned} \tag{4.13}$$

where $\overset{\circ}{\nabla} J_{k-1}(\widetilde{u}_{k-1})$ is given by (2.14). Furthermore, following the decompostition presented in (2.21), we know that

$$\overset{\circ}{\nabla} J_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}) = \nabla^2 \mathcal{F}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}) + \overset{\circ}{\nabla} \mathcal{G}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}), \tag{4.14}$$

It is well known that $\mathcal{F}$ is a strictly convex functional, which implies that ([5, 2, 11, 12])

$$\nabla^2 \mathcal{F}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}) \geq 0, \quad \forall w_{k-1} \in V_{k-1}^0 \setminus \{0\}. \tag{4.15}$$

Next, let us recall the expression $\overset{\circ}{\nabla} \mathcal{G}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1})$ given by

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1}) &= \int_{A_{\gamma,h}} g \frac{(\nabla w_{k-1} \cdot \nabla w_{k-1})}{|\nabla \widetilde{u}_{k-1}|} - \int_{A_{\gamma,h}} g \frac{(\nabla \widetilde{u}_{k-1} \cdot \nabla w_{k-1})^2}{|\nabla \widetilde{u}_{k-1}|^3} \\ &\quad + \int_{\Omega_{k-1} \setminus A_{\gamma,h}} \gamma (\nabla w_{k-1} \cdot \nabla w_{k-1}), \ \forall w_{k-1} \in V_{k-1}^0. \end{aligned} \tag{4.16}$$

Applying Cauchy-Schwarz to the right hand side in (4.16), we have that

$$\begin{aligned} \int_{A_\gamma} g \frac{(\nabla \widetilde{u}_{k-1} \cdot \nabla w_{k-1})^2}{|\nabla \widetilde{u}_{k-1}|^3} &\leq \int_{A_\gamma} g \frac{|\nabla \widetilde{u}_{k-1}|^2 |\nabla w_{k-1}|^2}{|\nabla \widetilde{u}_{k-1}|^3} \\ &= \int_{A_\gamma} g \frac{|\nabla w_{k-1}|^2}{|\nabla \widetilde{u}_{k-1}|} \\ &= \int_{A_\gamma} g \frac{(\nabla w_{k-1} \cdot \nabla w_{k-1})}{|\nabla \widetilde{u}_{k-1}|}, \end{aligned}$$

which implies that

$$(\overset{\circ}{\nabla} \mathcal{G}_{k-1}(\widetilde{u}_{k-1})(w_{k-1}, w_{k-1})) \geq \int_{\Omega_{k-1} \setminus A_\gamma} \gamma (\nabla w_{k-1} \cdot \nabla w_{k-1}) > 0, \text{ since } w_{k-1} \neq 0. \tag{4.17}$$

Summarizing (4.13), (4.14), (4.15) and (4.17) yield that

$$(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\widetilde{u}_{k-1}))w_{k-1} < 0. \tag{4.18}$$

To check that e is a descent direction, we still need to prove that the third term of the right hand side in (4.11) satifies that

$$z^\top w_{k-1} = z^\top(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}) < 0.$$

Note that $\widetilde{u}_{k-1}$ is the solution of the problem

$$\min_{u_{k-1}} \left(J_{k-1}(u_{k-1}) - \tau_{k-1}^\top u_{k-1} \right).$$

Therefore,

$$J_{k-1}(\widetilde{u}_{k-1}) - \tau_{k-1}^\top \widetilde{u}_{k-1} < J_{k-1}(u_{k-1}^{\nu_1}) - \tau_{k-1}^\top u_{k-1}^{\nu_1},$$

which is equivalent to

$$J_{k-1}(\widetilde{u}_{k-1}) - J_{k-1}(u_{k-1}^{\nu_1}) < \tau_{k-1}^\top(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}), \tag{4.19}$$

since the optimization algorithm was inicialized with $u_{k-1}^{\nu_1}$. Further, the functional J_{k-1} is also slantly differentiable since it is Gâteaux differentiable and, moreover $J_k^\circ = \nabla J_k$ (see Proposition 2.1.5). Thus, we can take its Gâteaux derivative as a slanting function of J_{k-1}. Then again, from Theorem 2.1.9, we have that

$$(J_{k-1}(\widetilde{u}_{k-1}) - J_{k-1}(u_{k-1}^{\nu_1})) = \nabla J_{k-1}(\widetilde{u}_{k-1})^\top(w_{k-1}). \tag{4.20}$$

Hence, from the inequality (4.19) and equation (4.20) we have that

$$\begin{aligned} \nabla J_{k-1}(\widetilde{u}_{k-1})(w_{k-1}) &< \tau_{k-1}^\top(\widetilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\ &= \tau_{k-1}^\top w_{k-1} \end{aligned}$$

which implies that

$$\nabla J_{k-1}(\widetilde{u}_{k-1})w_{k-1} - \tau_{k-1}^\top w_{k-1} < 0. \tag{4.21}$$

Next, from (4.9) and (4.21) we obtain that

$$z^\top w_{k-1} < 0. \tag{4.22}$$

Then, from (4.11), (4.18) and (4.22) we have that

$$\nabla J_k(u_k^{\nu_1})^\top e < 0,$$

and we can conclude that e is a descent direction.

Finally, if at least one of the parameters ν_1 or ν_2 is positive, at least one iteration of the optimization algorithm is performed at every cycle of the MG/OPT. Consequently, as the underlying optimization algorithm is globally convergent, the multigrid optimization algorithm is globally convergent. □

4.4 Implementation

4.4.1 Optimization algorithm

In this section, we briefly discuss the underlying optimization algorithms. We implement the MG/OPT algorithm using two versions of the steepest descent algorithm: the gradient method [26] and the preconditioned descent algorithm proposed in [12].

Generally speaking, a descent method starts with an inicial point u_0 and, with information of first order, the algorithm finds directions that lead us to the minimum of the objective functional. Also, the method must find the length of the step along the choosen direction. The basic idea consists in finding α_r and w_r such that:

$$J(u_r + \alpha_r w_r) < J(u_r), \text{ for } \alpha_r > 0$$

in each iteration of the method. The descent direction corresponds to w_k, and α_k is the length step. In other words, descent methods works by finding w_k such that:

$$\nabla J(u_r)^\top w_r < 0.$$

and the approximation of the solution, in each step, is given by:

$$u_{r+1} = u_r + \alpha_r w_r. \tag{4.23}$$

In the gradient method, the search direction w_r is determined by

$$w_r = -\nabla J(u_r).$$

On the other hand, the search direction w_r of the preconditioned descent algorithm for problem (2.10) is calculated taking into account the difficulties associated to the structure of the p-Laplace operator . In order to deal with this issues, in [16] a preconditioner is implemented for the computation of the search direction for the p-Laplacian problem in finite dimension. In [12] a preconditioned descent algorithm

is presented in function spaces but, for problem (2.10) .The idea of the preconditioner is to determine the search direction by solving the following equation

$$P_r(w_r, v_r) = -\nabla J(u_r)^\top v_r, \ \forall v_r \in V_h^0,$$

where the form $P_r : V_h^0 \times V_h^0 \to \mathbb{R}$ is chosen as a variational approximation of the p-Laplace operator. In what follows we present a general descent algorithm.

