

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

**Dottorato di Ricerca in Ingegneria Energetica,
Nucleare e del Controllo Ambientale**

Ciclo XXIV

Settore concorsuale di afferenza: 09/C3
Settore scientifico-disciplinare: ING-IND/19

**ANALYSIS OF OPTIMAL CONTROL PROBLEMS
FOR THE INCOMPRESSIBLE MHD EQUATIONS
AND IMPLEMENTATION IN A FINITE ELEMENT
MULTIPHYSICS CODE**

Presentata da: GIORGIO BORNIA

Coordinatore del Dottorato:
Prof. ANTONIO BARLETTA

Relatore:
Dott.Ing. SANDRO MANSERVISI

Esame finale anno 2012

Contents

Introduction	5
1 Abstract setting for optimal control problems	7
1.1 Normed spaces	7
1.2 Sobolev spaces	9
1.3 Abstract frameworks for mixed problems	16
1.3.1 Abstract linear mixed problem	16
1.3.2 Abstract nonlinear mixed problem	20
1.4 Constrained nonlinear optimal control problems	22
2 Lifting function approach for boundary optimal control	27
2.1 Optimal control of the MHD equations	27
2.2 Description of the optimal control problem	30
2.3 Weak formulation of the MHD state problem	34
2.4 Existence of a solution to the state equations	39
2.4.1 Reduction to homogeneous boundary conditions	39
2.4.2 Coercivity property	41
2.4.3 Existence	43
2.5 Optimal control problem	45
2.5.1 Existence of an optimal solution	45
2.5.2 First-order necessary condition	47
2.5.3 Optimality system	52
3 Finite element approximation of the optimality system	55
3.1 Finite element method	55
3.2 Finite element multigrid algorithm	64
3.3 Approximation of the optimality system	67
3.4 Gradient algorithm for the optimality system	69
4 Computational implementation of the optimality system	73
4.1 The <i>FEMuS</i> library	73

4.2	Configuration of the <i>FEMuS</i> library	76
4.2.1	External libraries	76
4.2.2	Internal configuration	78
4.3	Library main classes	80
4.3.1	The <i>EquationsMap</i> class	81
4.3.2	The <i>EqnBase</i> class	83
4.3.3	The <i>Quantity</i> class	85
4.3.4	The <i>Vect</i> class	86
4.3.5	The <i>Mesh</i> , <i>GeomEl</i> and <i>Domain</i> classes	87
4.3.6	The <i>GenCase</i> class	88
4.3.7	The <i>FEElemBase</i> and <i>FEGauss</i> classes	90
4.3.8	The <i>RunTimeMap</i> class	92
4.3.9	The <i>Physics</i> class	92
4.3.10	The <i>Utils</i> and <i>Files</i> classes	93
4.4	Multiphysics application structure	93
4.4.1	Basic structure	93
4.4.2	Main function	100
4.4.3	Equation implementation	102
5	Numerical results	105
5.1	Code verification test for the lifting function	105
5.2	Two-dimensional optimal control case	111
5.3	Three-dimensional results	118
5.3.1	Hartmann flow optimal control case	118
5.3.2	Flow inversion optimal control case	122
	Conclusions	137
	List of Figures	142
	List of Tables	143
	Bibliography	148

Introduction

This thesis deals with the study of optimal control problems for the incompressible Magnetohydrodynamics (MHD) equations. Particular attention to these problems arises from several applications in science and engineering, such as fission nuclear reactors with liquid metal coolant, aluminum casting in metallurgy and crystal growth in semiconductor industry. In such applications it is of great interest to achieve the control on the fluid state variables through the action of the magnetic Lorentz force. In this thesis we investigate a class of boundary optimal control problems, in which the flow is controlled through the boundary conditions of the magnetic field. We consider a systematic mathematical treatment for the analysis, discretization and numerical solution of these problems, which are expressed in terms of the constrained minimization of a cost functional. Due to their complexity, boundary optimal control problems present a wide variety of challenges in the definition of an adequate solution approach, both from a theoretical and from a computational point of view. In this thesis we propose a new boundary control approach, based on lifting functions of the boundary conditions, which yields both theoretical and numerical advantages. With the introduction of lifting functions, boundary control problems can be formulated as extended distributed problems.

The work is organized in five chapters as follows. In Chapter 1 we introduce the basic mathematical concepts for the analysis of the optimal control problems considered in this work. After recalling some general definitions about normed spaces, we illustrate the Sobolev function spaces and we quote some fundamental properties related to them. Then, we report two standard abstract results about the existence of solutions to linear and nonlinear mixed variational problems. The MHD equations we consider in this thesis fall within the nonlinear abstract framework. Furthermore, we describe an abstract formulation of constrained nonlinear optimal control problems upon which our subsequent analyses are built.

In Chapter 2 the lifting function approach proposed in this work for the treatment of boundary optimal control problems is illustrated. The optimal control problem is formulated in terms of the minimization of a cost functional with the nonlinear constraints of the MHD equations. A weak formulation of these equations is given and the existence of a solution is proved. Then, the existence of

a solution to the optimal control problem is shown. By means of the Lagrange multiplier principle, a first-order necessary condition is derived from which an optimality system is obtained. A solution of the optimality system is a candidate solution for the original optimal control problem.

Chapter 3 describes the main principles about the finite element method that is used for the discretization of the optimality system. It also illustrates the multigrid algorithm for the solution of linear systems arising from finite element approximation. In order to solve the optimality system in a robust and accurate manner, a gradient algorithm is described.

Chapter 4 is devoted to the description of the finite element object-oriented library that has been implemented for the numerical solution of the optimality system. A description of the library structure, functionalities and main classes is given. The library can perform the solution of the optimality system for parallel architectures and multigrid solvers. A particular attention in the library development was given to a modular and flexible implementation of a set of coupled equations in a multi-physics framework, such as the optimality system here considered.

In Chapter 5 we show the results of numerical solutions of the optimality system for two- and three-dimensional cases. These results show that a candidate solution of the optimal control problem here considered can be computed in a robust and effective manner.

Chapter 1

Abstract setting for optimal control problems

1.1 Normed spaces

We use the terminology *normed vector space* or more concisely *normed space* to indicate a vector space endowed with a norm. The definitions we provide here regarding normed spaces are consistent with their underlying counterparts in the context of topology.

A set V is a *neighborhood* of a point p of a normed space if there exists an open ball with center p and radius $r > 0$ that is contained in V . Given a subset S of a normed space X , a neighborhood of a subset is a neighborhood of all of its points. We say that a point $x \in X$ is an *accumulation point* (or *limit point*) of S in X if every neighborhood of x contains at least one element $m \in S$ different from x . Note that x need not belong to S . We define the *closure of S in X* as the set given by the union of S and all the accumulation points of S in X . We denote it as \bar{S}^X or simply \bar{S} if no ambiguity arises. The subset S is said to be *closed in X* if it contains all its accumulation points, namely, $S = \bar{S}$. A subset S of X is said to be *dense in X* if $\bar{S} = X$. In this case, each $x \in X$ is the limit of a sequence of elements of S . Furthermore, a subset S is *convex* if, for all $x, y \in S$ and all $\alpha \in [0, 1]$, one has that $\alpha x + (1 - \alpha)y \in S$. Note that both closed and convex subsets are not vector spaces in general. The normed space X is *separable* if it contains a countable dense subset.

A normed space is said to be a *complete* if every Cauchy sequence is convergent in the norm of the space. A complete normed space is commonly referred to as *Banach space*. Every normed space X is either a Banach space or a dense subset of a Banach space Y , called the *completion* of X , whose norm satisfies $\|x\|_Y = \|x\|_X$ for all $x \in X$. If an inner product (\cdot, \cdot) is defined on a vector space, a norm

induced by this inner product can always be defined as $\|x\| = \sqrt{(x, x)}$. If the space endowed with such a norm is complete, it is said to be a *Hilbert space*.

Given two normed spaces X and Y with norms $\|\cdot\|_X$ and $\|\cdot\|_Y$ respectively, we say that an operator $f : X \rightarrow Y$ is *continuous* if, for any sequence $\{x_n\}$ in X such that $\|x_n - x\|_X \rightarrow 0$ as $n \rightarrow \infty$, one has $\|f(x_n) - f(x)\|_Y \rightarrow 0$. When f is linear, it can be shown that the continuity of f is equivalent to the existence of a constant C such that $\|f(x)\|_Y \leq C\|x\|_X$ for all $x \in X$. This last property is referred to by saying that f is a *bounded* linear operator. A linear continuous bijection between two spaces with a continuous inverse is called an *isomorphism*.

We now give some notions of compactness. A subset A of X is said to be *compact* if every sequence in A has a subsequence converging to some element of A for the norm of X . The set A is called *precompact* in X if its closure \bar{A} in the norm topology of X is compact. An operator f between X and Y is said to be *compact* if $f(B)$ is precompact whenever B is bounded in X (i.e., whenever B is contained in a ball with some finite radius R centered in the zero element of X).

Given a vector space V , we define the *dual space* V' of V as the space of all linear functionals $g : V \rightarrow \mathbb{R}$ that are continuous with respect to the topology induced by the norm of V . The dual space is itself a vector space under straightforward definitions of pointwise addition and scalar multiplication. We denote with $\langle \cdot, \cdot \rangle$ the duality pairing between V' and V . If V is endowed with a norm $\|\cdot\|_V$, a norm can be defined on the dual space V' as

$$\|f\|_{V'} = \sup_{\|v\|_V \leq 1} |\langle f, v \rangle|. \quad (1.1)$$

It can be shown that this definition is well posed due to the continuity and linearity of the functionals. The topology induced by this norm or by equivalent norms turns V' into a Banach space, whether or not V is Banach. Then, V' is called the *normed dual* of V .

Given the definition of dual space, we may define the concept of *weak convergence*. A sequence $\{v_n\}$ in V is said to *converge weakly* to some element $v \in V$ if the sequence $\{\langle f, v_n \rangle\}$ converges to $\langle f, v \rangle$ for all $f \in V'$. The corresponding underlying topology is the weakest topology (i.e., the topology with the fewest open sets) that still renders continuous each element of the normed dual V' . We denote weak convergence as $v_n \rightharpoonup v$, while the usual convergence in the norm $\|\cdot\|_V$ is also called *strong convergence* and is denoted as $v_n \rightarrow v$. Since by the continuity of the dual elements we have $|\langle f, v_n \rangle| \leq \|f\|_{V'} \|v_n\|_V$, strong convergence implies weak convergence. The converse is generally not true, unless V is finite-dimensional.

Let us also endow normed spaces with a definition of *reflexivity*. Consider a normed space V and its second dual $V'' = (V')'$, consisting in the space of linear functionals $f : V' \rightarrow \mathbb{R}$ that are continuous with respect to the topology induced

by the norm of V' . Let us denote with $\langle \cdot, \cdot \rangle_V$ the duality pairing between V' and V and with $\langle \cdot, \cdot \rangle_{V'}$ the duality pairing between V'' and V' . A linear injection $J : V \rightarrow V''$ can be defined by associating to every element $v_0 \in V$ an element of the second dual $Jv_0 \in V''$, given by the evaluation at the point v_0 of any $v' \in V'$, namely

$$\langle Jv_0, v' \rangle_{V'} = \langle v', v_0 \rangle_V, \quad v' \in V'. \quad (1.2)$$

It can be shown via the Hahn-Banach extension theorem that J is an isometric isomorphism of V into V'' . If the range of J is the entire space V'' , then the space V is said to be *reflexive*.

Finally, we provide the notion of embedding between two normed spaces. We say that X is *embedded* into Y if X is a vector subspace of Y , and the operator $I : X \rightarrow Y$ defined by $Ix = x$ for all $x \in X$ is continuous. The operator I is called *embedding* and it is evidently linear by definition. Therefore, embeddings are bounded operators. An embedding I is said to be *dense* if X is dense in Y , and *compact* if the embedding operator I is compact.

1.2 Sobolev spaces

In order to establish the mathematical formulation of an optimal control problem, the definition of appropriate function spaces is in order. The *Sobolev spaces* and some of their properties that will be used in the following are briefly introduced in this section. For a thorough discussion on these spaces we refer to [1]. We assume that the reader is familiar with the following definitions concerning subsets of the Euclidean space \mathbb{R}^n . Unless otherwise stated, we consider a *Lipschitz continuous simply connected convex* open bounded domain $\Omega \subset \mathbb{R}^n$. Lack of convexity typically brings to a lack of regularity in the solution of differential problems defined on such domains. Since we are interested in boundary optimal controls, the regularity properties of the boundary are also important for the statement of some results and the *Lipschitz continuous* hypothesis is often necessary.

We denote by $\mathcal{D}(\Omega)$ or $\mathcal{C}_0^\infty(\Omega)$ the vector space of infinitely differentiable functions with compact support in Ω . In particular we define the set $\mathcal{D}(\bar{\Omega})$ by

$$\mathcal{D}(\bar{\Omega}) = \{u|_\Omega; u \in \mathcal{D}(\mathbb{R}^N)\} = \{u|_\Omega; u \in \mathcal{D}(\mathcal{O})\}, \quad (1.3)$$

where \mathcal{O} is any open subset of \mathbb{R}^N such that $\bar{\Omega} \subset \mathcal{O}$. The space of linear functionals on $\mathcal{D}(\Omega)$ that are continuous with respect to the usual definition of convergence in $\mathcal{D}(\Omega)$ is denoted as $\mathcal{D}'(\Omega)$ and is called the *space of distributions* on Ω . We denote by $\langle \cdot, \cdot \rangle$ the duality pairing between $\mathcal{D}'(\Omega)$ and $\mathcal{D}(\Omega)$. In order to define the *derivative of a distribution*, we introduce a multi-index $\alpha = (\alpha_1, \dots, \alpha_N) \in \mathbb{N}^N$

together with the notation

$$|\alpha| = \sum_{i=1}^N \alpha_i. \quad (1.4)$$

We define for $\alpha \in \mathbb{N}^N$ the derivative of a distribution $u \in \mathcal{D}'(\Omega)$ as the distribution $\partial^\alpha u \in \mathcal{D}'(\Omega)$ such that

$$\langle \partial^\alpha u, \phi \rangle = (-1)^{|\alpha|} \langle u, \partial^\alpha \phi \rangle \quad \forall \phi \in \mathcal{D}(\Omega). \quad (1.5)$$

Let us now define *Lebesgue spaces* or L^p spaces, upon which lies the construction of Sobolev spaces. For every real number p such that $1 \leq p < \infty$, we indicate with the usual notation $L^p(\Omega)$ the vector space whose elements are the equivalence classes of Lebesgue integrable functions $u : \Omega \rightarrow \mathbb{R}$ such that

$$\left(\int_{\Omega} |u|^p d\Omega \right)^{\frac{1}{p}} < \infty, \quad (1.6)$$

with respect to the following equivalence relation: two functions are identified if they are equal *almost everywhere* in Ω , namely, up to any subset of Ω with Lebesgue measure zero. The space $L^\infty(\Omega)$ is defined as the vector space of Lebesgue integrable functions such that

$$\text{ess sup}_{x \in \Omega} |u(x)| < \infty, \quad (1.7)$$

where “ess sup” denotes the essential supremum, i.e. the supremum up to a set of measure zero. Hence, these functions are said to be *essentially bounded*.

For $1 \leq p < \infty$, $L^p(\Omega)$ is a Banach space for the norm

$$\|u\|_{L^p} = \left(\int_{\Omega} |u|^p d\Omega \right)^{\frac{1}{p}}. \quad (1.8)$$

Also $L^\infty(\Omega)$ is a Banach space with the norm

$$\|u\|_{L^\infty} = \text{ess sup}_{x \in \Omega} |u(x)|. \quad (1.9)$$

The space $L^2(\Omega)$ is also a Hilbert space for the inner product

$$(u, v)_{L^2} = \int_{\Omega} u v d\Omega. \quad (1.10)$$

Now we recall some fundamental results concerning Lebesgue spaces. The following inequalities are often used for proving important properties of variational problems, such as those considered in this thesis. We begin with the *Minkowski* inequality, which is a necessary condition for $\|\cdot\|_{L^p}$ to be a norm for L^p spaces.

Theorem 1. *Let $u, v \in L^p(\Omega)$. Then*

$$\|u + v\|_{L^p} \leq \|u\|_{L^p} + \|v\|_{L^p}. \quad (1.11)$$

The Minkowski inequality is proved by using the well-known *Hölder's inequality*, which yields a characterization of the product of two functions belonging to different Lebesgue spaces. In the following, we say that two indices $p, q \in \mathbb{N}$ are *Hölder conjugate* provided that

$$\frac{1}{p} + \frac{1}{q} = 1. \quad (1.12)$$

When either p or q are taken as infinity, $1/\infty$ means zero.

Theorem 2 (Hölder's inequality). *Let $p, q \in \mathbb{N}$, $1 \leq p, q \leq \infty$ be Hölder conjugate indices. If $u \in L^p(\Omega)$ and $v \in L^q(\Omega)$, then $uv \in L^1(\Omega)$ and*

$$\|uv\|_{L^1} \leq \|u\|_{L^p} \|v\|_{L^q}. \quad (1.13)$$

For $p = q = 2$ Hölder's inequality is referred to as *Cauchy-Schwarz inequality*, namely

$$|(u, v)_{L^2}| \leq \|u\|_{L^2} \|v\|_{L^2}. \quad (1.14)$$

Hölder's inequality can be generalized to more than two functions as follows.

Theorem 3 (Generalized Hölder's inequality). *Let the numbers $p_1, \dots, p_n \in \mathbb{N}$ be chosen such that $\sum_{i=1}^n \frac{1}{p_i} = 1$. Let $u_i \in L^{p_i}(\Omega)$, $i = 1, \dots, n$. Then*

$$\left\| \prod_{i=1}^n u_i \right\|_{L^1} \leq \prod_{i=1}^n \|u_i\|_{L^{p_i}}. \quad (1.15)$$

Also, let us recall an embedding property concerning L^p spaces.

Theorem 4. *Suppose that Ω has a finite measure. Then, if $1 \leq p \leq q \leq \infty$, the space $L^q(\Omega)$ is continuously embedded in $L^p(\Omega)$.*

It can be shown that the L^p spaces are separable for $1 \leq p < \infty$ and reflexive for $1 < p < \infty$.