Algorithm 8 General descent algorithm

1: Initialize $u_0 \in V_h^0$ and set $r = 0$.
2: **for** $r = 1, 2, \cdots$ **do**

1. **if** $\nabla J(u_r) = 0$ **then**, STOP.
2. **end if**
3. Find a descent direction w_r.
4. Perform an efficient line search technique to obtain α_r.
5. Update $u_{r+1} := u_r + \alpha_r w_r$ and set $r = r + 1$.
6. **end for**

Global convergence discussion

The global convergence of a descent algorithm depends on the admissibility of the search direction w_r [15, Sec. 2.2.1], in the case of the preconditioned descent algorithm the search direction depends on how P_r is defined.

DEFINITION 4.4.1. *Admissibility of the search direction:*

$$\frac{\nabla J(u_r)^\top w_r}{\|w_r\|} \overset{k\to\infty}{\longrightarrow} 0 \Longrightarrow \ \|\nabla J(u_r)\| \overset{k\to\infty}{\longrightarrow} 0 \tag{4.24}$$

Thus, the preconditioned descent algorithm is considered and analyzed in two separated cases, when $1 < p < 2$ and $p \geq 2$ because, as it is known, the behaviour of J depends on the value of p.

Case $1 < p < 2$

The $1 < p < 2$ *case* is analyzed in function spaces, then we use the notation $J'(u)$ for the derivative of functional $J(u)$. In what follows we summarize the main results presented in [12] for proving global convergence of the preconditioned de-

scent algorithm. First, the author introduce the following Hilbert space.

THEOREM 4.4.2. *Let $1 < p < 2$, $\epsilon > 0$ and $u \in W_0^{1,p}(\Omega)$. Then, $H_0^u(\Omega)$ is a Hilbert space with the inner product*

$$(z, w)_{H_0^u} = \int_\Omega (\epsilon + |\nabla u|)^{p-2}(\nabla z, \nabla w)dx.$$

Furthermore, the following inclusion holds, with continuous injections

$$H_0^1(\Omega) \subset H_0^u(\Omega) \subset W_0^{1,p}(\Omega).$$

Proof. [12, Sec. 3.1] □

The preconditioned algorithm considers the function space $H_0^{\hat{u}}(\Omega)$, for some suitable $\hat{u} \in W_0^{1,p}(\Omega)$ and defines the form P_r as follows.

$$P_r(w, v) := \int_\Omega (\epsilon + |\nabla \hat{u}|)^{p-2}(\nabla w, \nabla v)).$$

As we mentioned before, the finite element approximation of the p- Laplacian derives in a degenerate algebraic system then, the parameter ϵ helps to deal with this issue when $\nabla \hat{u} = 0$.

Next, since $H_0^{\hat{u}}(\Omega) \subset W_0^{1,p}(\Omega)$, the author define the restriction of $J'(\hat{u})$ to $H_0^{\hat{u}}(\Omega)$ by $\hat{J}'(\hat{u})$ and states that

$$\left\langle \hat{J}'(u), v \right\rangle_{H_0^{\hat{u}*}, H_0^{\hat{u}}} = \left\langle J'(u), v \right\rangle_{W^{-1,p'}, W_0^{1,p}} \quad \forall v \in H_0^{\hat{u}}(\Omega),$$

where $\hat{J}'(\hat{u}) \in H_0^{\hat{u}}(\Omega)^*$. Then, the following variational equation is presented

$$\int_\Omega (\epsilon + |\nabla \hat{u}|)^{p-2}(\nabla w, \nabla v))dx = -\left\langle \hat{J}'(u), v \right\rangle_{H_0^{\hat{u}*}, H_0^{\hat{u}}}, \quad \forall v \in H_0^{\hat{u}}(\Omega) \tag{4.25}$$

and the author proves the existence of a unique w in $H_0^{\hat{u}}(\Omega) \subset W_0^{1,p}(\Omega)$ that verifies (4.25). With this result, the algorithm proposed finds a direction $w_r \in H_0^{u_r}(\Omega) \subset W_0^{1,p}(\Omega)$ by solving the following variational equation

$$\begin{aligned} \textstyle\int_\Omega (\epsilon + |\nabla u_r|)^{p-2}(\nabla w_r, \nabla v)dx &= -\left\langle \hat{J}'_\gamma(u_r), v \right\rangle_{H_0^{u_r*}, H_0^{u_r}} \\ &= -\textstyle\int_\Omega |\nabla u_r|^{p-2}(\nabla u_r, \nabla v)dx - g\gamma \int_\Omega \frac{(\nabla u_r, \nabla v)}{\max(g, \gamma|\nabla u_r|)}dx \\ &\quad + \textstyle\int_\Omega f v dx, \quad \forall v \in H_0^{u_r}(\Omega). \end{aligned}$$

The direction found satisfy the following condition.

$$\left\langle \hat{J}'_\gamma(u_k), v \right\rangle_{H_0^{u_{k^*}}, H_0^{u_k}} = \left\langle J'_\gamma(u_k), v \right\rangle_{W^{-1,p'}, W_0^{1,p}} < 0.$$

Then, the search direction is a descent direction. The author, also shows the admissibility of w_r and thanks to this result, the global convergence of the algorithm is also proved. In order to prove the admissibility of the search direction, the author assumes that, using interpolation theory, there exists q, $1 < p < q < 2$, such that $\cup_{j\in\mathbb{N}} H_0^{u_j}(\Omega) \subset W_0^{1,q}(\Omega) \subset W_0^{1,p}(\Omega)$. Hence, a sequence $\{u_l\}$ generated by Algorithm 8 with w_r given by solving (4.25), yields that $\{u_l\} \in \cup_{j\in\mathbb{N}} H_0^{u_j}(\Omega) \subset W_0^{1,q}(\Omega) \subset W_0^{1,p}(\Omega)$. Furthermore, the admissibility of w_r is proved by the following result.

PROPOSITION 4.4.3. *Let $\{u_l\}$ be the sequence generated by Algorithm 8 and suppose that the step length α_k satisfies the Wolfe-Powell conditions:*

$$J(u_r + \alpha w_r) \leq J(u_r) + \sigma_1 \alpha_r \left\langle J'(u_r), w_r \right\rangle_{W^{-1,p'}, W_0^{1,p}}, \tag{4.26}$$

$$\left\langle J'(u_r + \alpha w_r), w_r \right\rangle_{W^{-1,p'}, W_0^{1,p}} \geq \sigma_2 \left\langle J'(u_r), w_r \right\rangle_{W^{-1,p'}, W_0^{1,p}}. \tag{4.27}$$

Furthermore, let us suppose that exists q, $1 < p < q < 2$, such that $\cup_{j\in\mathbb{N}} H_0^{u_j}(\Omega) \subset W_0^{1,q}(\Omega) \subset W_0^{1,p}(\Omega)$. Then, the Zoutendijk codition is verified, i.e.,

$$\sum_{k=0}^{\infty} cos^2 \phi_r = \infty, \tag{4.28}$$

where $cos\phi_r = -\frac{\langle J'(u_r), w_r \rangle_{W^{-1,p'}, W_0^{1,p}}}{\|J'(u_r)\|_{W^{-1,p'}} \|w_r\|_{W_0^{1,p}}}$.