The transition from Lebesgue spaces towards Sobolev spaces goes through the notion of *weak derivative* or *distributional derivative* of $L^p(\Omega)$ functions. This stems from the previous definition of derivative of a distribution. We say that a function u is *locally integrable* on Ω if $u \in L^1(U)$ for every open set U such that its closure \bar{U} is contained in Ω and \bar{U} is compact. We denote as $L^1_{loc}(\Omega)$ the space of

locally integrable functions on Ω . To every function $u \in L^1_{loc}(\Omega)$ we can associate a distribution indicated as $T_u \in \mathcal{D}'(\Omega)$, which is defined as

$$\langle T_u, \phi \rangle = \int_{\Omega} u \phi \, d\Omega \quad \forall \phi \in \mathcal{D}(\Omega). \quad (1.16)$$

It can be easily shown that this definition is well posed. A function $u \in L^1_{loc}(\Omega)$ is then said to have a *weak derivative of order α* if there exists a function $v \in L^1_{loc}(\Omega)$ whose associated distribution T_v is equal to the derivative of order α of the distribution T_u associated to u , i.e., $T_v = \partial^\alpha T_u \in \mathcal{D}'(\Omega)$. If the weak derivative of u exists, it is unique up to a set of measure zero and it will be denoted again as $\partial^\alpha u$. Since for $1 \leq p \leq \infty$ we have $L^p(\Omega) \subset L^1_{loc}(\Omega)$, the definition of weak derivative also holds for $L^p(\Omega)$ functions.

We are now ready to define the *Sobolev space* $W^{m,p}(\Omega)$ for every integer $m \geq 0$ and every real $1 \leq p < \infty$ as

$$W^{m,p}(\Omega) = \{u \in L^p(\Omega) \mid \partial^\alpha u \in L^p(\Omega) \quad \forall |\alpha| \leq m\}. \quad (1.17)$$

If we equip $W^{m,p}(\Omega)$ with the norm

$$\|u\|_{m,p} = \left(\sum_{|\alpha| \leq m} \int_{\Omega} |\partial^\alpha u|^p \, d\Omega \right)^{1/p} = \left(\sum_{|\alpha| \leq m} \|\partial^\alpha u\|_{L^p}^p \right)^{1/p}, \quad (1.18)$$

it can be proved to be a Banach space for all $p < \infty$. When $p = 2$ the Sobolev space $W^{m,2}(\Omega)$ will be denoted as $H^m(\Omega)$ and its norm as $\|\cdot\|_m$ as a shorthand for $\|\cdot\|_{m,2}$. This is a Hilbert space with the inner product defined by

$$(u, v)_m = \sum_{|\alpha| \leq m} \int_{\Omega} (\partial^\alpha u) (\partial^\alpha v) \, d\Omega. \quad (1.19)$$

For $p = \infty$ the space $W^{m,p}$ is still a Banach space when the norm is defined as

$$\|u\|_{m,\infty} = \max_{|\alpha| \leq m} \left(\operatorname{ess\,sup}_{x \in \Omega} |\partial^\alpha u(x)| \right) = \max_{|\alpha| \leq m} \|\partial^\alpha u\|_{L^\infty}. \quad (1.20)$$

As $\mathcal{D}(\Omega) \subset W^{m,p}(\Omega)$, we can define the closure of $\mathcal{D}(\Omega)$ in $W^{m,p}(\Omega)$ with respect to the norm $\|\cdot\|_{m,p}$. We write

$$W_0^{m,p}(\Omega) = \overline{\mathcal{D}(\Omega)}^{W^{m,p}}. \quad (1.21)$$

We denote $H_0^m(\Omega) = W_0^{m,2}(\Omega)$.

The spaces $W_0^{m,p}(\Omega)$ are endowed with a number of interesting properties. An important inequality that involves them is the *Poincaré-Friedrichs* inequality. It

comes in handy especially when proving the coercivity of variational forms. We consider the seminorm $|\cdot|_{m,p}$ in $W^{m,p}(\Omega)$ given by

$$|u|_{m,p} = \left(\sum_{|\alpha|=m} \int_{\Omega} |\partial^{\alpha} u|^p d\Omega \right)^{1/p}. \quad (1.22)$$

Clearly, all the multi-indices with $|\alpha| = m$ are considered. We immediately have that $|u|_{m,p} \leq \|u\|_{m,p}$. On the space $W_0^{m,p}(\Omega)$ the following theorem yields an inequality in the opposite sense, so that the seminorm is also a norm.

Theorem 5 (Poincaré-Friedrichs inequality). *Let Ω be connected and bounded in at least one direction. Then, for each integer $m \geq 0$, there exists a constant $K = K(m, \Omega)$ such that*

$$\|u\|_{m,p} \leq K |u|_{m,p} \quad \forall u \in W_0^{m,p}(\Omega). \quad (1.23)$$

As for Lebesgue spaces, it can be shown that $W^{m,p}(\Omega)$ is separable for $1 \leq p < \infty$ and reflexive for $1 < p < \infty$. The separability property is important for the analysis of the nonlinear problems that will be considered in this thesis.

Some fundamental properties of the Sobolev function spaces can be stated through the *Sobolev Embedding Theorem*. This is a fundamental result in the analysis of partial differential equations and often a crucial point in the analysis of nonlinear problems. In the following, we denote by $W_{loc}^{m,p}(\Omega)$ the space of functions belonging to $W^{m,p}(\Omega)$ for any open set U such that $\bar{U} \subset \Omega$ and \bar{U} is compact. The symbol $C^n(\Omega)$ indicates the space of functions with continuous derivatives up to order n .

Theorem 6 (Sobolev Embedding Theorem). *Let Ω be an open Lipschitz-continuous subset of \mathbb{R}^N and let $p \in \mathbb{R}$ with $1 \leq p < \infty$ and m and $n \in \mathbb{N}$ with $n \leq m$. The following embeddings hold algebraically and topologically:*

$$W^{m,p}(\Omega) \subset \begin{cases} W^{n,q}(\Omega) & \text{if } \frac{1}{p} - \frac{m-n}{N} = \frac{1}{q} > 0, \\ W_{loc}^{n,q}(\Omega) & \text{if } \frac{1}{p} = \frac{m-n}{N} \quad \forall q \in [1, \infty), \\ C^n(\Omega) & \text{if } \frac{1}{p} < \frac{m-n}{N}. \end{cases} \quad (1.24)$$

Moreover, if Ω is bounded, the last inclusion holds in $C^n(\bar{\Omega})$ and the embedding of $W^{m,p}(\Omega)$ into $W^{n,q'}(\Omega)$ is compact for all real q' that satisfy

$$1 \leq q' < \frac{Np}{N - (m-n)p} \quad \text{whenever} \quad N > (m-n)p \quad (1.25)$$

or

$$1 \leq q' < \infty \quad \text{when} \quad N = (m-n)p. \quad (1.26)$$

In addition, these compact embeddings are also valid for negative n or m .

We know that the existence of an embedding (which is a linear continuous operator) between two spaces X and Y implies the simple containment $X \subset Y$. In the case of Sobolev spaces, it can be shown that the converse holds as well, as a consequence of the completeness of L^p spaces and of the closed graph theorem of functional analysis. Hence, the simple containment is also said to be an embedding in the *algebraic* sense, in order to distinguish it from the *topological* embedding given by the existence of a continuous embedding operator. We also observe that in the framework of the Sobolev Embedding Theorem we can retrieve as a special case the embedding of L^p spaces given by Theorem 4.

We also introduce Sobolev spaces for vector-valued functions. The vector valued Sobolev space $\mathbf{W}^{m,p}(\Omega)$ can be defined simply as the Cartesian product of the previously defined Sobolev space $W^{m,p}(\Omega)$, i.e. $\mathbf{W}^{m,p}(\Omega) = W^{m,p}(\Omega)^N$ where N is the spatial dimension. A vector valued function $u : \Omega \rightarrow \mathbb{R}^N$ belongs to this space if each component belongs to the corresponding Sobolev space with the same indices m and p . The vector valued Sobolev space is a Banach space for the norm ($p < \infty$)

$$\|\mathbf{u}\|_{m,p} = \left(\sum_{i=1}^N \|u_i\|_{m,p}^p \right)^{1/p}. \quad (1.27)$$

For $p = \infty$ the standard norm is defined by

$$\|\mathbf{u}\|_{m,\infty} = \max_{1 \leq i \leq N} \|u_i\|_{m,\infty}. \quad (1.28)$$

Like in the scalar case, for $p = 2$ we have the Hilbert space $\mathbf{W}^{m,2}(\Omega)$ which will be denoted as $\mathbf{H}^m(\Omega)$. Consistently, we also adopt the boldface notation for vector-valued function spaces whose components are in $W_0^{m,p}(\Omega)$.

The most important dual spaces of Sobolev spaces are the dual spaces of $W_0^{m,p}(\Omega)$. The dual space of $W_0^{m,p}(\Omega)$ is denoted by $W^{-m,p'}(\Omega)$ with $\frac{1}{p} + \frac{1}{p'} = 1$. $W^{-m,p'}(\Omega)$ is a Banach space when endowed with the norm

$$\|f\|_{-m,p'} = \sup_{0 \neq v \in W_0^{m,p}(\Omega)} \frac{\langle f, v \rangle}{\|v\|_{m,p}}. \quad (1.29)$$

When $p = p' = 1/2$, we drop the p and p' indices in the notations of the spaces and norms. The reason for the notations with Hölder conjugate indices p and p' lies in the fact that the functionals in $W^{-m,p'}(\Omega)$ can be given a characterization in terms of derivatives of $L^{p'}(\Omega)$ functions.

Theorem 7. *Let p and p' be Hölder conjugate with $1 \leq p < \infty$. A distribution f belongs to $W^{-m,p'}(\Omega)$ if and only if there exist functions $f_\alpha \in L^{p'}(\Omega)$, for $|\alpha| \leq m$, such that*

$$f = \sum_{|\alpha| \leq m} \partial^\alpha f_\alpha. \quad (1.30)$$

As a particular case of the previous theorem, one also retrieves that when $\Omega \subset \mathbb{R}^N$ the well-known *Dirac delta function* $\delta \in H^{-1}(\Omega)$ is the distributional derivative of the *Heaviside unit step function* belonging to $L^2(\Omega)$.

We will also make use of the *trace spaces* of Sobolev spaces. These spaces provide a generalization for functions in $W^{m,p}(\Omega)$ of the concept of restriction to boundary values that is straightforward for smooth functions. As we know, the value of a function in a Lebesgue space on a set with measure zero is not determined. However, one can define a notion of *trace* of a function u by approximating it with a sequence of functions in $\mathcal{D}(\bar{\Omega})$ and by restricting these functions to the boundary. More precisely, let us consider $u \in \mathcal{D}(\bar{\Omega})$ and let us define the linear mapping $u \rightarrow \gamma_0 u$ that yields the boundary values of u . This mapping has a unique linear continuous extension as can be stated in the following theorem. We precise that the following statement requires a definition of the Sobolev spaces $W^{m,p}(\Omega)$ or $W^{m,p}(\Gamma)$ in cases when m is a real-valued index. We refer the reader to [1, 13] for a definition of these spaces.

Theorem 8. *Let Ω be a bounded open subset of \mathbb{R}^N with boundary Γ of class $\mathcal{C}^{k,1}$ for some integer $k \geq 0$. Let $p \geq 1$ and $s \geq 0$ be two real numbers such that*

$$s \leq k + 1, \quad s - \frac{1}{p} = l + \sigma \quad (1.31)$$

where $l \geq 0$ is an integer and $0 < \sigma < 1$. The mapping $u \rightarrow \gamma_0 u$ defined on $\mathcal{D}(\bar{\Omega})$ has a unique linear continuous extension from $W^{s,p}(\Omega)$ onto $W^{s-1/p,p}(\Gamma)$. Furthermore, for $s = 1$ one has

$$\ker(\gamma_0) = W_0^{1,p}(\Omega). \quad (1.32)$$

In order to prove this result one can use the fact that the space $\mathcal{D}(\bar{\Omega})$ is dense in $W^{m,p}(\Omega)$ for all integers $m \geq 0$ and real p with $1 \leq p < \infty$. Thanks to Theorem 8, we can endow the space $W^{s-1/p,p}(\Gamma)$ with a Banach norm of the type

$$\|u\|_{s-1/p,p,\Gamma} = \inf_{v \in W^{s,p}(\Omega), \gamma_0 v = u} \|v\|_{s,p,\Omega} \quad (1.33)$$

When $s = \frac{1}{p}$, the space $W^{0,p}(\Gamma) = L^p(\Gamma)$ can be endowed with a more familiar norm

$$\|u\|_{0,p,\Gamma} = \left(\int_{\Gamma} |u|^p d\Gamma \right)^{1/p}. \quad (1.34)$$

Boldface notation will be adopted for trace spaces of vector-valued functions. We also recall a generalized form of Theorem 5 which can be formulated for functions with vanishing trace only on part of the boundary. The following result can be useful in the analysis of problems with Dirichlet boundary conditions enforced only on a portion of the boundary.

Theorem 9 (Generalized Poincaré inequality). *Let Γ_0 be a portion of Γ with strictly positive measure. Then, for any real $r > 1$, $|\cdot|_{1,r}$ and $\|\cdot\|_{1,r}$ are two equivalent norms on the space*

$$W_{\Gamma_0}^{1,r}(\Omega) = \{u \in W^{1,r}(\Omega); u|_{\Gamma_0} = 0\}. \quad (1.35)$$

1.3 Abstract frameworks for mixed problems

In this section we outline two general abstract frameworks for the analysis of linear and nonlinear mixed problems. These abstract settings are often used for studying problems like the Stokes and Navier-Stokes equations. The MHD system considered in this thesis fits into the nonlinear abstract framework described in the following.

1.3.1 Abstract linear mixed problem

Let us first introduce an abstract linear problem of *mixed* type. Mixed problems occur when the unknown functions belong to different function spaces. They often arise in conjunction with the enforcement of constraints on the unknowns by the use of Lagrange multipliers. Let X and M be two Hilbert spaces with norms $\|\cdot\|_X$ and $\|\cdot\|_M$ respectively. Let X' and M' be their corresponding dual spaces endowed with the dual norms

$$\|f\|_{X'} = \sup_{u \in X} \frac{\langle f, u \rangle}{\|u\|_X}, \quad \|f\|_{M'} = \sup_{u \in M} \frac{\langle f, u \rangle}{\|u\|_M}, \quad (1.36)$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing between each space and the corresponding dual. We define the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ as

$$a(u, v) : X \times X \rightarrow \mathbb{R}, \quad (1.37)$$

$$b(u, \lambda) : X \times M \rightarrow \mathbb{R}. \quad (1.38)$$

We recall that the bilinear property of a form $f : X_1 \times X_2 \rightarrow \mathbb{R}$ means that

$$f(\alpha_1 u_1 + \alpha_2 u_2, v) = \alpha_1 f(u_1, v) + \alpha_2 f(u_2, v) \quad \forall u_1, u_2 \in X_1, v \in X_2, \alpha_1, \alpha_2 \in \mathbb{R},$$

$$f(u, \beta_1 v_1 + \beta_2 v_2) = \beta_1 f(u, v_1) + \beta_2 f(u, v_2) \quad \forall v_1, v_2 \in X_2, u \in X_1, \beta_1, \beta_2 \in \mathbb{R}.$$

Thanks to the continuity property, there exist two positive constants C_a and C_b such that

$$|a(u, v)| \leq C_a \|u\|_X \|v\|_X, \quad (1.39)$$

$$|b(u, p)| \leq C_b \|u\|_X \|p\|_M. \quad (1.40)$$

Hence, the following definitions of norm for the forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are well-posed:

$$\|a\| = \sup_{(u,v) \in X \times X} \frac{a(u,v)}{\|u\|_X \|v\|_X}, \quad (1.41)$$

$$\|b\| = \sup_{(u,\mu) \in X \times M} \frac{b(u,\mu)}{\|u\|_X \|\mu\|_M}. \quad (1.42)$$

We now state the *mixed linear problem* that we call *Problem Q_l* as

Problem 1. (*Problem Q_l*). Given $l \in X'$ and $\chi \in M'$, find a pair $(u, \lambda) \in X \times M$ satisfying

$$a(u, v) + b(v, \lambda) = \langle l, v \rangle \quad \forall v \in X, \quad (1.43)$$

$$b(u, \mu) = \langle \chi, \mu \rangle \quad \forall \mu \in M. \quad (1.44)$$

We can now introduce the linear operators $A \in \mathcal{L}(X, X')$ and $B \in \mathcal{L}(X, M')$ (the symbol $\mathcal{L}(X, Y)$ denoting the space of linear operators from X to Y) such that

$$\langle Au, v \rangle = a(u, v) \quad \forall u, v \in X, \quad (1.45)$$

$$\langle Bu, \mu \rangle = b(u, \mu) \quad \forall u \in X, \mu \in M. \quad (1.46)$$

We indicate with B' the *adjoint* operator of B , i.e. the operator such that

$$\langle Bu, \mu \rangle = \langle u, B'\mu \rangle. \quad (1.47)$$

The problem can be restated in operator form as

$$Au + B'p = l \quad \text{in } X' \quad (1.48)$$

$$Bu = \chi \quad \text{in } M'. \quad (1.49)$$

Let $\Phi \in \mathcal{L}(X \times M, X' \times M')$ be the operator given by

$$\Phi(u, p) = (Au + B'p, Bp). \quad (1.50)$$

We say that Problem Q_l is *well-posed* if the operator Φ is an isomorphism from $X \times M$ to $X' \times M'$. In this case, the solution to Problem Q_l exists, is unique and depends continuously the data l and χ .

Now we introduce the affine space

$$V(\chi) = \{u \in X \mid b(u, \mu) = \langle \chi, \mu \rangle \quad \forall \mu \in M\}. \quad (1.51)$$

We denote $V = V(0) = \ker(B)$. We can associate to Problem Q_l the following *Problem P_l* :

Problem 2. (Problem P_l). Given $l \in X'$, find $u \in V(\chi)$ satisfying

$$a(u, v) = \langle l, v \rangle \quad \forall v \in V. \quad (1.52)$$

Clearly, if $(u, \lambda) \in X \times M$ satisfies Problem Q_l , then u is also solution of Problem P_l . We intend to determine under which conditions the converse statement holds.

If we introduce the operator $\pi \in \mathcal{L}(X', V')$ such that

$$\langle \pi f, v \rangle = \langle f, v \rangle \quad \forall f \in X', \quad \forall v \in V, \quad (1.53)$$

the corresponding operator form for Problem P_l is

$$\pi Au = \pi l \quad \text{in } V'. \quad (1.54)$$

We can now state the well-posedness result for Problem Q_l . It can be considered as a generalization of the well-known Lax-Milgram existence theorem concerning bilinear elliptic forms.