Proof. [12, Sec.3] □

The last result shows the admissibility of w_r. Finally, the global convergence of the algorithm is given by the following theorem.

THEOREM 4.4.4. *Let $\{u_r\}$ be the sequence generated by Algorithm 8 and suppose that the step lenght α_r satisfies the Wolfe-Powell coditions* (4.26) *and* (4.4.3)*. Then, the sequence $\{u_r\}$ converges to the uniquely determined global minimum of J.*

Proof. [12, Sec. 3, Th. 3.8] □

For a deeper analysis of the algorithm and the proofs of the previous results, see [12] and the references cited there in.

Case $p \geq 2$

For the $p \geq 2$ *case*, the author shows that it is not natural to construct a descent direction in function spaces due to the structure of the problem and because there exist regularity issues regarding the search direction. Then, it is considered that the best solution is to analyze the problem with a "discretize then optimize" approach. Thus, it is proposed a preconditioned algorithm in finite dimension spaces, i.e., it is proposed to find a search direction $w_{r,h}$, for the discrete problem (2.12), by solving

$$\begin{aligned} \int_\Omega (\nabla w_{r,h}, \nabla v) dx &= -\nabla \hat{J}_h(u_{r,h})^\top v \\ &= -\int_\Omega |\nabla u_{r,h}|^{p-2} (\nabla u_{r,h}, \nabla v_r) dx - g\gamma \int_\Omega \frac{(\nabla u_{r,h}, \nabla v_r)}{\max(g, \gamma |\nabla u_{r,h}|)} dx \\ &+ \int_\Omega f v dx, \quad \forall v \in V^0(\Omega). \end{aligned} \tag{4.29}$$

In this case, it is known [6] that

$$V_h^0 \subset W_0^{1,p}(\Omega) \subset H_0^1(\Omega).$$

Thanks to this result, one can consider V_0^h a Hilbert space with the norm induced by $H_0^1(\Omega)$. This constitutes a key result for the existence of a unique solution $w_{r,h} \in V_h^0$ for equation (4.29). In [12] it is proved that this unique solution $w_{r,h}$ is an admissible descent direction for $\nabla J_h(u_{r,h})$. Hence,

$$P_r(w_{r,h}, v_r) = \int_\Omega (\nabla w_{r,h}, \nabla v_r) dx.$$

Finally, the following result is presented.

Proposition 4.4.5. *The equation* (4.29) *has a unique solution* $w_{r,h} \in V_h^0(\Omega)$. *Furthermore, this solution* w_k^h *is an admissible direction (satisfies condition* (4.24)*) for* $\nabla J_h(u_{r,h})$ *and satisfies*

$$\nabla J_h(u_r)^\top w_{r,h} < 0, \quad \forall r \in \mathbb{N}.$$

Proof. [12, Sec. 3.2] □

With this last result the author concludes that the algorithm is globally convergent.

We have summarized the main results presented in [12]. As it was discussed above, the analysis of the search direction for the $1 < p < 2$ case was carried out

in function spaces and for $p \geq 2$ it was analyzed with a "discretize then optimize" approach in finite dimension spaces. Since the multigrid optimization algorithm is defined in finite dimension spaces, we present the preconditioned descent algorithm in finite dimension as well. Thus, the general preconditioned descent algorithm reads as follows.

Algorithm 9 General Preconditioned descent algorithm

1: Initialize $u_0 \in V_h^0$ and set $r = 0$.
2: **for** $r = 1, 2, \cdots$ **do**

1. **if** $\nabla J(u_{r,h}) = 0$ **then**, STOP.
2. **end if**
3. Find a descent direction $w_{r,h}$ by solving the following equation

$$P_r(w_{r,h}, v) = -\nabla J(u_{r,h})^\top v, \ \ \forall v \in V_h^0,$$

if $1 < p < 2$,

$$P_r(w_{r,h}, v) = \int_{\Omega_h} (\epsilon + |\nabla u_{r,h}|)^{p-2} \nabla w_{r,h} \nabla v \ dx, \ \forall v \in V_h^0,$$

else if $p > 2$,

$$P_r(w_{r,h}, v) = \int_{\Omega_h} \nabla w_{r,h} \nabla v \ dx, \ \forall v \in V_h^0,$$

end.

4. Perform an efficient line search technique to obtain α_r.
5. Update $u_{r+1,h} := u_{r,h} + \alpha_r w_{r,h}$ and set $r = r + 1$.
6. **end for**

For a deeper analysis of the algorithm, see [12, Sec. 3] and the references therein.

4.4.2 Line search strategy

We propose to use a line search algorithm in order to choose the steph lenght α_r. This strategy selects the step by backtracking. i.e., the procedure starts with $\alpha_r = 1$, if $u_{r+1} = u_r + w_r$ does not satisfy the descent condition on $J(u_{r+1})$, the strategy is to reduce (or backtrack) α_r until $u_{r+1} = u_r + \alpha_r w_r$ satisfies the descent condition. In this section, we describe a line search algorithm proposed in [8, Sec. 6.3.2] which uses polynomial models of the objective functional for backtracking.

First, let us introduce the following function

$$\varphi_r(\alpha) := J(u_r + \alpha w_k)$$

Initially, we know that

$$\varphi_r(0) = J(u_r) \text{ and } \varphi_r'(0) = \nabla J(u_r)^\top w_r. \tag{4.30}$$

If we start with $\alpha = 1$, we have that

$$\varphi_r(1) = J(u_r + w_r). \tag{4.31}$$

Then, if $\varphi_r(1)$ does not satisfy the descent condition

$$\varphi_r(1) > \varphi_r(0) + \sigma\varphi_r'(0),$$

the algorithm construct the backtracking function φ_r, using (4.30) and (4.31), with the following quadratic model

$$m_2(\alpha) := (\varphi_r(1) - \varphi_r(0) - \varphi_r'(0))\alpha^2 + \varphi_r'(0)\alpha + \varphi_r(0).$$

The stationary point $\bar{\alpha_2}$ of m_2 is given by

$$\bar{\alpha}_2 = \frac{-\varphi_r'(0)}{2(\varphi_r(1) - \varphi_r(0) - \varphi_r'(0))}, \tag{4.32}$$

furthermore, we have that

$$m_2''(\alpha) = 2(\varphi_r(1) - \varphi_r(0) - \varphi_r'(0)) > 0$$

due to the fact that $\varphi_r(1) > \varphi_r(0) + \sigma\varphi_r'(0) > \varphi_r(0) + \varphi_r'(0)$. Therefore, we can conclude that $\bar{\alpha_2} = arg\min \varphi_r(\alpha)$. Also, $\bar{\alpha}_2 > 0$, because $\varphi_r'(0) < 0$. Then, we take $\alpha_r = \bar{\alpha_2}$.

Now, from (4.32) and since $\varphi_r(1) > \varphi_r(0) + \sigma\varphi_r'(0)$ we have

$$\bar{\alpha}_2 < \frac{1}{2(1-\sigma)}.$$

In fact, if $\varphi_r(1) \geq \varphi_r(0)$, then $\bar{\alpha} \leq \frac{1}{2}$. Hence, (4.32) gives an implicit upper bound of $\approx \frac{1}{2}$ for $\bar{\alpha}_2$ on the first backtrack. Morover, if $\varphi_r(1)$ is much larger than $\varphi_r(0)$, $\bar{\alpha}$ can be very small. Then, $\varphi_r(\alpha)$ can be poorly modeled by m_2 and the steps can be very small. Therefore, the algorithm impose a lower bound of $\frac{1}{10}$. Now, suppose

that $\sigma(\bar{\alpha}_2)$ does not satisfy the following Wolfe-Powell condition

$$J(u_r + \alpha w_r) \leq J(u_r) + \sigma \nabla J(u_r)^\top w_k, \tag{4.33}$$

then, we need to backtrack again. However, we have the following information available $\varphi_r(0) = J(u_r), \varphi_r'(0) = \nabla J(u_r)^\top w_r$ and the last two values of $\varphi(\alpha)$. Following the same previous idea, now we can model φ by a cubic model as follows: let α_p and α_{2p} be the last two previous values of α_r, the cubic model is given by

$$m_3(\alpha) := c\alpha^3 + d\alpha^2 + \varphi_r'(0)\alpha + \varphi_r(0),$$

where

$$\begin{pmatrix} c \\ d \end{pmatrix} = \frac{1}{\alpha_p - \alpha_{2p}} \begin{pmatrix} \frac{1}{\alpha_p^2} & \frac{-1}{\alpha_{2p}^2} \\ \frac{-\alpha_{2p}}{\alpha_p^2} & \frac{\alpha_p}{\alpha_{2p}^2} \end{pmatrix} \begin{pmatrix} \varphi_r(\alpha_p) - \alpha_r(0) - \alpha_r'(0)\alpha_p \\ \alpha_r(\alpha_{2p}) - \alpha_r(0) - \alpha_r'(0)\alpha_{2p}. \end{pmatrix}$$

The minimizer of the cubic model is given by

$$\tilde{\alpha_3} = \frac{-d + \sqrt{d^2 - 3c\varphi_r'(0)}}{3c} \tag{4.34}$$

If $\varphi(\alpha_p) \geq \varphi(0)$, then $\tilde{\alpha_3} < \frac{2}{3}\alpha_p$, however, as this reduction is considered too small [8], the upper bound of 0.5 is imposed, i.e. $\alpha_r = \frac{1}{2}\alpha_p$. Also, if $\tilde{\alpha}_3$ is a small fraction of α_r, once again the lower bound of $\frac{1}{10}$ is imposed and the algorithm sets $\alpha_k = \frac{1}{10}\alpha_p$.

Chapter 5

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 [9, 12, 17]. 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 5.1). 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.

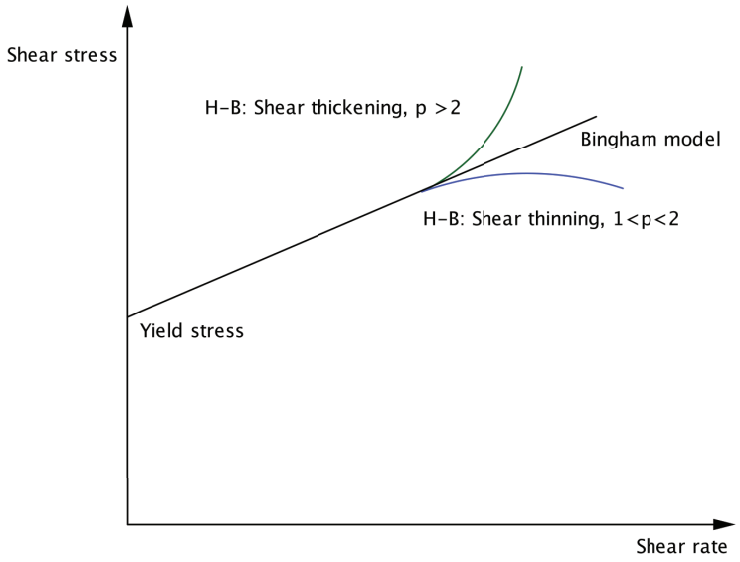

Figure 5.1: Viscoplastic models

In this work we simulate the pipe flow for the classic viscoplastic models: Herschel-Bulkley, Bingham and Casson. The three models 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:

$$\min_{u_h \in V_h^0} J_h(u_h) := \phi(\nabla u_h) + \int_{\Omega_h} \psi_\gamma(\nabla u_h)\, dx - \int_{\Omega_h} f u_h\, dx, \tag{5.1}$$

where

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

In the coming numerical experiments, we present the results of the MG/OPT algorithm applied to the numerical solution of (5.1). 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

Table 5.1: 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

Table 5.2: Unit square mesh information.

5.0.1 Herschel-Bulkley: case $1 < p < 2$

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 5.1). 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 [12]) 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 5.3, 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.

g	Finest m.	Coarsest m.	V-cycles	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
0.2	Ω_6	Ω_1	11	7.38e-07	965.49	0.0527
	Ω_5	Ω_1	15	3.75e-07	258.9	0.0520
	Ω_4	Ω_1	13	4.70e-07	51.2	0.0523
	Ω_3	Ω_1	10	3.75e-07	10.04	0.0522
0.4	Ω_6	Ω_1	26	1.08e-07	3914.61	0.0043
	Ω_5	Ω_1	17	3.79e-07	329.07	0.0046
	Ω_4	Ω_1	13	7.23e-07	51.33	0.0047
	Ω_3	Ω_1	23	5.00 e-07	24.07	0.0037

Table 5.3: Results of the resolution of problem (5.1) with $p = 1.75$, $\gamma = 10^3$ and $f = 1$.

This experiment was initialized with the solution of the Poisson problem, $-\Delta u_h = f$. From Table 5.3 we can notice that the number of V-cycles is similar when solving the problem at the different levels of discretization. In Figure 5.3 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 [12, Sec. 4.3.1]). The resulting velocity field is displayed in Figure 5.2. The flattening of the velocity in the center of the pipe corresponds to the plug flow velocity, where the material presents rigid zones.