Theorem 10. *Problem Q_l is well-posed if and only if the following conditions hold:*

- i) *the operator πA is an isomorphism from V onto V' ;*
- ii) *the inf-sup condition holds, i.e.*

$$\inf_{\mu \in M} \sup_{u \in X} \frac{b(u, \mu)}{\|u\|_X \|\mu\|_M} \geq \beta. \quad (1.55)$$

When Problem Q_l is well-posed, its solution (u, λ) depends continuously on the data, namely

$$\|u\|_X + \|\lambda\|_M \leq C(\|l\|_{X'} + \|\chi\|_{M'}). \quad (1.56)$$

Property i) in Theorem 10 can be equivalently claimed in terms of the *ellipticity* or *coercivity* of the bilinear form $a(\cdot, \cdot)$ on the space V . Therefore, we have the following alternative statement of well-posedness for Problem Q_l .

Corollary 1. *Assume that the bilinear form $a(\cdot, \cdot)$ is V -elliptic, i.e. there exists a constant $\alpha > 0$ such that*

$$a(u, u) \geq \alpha \|u\|_X^2 \quad \forall u \in V. \quad (1.57)$$

Then, Problem Q_l is well posed if and only if the bilinear form $b(\cdot, \cdot)$ satisfies the inf-sup condition (1.55).

Therefore, the *inf-sup* condition determines the equivalence between Problem Q_l and Problem P_l .

By way of example, we see how the *Stokes equations* fit within the abstract linear mixed framework previously outlined. The Stokes equations are often encountered in practice to describe the motion of an incompressible fluid in the case of negligible convection. If \mathbf{u} and p denote the velocity and pressure of the fluid respectively and we enforce homogeneous Dirichlet boundary conditions for velocity, the strong form of the Stokes equations is given by:

$$\begin{aligned} -\nu\Delta\mathbf{u} + \text{grad } p &= \mathbf{f} && \text{in } \Omega, \\ \text{div } \mathbf{u} &= 0 && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{0} && \text{on } \Gamma. \end{aligned}$$

Here, ν is a parameter which is equal, in the nondimensional form of the equations, to the inverse of the Reynolds number. The term \mathbf{f} represents a general body force. A variational form of this problem can be obtained in the framework of Problem Q_l . To this end, we set the choices

$$X = \mathbf{H}_0^1(\Omega), \quad M = L_0^2(\Omega), \quad (1.58)$$

$$\|\cdot\|_X = \|\cdot\|_1, \quad \|\cdot\|_M = \|\cdot\|_0, \quad (1.59)$$

$$a(\mathbf{u}, \mathbf{v}) = \nu \int_{\Omega} \text{grad } \mathbf{u} : \text{grad } \mathbf{v} \, d\Omega, \quad (1.60)$$

$$b(\mathbf{v}, q) = \int_{\Omega} q \, \text{div } \mathbf{v} \, d\Omega, \quad (1.61)$$

$$l = \mathbf{f}, \quad \chi = 0. \quad (1.62)$$

Here L_0^2 denotes the subspace of L^2 with zero mean over Ω , namely

$$L_0^2(\Omega) = \left\{ p \in L^2(\Omega) \mid \int_{\Omega} p \, d\mathbf{x} = 0 \right\}. \quad (1.63)$$

Owing to these choices, the space V is given by

$$V = \left\{ \mathbf{v} \in \mathbf{H}_0^1(\Omega) \mid \int_{\Omega} q \, \text{div } \mathbf{v} \, d\Omega = 0 \quad \forall q \in L_0^2(\Omega) \right\}. \quad (1.64)$$

The variational Q_l form of the Stokes problem can be given as follows:

Problem 3. Given $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$, find a pair $(\mathbf{u}, p) \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle \mathbf{f}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \quad (1.65)$$

$$b(\mathbf{u}, q) = 0 \quad \forall q \in L_0^2(\Omega). \quad (1.66)$$

The corresponding P_l form is

Problem 4. Find $\mathbf{u} \in V$ such that

$$a(\mathbf{u}, \mathbf{v}) = \langle \mathbf{f}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in V. \quad (1.67)$$

Given the assumed definitions, it can be shown that all the hypotheses invoked for the abstract existence Theorem 10 are fulfilled [13]. Hence, the variational Stokes problem is well posed and the following existence theorem can be claimed.

Theorem 11. Let Ω be a bounded and connected open subset of \mathbb{R}^N with Lipschitz continuous boundary. There exists a unique pair $(\mathbf{u}, p) \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$ which is solution of Problem 3. This solution depends continuously on the data, i.e.

$$\|\mathbf{u}\|_1 + \|p\|_0 \leq C \|\mathbf{f}\|_{-1}. \quad (1.68)$$

1.3.2 Abstract nonlinear mixed problem

Let us now introduce a general *nonlinear mixed* problem. We assume the same definitions for the Hilbert spaces X and M and for the bilinear form $b(\cdot, \cdot)$ as in Section 1.3.1. The idea is to replace the bilinear form $a(\cdot, \cdot)$ with the continuous trilinear form

$$d(u, v, w) : X \times X \times X \rightarrow \mathbb{R}. \quad (1.69)$$

Then we can define the following nonlinear mixed problem:

Problem 5. (Problem Q_n). Given $l \in X'$, find a pair $(u, \lambda) \in X \times M$ satisfying

$$d(u, u, v) + b(v, \lambda) = \langle l, v \rangle \quad \forall v \in X \quad (1.70)$$

$$b(u, \mu) = 0 \quad \forall \mu \in M. \quad (1.71)$$

Hence, Problem Q_n is a nonlinear extension of Problem Q_l . We underline that, unlike (1.44), the right-hand side in (1.71) is homogeneous.

We can now introduce two linear operators $D(w) \in \mathcal{L}(X, X')$ with $w \in X$ and $B \in \mathcal{L}(X, M')$ such that

$$\langle D(w)u, v \rangle = d(w, u, v) \quad \forall u, v \in X \quad (1.72)$$

$$\langle Bu, \mu \rangle = b(u, \mu) \quad \forall u \in X, \mu \in M. \quad (1.73)$$

The problem can then be restated in operator form as

$$D(u)u + B'p = l \quad \text{in } X' \quad (1.74)$$

$$Bu = 0 \quad \text{in } M'. \quad (1.75)$$

As usual, we indicate with B' the adjoint operator of B , i.e. the operator such that

$$\langle Bu, \mu \rangle = \langle u, B'\mu \rangle . \quad (1.76)$$

By using the set $V = V(0)$ from (1.51), the problem can be formulated in another form, i.e.

Problem 6. (Problem P_n). Given $l \in X'$, find $u \in V$ satisfying

$$d(u, u, v) = \langle l, v \rangle \quad \forall v \in V . \quad (1.77)$$

In operator form, we can write this problem as

$$D(u)u = l \quad \text{in } X' . \quad (1.78)$$

The next classical abstract result (see [13]) is a useful tool for studying the existence of solutions to nonlinear variational problems. Its proof relies on the construction of a sequence of finite-dimensional Galerkin approximations of the original problem. Moreover, compactness arguments and fixed point theorems play an essential role for nonlinear problems.

Theorem 12. Consider Problem P_n . Suppose that:

- the space V is separable;
- $d(u, v, w)$ is continuous, i.e. there exists a constant C such that

$$|d(u, v, w)| \leq C \|u\|_V \|v\|_V \|w\|_V \quad \forall u, v, w \in V ; \quad (1.79)$$

- $d(u, u, v)$ is V -elliptic, i.e. there exists a constant $\alpha > 0$ such that

$$d(u, u, u) \geq \alpha \|u\|_V^2 \quad \forall u \in V ; \quad (1.80)$$

- $d(u, v, w)$ is weakly sequentially continuous on V , i.e. if for any sequence u_n in V such that

$$\lim_{n \rightarrow \infty} \langle g, u_n \rangle = \langle g, u \rangle \quad \forall g \in V' \quad (1.81)$$

then

$$\lim_{n \rightarrow \infty} d(u_n, u_n, v) = d(u, u, v) \quad \forall v \in V . \quad (1.82)$$

Then, Problem P_n has at least one solution $u \in V$. Any solution depends continuously on the data by the stability bound

$$\|u\|_V \leq \alpha^{-1} \|l\|_{V'}. \quad (1.83)$$

Furthermore, if the smallness condition $C\alpha^{-2} \|l\|_{V'} < 1$ is satisfied, Problem P_n has a unique solution $u \in V$.

Theorem 12 claims the existence of a solution for Problem P_n . If Problem P_n has a solution, then it can be proved that the initial mixed form given by Problem Q_n has a solution.

Theorem 13. *If the bilinear form $b(\cdot, \cdot)$ satisfies the condition*

$$\inf_{\mu \in M} \sup_{u \in X} \frac{b(u, \mu)}{\|u\|_X \|\mu\|_M} \geq \beta \quad (1.84)$$

for some constant $\beta > 0$, then Problem Q_n has at least one solution (u, λ) , where u is a solution of Problem P_n .

The variational form of the Navier-Stokes and MHD equations considered in this thesis can be cast in a form which fits into the above nonlinear abstract framework, as will be illustrated in Chapter 2.

1.4 Constrained nonlinear optimal control problems

The analysis of optimal control problems for nonlinear partial differential equations is a nontrivial task. In order to point out some of the most important concepts and ideas about the study of optimal control problems, we report an abstract analysis proposed in [17]. The analysis of the optimal control problems considered in this thesis is based on a similar formulation.

Let G, X and Y be reflexive Banach spaces whose norms are denoted by $\|\cdot\|_G$, $\|\cdot\|_X$ and $\|\cdot\|_Y$ respectively. Let $\langle \cdot, \cdot \rangle$ denote the duality pairing between any of the previous spaces and their corresponding dual. Let Θ , the *control set*, be a closed convex subset of G . Let Z be a subspace of Y with compact embedding. We remark that the compactness of the embedding $Z \subset Y$ plays an important role.

An *optimal control problem* is usually characterized by the minimization of a *cost functional* under the presence of *constraints*. Here we assume that the functional to be minimized has the form

$$\mathcal{J}(v, z) = \lambda \mathcal{F}(v) + \lambda \mathcal{E}(z) \quad \forall (v, z) \in Z \times \Theta. \quad (1.85)$$

The quantity v is the *state* variable, z is the *control* variable, \mathcal{F} is a functional on X , \mathcal{E} is a functional on Θ and λ is a given parameter that is assumed to belong to a compact interval $\Lambda \subset \mathbb{R}_+$.

The *constraint* equation $M(v, z) = 0$ relating the state and control variables is defined as follows. Let $N : X \rightarrow Y$ be a (possibly nonlinear) differentiable mapping, $K : \Theta \rightarrow Y$ and $T : Y \rightarrow X$ be two continuous linear operators. For any $\lambda \in \Lambda$, we define the mapping $M : X \times \Theta \rightarrow X$ by

$$M(v, z) = v + \lambda T N(v) + \lambda T K(z) \quad \forall (v, z) \in X \times \Theta. \quad (1.86)$$

With these definitions we now consider the constrained minimization problem

$$\min_{(v, z) \in X \times \Theta} \mathcal{J}(v, z) \quad \text{subject to} \quad M(v, z) = 0. \quad (1.87)$$

We seek a *global* minimizer (v^*, z^*) with respect to the set

$$C = \{(v, z) \in X \times \Theta : M(v, z) = 0\}. \quad (1.88)$$

Although we can show that a global minimizer for (1.87) exists, in practice one can only determine *local* minima, i.e. points $(u, g) \in X \times \Theta$ such that, for some $\epsilon > 0$,

$$\mathcal{J}(u, g) \leq \mathcal{J}(v, z) \quad \forall (v, z) \in C \quad \text{and} \quad \|u - v\|_X \leq \epsilon. \quad (1.89)$$

The first step in the analysis of an optimal control problem is to prove the existence of a solution to the problem, i.e. the existence of a so-called *optimal solution*. Let us introduce a first set of hypotheses that will be invoked to this purpose. We have (the symbol \rightharpoonup denotes weak convergence):

- (H1) $\inf_{v \in X} \mathcal{F}(v) > -\infty$;
- (H2) there exist constants $\alpha, \beta > 0$ such that $\mathcal{E}(z) \geq \alpha \|z\|^\beta \quad \forall z \in \Theta$;
- (H3) there exists a $(v, z) \in X \times \Theta$ satisfying $M(v, z) = 0$;
- (H4) if $u^{(n)} \rightharpoonup u \in X$ and $g^{(n)} \rightharpoonup g \in G$ where $\{(u^{(n)}, g^{(n)})\} \in X \times \Theta$, then $Nu^{(n)} \rightharpoonup Nu \in X$ and $Kg^{(n)} \rightharpoonup Kg \in Y$;
- (H5) $\mathcal{J}(\cdot, \cdot)$ is weakly lower semicontinuous on $X \times \Theta$, i.e., for every sequence $\{(u^{(n)}, g^{(n)})\}$ that is weakly convergent to some $(u, g) \in X \times \Theta$, one has

$$\mathcal{J}(u, g) \leq \liminf_{n \rightarrow \infty} \mathcal{J}(u^{(n)}, g^{(n)}); \quad (1.90)$$

- (H6) if $\{(u^{(n)}, g^{(n)})\} \in X \times \Theta$ is such that $\{\mathcal{F}(u^{(n)})\}$ is a bounded set in \mathbb{R} and $M(u^{(n)}, g^{(n)}) = 0$, then $\{u^{(n)}\}$ is a bounded set in X .

A second set of hypotheses can be introduced in order to justify the use of Lagrange multiplier principles and to derive an optimality system from which optimal states and controls can be determined. This second set is as follows:

(H7) for each $z \in \Theta$, $\mathcal{J}(v, z)$ and $\mathcal{M}(v, z)$ are Fréchet differentiable with respect to v ;

(H8) the mapping $z \mapsto \mathcal{E}(z)$ is convex, i.e.

$$\mathcal{E}(\gamma z_1 + (1-\gamma)z_2) \leq \gamma \mathcal{E}(z_1) + (1-\gamma)\mathcal{E}(z_2), \quad \forall z_1, z_2 \in \Theta, \forall \gamma \in [0, 1]; \quad (1.91)$$

(H9) for $v \in X$, $N'(v)$ maps X into Z (i.e., $\text{Range}(N') = Z$).

Let us use hypotheses (H1)-(H6) to establish that optimal solutions exist.

Theorem 14. *Given the minimization problem (1.87) for the functional (1.85) and the constraint defined by (1.86), assume that hypotheses (H1)-(H6) hold. Then, there exists a solution to this minimization problem.*

Proof. We follow the proof as in [17]. By assumption (H3), there exists at least one element of $X \times \Theta$ that satisfies the constraint. Thus, we may choose a *minimizing sequence* $\{(u^{(n)}, g^{(n)})\}$ in C for $\mathcal{J}(\cdot, \cdot)$, i.e. a sequence such that

$$\lim_{n \rightarrow \infty} \mathcal{J}(u^{(n)}, g^{(n)}) = \inf_{(v,z) \in C} \mathcal{J}(v, z).$$

By (H1) and (H2), the boundedness of $\{\mathcal{J}(u^{(n)}, g^{(n)})\}$ implies the boundedness of the sequences $\|g^{(n)}\|_G$ and $\{\mathcal{F}(u^{(n)})\}$. Then, by (H6), we deduce that $\|u^{(n)}\|_X$ is bounded. Thus, we may extract a weakly convergent subsequence $\{(u^{(n)}, g^{(n)})\}$ to some element $(u, g) \in X \times G$. Since Θ is convex, we have that $g \in \Theta$. We now have to pass the limit inside the functional $\mathcal{J}(\cdot, \cdot)$ in order to show that (u, g) satisfies the constraint. By (H4), we have that, for all $f \in X^*$,

$$\begin{aligned} \lim_{n \rightarrow \infty} \langle TNu^{(n)}, f \rangle &= \lim_{n \rightarrow \infty} \langle Nu^{(n)}, T^*f \rangle = \langle Nu, T^*f \rangle = \langle TNu, f \rangle \\ \lim_{n \rightarrow \infty} \langle TKg^{(n)}, f \rangle &= \lim_{n \rightarrow \infty} \langle Kg^{(n)}, T^*f \rangle = \langle Kg, T^*f \rangle = \langle TKg, f \rangle. \end{aligned}$$

Thus, we have

$$0 = \lim_{n \rightarrow \infty} \langle M(u^{(n)}, g^{(n)}), f \rangle = \langle u + \lambda TN(u) + \lambda TK(g), f \rangle \quad \forall f \in X^*,$$

i.e. $M(u, g) = 0$. Therefore, the subsequence whose elements satisfy the constraint has a weak limit that also satisfies the constraint. Finally, we can show that (u, g) minimizes the functional. In fact, by the weak lower semicontinuity of \mathcal{J} , given by (H5), we conclude that

$$\inf_{(v,z) \in C} \mathcal{J}(v, z) \leq \mathcal{J}(u, g) \leq \liminf_{n \rightarrow \infty} \mathcal{J}(u^{(n)}, g^{(n)}) = \inf_{(v,z) \in C} \mathcal{J}(v, z).$$

Indeed, (u, g) is a minimizer. □

Now let us consider the additional hypotheses (H7)-(H9) to show that the Lagrange multiplier rule can be used to turn the constrained minimization problem into an unconstrained one. We remark that the existence of an optimal solution required by the following result is guaranteed by Theorem 14. Also, we denote by $\sigma(-TN'(v))$ the spectrum of the operator $-TN'(v)$.

Theorem 15. *Let $\lambda \in \Lambda$ be given. Assume that hypotheses (H1)-(H9) hold. Let $(u, g) \in X \times \Theta$ be an optimal solution satisfying (1.89). Then there exists $k \in \mathbb{R}$ and $\mu \in X^*$ not both equal to zero such that*

$$k \langle \mathcal{J}_u(u, g), w \rangle - \langle \mu, M_u(u, g) \cdot w \rangle = 0 \quad \forall w \in X \quad (1.92)$$

$$\min_{z \in \Theta} \mathcal{L}(u, z, \mu, k) = \mathcal{L}(u, g, \mu, k), \quad (1.93)$$

where $\mathcal{L}(u, z, \mu, k) = k\mathcal{J}(u, z) - \langle \mu, M(u, z) \rangle$. Furthermore, if $\frac{1}{\lambda} \notin \sigma(-TN'(u))$, we may choose $k = 1$, so that there exists $\mu \in X^*$ that satisfies

$$\langle \mathcal{J}_u(u, g), w \rangle - \langle \mu, M_u(u, g) \cdot w \rangle = 0 \quad \forall w \in X \quad (1.94)$$

$$\min_{z \in \Theta} \mathcal{L}(u, z, \mu, 1) = \mathcal{L}(u, g, \mu, 1). \quad (1.95)$$

Proof. See [17] and [36]. □

So far, no specific assumptions were made on the control variable g in the functional or in the constraint and the control set Θ was only assumed to be a closed and convex subset of G without any topological structure. If we now turn to the case $\Theta = G$, we can also consider differentiation of the functional $\mathcal{E}(z)$. An additional hypothesis may then be assumed, namely

(H10) $\Theta = G$ and the mapping $z \mapsto \mathcal{E}(z)$ is Fréchet differentiable on G .