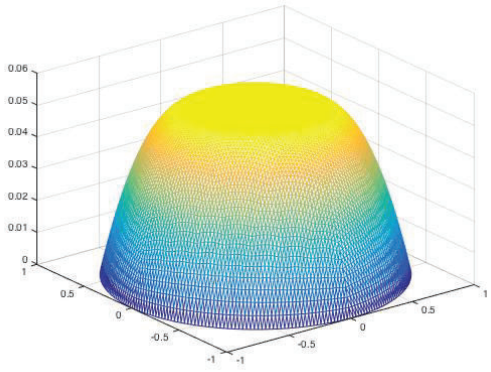

Figure 5.2: Calculated velocity u for mesh Ω_6. Parameters: $p = 1.75, g = 0.2, \gamma = 10^3$ and $\epsilon = 10^{-6}$

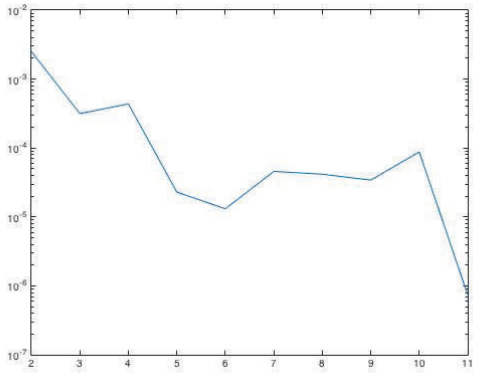

Figure 5.3: Calculated $|\nabla J_\gamma^\top e|$ for mesh Ω_6 and $g = 0.2$.

In Tables 5.4, 5.5, 5.6 and 5.7, we present the performance of the line search globalization technique, described in Section 4.4.2, 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 proce-

dures 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

Table 5.4: 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

Table 5.5: 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

Table 5.6: 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

Table 5.7: 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 5.8 and 5.9 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

Table 5.8: 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

Table 5.9: 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 algorithm
Ω_6	**965.49**	2146.37
Ω_5	258.95	71.67
Ω_4	**51.23**	63.11
Ω_3	10.04	8.30

Table 5.10: Time comparison, Experiment 1

$\lvert\nabla J_\gamma^\top e\rvert$		
Mesh	MG/OPT	preconditioned descent algorithm
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

Table 5.11: Norm $|\nabla J_\gamma^\top e|$comparison, Experiment 1

From Tables 5.10 and 5.11 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 5.10, 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.

Experiment 3

In this experiment, we set $p = 1.5$ and $f = 1$. We tested the algorithm with two different values of g. At higher values of g the plug flow zone is bigger and the execution time of the algorithm increases. Also, as in the previous experiment, the number of iterations is very stable as the mesh resolution is higher. This results are shown in Table 5.12.

g	Finest m.	Coarsest m.	V-cycles	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
0.1	Ω_6	Ω_1	5	8.28e-08	387.50	0.0428
	Ω_5	Ω_1	5	2.46e-08	83.34	0.0427
	Ω_4	Ω_1	3	2.68e-07	11.52	0.0427
	Ω_3	Ω_1	4	5.45e-07	4.02	0.0422
0.4	Ω_6	Ω_1	7	1.29e-07	683.96	9.08e-04
	Ω_5	Ω_1	6	1.63e-07	94.38	9.71e-04
	Ω_4	Ω_1	8	2.57e-08	29.23	8.85e-04
	Ω_3	Ω_1	12	3.02e-07	10.63	7.95e-04

Table 5.12: Results of the resolution of problem (5.1) with $p = 1.5$, $\gamma = 10^3$ and $f = 1$.

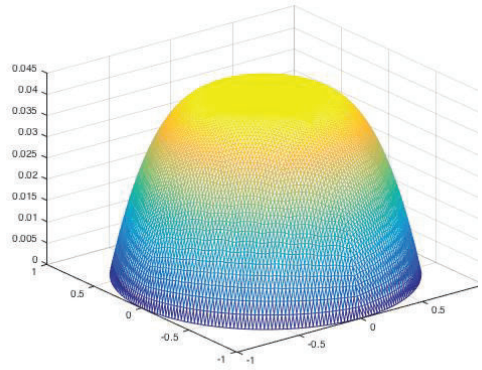

Figure 5.4: Calculated velocity u for mesh Ω_6. Parameters: $p = 1.5, g = 0.1, \gamma = 10^3$ and $\epsilon = 10^{-6}$

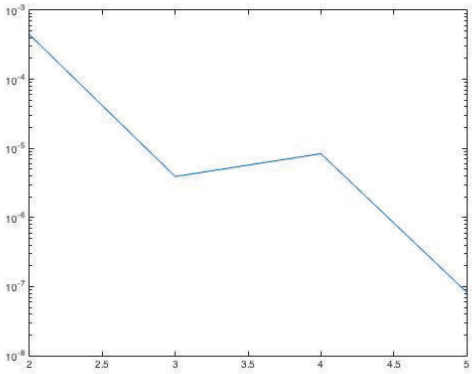

Figure 5.5: Calculated $|\nabla J_\gamma e|$ for mesh Ω_6.

The resulting velocity function is displayed in Figure 5.4 for $g = 0.1$. Here, we can see that the shear stress transmitted by a fluid layer decreases toward the center of the pipe, which provokes the solid-rigid movement in that area. In Figure 5.5 the decay of the norm $|\nabla J_\gamma e|$ is plotted in a logarithmic scale.

Experiment 4

For this experiment, we present the behaviour of the algorithm in the unit square domain for the following parameters: $p = 1.25$, $g = 0.2, \gamma = 10^3$ and $f = 3$. In Figure 5.6, we can see that in a square domain, the plug flow region is bigger due to the complexity of the domain. In this case, we obtained better results initializing the algorithm with an estimated solution given by one iteration of the full multigrid (FMG) scheme (see [29, Sec. 2.6]) at all refinement levels. We denote by r the number of MG/OPT cycles inside the FMG approach. All these results are presented in Table 5.13.

Finest mesh	Coarsest mesh	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
Ω_5	Ω_1	1	1	2.07e-07	101.74	0.011
Ω_4	Ω_1	2	1	4.83e-08	26.88	0.011
Ω_3	Ω_1	2	1	7.31e-08	6.88	0.010

Table 5.13: Results of the resolution of problem (5.1) with $p = 1.25, g = 0.2, \gamma = 10^3$ and $f = 3$.

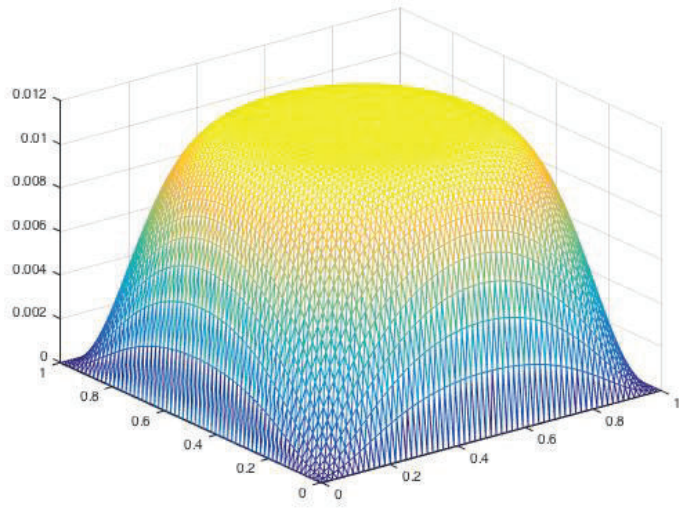

Figure 5.6: Calculated velocity u for mesh Ω_5. Parameters: $p = 1.25, g = 0.2, \gamma = 10^3$ and $f = 3$

Experiment 5: Comparison between MG/OPT and descent algorithms

In this experiment, we compare the behaviour of the MG/OPT algorithm versus the preconditioned descent algorithm for solving the same problem described in Experiment 4, in the finest grid.