In this case, we can give a more practical expression than (1.93) for the variation with respect to the control variable.

Theorem 16. *Let $\lambda \in \Lambda$ be given. Assume that hypotheses (H1)-(H10) hold. Let $(u, g) \in X \times G$ be an optimal solution satisfying (1.89). Then there exists $k \in \mathbb{R}$ and $\mu \in X^*$ not both equal to zero such that*

$$k \langle \mathcal{J}_u(u, g), w \rangle - \langle \mu, (I + \lambda TN'(u)) \cdot w \rangle = 0 \quad \forall w \in X \quad (1.96)$$

$$k \langle \mathcal{E}'(g), z \rangle - \langle \mu, TKz \rangle = 0 \quad \forall z \in G. \quad (1.97)$$

Furthermore, if $\frac{1}{\lambda} \notin \sigma(-TN'(u))$, we may choose $k = 1$, so that there exists $\mu \in X^*$ that satisfies

$$\langle \mathcal{J}_u(u, g), w \rangle - \langle \mu, (I + \lambda TN'(u)) \cdot w \rangle = 0 \quad \forall w \in X \quad (1.98)$$

$$\langle \mathcal{E}'(g), z \rangle - \langle \mu, TKz \rangle = 0 \quad \forall z \in G. \quad (1.99)$$

Equation (1.98) is the so-called *adjoint equation* of the optimal control problem, while Equation (1.99) is usually referred to as the *optimality condition*. These equations represent a *first-order necessary condition* for an optimal solution. Together with the state equation given by $M(u, g) = 0$, they form the so-called *optimality system*. A solution of the optimality system is then a candidate solution for the original optimal control problem.

Chapter 2

Lifting function approach for boundary optimal control

We illustrate in this chapter the mathematical formulation of the optimal control problem considered in this thesis. First, we give a survey on the literature concerning the MHD equations and related optimal control problems in Section 2.1. Section 2.2 describes the lifting function approach that we propose for the study of boundary optimal control problems for the MHD equations. In Section 2.3, a weak formulation of the MHD state equations suitable for our purposes is discussed. The existence of a solution to these equations is proved in Section 2.4. Section 2.5 deals with the mathematical formulation of the optimal control problem. Existence of a solution for this problem is shown in Section 2.5.1. The technique of Lagrange multipliers, the first-order necessary condition and the resulting optimality system are illustrated in Sections 2.5.2 and 2.5.3. A solution of the optimality system is a candidate solution for the optimal control problem. Section 3.3 is devoted to the finite element approximation of the optimality system. A gradient algorithm for a robust numerical solution of this system is illustrated in Section 3.4.

2.1 Optimal control of the MHD equations

The study of control problems for fluid flows interacting with magnetic fields is a very attractive feature in many science and engineering settings such as fusion technology, fission nuclear reactors cooled using liquid metals, aluminum casting in metallurgy and crystal growth in semiconductor industry [9, 27]. Since the magnetic field can provide a very effective means for obtaining flow control in applications involving electrically conductive fluids, it is very interesting to study such a possibility in order to achieve desired flow configurations. An externally imposed magnetic field may interact with the fluid and modify the Lorentz force,

so that one can achieve practical goals such as steering the velocity profile to a desired one or minimizing quantities of physical interest, like drag or vorticity.

The behavior of fluid flows interacting with magnetic fields is described by the equations of Magnetohydrodynamics (MHD). Numerous formulations are proposed and analyzed in literature for these equations [14, 20, 30, 34, 32]. Attention is focused on various aspects such as physical modeling, mathematical formulation, numerical approximation and computational procedures. It is well-known that the MHD problems can be formulated in terms of different sets of unknowns. The mechanical behavior of the fluid flow is often described by the Navier-Stokes equations in the velocity and pressure unknowns [20, 34, 30, 23]. A greater variety occurs in the choice of the electromagnetic variables, for which one combination or another of quantities such as the magnetic field, the current density, the electric field, and the electric potential is used [14]. In [20] and [34], two analyses on the existence, uniqueness and finite element approximation of the MHD equations are carried out. The paper [20] is based on a formulation involving the commonly used velocity, pressure and magnetic field variables. An H^1 -conforming finite element approximation for the magnetic field is considered, which is convergent on standard convex domains or with $C^{1,1}$ boundary smoothness assumption. An analysis for the case of general Lipschitz polyhedron is given in [34], accounting for possible low regularity of the magnetic field on non-convex domains. This generalization is performed with the introduction of an auxiliary Lagrange multiplier variable for the divergence-free constraint on the magnetic field, together with the use of the $H(\text{curl})$ -conforming family of Nédélec finite element spaces. In [23], convergent finite element approximations on non-convex domains are instead obtained with a modification of the magnetic bilinear form in the weak formulation via a weighted regularization approach, and no auxiliary multiplier is introduced. Another particular choice in the weak formulation leads to stabilized finite element methods, analyzed in [11], which permit to circumvent the requirements of the inf-sup condition.

Although the literature on optimal control problems for the equations of Magnetohydrodynamics is not particularly broad to our best knowledge, a heterogeneous range of approaches and methodologies is covered. Amongst the variety of possibilities, some general classifications can be typically made. In particular, *stationary* and *non-stationary* problems can be distinguished, as well as *distributed* and *boundary* problems. The former distinction is clearly related to the dependency on time of the state variables, while the latter identifies situations in which the control variable is applied either in the interior or on the boundary of the domain, respectively. In [14], a stationary optimal control problem is taken into account following a velocity-current formulation proposed in [31], which allows to consider the MHD problem on a bounded domain without enforcing artificial

shielding boundary conditions. External or injected currents are considered along with magnetic fields as control physical mechanisms; these control quantities are assumed in lumped parameter form for a more direct link with applications. The convergence of an operator splitting scheme for the numerical solution of the MHD state equations is also studied. In [25] and [24], two analyses of boundary optimal control problems for the stationary MHD equations with infinite-dimensional controls are carried out. Focus is given on the theoretical formulation but no computational algorithms are considered nor results given. In [25] an optimal control problem is studied with a desired electrical current as target and a boundary electrical potential as control. Both the magnetic field and the electric potential are considered as electromagnetic variables. In [24], the normal magnetic field on part of the boundary is taken as control and the objectives are desired velocity and magnetic field volume distributions. As a common practice, the cost functional is of the tracking type; a penalty term is added in order to limit the size of controls and avoid the use of inequality constraints. Some computational results for different objective functionals are reported in [26] in the case of boundary normal electrical current as control. Time-dependent analyses are treated in [21, 22, 32]. In [21], a theoretical formulation of a control problem for the time-periodic MHD equations with volume periodic control inputs is considered, and semidiscrete time approximations are defined. The paper [22] considers a modified form of the Navier-Stokes equations due to Ladyzhenskaya, which guarantees global unique solvability for three-dimensional domains. In [32], focus is devoted to a fast and reliable real-time algorithm using a reduced order POD model of an MHD flow controlled with boundary potential.

In this work we consider a boundary optimal control problem for the stationary MHD equations and we propose a new approach for its solution, based on the transformation of the boundary control into an extended distributed one. Compared to the case of distributed controls, standard approaches for treating boundary control problems are in fact not entirely straightforward to implement numerically. For example, boundary controls involve normal or tangential components of the magnetic field so that the direct implementation of such controls causes substantial difficulties on general domains. Furthermore, such implementations often lead to unnecessarily smooth controls [19, 30] or involve penalization terms that can adversely affect accuracy and also the conditioning of discretized systems. The boundary optimal control problem under consideration is characterized by a cost functional representing the error between the fluid velocity and a desired velocity profile. The control variable is given by the boundary values of the magnetic field. In order to deal with this type of boundary control, we choose to split the magnetic field in the domain into two parts: a lifting function of the magnetic field boundary conditions and an auxiliary field with homogeneous boundary conditions. The

lifting function can be considered as an extension of the magnetic field from the boundary to the interior of the domain. The optimal solution is then searched for by exploring all possible extended functions. This approach brings several advantages from both a theoretical and a computational point of view [3, 19, 30]. The boundary control algorithm can be solved by robust distributed control techniques over the inner part of the domain. Furthermore, the optimal boundary controls can be determined in natural half-integer Sobolev spaces as traces of the optimal extended functions. Also, both Dirichlet and Neumann boundary conditions can be taken into account with this approach.

In the following sections, we provide a mathematical description of the optimal control problem at hand and we illustrate the proposed lifting function approach.

2.2 Description of the optimal control problem

The optimal control problem we consider consists of a *cost* or *objective* functional, a set of *control* functions, and a set of state equations that act as *constraints*.

Let $\Omega \subset \mathbb{R}^3$ denote an open bounded connected domain with $C^{1,1}$ boundary Γ . We denote by Γ_1 a subset of Γ with positive surface measure. For the constraint or state equations, we have the steady-state MHD model given in nondimensional form by [10, 29, 33]

$$\left\{ \begin{array}{l} -\frac{1}{Re} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - S_1 (\nabla \times \mathbf{B}) \times \mathbf{B} - \mathbf{f} = \mathbf{0} \\ \nabla \cdot \mathbf{u} = 0 \\ \frac{1}{Re_m} \nabla \times (\nabla \times \mathbf{B}) - \nabla \times (\mathbf{u} \times \mathbf{B}) + \nabla \sigma = \mathbf{0} \\ \nabla \cdot \mathbf{B} = 0 \end{array} \right. \quad \text{on } \Omega. \quad (2.1)$$

Here, \mathbf{u} and p denote the fluid velocity and pressure, respectively, \mathbf{B} the magnetic field, σ the Lagrange multiplier associated to the divergence-free constraint of the magnetic field, $Re = \rho UL/\mu$ the viscous Reynolds number, $Re_m = \mu_0 \sigma_c UL$ the magnetic Reynolds number, $H_m = BL\sqrt{\sigma_c/\mu}$ the Hartmann number, and $S_1 = H_m^2/Re Re_m$. We denote by U , B and L the reference values for velocity, magnetic field and length. The fluid density, dynamic viscosity, magnetic permeability and electrical conductivity are indicated as ρ , μ , μ_0 , and σ_c , respectively. The MHD system (2.1) is completed with appropriate boundary conditions for the velocity and magnetic fields which are discussed below.

By using the well-known vector identities

$$\begin{aligned} (\nabla \times \mathbf{B}) \times \mathbf{B} &= (\mathbf{B} \cdot \nabla) \mathbf{B} - \frac{1}{2} \nabla B^2, \\ \nabla(\mathbf{A} \cdot \mathbf{B}) &= (\mathbf{A} \cdot \nabla) \mathbf{B} + (\mathbf{B} \cdot \nabla) \mathbf{A} + \mathbf{A} \times (\nabla \times \mathbf{B}) + \mathbf{B} \times (\nabla \times \mathbf{A}), \\ \nabla \times (\mathbf{A} \times \mathbf{B}) &= \mathbf{A}(\nabla \cdot \mathbf{B}) - \mathbf{B}(\nabla \cdot \mathbf{A}) + (\mathbf{B} \cdot \nabla) \mathbf{A} - (\mathbf{A} \cdot \nabla) \mathbf{B}, \end{aligned}$$

the system (2.1) takes the form

$$\begin{cases} -\frac{1}{Re} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla r - S_1 (\mathbf{B} \cdot \nabla) \mathbf{B} - \mathbf{f} = \mathbf{0} \\ \nabla \cdot \mathbf{u} = 0 \\ \frac{1}{Re_m} \nabla \times (\nabla \times \mathbf{B}) + (\mathbf{u} \cdot \nabla) \mathbf{B} - (\mathbf{B} \cdot \nabla) \mathbf{u} + \nabla \sigma = \mathbf{0} \\ \nabla \cdot \mathbf{B} = 0 \end{cases} \quad \text{on } \Omega, \quad (2.2)$$

where the modified pressure r is defined as

$$r = p + \frac{S_1}{2} |\mathbf{B}|^2. \quad (2.3)$$

To the system (2.2) we append the boundary conditions

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_1, \quad (2.4)$$

$$\boldsymbol{\tau}(\mathbf{u}, r) = \mathbf{t} \quad \text{on } \Gamma \setminus \Gamma_1, \quad (2.5)$$

$$\mathbf{B} = \boldsymbol{\Phi}_0 \quad \text{on } \Gamma, \quad (2.6)$$

where

$$\boldsymbol{\tau}(\mathbf{u}, r) = -\frac{1}{Re} \frac{\partial \mathbf{u}}{\partial n} + r \mathbf{n} + \frac{1}{2} (\mathbf{u} \cdot \mathbf{n}) \mathbf{u}. \quad (2.7)$$

Due to the divergence-free constraints on the velocity and magnetic fields, the data \mathbf{g} and $\boldsymbol{\Phi}_0$ must satisfy the compatibility conditions

$$\int_{\Gamma} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0, \quad \int_{\Gamma} \boldsymbol{\Phi}_0 \cdot \mathbf{n} \, d\Gamma = 0. \quad (2.8)$$

The boundary conditions (2.4) and (2.5) are standard velocity and stress boundary conditions, respectively, for the velocity \mathbf{u} and modified pressure r . The definition (2.7) can be physically interpreted as a *generalized stress* at the boundary that takes into account both dissipative (due to viscous dissipation) and conservative energy (due to mechanical pressure, magnetic energy and kinetic energy) contributions at the boundary. The boundary condition (2.6) is not the most commonly

used boundary condition for the magnetic field \mathbf{B} . Instead, one of the pairs of boundary conditions

$$\mathbf{B} \cdot \mathbf{n} = \Phi_1 \quad \text{and} \quad \mathbf{E} \times \mathbf{n} = \Psi_1 \quad \text{on } \Gamma \quad (2.9)$$

or

$$\mathbf{E} \cdot \mathbf{n} = \Psi_2 \quad \text{and} \quad \mathbf{B} \times \mathbf{n} = \Phi_2 \quad \text{on } \Gamma, \quad (2.10)$$

where

$$\mathbf{E} = \frac{1}{Re_m} \nabla \times \mathbf{B} - \mathbf{u} \times \mathbf{B} \quad (2.11)$$

denotes the electric field, is usually applied on Γ . However, (2.6) (or equivalently, boundary conditions on the pair $\mathbf{B} \cdot \mathbf{n}$ and $\mathbf{B} \times \mathbf{n}$) is, for us, an expedient boundary condition to use in the boundary optimal control problem we consider; its use does not engender any loss of generality, as will become evident later.

The *controls* in the optimal control problem we consider can be chosen to be the data pair Φ_1 and Ψ_1 in (2.9) or the data pair Φ_2 and Ψ_2 in (2.10) or even the data Φ_0 in (2.6). However, we do not deal with these control choices directly. Instead, we determine an optimal *distributed* magnetic field control from which any and all of the possible boundary controls can be determined by restriction. To this end, we introduce the decomposition of the magnetic field given by

$$\mathbf{B} = \mathbf{b} + \mathbf{B}_e \quad \text{on } \Omega, \quad (2.12)$$

with

$$\nabla \cdot \mathbf{B}_e = 0 \quad \text{in } \Omega \quad \text{and} \quad \mathbf{B}_e = \mathbf{B} = \Phi_0 \quad \text{on } \Gamma, \quad (2.13)$$

so that \mathbf{b} is solenoidal and satisfies homogeneous boundary conditions on Γ , i.e., we have $\nabla \cdot \mathbf{b} = 0$ in Ω and $\mathbf{b} = \mathbf{0}$ on Γ . The function \mathbf{B}_e is the so-called *lifting function* of the Dirichlet boundary conditions on the magnetic field. Note that the fourth equation in (2.1) or (2.2) implies that $\int_{\Gamma} \mathbf{B} \cdot \mathbf{n} \, d\Gamma = 0$ so that, from (2.12) and (2.13), the compatibility condition on the boundary datum Φ_0 , namely

$$\int_{\Gamma} \Phi_0 \cdot \mathbf{n} \, d\Gamma = \int_{\Gamma} \mathbf{B}_e \cdot \mathbf{n} \, d\Gamma = \int_{\Omega} \nabla \cdot \mathbf{B}_e \, dx = 0, \quad (2.14)$$

is automatically satisfied thanks to the divergence-free condition.

In view of the previous decomposition, we can say that \mathbf{u} , r , \mathbf{b} and σ are the *state functions* in our control problem and \mathbf{B}_e is the *distributed magnetic field control function*. Substitution of (2.12) into (2.2) results in the final form of the

constraint or state equations

$$\left\{ \begin{array}{l} -\frac{1}{Re} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla r \\ \qquad \qquad \qquad - S_1((\mathbf{b} + \mathbf{B}_e) \cdot \nabla)(\mathbf{b} + \mathbf{B}_e) = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \\ \frac{1}{Re_m} \nabla \times (\nabla \times (\mathbf{b} + \mathbf{B}_e)) + (\mathbf{u} \cdot \nabla)(\mathbf{b} + \mathbf{B}_e) \\ \qquad \qquad \qquad - ((\mathbf{b} + \mathbf{B}_e) \cdot \nabla) \mathbf{u} + \nabla \sigma = \mathbf{0} \\ \nabla \cdot \mathbf{b} = 0 \end{array} \right. \quad \text{on } \Omega \quad (2.15)$$

along with

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_1, \quad (2.16)$$

$$\boldsymbol{\tau}(\mathbf{u}, r) = \mathbf{t} \quad \text{on } \Gamma \setminus \Gamma_1, \quad (2.17)$$

$$\mathbf{b} = \mathbf{0} \quad \text{on } \Gamma, \quad (2.18)$$

which candidate optimal states \mathbf{u} , r , \mathbf{b} and σ and optimal controls \mathbf{B}_e are required to satisfy. Here, \mathbf{f} and \mathbf{t} are given data functions, although, using well-known approaches [15], these could be used as control functions as well.

Clearly, for every solution $(\mathbf{u}, r, \mathbf{B}, \sigma)$ of the MHD state problem (2.2), one can find a lifting function \mathbf{B}_e as above and a quadruple $(\mathbf{u}, r, \mathbf{b}, \sigma)$ that solves the problem (2.15). Of course, the lifting function is not uniquely defined, so that different lifting functions will produce different solutions $(\mathbf{u}, r, \mathbf{b}, \sigma)$ of (2.15). However, the sum $\mathbf{B} = \mathbf{b} + \mathbf{B}_e$ is uniquely determined and, along with r and σ , solves (2.2).

The goal of the optimal control problem we consider is to match, as well as possible, the velocity \mathbf{u} to a prescribed velocity \mathbf{u}_d . To this end, we define the *cost* or *objective functional*

$$\mathcal{J}(\mathbf{u}, \mathbf{B}_e) = \frac{\alpha}{2} \int_{\Omega} |\mathbf{u} - \mathbf{u}_d|^2 d\mathbf{x} + \frac{\beta}{2} \int_{\Omega} |\mathbf{B}_e|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\nabla \mathbf{B}_e|^2 d\mathbf{x}, \quad (2.19)$$

where α , β and γ denote positive constants. The first term in (2.19) embodies the objective of the control problem. The other terms are added to limit the cost of control and regularize it. The values of the constants α , β and γ can be chosen to adjust the relative importance of the terms in the functional. Large values of α compared to β and γ allow for better velocity matching whereas small values of α result in poorer velocity matching. The presence of the γ -term is motivated by the need to have a sufficiently smooth control function so that the optimal control problem is well posed.

Now that we have defined all its ingredients, we can state the optimal control problem we intend to study.

Problem 7. Find an optimal control \mathbf{B}_e^* and an optimal state $\{\mathbf{u}^*, r^*, \mathbf{b}^*, \sigma^*\}$ such that the functional $\mathcal{J}(\cdot, \cdot)$ given in (2.19) is minimized and the state system (2.15)–(2.18) is satisfied.

Once the optimal state and control functions have been determined, optimal controls of other types can be determined. For example, if we wish to know the optimal boundary control of type (2.9), we have, from (2.11), (2.12), (2.16), and (2.18),

$$\begin{cases} \Phi_1^* = \mathbf{B}^* \cdot \mathbf{n} = \mathbf{B}_e^* \cdot \mathbf{n} \\ \Psi_1^* = \mathbf{E}^* \times \mathbf{n} = \left(\frac{1}{Re_m} \nabla \times (\mathbf{b}^* + \mathbf{B}_e^*) - \mathbf{u}^* \times \mathbf{B}_e^* \right) \times \mathbf{n} \end{cases} \quad \text{on } \Gamma, \quad (2.20)$$

whereas optimal boundary controls of the type (2.10) would be determined as

$$\begin{cases} \Psi_2^* = \mathbf{B}^* \times \mathbf{n} = \mathbf{B}_e^* \times \mathbf{n} \\ \Phi_2^* = \mathbf{E}^* \cdot \mathbf{n} = \left(\frac{1}{Re_m} \nabla \times (\mathbf{b}^* + \mathbf{B}_e^*) - \mathbf{u}^* \times \mathbf{B}_e^* \right) \cdot \mathbf{n} \end{cases} \quad \text{on } \Gamma. \quad (2.21)$$

Of course, optimal boundary controls of the type (2.6) are simply determined as $\Phi_0^* = \mathbf{B}^* = \mathbf{B}_e^*$ on Γ . Thus we see that obtaining the single optimal distributed control \mathbf{B}_e^* along with the optimal state variables enables one to determine optimal controls of other types.

We also remark that in practical situations the boundary control given by Φ_0 (or by the pairs (Φ_1, Ψ_1) or (Φ_2, Ψ_2)) acts only on part of the boundary $\Gamma_c \subset \Gamma$ [15]. In those cases, the corresponding distributed control function \mathbf{B}_e is fixed on the remaining portion $\Gamma_d = \Gamma \setminus \Gamma_c$. Thus, we distinguish the control and fixed parts of the boundary condition for the magnetic field as $\Phi_{0,c}$ on Γ_c and $\Phi_{0,d}$ on Γ_d (and similarly for the other boundary conditions).

2.3 Weak formulation of the MHD state problem

In order to formulate the optimal control problem under consideration we introduce some function spaces. We denote with $H^m(\Omega) = W^{m,2}(\Omega)$ and $\mathbf{H}^m(\Omega) = \mathbf{W}^{m,2}(\Omega)$ the usual scalar and vector-valued Sobolev spaces endowed with the standard norm $\|\cdot\|_m$ and by $\mathbf{H}^0(\Omega)$ the space of square-integrable functions $\mathbf{L}^2(\Omega)$ with norm $\|\cdot\| = \|\cdot\|_0$. The scalar product in $\mathbf{L}^2(\Omega)$ will be denoted by (\cdot, \cdot) . Let $\mathbf{H}_0^m(\Omega)$ denote the closure of $\mathcal{C}_0^\infty(\Omega)$ with respect to the norm $\|\cdot\|_m$ and $\mathbf{H}^{-m}(\Omega)$ denote the dual space of $\mathbf{H}_0^m(\Omega)$. The dual space of $\mathbf{H}^1(\Omega)$ is denoted by $\mathbf{H}^1(\Omega)^*$. We

define the spaces $\mathbf{V}(\Omega)$, $\mathbf{V}_0(\Omega)$ and $L_0^2(\Omega)$ as

$$\mathbf{V}(\Omega) = \{\mathbf{u} \in \mathbf{H}^1(\Omega) \mid \nabla \cdot \mathbf{u} = 0\}, \quad (2.22)$$

$$\mathbf{V}_0(\Omega) = \{\mathbf{u} \in \mathbf{H}_0^1(\Omega) \mid \nabla \cdot \mathbf{u} = 0\}, \quad (2.23)$$

$$L_0^2(\Omega) = \{p \in L^2(\Omega) \mid \int_{\Omega} p \, d\mathbf{x} = 0\}. \quad (2.24)$$

Also, given any subset $\Gamma_s \subset \Gamma$, we denote the space $\mathbf{H}_{\Gamma_s}^1(\Omega)$ as

$$\mathbf{H}_{\Gamma_s}^1(\Omega) = \{\mathbf{u} \in \mathbf{H}^1(\Omega) \mid \mathbf{u} = 0 \text{ on } \Gamma_s\}. \quad (2.25)$$

The trace operator acting on a function or on a function space is denoted by γ_0 , i.e. $\gamma_0 \mathbf{f} := \mathbf{f}|_{\Gamma}$. For smooth functions, the trace operator simply restricts the function to its boundary values. The trace space of $\mathbf{H}^1(\Omega)$ is denoted by $\mathbf{H}^{1/2}(\Gamma)$ and its dual by $\mathbf{H}^{-1/2}(\Gamma)$.

For all functions $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega)$ and $q \in L_0^2(\Omega)$ we introduce the bilinear forms

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} \, d\mathbf{x}, \quad (2.26)$$

$$a_m(\mathbf{u}, \mathbf{v}) = \int_{\Omega} (\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) \, d\mathbf{x}, \quad (2.27)$$

$$d(\mathbf{v}, q) = - \int_{\Omega} q \nabla \cdot \mathbf{v} \, d\mathbf{x}, \quad (2.28)$$

and the trilinear forms

$$\tilde{c}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, d\mathbf{x}, \quad (2.29)$$

$$\begin{aligned} c(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \frac{1}{2} \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, d\mathbf{x} - \frac{1}{2} \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{w} \cdot \mathbf{v} \, d\mathbf{x} = \\ &= \frac{1}{2} (\tilde{c}(\mathbf{u}, \mathbf{v}, \mathbf{w}) - \tilde{c}(\mathbf{u}, \mathbf{w}, \mathbf{v})). \end{aligned} \quad (2.30)$$

The form c is the usual anti-symmetric form for the nonlinear term of the Navier-Stokes equations [34, 35]. We recall the following result.

Lemma 1. *Let $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega)$. Then*

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -c(\mathbf{u}, \mathbf{w}, \mathbf{v}) \quad (2.31)$$

and

$$\tilde{c}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -\tilde{c}(\mathbf{u}, \mathbf{w}, \mathbf{v}) + \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n})(\mathbf{w} \cdot \mathbf{v}) \, ds. \quad (2.32)$$

Let $\mathbf{u} \in \mathbf{H}^1(\Omega)$ with $\nabla \cdot \mathbf{u} = 0$ and $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ , $\mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega)$. Then

$$\tilde{c}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -\tilde{c}(\mathbf{u}, \mathbf{w}, \mathbf{v}). \quad (2.33)$$

Proof. Equation (2.31) is obvious. For (2.32) we have

$$\begin{aligned}
\int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, d\mathbf{x} &= \sum_{i=1}^N \sum_{j=1}^N \int_{\Omega} w_i u_j \frac{\partial v_i}{\partial x_j} \, d\mathbf{x} \\
&= \sum_{i=1}^N \sum_{j=1}^N \left(- \int_{\Omega} \frac{\partial w_i u_j}{\partial x_j} v_i \, d\mathbf{x} + \int_{\Gamma} w_i u_j v_i n_j \, ds \right) \\
&= \sum_{i=1}^N \sum_{j=1}^N \left(- \int_{\Omega} v_i u_j \frac{\partial w_i}{\partial x_j} \, d\mathbf{x} - \int_{\Omega} v_i w_i \frac{\partial u_j}{\partial x_j} \, d\mathbf{x} + \int_{\Gamma} u_j n_j w_i v_i \, ds \right) \\
&= - \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{w} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n})(\mathbf{w} \cdot \mathbf{v}) \, ds.
\end{aligned}$$

Eq. (2.33) follows from the definition of c and from (2.32). \square

The operators a and a_m have the following properties:

a) continuity: there exist $C_a, C_m > 0$ such that

$$|a(\mathbf{u}, \mathbf{v})| \leq C_a \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 \quad \forall (\mathbf{u}, \mathbf{v}) \in \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega), \quad (2.34)$$

$$|a_m(\mathbf{u}, \mathbf{v})| \leq C_m \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 \quad \forall (\mathbf{u}, \mathbf{v}) \in \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega); \quad (2.35)$$

b) coercivity: there exist $\alpha_a, \alpha_m > 0$ such that, with $\text{meas}(\Gamma_1) > 0$,

$$a(\mathbf{u}, \mathbf{u}) \geq \alpha_a \|\mathbf{u}\|_1^2 \quad \forall \mathbf{u} \in \mathbf{H}_{\Gamma_1}^1(\Omega), \quad (2.36)$$

$$a_m(\mathbf{u}, \mathbf{u}) \geq \alpha_m \|\mathbf{u}\|_1^2 \quad \forall \mathbf{u} \in \mathbf{V}(\Omega). \quad (2.37)$$

The bilinear form d satisfies the following properties:

a) continuity: there exists $D > 0$ such that

$$|d(\mathbf{u}, q)| \leq D \|\mathbf{u}\|_1 \|q\| \quad \forall (\mathbf{u}, q) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega); \quad (2.38)$$

b) weak coercivity or the inf-sup condition: there exists $\beta_1 > 0$ such that

$$\inf_{0 \neq q \in L_0^2(\Omega)} \sup_{0 \neq \mathbf{u} \in \mathbf{H}^1(\Omega)} \frac{d(\mathbf{u}, q)}{\|\mathbf{u}\|_1 \|q\|_0} \geq \beta_1. \quad (2.39)$$

The trilinear form $c(\mathbf{u}, \mathbf{v}, \mathbf{w})$ satisfies the following properties (see [13, 12, 35]):

a) continuity: there exists $K > 0$ independent of the functions \mathbf{u}, \mathbf{v} and \mathbf{w} such that

$$|c(\mathbf{u}, \mathbf{v}, \mathbf{w})| \leq K \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 \|\mathbf{w}\|_1 \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega); \quad (2.40)$$

b) antisymmetric property with respect to the last two arguments:

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -c(\mathbf{u}, \mathbf{w}, \mathbf{v}) \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega). \quad (2.41)$$

An obvious consequence of (2.41) is that

$$c(\mathbf{w}, \mathbf{u}, \mathbf{u}) = 0 \quad \forall \mathbf{u}, \mathbf{w} \in \mathbf{H}^1(\Omega). \quad (2.42)$$

The advantage that using the form $c(\cdot, \cdot, \cdot)$ has over using the form $\tilde{c}(\cdot, \cdot, \cdot)$ is that the latter satisfies (2.41) and (2.42) only if $\nabla \cdot \mathbf{w} = 0$ on Ω and one of $\mathbf{w} \cdot \mathbf{n} = 0$ or $\mathbf{u} = \mathbf{0}$, or, for (2.41), $\mathbf{v} = \mathbf{0}$ hold on Γ .

Given the previous framework, $(\mathbf{u}, r, \mathbf{B}, \sigma) \in \mathbf{H}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ is called a *weak solution for the MHD equations* if it satisfies the weak form of the steady MHD system (2.2) given by

$$\begin{aligned} \frac{1}{Re} a(\mathbf{u}, \mathbf{v}_1) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}_1) - S_1 c(\mathbf{B}, \mathbf{B}, \mathbf{v}_1) + d(\mathbf{v}_1, r) \\ = \langle \mathbf{f}, \mathbf{v}_1 \rangle - \langle \mathbf{t}, \mathbf{v}_1 \rangle_{\Gamma \setminus \Gamma_1}, \\ d(\mathbf{u}, q_1) = 0, \\ \frac{1}{Re_m} a_m(\mathbf{B}, \mathbf{v}_2) + c(\mathbf{u}, \mathbf{B}, \mathbf{v}_2) - c(\mathbf{B}, \mathbf{u}, \mathbf{v}_2) + d(\mathbf{v}_2, \sigma) = 0, \\ d(\mathbf{B}, q_2) = 0, \end{aligned} \quad (2.43)$$

along with boundary conditions

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_1, \quad (2.44)$$

$$\mathbf{B} = \Phi_0 \quad \text{on } \Gamma, \quad (2.45)$$

for all test functions $(\mathbf{v}_1, q_1, \mathbf{v}_2, q_2) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$.

The choice of these boundary conditions is of interest for our boundary optimal control problem. From a physical point of view, a portion of the domain can have non-homogeneous velocity boundary conditions while the remaining part can have stress type boundary conditions. The magnetic field on the boundary can act either as a known, fixed quantity or as a control variable and in both cases it may assume non-homogeneous values.

Clearly, if $(\mathbf{u}, r, \mathbf{B}, \sigma)$ is a solution of the strong problem, then it is also a solution of the weak formulation (2.43), while the converse may or may not be true.

Now, with the decomposition of the magnetic field \mathbf{B} given by (2.12), the weak form of the MHD equations can be recast as follows: given a lifting function $\mathbf{B}_e \in \mathbf{V}(\Omega)$ such that $\mathbf{B}_e = \Phi_0$ on Γ , we seek $(\mathbf{u}, r, \mathbf{b}, \sigma) \in \mathbf{H}^1(\Omega) \times L^2(\Omega) \times$

$\mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$ such that

$$\begin{aligned} \frac{1}{Re} a(\mathbf{u}, \mathbf{v}_1) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) + \\ d(\mathbf{v}_1, r) = \langle \mathbf{f}, \mathbf{v}_1 \rangle - \langle \mathbf{t}, \mathbf{v}_1 \rangle_{\Gamma \setminus \Gamma_1}, \\ d(\mathbf{u}, q_1) = 0, \\ \frac{1}{Re_m} a_m(\mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) + c(\mathbf{u}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) - c(\mathbf{b} + \mathbf{B}_e, \mathbf{u}, \mathbf{v}_2) + d(\mathbf{v}_2, \sigma) = 0, \\ d(\mathbf{b}, q_2) = 0, \end{aligned} \tag{2.46}$$

along with boundary conditions

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_1, \tag{2.47}$$

$$\mathbf{b} = \mathbf{0} \quad \text{on } \Gamma, \tag{2.48}$$

for all test functions $(\mathbf{v}_1, q_1, \mathbf{v}_2, q_2) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$.

Note that the boundary condition (2.17) does not have to be explicitly enforced; it is, in fact, a *natural* boundary condition imposed weakly through the boundary integral appearing in the first equation in (2.46). On the other hand, (2.16) and (2.18) are *essential* boundary conditions for the weak formulation (2.46) so that they must be explicitly imposed on candidate solutions. Also note that the third equation in (2.46) does not contain any boundary integrals because an essential boundary condition for the magnetic field variable \mathbf{b} is imposed on all of Γ , i.e., see (2.18).

It is a straightforward matter to verify that if $(\mathbf{u}, r, \mathbf{b}, \sigma)$ is a solution of (2.15)–(2.18), then it satisfies (2.46), (2.16), and (2.18). On the other hand, the converse is true only for solutions of (2.46), (2.16), and (2.18) that are sufficiently smooth, i.e., the weak formulation (2.46), (2.16), and (2.18) admits solutions that are not sufficiently regular to satisfy (2.15)–(2.18). For this reason, solutions $(\mathbf{u}, r, \mathbf{b}, \sigma)$ of (2.46), (2.16), and (2.18) are not only referred to as *weak* solutions of the MHD system (2.15)–(2.18), but are also referred to as *generalized* solutions.

Note that our indirect approach towards finding optimal boundary controls allows for those controls to belong to weaker spaces than would be practical in a direct boundary approach. In the latter case, one would replace the regularization term in the functional (2.19) by the square of a norm of the boundary controls. For this purpose, one would want to use the weakest norms possible that still result in well posedness; these norms are fractional Sobolev space norms, i.e., $H^{1/2}(\Gamma)$ for components of the magnetic field \mathbf{B} and $H^{-1/2}(\Gamma)$ for components of the electric field \mathbf{E} ; see [19]. In our approach we instead find a distributed control \mathbf{B}_e for which the square of the $\mathbf{H}^1(\Omega)$ norm is appropriate for regularization; see (2.19).

Thus, one does not have to deal with fractional Sobolev space norms. Moreover, with $\mathbf{B}_\epsilon, \mathbf{b}, \mathbf{u} \in \mathbf{H}^1(\Omega)$, we have, from (2.20) and (2.21), that the components of the corresponding optimal boundary magnetic and electric field controls do belong to the appropriate trace spaces $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$, respectively, so that our approach results in boundary controls that are optimal with respect to larger function spaces.