	Time (s)	
Mesh	MG/OPT	preconditioned descent algorithm
Ω_5	**101.74**	143.53
Ω_4	**26.88**	28.08
Ω_3	**6.88**	8.09

Table 5.14: Time comparison, Experiment 4

	$\lvert\nabla J_\gamma^\top e\rvert$	
Mesh	MG/OPT	preconditioned descent algorithm
Ω_5	2.07e-07	1.38e-07
Ω_4	4.83e-08	5.94e-07
Ω_3	7.31e-08	7.82e-08

Table 5.15: Norm $|\nabla J_\gamma^\top e|$ comparison, Experiment 4

Tables 5.14 and 5.15 show that we have CPU time savings at all resolution levels.

Particularly, we can see a significant reduction of time at the highest resolution level Ω_5. The time reduction with respect to the execution time of the preconditioned algorithm is about of the 30%. Regarding the tolerance order achieved at the stopping criteria, both algorithms behaves similarly, with a slight advantage of the MG/OPT method.

5.0.2 Herschel-Bulkley: case $p \geq 2$

Experiment 6

We now analyze the behaviour of the MG/OPT algorithm in the case $p > 2$. The fact that p increases usually implies instabilities in the performance of the numerical algorithms (see [5, 2, 11, 16]). Also high values of g are difficult to consider, since the fluid presents more rigid zones along the cross section of the pipe.

Unlike the previous cases, where the MG/OPT algorithm was initialized with the solution of the Poisson problem, in this case we initialize the MG/OPT algorithm with an approximated solution given by the full multigrid (FMG) scheme. In the FMG scheme we apply r MG/OPT cycles at each level of the algorithm. We set the tolerance in 10^{-7} and the parameters $p = 4$, $\gamma = 10^3$ and $f = 3$.

g	Finest m.	Coarsest m.	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
0.2	Ω_5	Ω_1	8	1	9.22e-07	398.47	0.258
	Ω_4	Ω_1	31	1	3.48e-07	301.23	0.258
	Ω_3	Ω_1	9	2	3.11e-07	21.84	0.258
0.4	Ω_5	Ω_1	175	2	5.59e-07	7826.02	0.153
	Ω_4	Ω_1	51	2	1.80e-07	455.72	0.154
	Ω_3	Ω_1	28	2	8.33e-07	58.47	0.154

Table 5.16: MG/OPT results of problem (5.1) with $p = 4$, $\gamma = 10^3$ and $f = 3$.

The last column of Table 5.16 helps us to test the accuracy of the algorithm when estimating the plug flow velocity with different mesh sizes. The resulting velocity field is displayed in Figure 5.7. Now we are in the case of a shear-thickening material. Since the shear stress transmitted by a fluid layer decreases toward the center of the pipe, the velocity takes a conical form. The residual presented in Figure 5.8 shows local convergence in the last iterations.

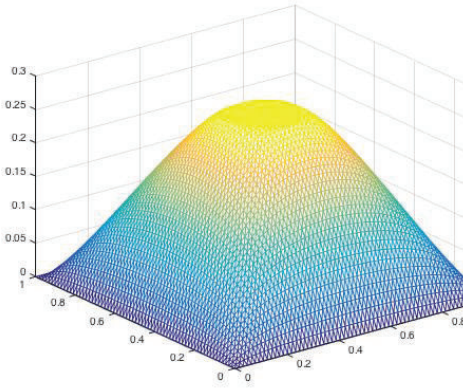

Figure 5.7: Calculated velocity u for mesh Ω_5. Parameters: $p = 4$, $g = 0.2, \gamma = 10^3$.

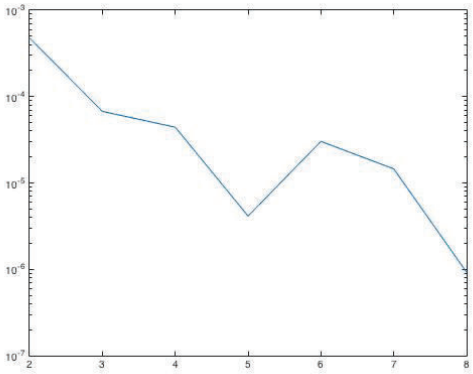

Figure 5.8: Calculated $|\nabla J_\gamma e|$ for mesh Ω_5.

5.0.3 Bingham

Bingham fluids are viscoplastic materials that can be seen as a particular case of the Herschel-Bulkley model when $p = 2$. The main characteristic of Bingham fluids is that when the shear stress exceeds the yield stress, the material presents a linear stress-shear rate relationship (see Figure 5.1).

Experiment 7

The following experiment was carried out in a circular geometry representing the cross section of a pipe, and we consider the following parameters: $p = 2$, $\gamma = 10^3$ and $f = 1$. In Table 5.17, we can see an example of the algorithm accuracy when determining the plug flow velocity using different mesh sizes. When finer grids are used, there is no difference in the plug flow velocity and we obtain a faster decay of the norm $|\nabla J_\gamma^\top e|$. The resulting velocity function is displayed in Figure 5.9, and we can see the classic behaviour of these materials.

g	Finest m.	Coarsest m.	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
	Ω_5	Ω_1	94	1	4.18e-07	1723.07	0.090
0.2	Ω_4	Ω_1	13	1	4.96e-07	60.02	0.090
	Ω_3	Ω_1	17	1	6.79e-07	18.64	0.089

Table 5.17: MG/OPT results for problem (5.1) with $p = 2$, $\gamma = 10^3$ and $f = 1$.

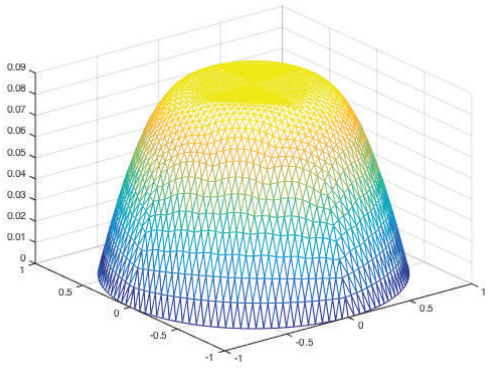

Figure 5.9: Calculated velocity u for mesh Ω_5. Parameters: $p = 2, g = 1, \gamma = 10^3$

Experiment 8

As g increases, the plug flow zone increases as well. Thus, we test the MG/OPT algorithm with $g = 1$ to show this phenomenon in a square cross section. In this experiment we fix the parameters $\gamma = 10^3$ and $f = 10$, and we set the tolerance in 10^{-5}. This experiment was initialized with an estimated solution given by the FMG scheme with r MG/OPT cycles at all levels of refinement. In Figures 5.10 and 5.11 we can notice the difference of the plug flow zone when $g = 1$ and $g = 0.2$, respectively. Also, the execution time when solving this experiment is longer when $g = 1$ than $g = 0.2$. This fact is due to the increasing plug flow and the complexity of the geometry.

g	Finest m.	Coarsest m.	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
	Ω_5	Ω_2	2	1	9.42e-05	152.95	0.2920
1	Ω_4	Ω_1	8	1	9.13e-05	85.7	0.2920
	Ω_3	Ω_1	3	2	7.57e-05	9.4	0.2925
	Ω_5	Ω_2	0	1	9.83e-06	56.26	0.6326
0.2	Ω_4	Ω_1	0	1	4.88e-05	10.87	0.6319
	Ω_3	Ω_1	0	1	4.93e-05	9.93	0.6319

Table 5.18: MG/OPT results of problem (5.1) with $p = 2, g = 1, \gamma = 10^3$ and $f = 10$.