2.4 Existence of a solution to the state equations

2.4.1 Reduction to homogeneous boundary conditions

Before stating the existence theorem for the state equations, some technical lemmas are required in substitution to standard results. It is well-known that the global solvability (i.e., without any smallness condition on the data) of the Navier-Stokes equations in the case of nonhomogeneous Dirichlet boundary conditions can be proved by the Hopf's lemma also known as Leray's inequality (see [13, 35]). For the MHD equations, local solvability was proved in [20] with a smallness assumption on the boundary velocity but no bound on the boundary magnetic field. This smallness requirement was removed in [39], where a continuation of the boundary data for the velocity field was constructed whose \mathbf{L}^3 -norm can be made arbitrarily small. We recall the result.

Lemma 2. *For any given $\mathbf{g} \in \mathbf{H}^{1/2}(\Gamma)$ with $\int_\Gamma \mathbf{g} \cdot \mathbf{n} = 0$ and for any $\epsilon > 0$ there exists a function $\mathbf{u}_\epsilon \in \mathbf{H}^1(\Omega)$ such that*

$$\mathbf{u}_\epsilon = \mathbf{g} \quad \text{on } \Gamma, \quad \nabla \cdot \mathbf{u}_\epsilon = 0, \quad \|\mathbf{u}_\epsilon\|_{\mathbf{L}^3(\Omega)} \leq \epsilon \|\mathbf{g}\|_{1/2, \Gamma}. \quad (2.49)$$

Proof. See [39]. □

We will use this result to prove the global solvability of our problem. This allows us to avoid having *a priori* bounds on the boundary conditions and controls. A generalization of this for the case of nonhomogeneous mixed Dirichlet-Neumann boundary conditions is straightforward and similar results can be found in [7, 2].

Remark 1. For the proof of existence, we will need to perform splittings of the type $\mathbf{u} = \hat{\mathbf{u}} + \mathbf{u}_0$ and $\mathbf{B} = \hat{\mathbf{b}} + \mathbf{B}_0$ where \mathbf{u}_0 and \mathbf{B}_0 are lifting functions of the Dirichlet boundary conditions whose existence for sufficiently small values of ϵ is guaranteed by Lemma 2. The set of boundary conditions given by (2.47) - (2.48) does not assure that $\hat{\mathbf{u}}$ is homogeneous over the whole boundary Γ . Therefore, the property $\hat{\mathbf{u}} \cdot \mathbf{n} = 0$ is fulfilled only on Γ_1 : this is the reason for the use of the trilinear form $c(\mathbf{u}, \mathbf{v}, \mathbf{w})$ which is antisymmetric for any $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega)$, independent of their values at the boundary Γ (see Lemma 1).

The proof of existence requires the transformation of the non-homogeneous boundary problem into a problem with homogeneous boundary conditions. Therefore let us split the velocity and magnetic fields as

$$\mathbf{u} = \hat{\mathbf{u}} + \mathbf{u}_0, \quad \mathbf{B} = \hat{\mathbf{b}} + \mathbf{B}_0, \quad (2.50)$$

with

$$\begin{aligned} \nabla \cdot \mathbf{u}_0 &= 0 & \mathbf{u}_0 &= \mathbf{g} & \text{on } \Gamma_1, \\ \nabla \cdot \mathbf{B}_0 &= 0 & \mathbf{B}_0 &= \Phi_0 & \text{on } \Gamma. \end{aligned}$$

After the splitting of the lifting functions, we note that the MHD problem can be put in the framework of the abstract setting for nonlinear mixed problems like the Navier-Stokes equations as described in Section 1.3 ([13]). In order to see this let us consider the bilinear forms

$$a_0((\mathbf{u}, \mathbf{B}), (\mathbf{v}_1, \mathbf{v}_2)) = \frac{1}{Re} a(\mathbf{u}, \mathbf{v}_1) + \frac{S_1}{Re_m} a_m(\mathbf{B}, \mathbf{v}_2)$$

and

$$\hat{d}((\mathbf{v}_1, \mathbf{v}_2), (r, \sigma)) = d(\mathbf{v}_1, r) + S_1 d(\mathbf{v}_2, \sigma).$$

We also define the trilinear forms

$$\begin{aligned} a_1((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) &= \\ c(\mathbf{u}, \mathbf{w}, \mathbf{v}_1) - S_1 c(\mathbf{B}, \mathbf{C}, \mathbf{v}_1) + S_1 c(\mathbf{u}, \mathbf{C}, \mathbf{v}_2) - S_1 c(\mathbf{B}, \mathbf{w}, \mathbf{v}_2) \end{aligned}$$

such that

$$\begin{aligned} a_1((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{v}_1, \mathbf{v}_2)) &= \\ c(\mathbf{u}, \mathbf{u}, \mathbf{v}_1) - S_1 c(\mathbf{B}, \mathbf{B}, \mathbf{v}_1) + S_1 c(\mathbf{u}, \mathbf{B}, \mathbf{v}_2) - S_1 c(\mathbf{B}, \mathbf{u}, \mathbf{v}_2) \end{aligned}$$

and

$$\begin{aligned} \hat{a}((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) &:= \\ a_0((\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) + a_1((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)). \end{aligned}$$

For a given $(\mathbf{u}_0, \mathbf{B}_0)$ we set

$$\langle \hat{\mathbf{F}}, (\mathbf{v}_1, \mathbf{v}_2) \rangle = \langle \mathbf{f}, \mathbf{v}_1 \rangle - \hat{a}((\mathbf{u}_0, \mathbf{B}_0), (\mathbf{u}_0, \mathbf{B}_0), (\mathbf{v}_1, \mathbf{v}_2))$$

and

$$\begin{aligned} \tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) &:= \hat{a}((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) + \\ a_1((\mathbf{u}_0, \mathbf{B}_0), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2)) + a_1((\mathbf{w}, \mathbf{C}), (\mathbf{u}_0, \mathbf{B}_0), (\mathbf{v}_1, \mathbf{v}_2)). \end{aligned}$$

After multiplying the MHD equation by S_1 , summing the equations and bringing the lifting functions to the right hand side, the *homogenized MHD problem* in the so-called *Q form* [13] becomes: seek $(\hat{\mathbf{u}}, \hat{\mathbf{b}}) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times \mathbf{H}_0^1(\Omega)$ and $(r, \sigma) \in L^2(\Omega) \times L_0^2(\Omega)$ so as to satisfy the equation

$$\begin{aligned} \tilde{a}((\hat{\mathbf{u}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{b}}), (\mathbf{v}_1, \mathbf{v}_2)) + \hat{d}((\mathbf{v}_1, \mathbf{v}_2), (r, \sigma)) = \\ \langle \hat{\mathbf{F}}, (\mathbf{v}_1, \mathbf{v}_2) \rangle - \langle \mathbf{t}, \mathbf{v}_1 \rangle_{\Gamma \setminus \Gamma_1} \quad \forall (\mathbf{v}_1, \mathbf{v}_2) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times \mathbf{H}_0^1(\Omega), \\ \hat{d}((\hat{\mathbf{u}}, \hat{\mathbf{b}}), (z_1, z_2)) = 0 \quad \forall (z_1, z_2) \in L^2(\Omega) \times L_0^2(\Omega), \\ \hat{\mathbf{u}} = \mathbf{0} \quad \text{on } \Gamma_1, \\ \hat{\mathbf{b}} = \mathbf{0} \quad \text{on } \Gamma. \end{aligned}$$

Let \mathbf{Z} be the kernel of \hat{d} , namely,

$$\mathbf{Z} = \{(\mathbf{u}, \mathbf{B}) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times \mathbf{H}_0^1(\Omega) \mid \hat{d}((\mathbf{u}, \mathbf{B}), (z_1, z_2)) = 0 \\ \forall (z_1, z_2) \in L^2(\Omega) \times L_0^2(\Omega)\}, \quad (2.51)$$

with the norm naturally inherited by its parent space (see [20])

$$\|(\mathbf{u}, \mathbf{B})\|_{\mathbf{Z}} = (\|\mathbf{u}\|_1^2 + \|\mathbf{B}\|_1^2)^{1/2}. \quad (2.52)$$

We can rewrite the above problem in *P form* as

$$\tilde{a}((\hat{\mathbf{u}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{b}}), (\mathbf{v}_1, \mathbf{v}_2)) = \langle \hat{\mathbf{F}}, (\mathbf{v}_1, \mathbf{v}_2) \rangle - \langle \mathbf{t}, \mathbf{v}_1 \rangle_{\Gamma \setminus \Gamma_1} \quad \forall (\mathbf{v}_1, \mathbf{v}_2) \in \mathbf{Z}. \quad (2.53)$$

It is well-known that the *P* and *Q* forms are equivalent and the former is used for the proof of existence in the abstract nonlinear framework described in Section 1.3.2.

2.4.2 Coercivity property

We formulate the coercivity property of the form $\tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{w}, \mathbf{C}), (\mathbf{v}_1, \mathbf{v}_2))$ on the space \mathbf{Z} . Note that unlike [20] the coercivity will be proved without any conditions on the data (see [39]).

Lemma 3. *For all $(\mathbf{u}, \mathbf{B}) \in \mathbf{Z}$ there exists a constant $K > 0$ such that*

$$\tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) \geq K \|(\mathbf{u}, \mathbf{B})\|_{\mathbf{Z}}^2. \quad (2.54)$$

Proof. We have

$$\begin{aligned}
& \tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) \\
& \quad := a_0((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) + a_1((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) \\
& \quad + a_1((\mathbf{u}_0, \mathbf{B}_0), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) + a_1((\mathbf{u}, \mathbf{B}), (\mathbf{u}_0, \mathbf{B}_0), (\mathbf{u}, \mathbf{B})) \\
& \quad = \frac{1}{Re} a(\mathbf{u}, \mathbf{u}) + c(\mathbf{u}, \mathbf{u}_0, \mathbf{u}) - S_1 c(\mathbf{B}, \mathbf{B}_0, \mathbf{u}) \\
& \quad + \frac{S_1}{Re_m} a_m(\mathbf{B}, \mathbf{B}) + S_1 c(\mathbf{u}, \mathbf{B}_0, \mathbf{B}) - S_1 c(\mathbf{B}, \mathbf{u}_0, \mathbf{B}),
\end{aligned}$$

where some of the terms vanish due to the antisymmetry property of the form c with respect to the last two arguments for all functions in $\mathbf{H}^1(\Omega)$. The coercivity of the forms yields

$$\begin{aligned}
a(\mathbf{u}, \mathbf{u}) & \geq \alpha \|\mathbf{u}\|_1^2 \\
a_m(\mathbf{B}, \mathbf{B}) & \geq \alpha_m \|\mathbf{B}\|_1^2.
\end{aligned}$$

On the other hand, we can bound the remaining terms with arbitrarily small coefficients. By using the generalized Hölder inequality, the embedding of \mathbf{H}^1 into \mathbf{L}^6 , the generalized Poincaré inequality for \mathbf{u} and Lemma 2, we have that

$$\begin{aligned}
|c(\mathbf{u}, \mathbf{u}_0, \mathbf{u})| & = |c(\mathbf{u}, \mathbf{u}, \mathbf{u}_0)| \\
& \leq \|\mathbf{u}\|_{\mathbf{L}^6} \|\nabla \mathbf{u}\|_{\mathbf{L}^2} \|\mathbf{u}_0\|_{\mathbf{L}^3} \\
& \leq \epsilon_1 \|\mathbf{u}\|_1^2
\end{aligned}$$

for an arbitrarily small constant $\epsilon_1 > 0$. For the second term we have

$$\begin{aligned}
|c(\mathbf{B}, \mathbf{B}_0, \mathbf{u})| & = |c(\mathbf{B}, \mathbf{u}, \mathbf{B}_0)| \\
& \leq \|\mathbf{B}\|_{\mathbf{L}^6} \|\nabla \mathbf{u}\|_{\mathbf{L}^2} \|\mathbf{B}_0\|_{\mathbf{L}^3} \\
& \leq \epsilon_2 \|\mathbf{B}\|_1 \|\mathbf{u}\|_1
\end{aligned}$$

for an arbitrarily small constant $\epsilon_2 > 0$. The third term can be treated in a similar way to obtain

$$|c(\mathbf{u}, \mathbf{B}_0, \mathbf{B})| = |c(\mathbf{u}, \mathbf{B}, \mathbf{B}_0)| \leq \epsilon_3 \|\mathbf{B}\|_1 \|\mathbf{u}\|_1$$

for an arbitrarily small constant $\epsilon_3 > 0$. The fourth term yields

$$|c(\mathbf{B}, \mathbf{u}_0, \mathbf{B})| = |c(\mathbf{B}, \mathbf{B}, \mathbf{u}_0)| \leq \epsilon_4 \|\mathbf{B}\|_1^2$$

Finally, gathering all the terms and using Young's inequality $ab \leq a^2/\gamma + \gamma b^2/4$ we have

$$\begin{aligned}
& \tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B})) = \\
& \frac{1}{Re} a(\mathbf{u}, \mathbf{u}) + c(\mathbf{u}, \mathbf{u}_0, \mathbf{u}) - S_1 c(\mathbf{B}, \mathbf{B}_0, \mathbf{u}) \\
& + \frac{S_1}{Re_m} a_m(\mathbf{B}, \mathbf{B}) + S_1 c(\mathbf{u}, \mathbf{B}_0, \mathbf{B}) - S_1 c(\mathbf{B}, \mathbf{u}_0, \mathbf{B}) \\
& \geq Re\alpha \|\mathbf{u}\|_1^2 + \frac{Re_m}{S_1} \alpha_m \|\mathbf{B}\|_1^2 - |c(\mathbf{u}, \mathbf{u}_0, \mathbf{u})| \\
& - |S_1 c(\mathbf{B}, \mathbf{B}_0, \mathbf{u})| - |S_1 c(\mathbf{u}, \mathbf{B}_0, \mathbf{B})| - |S_1 c(\mathbf{B}, \mathbf{u}_0, \mathbf{B})| \\
& \geq Re\alpha \|\mathbf{u}\|_1^2 + \frac{Re_m}{S_1} \alpha_m \|\mathbf{B}\|_1^2 \\
& - \epsilon_1 \|\mathbf{u}\|_1^2 - \epsilon_2 \|\mathbf{B}\|_1 \|\mathbf{u}\|_1 - \epsilon_3 \|\mathbf{B}\|_1 \|\mathbf{u}\|_1 - \epsilon_4 \|\mathbf{B}\|_1^2 \\
& \geq Re\alpha \|\mathbf{u}\|_1^2 + \frac{Re_m}{S_1} \alpha_m \|\mathbf{B}\|_1^2 \\
& - \epsilon_1 \|\mathbf{u}\|_1^2 - (\epsilon_2 + \epsilon_3) \|\mathbf{B}\|_1 \|\mathbf{u}\|_1 - \epsilon_4 \|\mathbf{B}\|_1^2 \\
& \geq Re\alpha \|\mathbf{u}\|_1^2 + \frac{Re_m}{S_1} \alpha_m \|\mathbf{B}\|_1^2 \\
& - \epsilon_1 \|\mathbf{u}\|_1^2 - \frac{(\epsilon_2 + \epsilon_3)}{\epsilon_5} \|\mathbf{B}\|_1^2 - (\epsilon_2 + \epsilon_3) \frac{\epsilon_5}{4} \|\mathbf{u}\|_1^2 - \epsilon_4 \|\mathbf{B}\|_1^2 \\
& = (Re\alpha - \epsilon_1 - (\epsilon_2 + \epsilon_3) \frac{\epsilon_5}{4}) \|\mathbf{u}\|_1^2 + (\frac{Re_m}{S_1} \alpha_m - \frac{(\epsilon_2 + \epsilon_3)}{\epsilon_5}) \|\mathbf{B}\|_1^2 \\
& \geq \min\{(Re\alpha - \epsilon_1 - (\epsilon_2 + \epsilon_3) \frac{\epsilon_5}{4}), (\frac{Re_m}{S_1} \alpha_m - \frac{(\epsilon_2 + \epsilon_3)}{\epsilon_5})\} \|(\mathbf{u}, \mathbf{B})\|_{\mathbf{Z}}^2,
\end{aligned}$$

where $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5$ are arbitrarily small constants that can be chosen without any loss of generality so that

$$(Re\alpha - \epsilon_1 - (\epsilon_2 + \epsilon_3) \frac{\epsilon_5}{4}) > 0, \quad (\frac{Re_m}{S_1} \alpha_m - \frac{(\epsilon_2 + \epsilon_3)}{\epsilon_5}) > 0.$$

The proof is now complete. \square

2.4.3 Existence

The existence theorem for Problem P (2.53) and hence Problem Q can now be summarized in the following theorem.

Theorem 17. *Given $(\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2) \in \mathbf{H}^1(\Omega)^* \times \mathbf{H}^1(\Omega)^*$ and $\mathbf{t} \in \mathbf{H}^{-1/2}(\Gamma \setminus \Gamma_1)$ there exists at least one solution $((\hat{\mathbf{u}}, \hat{\mathbf{b}}), (r, \sigma)) \in (\mathbf{H}_{\Gamma_1}^1(\Omega) \times \mathbf{H}_0^1(\Omega)) \times (L^2(\Omega) \times L_0^2(\Omega))$*

of Problem Q , where $(\hat{\mathbf{u}}, \hat{\mathbf{b}})$ solves Problem P . Thus, the nonhomogeneous MHD problem 2.43 has a solution $(\mathbf{u}, r, \mathbf{B}, \sigma) \in (\mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}^1(\Omega) \times L_0^2(\Omega))$.

Proof. Problem P has a solution. In fact we can refer to the standard abstract setting for nonlinear mixed problems described in Section 1.3 and prove the following properties (see [13]).

- i) *Separability.* \mathbf{Z} is a separable Hilbert space as a subspace of $\mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega)$.
- ii) *Weak sequential continuity.* The mapping $\tilde{a}((\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}), (\mathbf{u}, \mathbf{B}))$ is weakly sequentially continuous on \mathbf{Z} . For a similar proof, see [13, 20].
- iii) *Continuity.* It is an immediate consequence of the continuity of the bilinear forms a, a_m, c, d .
- iv) *Coercivity.* See Lemma 3.

This implies by standard arguments that there exists at least one solution of Problem P .