An interesting fact in Experiment 8 is exhibited in the fourth column of Table 5.18 for $g = 0.2$. Since g is small the contribution of the less regular component $\int_\Omega \psi_\gamma(\nabla u)\, dx$ of the functional decreases, therefore the plug flow zone is not big with respect to the cross section area. Hence, the minimization problem was

uniquely solved by one iteration of the FMG scheme, i.e., the algorithm reached the convergence tolerance in only one iteration.

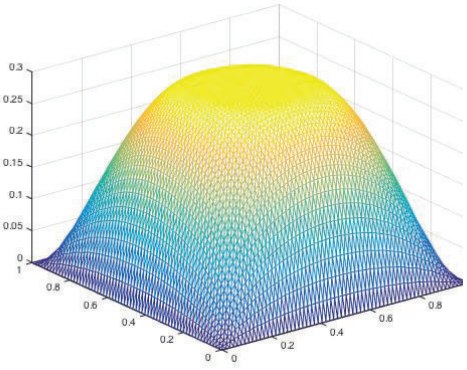

Figure 5.10: Calculated velocity u for mesh Ω_5. Parameters: $p = 2, g = 1, \gamma = 10^3$

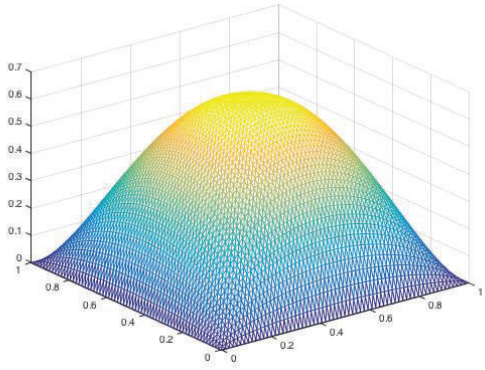

Figure 5.11: Calculated velocity u for mesh Ω_5. Parameters: $p = 2, g = 0.2, \gamma = 10^3$

Experiment 9: Comparison between MG/OPT and Descent algorithms

In this experiment, we compare the behaviour of the MG/OPT algorithm and the preconditioned descent algorithm when solving the same problem at the finest grid. In Tables 5.19 and 5.20, we compare the CPU time and the stopping criteria registered for Experiment 8, with parameters $p = 2$, $g = 1, \gamma = 10^3$ and $f = 10$.

	$\lvert\nabla J_\gamma^\top e\rvert$	
Mesh	MG/OPT	preconditioned descent algorithm
Ω_5	9.42e-05	5.62e-05
Ω_4	9.13e-05	9.87e-05
Ω_3	7.570e-05	5.61e-05

Table 5.19: Norm $\lvert\nabla J_\gamma^\top e\rvert$ comparison.

	Time (s)	
Mesh	MG/OPT	preconditioned descent algorithm
Ω_5	**152.95**	170.79
Ω_4	**85.7**	103.44
Ω_3	9.4	7.6

Table 5.20: Time comparison.

From Tables 5.19 and 5.20, we can conclude that we can not achieve a decrease of the norm $|\nabla J_\gamma^\top e|$ with respect to the results obtained by the preconditioned descent algorithm. However, in almost all discretization levels, we have CPU time savings of around the 14% with the MG/OPT algorithm with respect to the execution time of the preconditioned descent algorithm.

5.0.4 Casson

The Casson model is a viscoplastic model that was first developed for modeling printing inks. However, it has also been used to model food flow behaviour such as chocolate and cocoa products [28], and has been applied to biorheology models like hemodynamics and viscometric flows [32].

Experiment 10

In Tables 5.21 and 5.22, we present the results for the Casson fluid flow in a square and circle cross sections, respectively . Both experiments are set with the following parameters: $\gamma = 10^3$, $f = 1$ and tolerance 10^{-7}. The experiments are initialized with a FMG approximated solution.

g	Finest m.	Coarsest m.	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
	Ω_5	Ω_2	2	2	6.59e-07	321.98	2.68e-04
0.2	Ω_5	Ω_1	4	2	3.23e-07	458.00	2.36e-04
	Ω_4	Ω_1	1	2	7.50e-07	50.03	2.03e-04
	Ω_3	Ω_1	6	2	6.066e-07	28.55	2.06e-04

Table 5.21: MG/OPT results of problem (5.1) with $\gamma = 10^3$ and $f = 1$.

g	Finest m.	Coarsest m.	V-cycles	r	$\lvert\nabla J_\gamma^\top e\rvert$	Time(s)	Plug flow vel.
	Ω_6	Ω_2	18	2	9.78e-07	3138.30	0.0150
	Ω_6	Ω_1	30	2	3.25e-08	4553.74	0.0150
0.2	Ω_5	Ω_1	27	2	9.11e-07	829.65	0.0150
	Ω_4	Ω_1	36	2	6.78e-07	243.84	0.0148
	Ω_3	Ω_1	33	2	1.31e-07	56.65	0.0146
	Ω_6	Ω_2	0	2	6.50e-07	377.85	4.91e-04
	Ω_6	Ω_1	1	2	2.60e-07	528.69	4.79e-04
0.4	Ω_5	Ω_1	6	2	5.59e-07	220.84	4.89e-04
	Ω_4	Ω_1	9	2	1.67e-07	59.43	4.85e-04
	Ω_3	Ω_1	6	2	1.034e-07	11.04	4.54e-04

Table 5.22: MG/OPT results of problem (5.1) with $\gamma = 10^3$ and $f = 1$.

In these experiments we notice that, if we choose Ω_6 as the finest mesh and Ω_2 as the coarsest one instead of Ω_1, the algorithm is executed in less time for the same amount of nodes. Moreover, for the parameter $g = 0.4$, the algorithm only needs 2 iterations of the FMG scheme to reach the stopping tolerance of 10^{-7}. This behaviour tells us, at least experimentally, that beyond a number of grids used in the multigrid cycles we can not achieve more CPU time savings.