- v) *L.B.B. condition* Following the Ladyzhenskaya-Babuska-Brezzi theory for mixed problems (see [25, 13, 20]) one can prove that there exists a positive constant β_2 such that

$$\inf_{0 \neq (r, \sigma) \in L^2(\Omega) \times L_0^2(\Omega)} \sup_{0 \neq (\hat{\mathbf{u}}, \hat{\mathbf{b}}) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times \mathbf{H}_0^1(\Omega)} \frac{\hat{d}((\hat{\mathbf{u}}, \hat{\mathbf{b}}), (r, \sigma))}{\|(\hat{\mathbf{u}}, \hat{\mathbf{b}})\|_{\mathbf{H}^1 \times \mathbf{H}^1} \|(r, \sigma)\|_{L^2 \times L_0^2}} \geq \beta_2 > 0. \quad (2.55)$$

The proof of i)-v) assures the existence of a solution to Problems P and Q . Therefore the inhomogeneous Problem (2.43) has a solution. \square

Remark 2. The existence can be proved by similar arguments if the Navier-Stokes equations are supplemented with mixed boundary conditions on $\Gamma \setminus \Gamma_1$, such as

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{g} \cdot \mathbf{n}, \quad \boldsymbol{\tau} \times \mathbf{n} = \mathbf{t} \times \mathbf{n} \quad (2.56)$$

or

$$\mathbf{u} \times \mathbf{n} = \mathbf{g} \times \mathbf{n}, \quad \boldsymbol{\tau} \cdot \mathbf{n} = \mathbf{t} \cdot \mathbf{n}. \quad (2.57)$$

In our analysis it is required to have full Dirichlet boundary conditions for the velocity $\mathbf{u} = \mathbf{g}$ on Γ_1 and full Dirichlet boundary conditions for the magnetic field $\mathbf{B} = \boldsymbol{\Phi}_0$ on Γ . Neumann boundary conditions for the magnetic field are not addressed here (see [39]).

Remark 3. The two decompositions of the magnetic field $\mathbf{B} = \mathbf{b} + \mathbf{B}_e$ and $\mathbf{B} = \hat{\mathbf{b}} + \mathbf{B}_0$ serve different purposes and it is evident that there is no relationship between \mathbf{B}_e and \mathbf{B}_0 , the former being used for the treatment of boundary conditions for the optimal control problem, the latter instead being used only for the theoretical purpose of proving the existence of a magnetic field \mathbf{B} satisfying the inhomogeneous state problem. In other words, the lifting \mathbf{B}_0 can be chosen in

infinite manners with respect to a smallness parameter ϵ (see Lemma 2), with the only requirement that the coercivity property (3) holds. On the other hand, the function \mathbf{B}_ϵ is not chosen for theoretical purposes but it is rather a *computational tool* that is used for the solution of the boundary optimal control problem at hand.

2.5 Optimal control problem

After proving the existence of a solution to the state equations, which is necessary for the following steps, we perform here the analysis of the optimal control problem 7, much in the spirit of Section 1.4.

2.5.1 Existence of an optimal solution

We first state the optimal control problem in a more precise way by defining the *set of admissible target velocities* \mathbf{U}_{ad} and the *set of admissible solutions* \mathbf{A}_{ad} . The first is defined as

$$\mathbf{U}_{ad} = \{\mathbf{u} \in \mathbf{H}^1(\Omega) \mid -\frac{1}{Re}\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} \in \mathbf{L}^2(\Omega)\} \quad (2.58)$$

and the *set of admissible solutions* as

$$\begin{aligned} \mathbf{A}_{ad} = \{ & (\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_\epsilon) \in \mathbf{H}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times \mathbf{H}^1(\Omega) \mid \\ & (\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_\epsilon) \text{ satisfies (2.46) and (2.13) and } \mathcal{J}(\mathbf{u}, \mathbf{B}_\epsilon) < \infty\}. \end{aligned} \quad (2.59)$$

We also set the restriction $\bar{\mathbf{A}}_{ad}$ as

$$\bar{\mathbf{A}}_{ad} = \{(\mathbf{u}, \mathbf{B}_\epsilon) \mid (\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_\epsilon) \in \mathbf{A}_{ad}\}.$$

The optimal boundary control problem can then be formulated as follows.

Problem 8. *Given $\mathbf{u}_d \in \mathbf{U}_{ad}$ find a global minimum point $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_\epsilon) \in \mathbf{A}_{ad}$ of the objective functional*

$$\mathcal{J}(\mathbf{u}, \mathbf{B}_\epsilon) = \frac{\alpha}{2}\|\mathbf{u} - \mathbf{u}_d\|_0^2 + \frac{\beta}{2}\|\mathbf{B}_\epsilon\|_0^2 + \frac{\gamma}{2}a(\mathbf{B}_\epsilon, \mathbf{B}_\epsilon) \quad . \quad (2.60)$$

We now state the existence of a global minimizer in \mathbf{A}_{ad} .

Theorem 18. *Given $\mathbf{u}_d \in \mathbf{U}_{ad}$, there exists a solution $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_\epsilon) \in \mathbf{A}_{ad}$ of the optimal control problem.*

Proof. The proof follows standard techniques as those described in Section 1.4. Theorem 17 states the existence of a solution of the state MHD system, therefore the set of admissible solutions \mathbf{A}_{ad} is nonempty. We define

$$M := \inf_{(\mathbf{u}, \mathbf{B}_e) \in \mathbf{A}_{ad}} \mathcal{J}(\mathbf{u}, \mathbf{B}_e)$$

Clearly M exists and $M \geq 0$ since the set \mathbf{A}_{ad} is nonempty and the functional is non-negative. Thus let $\{(\mathbf{u}_n, r_n, \mathbf{b}_n, \sigma_n, \mathbf{B}_{en})\}$ be a minimizing sequence in \mathbf{A}_{ad} for the objective functional, i.e.,

$$\lim_{n \rightarrow \infty} \mathcal{J}(\mathbf{u}_n, \mathbf{B}_{en}) = M.$$

Now we show that the minimizing sequence $\{(\mathbf{u}_n, r_n, \mathbf{b}_n, \sigma_n, \mathbf{B}_{en})\}$ is uniformly bounded. As every convergent sequence of real numbers, the sequence $\mathcal{J}(\mathbf{u}_n, \mathbf{B}_{en})$ is bounded, hence the sequence $\{\mathbf{B}_{en}\}$ is uniformly bounded in $\mathbf{V}(\Omega)$. By the continuous dependence of the state solutions on the data and by using the triangle inequality, we also have that the sequences $\{\mathbf{u}_n\}$, $\{\mathbf{b}_n\}$, $\{r_n\}$ and $\{\sigma_n\}$ are uniformly bounded. Hence, we can extract a subsequence $\{(\mathbf{u}_m, r_m, \mathbf{b}_m, \sigma_m, \mathbf{B}_{em})\}$ that converges weakly to some $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_e)$. We write

$$\begin{array}{llll} \mathbf{u}_m \rightarrow \tilde{\mathbf{u}} & \text{weakly in} & \mathbf{V}(\Omega) \\ r_m \rightarrow \tilde{r} & \text{weakly in} & L^2(\Omega) \\ \mathbf{b}_m \rightarrow \tilde{\mathbf{b}} & \text{weakly in} & \mathbf{V}(\Omega) \\ \sigma_m \rightarrow \tilde{\sigma} & \text{weakly in} & L_0^2(\Omega) \\ \mathbf{B}_{em} \rightarrow \tilde{\mathbf{B}}_e & \text{weakly in} & \mathbf{V}(\Omega) \\ \mathbf{u}_m \rightarrow \tilde{\mathbf{u}} & \text{strongly in} & \mathbf{L}^2(\Omega) \\ \mathbf{b}_m \rightarrow \tilde{\mathbf{b}} & \text{strongly in} & \mathbf{L}^2(\Omega) \\ \mathbf{B}_{em} \rightarrow \tilde{\mathbf{B}}_e & \text{strongly in} & \mathbf{L}^2(\Omega). \end{array}$$

The strong convergence in $\mathbf{L}^2(\Omega)$ is a consequence of the compact imbedding from $\mathbf{H}^1(\Omega)$ to $\mathbf{L}^2(\Omega)$.

Now we pass to the limit in order to show that $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_e) \in \mathbf{A}_{ad}$, i.e. the weak limit of the subsequences satisfies the constraints. We can pass to the limit inside the linear and the nonlinear terms of the MHD equation. The result for the nonlinear term can be proved by compactness arguments (see for example [13, 20, 35]). Therefore the set \mathbf{A}_{ad} is closed in the weak topology. In order to see that $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_e) \in \mathbf{A}_{ad}$ is a solution for the optimal control problem we finally

have to show that $(\tilde{\mathbf{u}}, \tilde{\mathbf{B}}_e) \in \bar{\mathbf{A}}_{ad}$ minimizes the functional. In fact, by the weak lower semicontinuity of the functional, we have

$$M \leq \mathcal{J}(\tilde{\mathbf{u}}, \tilde{\mathbf{B}}_e) \leq \liminf_{m \rightarrow \infty} \mathcal{J}(\mathbf{u}_m, \mathbf{B}_{em}) \leq \lim_{k \rightarrow \infty} (\inf_{p \geq k} \mathcal{J}(\mathbf{u}_p, \mathbf{B}_{ep})) = M.$$

Therefore the infimum M of the functional is attained at the point $(\tilde{\mathbf{u}}, \tilde{\mathbf{B}}_e)$, which is indeed a global minimizer. \square

Remark 4. The existence of a global minimizer to the optimal control problem has been proved. In the following, the technique of Lagrange multipliers will be used in order to turn the constrained problem into an unconstrained one. Unfortunately, this method only allows us to search for local minima [16], i.e. to find solutions of the local problem

Problem 9. Given $\mathbf{u}_d \in \mathbf{U}_{ad}$ find a $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_e) \in \mathbf{A}_{ad}$ such that the objective functional

$$\mathcal{J}(\mathbf{u}, \mathbf{B}_e) = \frac{\alpha}{2} \|\mathbf{u} - \mathbf{u}_d\|_0^2 + \frac{\beta}{2} \|\mathbf{B}_e\|_0^2 + \frac{\gamma}{2} a(\mathbf{B}_e, \mathbf{B}_e) \quad (2.61)$$

is locally minimized, i.e. there exists $\epsilon > 0$ such that

$$\mathcal{J}(\tilde{\mathbf{u}}, \tilde{\mathbf{B}}_e) \leq \mathcal{J}(\mathbf{u}, \mathbf{B}_e) \quad \forall (\mathbf{u}, \mathbf{B}_e) \in \bar{\mathbf{A}}_{ad} \text{ with } \|\mathbf{u} - \tilde{\mathbf{u}}\|_1 + \|\mathbf{B}_e - \tilde{\mathbf{B}}_e\|_1 < \epsilon. \quad (2.62)$$

2.5.2 First-order necessary condition

In order to obtain the first order necessary conditions and the optimality system for the optimal control problem let us introduce the nonlinear continuous constraint operator $M : \mathbf{B}_1 \rightarrow \mathbf{B}_2$ defined between the spaces $\mathbf{B}_1 = \mathbf{H}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times \mathbf{H}^1(\Omega)$ and $\mathbf{B}_2 = \mathbf{H}_{\Gamma_1}^1(\Omega)^* \times L^2(\Omega) \times \mathbf{H}^{-1}(\Omega) \times L_0^2(\Omega) \times L_0^2(\Omega)$ so that $M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) = (\mathbf{f}_1, q_1, \mathbf{f}_2, q_2, q_3)$, where

$$\begin{aligned} \langle \mathbf{f}_1, \mathbf{v}_1 \rangle &:= \frac{1}{Re} a(\mathbf{u}, \mathbf{v}_1) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) + \\ &\quad d(\mathbf{v}_1, r) + \langle \mathbf{t}, \mathbf{v}_1 \rangle_{\Gamma \setminus \Gamma_1}, \\ (q_1, z_1) &:= d(\mathbf{u}, z_1), \\ \langle \mathbf{f}_2, \mathbf{v}_2 \rangle &:= \frac{1}{Re_m} a_m(\mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) + c(\mathbf{u}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) - c(\mathbf{b} + \mathbf{B}_e, \mathbf{u}, \mathbf{v}_2) + d(\mathbf{v}_2, \sigma), \\ (q_2, z_2) &:= d(\mathbf{b}, z_2), \\ (q_3, z_3) &:= d(\mathbf{B}_e, z_3) \end{aligned}$$

for all functions $(\mathbf{v}_1, z_1, \mathbf{v}_2, z_2, z_3) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times L_0^2(\Omega)$ and boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_1, \\ \mathbf{b} &= \mathbf{0} && \text{on } \Gamma, \\ \mathbf{B}_e &= \Phi_{0,d} && \text{on } \Gamma_d, \\ \mathbf{B}_e &= \Phi_{0,c} && \text{on } \Gamma_c. \end{aligned}$$

With the definition of the mapping $M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ the constraints can be expressed as $M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) = (\mathbf{f}, 0, \mathbf{0}, 0, 0)$. Through the usual method of Lagrange multipliers, we turn the constrained minimization problem into an unconstrained one. The new problem is then

Problem 10. *Find a stationary point $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e, \bar{a}, \bar{\boldsymbol{\lambda}}, \bar{\pi}_1, \bar{\boldsymbol{\xi}}, \bar{\pi}_2, \bar{\pi}_3)$ of the augmented functional*

$$\begin{aligned} \mathcal{J}_{aug}(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e, a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) = \\ a\mathcal{J}(\mathbf{u}, \mathbf{B}_e) + \langle (\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \rangle. \end{aligned} \quad (2.63)$$

Clearly, one does not know whether stationary points of \mathcal{J}_{aug} , i.e. points with vanishing Fréchet differential, yield a local minimum of the original cost functional. In the following we will only derive a first-order necessary condition.

At every point $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \in \mathbf{B}_1$ we introduce the bounded linear mapping $M' \in \mathcal{L}(\mathbf{B}_1, \mathbf{B}_2)$ as $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) = (\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3)$ for $(\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1$ and $(\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \mathbf{B}_2$ if and only if

$$\begin{aligned} \langle \hat{\mathbf{f}}_1, \mathbf{v}_1 \rangle &:= \frac{1}{Re} a(\hat{\mathbf{u}}, \mathbf{v}_1) + c(\hat{\mathbf{u}}, \mathbf{u}, \mathbf{v}_1) + c(\mathbf{u}, \hat{\mathbf{u}}, \mathbf{v}_1) \\ &\quad - S_1 c(\hat{\mathbf{b}}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{b}}, \mathbf{v}_1) \\ &\quad - S_1 c(\hat{\mathbf{B}}_e, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{B}}_e, \mathbf{v}_1) \\ &\quad + d(\mathbf{v}_1, \hat{r}), \\ (\hat{q}_1, z_1) &:= d(\hat{\mathbf{u}}, z_1), \\ \langle \hat{\mathbf{f}}_2, \mathbf{v}_2 \rangle &:= \frac{1}{Re_m} a_m(\hat{\mathbf{b}}, \mathbf{v}_2) + \frac{1}{Re_m} a_m(\hat{\mathbf{B}}_e, \mathbf{v}_2) \\ &\quad + c(\hat{\mathbf{u}}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) + c(\mathbf{u}, \hat{\mathbf{b}}, \mathbf{v}_2) + c(\mathbf{u}, \hat{\mathbf{B}}_e, \mathbf{v}_2) \\ &\quad - c(\hat{\mathbf{b}}, \mathbf{u}, \mathbf{v}_2) - c(\hat{\mathbf{B}}_e, \mathbf{u}, \mathbf{v}_2) - c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{u}}, \mathbf{v}_2) \\ &\quad + d(\mathbf{v}_2, \hat{\sigma}), \\ (\hat{q}_2, z_2) &:= d(\hat{\mathbf{b}}, z_2), \\ (\hat{q}_3, z_3) &:= d(\hat{\mathbf{B}}_e, z_3) \end{aligned}$$

for all functions $(\mathbf{v}_1, z_1, \mathbf{v}_2, z_2, z_3) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times L_0^2(\Omega)$ and homogeneous boundary conditions

$$\begin{aligned} \hat{\mathbf{u}} &= \mathbf{0} & \text{on} & \Gamma_1 \\ \hat{\mathbf{b}} &= \mathbf{0} & \text{on} & \Gamma \\ \hat{\mathbf{B}}_e &= \mathbf{0} & \text{on} & \Gamma_d. \end{aligned}$$

We have the following result.

Theorem 19. *Let $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \in \mathbf{B}_1$. The operator $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ has closed range and a finite-dimensional kernel.*

Proof. i) We can show that the operator $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ is a compact perturbation of an isomorphism; i.e., it can be decomposed as $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) = K'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) + S'$, where S' is an isomorphism and K' is a compact operator. The operator $S' \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) = (\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3)$ for $(\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1$ and $(\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \mathbf{B}_2$ is defined as

$$\begin{aligned} \langle \hat{\mathbf{f}}_1, \mathbf{v}_1 \rangle &:= \frac{1}{\text{Re}} a(\hat{\mathbf{u}}, \mathbf{v}_1) + d(\mathbf{v}_1, \hat{r}), \\ (\hat{q}_1, z_1) &:= d(\hat{\mathbf{u}}, z_1), \\ \langle \hat{\mathbf{f}}_2, \mathbf{v}_2 \rangle &:= \frac{1}{\text{Re}_m} a_m(\hat{\mathbf{b}}, \mathbf{v}_2) + \frac{1}{\text{Re}_m} a_m(\hat{\mathbf{B}}_e, \mathbf{v}_2) + d(\mathbf{v}_2, \hat{\sigma}), \\ (\hat{q}_2, z_2) &:= d(\hat{\mathbf{b}}, z_2), \\ (\hat{q}_3, z_3) &:= d(\hat{\mathbf{B}}_e, z_3), \end{aligned}$$

for all functions $(\mathbf{v}_1, z_1, \mathbf{v}_2, z_2, z_3) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times L_0^2(\Omega)$ and homogeneous boundary conditions. Given the unique solvability of the Stokes problem, we have that S' is an isomorphism. The operator $K' \in \mathcal{L}(\mathbf{B}_1, \mathbf{B}_2)$ is defined as $K'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) = (\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3)$ for $(\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1$ and $(\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \mathbf{B}_2$ if and only if

$$\begin{aligned} \langle \hat{\mathbf{f}}_1, \mathbf{v}_1 \rangle &:= +c(\hat{\mathbf{u}}, \mathbf{u}, \mathbf{v}_1) + c(\mathbf{u}, \hat{\mathbf{u}}, \mathbf{v}_1) \\ &\quad - S_1 c(\hat{\mathbf{b}}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{b}}, \mathbf{v}_1) \\ &\quad - S_1 c(\hat{\mathbf{B}}_e, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_1) - S_1 c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{B}}_e, \mathbf{v}_1), \\ (\hat{q}_1, z_1) &:= 0, \\ \langle \hat{\mathbf{f}}_2, \mathbf{v}_2 \rangle &:= +c(\hat{\mathbf{u}}, \mathbf{b} + \mathbf{B}_e, \mathbf{v}_2) + c(\mathbf{u}, \hat{\mathbf{b}}, \mathbf{v}_2) + c(\mathbf{u}, \hat{\mathbf{B}}_e, \mathbf{v}_2) \\ &\quad - c(\hat{\mathbf{b}}, \mathbf{u}, \mathbf{v}_2) - c(\hat{\mathbf{B}}_e, \mathbf{u}, \mathbf{v}_2) - c(\mathbf{b} + \mathbf{B}_e, \hat{\mathbf{u}}, \mathbf{v}_2), \\ (\hat{q}_2, z_2) &:= 0, \\ (\hat{q}_3, z_3) &:= 0, \end{aligned}$$

for all functions $(\mathbf{v}_1, z_1, \mathbf{v}_2, z_2, z_3) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times L_0^2(\Omega)$ and homogeneous boundary conditions. The compactness of Sobolev embeddings and the properties of the trilinear form $c(\mathbf{u}, \mathbf{v}, \mathbf{w})$ imply that the operator K' is compact [14, 25].