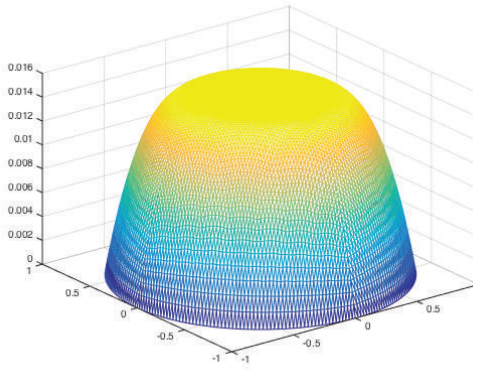

Figure 5.12: Calculated velocity u for mesh Ω_6. Parameters: $g = 0.2, \gamma = 10^3$

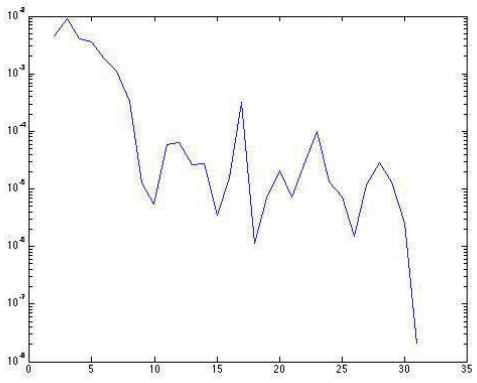

Figure 5.13: Calculated velocity u for mesh Ω_6. Parameters: $g = 0.2, \gamma = 10^3$

Experiment 11: Comparison between MG/OPT and Descent algorithms

In this experiment, we present the results of solving the same problem at the finest grid with the preconditioned descent algorithm and the MG/OPT algorithm. In the following tables we compare the results from Experiment 10, with $g = 0.4$, based on

the CPU time and the stopping criteria registered.

$\lvert\nabla J_\gamma^\top e\rvert$		
Mesh	MG/OPT	preconditioned descent algorithm
$\Omega_6 - \Omega_2$	6.50e-07	9.8e-07
$\Omega_6 - \Omega_1$	2.60e-07	9.8e-07
Ω_5	5.59 e-07	9.77e-07
Ω_4	1.67e-07	9.14e-07
Ω_3	1.34e-07	9.66e-07

Table 5.23: Norm $|\nabla J_\gamma^\top e|$ comparison.

Time (s)		
Mesh	MG/OPT	preconditioned descent algorithm
$\Omega_6 - \Omega_2$	**337.85**	2804.14
$\Omega_6 - \Omega_1$	**528.69**	2804.14
Ω_5	**220.84**	464.57
Ω_4	**59.43**	100.51
Ω_3	**11.04**	19.02

Table 5.24: Time comparison.

From Table 5.24, we can conclude that at all discretization levels, we have significant CPU time savings with the MG/OPT algorithm. The average of the time reduction regarding to the execution time of the preconditioned descent algorithm is about the 45% for meshes Ω_5, Ω_4 and Ω_3. Particularly at the finest level Ω_6 (with a V-cycle with no more than five grids), the time reduction with respect to the time of the preconditioned descent algorithm is about the 87.9%. Also, from Table 5.23, we notice that the MG/OPT algorithm reaches smaller values for the norm $|\nabla J_\gamma^\top e|$, at all the refinement levels, than the preconditioned descent algorithm.

Chapter 6

Conclusions

6.1 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, and we deeply analyzed the differentiability of the functional. In particular, we proved that the Jacobian of the discretized functional is slantly differentiable. The MG/OPT algorithm was presented and all of the involved transfer operators analyzed. The convergence of the MG/OPT algorithm was stablished by using the mean value theorem for slantly differentiable functions and the global convergence of the underlying optimization algorithms. The main issues regarding the implementation of the algorithm were explained, and we described the type of global convergent deepest descent methods used as underlying optimization algorithms. 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. Particularly, we computed the solution for the Herschel-Bulkley, Bingham and Casson models.

Bibliography

[1] J. ALBERTY, C. CARSTENSEN AND S. A. FUNKEN , *Remarks around 50 lines of Matlab: short finite element implementation* , Springer J. Numerical Algorithms, 20 (1999) 117-137.

[2] R. BERMEJO AND J-. A. INFANTE, *A Multigrid Algorithm for the p-Laplacian*, SIAM J. Sci. Comput., 21 (2000) 1774-1789.

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

[4] A. BORZÌ AND M. VALLEJOS, *Multigrid Optimization Methods for Linear and Bilinear Elliptic Optimal Control Problems*, Computing, 82 (2008) 31-52.

[5] J. W. BARRET AND W. B. LIU, *Finite Element Approximation of the p-Laplacian*, Mathematics of Computation, 61 (1993) 523-537.

[6] E. CASAS AND L. A. FERNÁNDEZ, *Distributed control of systems governed by a general class of quasilinear elliptic equations*, Journal of differential equations, 104 (1993) 20-47.

[7] X. CHEN, Z. NASHED AND L. QI, *Smoothing Methods and Semismooth Methods for Nondifferentiable Operator Equations*, SIAM J. Numer. Anal., 38 (2000) 1200-1216.

[8] J.E. DENNIS JR AND R.B. SCHNABEL, *Numerical methods for unconstrained optimization and nonlinear equations* , SIAM, 16 (1996)

[9] G. DUVANT AND J.L LIONS, *Inequalities in mechanics and physics*, Springer-Verlag, (1976).

[10] M. GIAQUINTA AND G. MODICA, *Mathematical analysis: an introduction to functions of several variables*, Springer Science & Business Media, (2010).

[11] 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.

[12] 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.

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

[14] 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.

[15] M. HINZE, R. PINNAU, M. ULBRICH AND S. ULBRICH, *Optimization with PDE constraints* , Springer Science & Business Media , 23 (2008).

[16] Y.Q. HUANG, R. LI AND W. LIU, *Preconditioned descent algorithms for p-Laplacian* , Springer, Journal of Scientific Computing, 32 (2007) 343-371.

[17] 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.

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

[19] R. KORNHUBER, *Monotone multigrid methods for elliptic variational inequalities I*, Numer. Math., 69 (1994) 167-184.

[20] R. KORNHUBER, *Monotone multigrid methods for elliptic variational inequalities II*, Numer. Math., 72 (1996) 481-499.

[21] R. KORNHUBER, *On Constrained Newton Linearization and Multigrid for Variational Inequalites*, Numer. Math., 91 (2002) 699-721.

[22] 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.

[23] 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.

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

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

[26] J. NOCEDAL AND S. WRIGHT, *Numerical optimization* , Springer Science & Business Media, (2006).

[27] 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.

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

[29] U. TROTTENBERG, C. W. OOSTERLEE AND A. SCHULLER, *Multigrid*, Academic press, 2000.

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

[31] 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.

[32] WP. WALAWENDER, TY. CHEN AND DF. CALA, *An approximate Casson fluid model for tube flow of blood*, Biorheology, 12 (1975) 111-119.

[33] Z. WEN AND D. GOLDFARB, *A Line Search Multigrid Method for Large-Scale Nonlinear Optimization*, SIAM J. Optim., 20 (2009) 1478-1503.