Finally, by the Fredholm alternative the operator M' is a semi-Fredholm operator, i.e. it has a closed range and a finite-dimensional kernel. \square

Let $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ denote an optimal solution in the local sense and $N : \mathbf{B}_1 \rightarrow \mathbb{R} \times \mathbf{B}_2$ the nonlinear mapping defined by $N(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) = (a, \mathbf{f}_1, q_1, \mathbf{f}_2, q_2, q_3)$ for $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \in \mathbf{B}_1$ and $(a, \mathbf{f}_1, q_1, \mathbf{f}_2, q_2, q_3) \in \mathbb{R} \times \mathbf{B}_2$ if and only if

$$\begin{aligned} a &:= \mathcal{J}(\mathbf{u}, \mathbf{B}_e) - \mathcal{J}(\bar{\mathbf{u}}, \bar{\mathbf{B}}_e), \\ (\mathbf{f}_1, q_1, \mathbf{f}_2, q_2, q_3) &:= M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e). \end{aligned}$$

As we did before, for any $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \in \mathbf{B}_1$ we introduce the bounded linear mapping $N' \in \mathcal{L}(\mathbf{B}_1, \mathbb{R} \times \mathbf{B}_2)$ defined as $N'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) = (a_0, \hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3)$ for $(\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1$ and $(a_0, \hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \mathbb{R} \times \mathbf{B}_2$ such that

$$\begin{aligned} a_0 &:= \mathcal{J}'(\mathbf{u}, \mathbf{B}_e) \cdot (\hat{\mathbf{u}}, \hat{\mathbf{B}}_e), \\ (\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) &:= M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e). \end{aligned}$$

Theorem 20. *Let $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ denote an optimal solution in the local sense.*

- i) The operator $N'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ has closed range.*
- ii) The operator $N'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ is not onto.*

Proof. i) Since the kernel of a continuous linear operator is closed in its domain, we have that $\text{Ker}(M')$ is closed in \mathbf{B}_1 . Clearly $\text{Ker}(M')$ is a Banach space as it is a closed subspace of a Banach space. It is also known that if f is a linear functional on a Banach space X , then either $f \equiv 0$ or $\text{Ran}(f) = \mathbb{R}$. Applying these results, one has that $J' \cdot \text{Ker}(M')$ is either 0 or \mathbb{R} , therefore $J' \cdot \text{Ker}(M')$ is closed in \mathbb{R} . Now we recall a well-known result [36]. Let X, Y, Z be Banach spaces, $A : X \rightarrow Y$ and $B : X \rightarrow Z$ be continuous linear operators. Let $C : X \rightarrow Y \times Z$ be defined as $C(x) = (A(x), B(x))$. If $\text{Ran}(A)$ is closed in Y and $B \cdot \text{Ker}(A)$ is closed in Z , then $\text{Ran}(C)$ is closed in $Y \times Z$. Setting $A = M'$, $B = J'$, $C = N'$, $X = \mathbf{B}_1$, $Y = \mathbf{B}_2$ and $Z = \mathbb{R}$ we have the result.

ii) By contradiction, if the operator N' were onto, there would exist by the implicit function theorem another optimal solution $(\tilde{\mathbf{u}}, \tilde{r}, \tilde{\mathbf{b}}, \tilde{\sigma}, \tilde{\mathbf{B}}_e) \in \mathbf{A}_{ad}$, different from the assumed optimal solution $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$, such that

$$\|\bar{\mathbf{u}} - \tilde{\mathbf{u}}\|_1 + \|\bar{\mathbf{B}}_e - \tilde{\mathbf{B}}_e\|_1 < \epsilon$$

and $\mathcal{J}(\tilde{\mathbf{u}}, \tilde{\mathbf{B}}_e) < \mathcal{J}(\bar{\mathbf{u}}, \bar{\mathbf{B}}_e)$. This is in contradiction of the hypothesis that $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ is an optimal solution. \square

Remark 5. We remark that the operator $N'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ is not onto in the more general case $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) \in \mathbf{B}_1$, but the previous result is sufficient for our purposes.

Now we derive the first-order necessary condition from which an optimality system may be derived.

Theorem 21 (First-order necessary condition). *Let $(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$ denote an optimal solution in the local sense. There exists a nonzero Lagrange multiplier $(a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \in \mathbb{R} \times \mathbf{B}_2^*$ that satisfies the Euler equations*

$$\begin{aligned} & \mathcal{J}'_{aug}(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e, a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e, 0, \mathbf{0}, 0, 0, 0, 0) = \\ & a \mathcal{J}'(\bar{\mathbf{u}}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{\mathbf{B}}_e) + \langle (\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), M'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \rangle = 0 \\ & \quad \forall (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1. \end{aligned} \quad (2.64)$$

Furthermore, if the operator M' is onto, we have $a \neq 0$; thus we may choose $a = 1$ so that there exists a nonzero Lagrange multiplier $(\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \in \mathbf{B}_2^*$ that satisfies the Euler equations

$$\begin{aligned} & \mathcal{J}'_{aug}(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e, 1, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e, 0, \mathbf{0}, 0, 0, 0, 0) = \\ & \mathcal{J}'(\bar{\mathbf{u}}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{\mathbf{B}}_e) + \langle (\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), M'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \rangle = 0 \\ & \quad \forall (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1. \end{aligned} \quad (2.65)$$

Proof. Since $\text{Ran}(N')$ is a closed and proper subspace of $\mathbb{R} \times \mathbf{B}_2$, the Hahn-Banach theorem (see [40], p. 109) implies that there exists a nonzero element of $(\mathbb{R} \times \mathbf{B}_2)^*$ that annihilates the range of $N'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)$, i.e., there exists $(a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \in (\mathbb{R} \times \mathbf{B}_2)^*$ such that

$$\begin{aligned} & \langle (a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), (a_0, \hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \rangle = 0 \\ & \quad \forall (a_0, \hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \text{Range}(N'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e)) \end{aligned} \quad (2.66)$$

Hence the result (2.64) from the definition of N' . Moreover, if the operator M' is onto, one has that $a \neq 0$; in fact, by contradiction, one would have $\langle (\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), (\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \rangle = 0$ for all $(\hat{\mathbf{f}}_1, \hat{q}_1, \hat{\mathbf{f}}_2, \hat{q}_2, \hat{q}_3) \in \text{Ran}(M') \equiv \mathbf{B}_2$, which would imply $(\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) = (\mathbf{0}, 0, \mathbf{0}, 0, 0)$ since M' is onto. This contradicts the previous result stating that $(a, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \neq (0, \mathbf{0}, 0, \mathbf{0}, 0, 0)$. Without any loss of generality, we may normalize so as to have $a = 1$, which yields (2.65). \square

Now we briefly discuss the case $a = 0$. Setting $a = 0$ in the first-order necessary condition 21, one would have $(\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3) \neq (\mathbf{0}, 0, \mathbf{0}, 0, 0)$ such that

$$\begin{aligned} & \langle (\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3), M'(\bar{\mathbf{u}}, \bar{r}, \bar{\mathbf{b}}, \bar{\sigma}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \rangle = 0 \\ & \quad \forall (\hat{\mathbf{u}}, \hat{r}, \hat{\mathbf{b}}, \hat{\sigma}, \hat{\mathbf{B}}_e) \in \mathbf{B}_1. \end{aligned} \quad (2.67)$$

This would bring to an optimality system without any contribution from the Fréchet differential of the cost functional $\mathcal{J}'(\bar{\mathbf{u}}, \bar{\mathbf{B}}_e) \cdot (\hat{\mathbf{u}}, \hat{\mathbf{B}}_e)$. Therefore such a system would retain no information about the objective of the control problem. Its solution would have no physical meaning. Thus, the case $a = 0$ must be avoided in practice. As we know, surjectivity of $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ is a sufficient condition for having $a \neq 0$. One is interested in finding whether surjectivity of $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ occurs without any “ad hoc” assumption. Unfortunately, this is not always possible and some appropriate assumptions may be required for specific optimal control problems [14, 25]. Here we follow an approach proposed in [14] and derive the conditions for M' to be surjective.

Theorem 22. *Except for a countable set of values $(Re, Re_m) \subset \mathbb{R}^2$, the operator $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ is an isomorphism. Moreover, $M'(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ is an isomorphism whenever Re and Re_m are sufficiently small.*

Proof. The proof follows the same lines as in [14], Proposition 3.7. The second result comes from a Neumann series argument. \square

Remark 6. If the values of Re and Re_m are sufficiently small, the operator M' is onto and one may choose $a = 1$ in the optimality system. This assumption is in agreement with the physical model of the Navier-Stokes equations in the absence of a turbulence model, which is known to be appropriate for small Reynolds numbers Re .

2.5.3 Optimality system

With the assumption of small Re and Re_m , one may choose $a = 1$. By studying the stationary points of the augmented functional, one has a possible candidate for the local optimal solution.

Theorem 23 (Optimality system). *Let $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e, 1, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3)$ be a stationary point of the augmented functional $\mathcal{J}_{aug}(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e, 1, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3)$. The variables (\mathbf{B}_e, π_3) in $\mathbf{H}^1(\Omega) \times L_0^2(\Omega)$ satisfy the system*

$$\begin{aligned} & \beta(\mathbf{B}_e, \delta\mathbf{B}_e) + \gamma a(\mathbf{B}_e, \delta\mathbf{B}_e) - S_1 c(\delta\mathbf{B}_e, \mathbf{b} + \mathbf{B}_e, \boldsymbol{\lambda}) - S_1 c(\mathbf{b} + \mathbf{B}_e, \delta\mathbf{B}_e, \boldsymbol{\lambda}) + \\ & \quad \frac{1}{Re_m} a_m(\delta\mathbf{B}_e, \boldsymbol{\xi}) + c(\mathbf{u}, \delta\mathbf{B}_e, \boldsymbol{\xi}) - c(\delta\mathbf{B}_e, \mathbf{u}, \boldsymbol{\xi}) + d(\delta\mathbf{B}_e, \pi_3) = 0, \\ & d(\mathbf{B}_e, \delta\pi_3) = 0 \end{aligned} \tag{2.68}$$

for all test functions $(\delta\mathbf{B}_e, \delta\pi_3) \in \mathbf{H}_{\Gamma_d}^1(\Omega) \times L_0^2(\Omega)$ along with boundary conditions

$\mathbf{B}_e = \Phi_{0,d}$ on Γ_d . The variables $(\boldsymbol{\lambda}, \pi_1)$ in $\mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega)$ are solutions of

$$\begin{aligned} \frac{1}{Re} a(\delta \mathbf{u}, \boldsymbol{\lambda}) + c(\delta \mathbf{u}, \mathbf{u}, \boldsymbol{\lambda}) + c(\mathbf{u}, \delta \mathbf{u}, \boldsymbol{\lambda}) + d(\delta \mathbf{u}, \pi_1) + c(\delta \mathbf{u}, \mathbf{b} + \mathbf{B}_e, \boldsymbol{\xi}) \\ - c(\mathbf{b} + \mathbf{B}_e, \delta \mathbf{u}, \boldsymbol{\xi}) + \alpha(\mathbf{u} - \mathbf{u}_d, \delta \mathbf{u}) = 0, \\ d(\boldsymbol{\lambda}, \delta r) = 0, \end{aligned} \quad (2.69)$$

for all test functions $(\delta \mathbf{u}, \delta r) \in \mathbf{H}_{\Gamma_1}^1(\Omega) \times L^2(\Omega)$. The variables $(\boldsymbol{\xi}, \pi_2)$ in $\mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$ satisfy

$$\begin{aligned} \frac{1}{Re_m} a_m(\delta \mathbf{b}, \boldsymbol{\xi}) + c(\mathbf{u}, \delta \mathbf{b}, \boldsymbol{\xi}) - c(\delta \mathbf{b}, \mathbf{u}, \boldsymbol{\xi}) - S_1 c(\delta \mathbf{b}, \mathbf{b} + \mathbf{B}_e, \boldsymbol{\lambda}) \\ - S_1 c(\mathbf{b} + \mathbf{B}_e, \delta \mathbf{b}, \boldsymbol{\lambda}) + d(\delta \mathbf{b}, \pi_2) = 0, \\ d(\boldsymbol{\xi}, \delta \sigma) = 0 \end{aligned} \quad (2.70)$$

for all test functions $(\delta \mathbf{b}, \delta \sigma) \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega)$.

Proof. It suffices to set to zero the Fréchet differential of the augmented functional $\mathcal{J}_{aug}(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e, 1, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3)$ as in Theorem 21 and then split the independent variations. One has

$$\mathcal{J}'(\mathbf{u}, \mathbf{B}_e) \cdot (\delta \mathbf{u}, \delta \mathbf{B}_e) = \alpha(\mathbf{u} - \mathbf{u}_d, \delta \mathbf{u}) + \beta(\mathbf{B}_e, \delta \mathbf{B}_e) + \gamma a(\mathbf{B}_e, \delta \mathbf{B}_e). \quad (2.71)$$

Clearly, the variations $(\delta \boldsymbol{\lambda}, \delta \pi_1, \delta \boldsymbol{\xi}, \delta \pi_2, \delta \pi_3)$ with respect to the Lagrange multipliers yield the constraints defined by $M(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e) = (\mathbf{f}, 0, \mathbf{0}, 0, 0)$, i.e., the state equations (2.43) and the divergence-free constraint on the control variable \mathbf{B}_e . For the other variables we can proceed in a standard way and obtain the corresponding Euler equations [18, 19, 20, 24, 14]. \square

Remark 7. From the first-order necessary condition, we know that to every local optimal solution $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e)$ there corresponds a stationary point $(\mathbf{u}, r, \mathbf{b}, \sigma, \mathbf{B}_e, 1, \boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2, \pi_3)$ of the augmented functional which satisfies the optimality system. Viceversa, a solution to the optimality system may or may not correspond to a solution to the original optimal control problem. In order to ensure that, one would require a second-order sufficient condition. In this work we choose to search for an optimal solution by numerically solving the optimality system and then by checking if the solution is a good candidate for the optimal control problem.

The optimality system is a very complex system and its numerical solution is a difficult and time-consuming task.

Chapter 3

Finite element approximation of the optimality system

3.1 Finite element method

The numerical solution of the optimal control problems considered in this thesis is very challenging. We consider a *finite element method* for their discretization.

Generally speaking, any finite element method for the solution of differential problems starts from a *weak formulation* or *variational formulation* for the problem at hand. This formulation is characterized by infinite-dimensional *operators* acting on infinite-dimensional *function spaces*. The finite element method consists in the determination of *finite-dimensional* counterparts for the operators and function spaces, where the finite dimensionality is achieved with a suitable geometric partition of the domain.

In order to illustrate the finite element approximation of a variational problem, for the sake of simplicity we refer to the linear mixed problem defined as Problem Q_l in Section 1.3.1. The finite element approximation of the nonlinear mixed Problem Q_n defined in Section 1.3.2 relies on the theory of *branches of nonsingular solutions*, which pertains a wide class of nonlinear problems, including the Navier-Stokes equations. We refer the reader to [13] for details on this theory, which is due to Brezzi, Rappaz and Raviart [6]. Nevertheless, the basic aspects that we illustrate here for the linear problem can be extended to the nonlinear case.

Let h denote a discretization parameter tending to zero and, for each h , let X_h and M_h be two *finite-dimensional* spaces such that

$$X_h \subset X, \quad M_h \subset M. \quad (3.1)$$

Since the finite-dimensional spaces are taken as subsets of the infinite-dimensional counterparts, the approximation is said to be *conforming*. Let X'_h and M'_h denote

the dual spaces of X_h and M_h with the dual norms

$$\|f_h\|_{X'_h} = \sup_{u_h \in X_h} \frac{\langle f_h, u_h \rangle}{\|u_h\|_X}, \quad \|f_h\|_{M'_h} = \sup_{u \in M_h} \frac{\langle f_h, u_h \rangle}{\|u_h\|_M}. \quad (3.2)$$

Clearly, since the supremum is taken over a smaller set in the discrete case,

$$\|l\|_{X'_h} \leq \|l\|_{X'}, \quad \|\chi\|_{M'_h} \leq \|\chi\|_{M'} \quad \forall l \in X', \quad \forall \chi \in M'. \quad (3.3)$$

Like in the continuous case, we associate with $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ the operators $A_h \in \mathcal{L}(X, X'_h)$, $B_h \in \mathcal{L}(X, M'_h)$ and $B'_h \in \mathcal{L}(M, X'_h)$ acting on the continuous spaces and with values in the duals of the discrete spaces, such that

$$\langle A_h u, v_h \rangle = a(u, v_h) \quad \forall u \in X, \quad \forall v_h \in X_h, \quad (3.4)$$

$$\langle B_h u, \mu_h \rangle = b(u, \mu_h) \quad \forall u \in X, \quad \forall \mu_h \in M_h, \quad (3.5)$$

$$\langle B'_h \mu, u_h \rangle = b(u_h, \mu) \quad \forall u_h \in X_h, \quad \forall \mu \in M. \quad (3.6)$$

Strictly speaking, B'_h is not the dual operator of B_h , but, if B_h is restricted to X_h and B'_h to M_h , then B_h and B'_h are indeed dual operators. Also, we have

$$\|A_h u\|_{X'_h} \leq \|A_h u\|_{X'} \quad \forall u \in X, \quad (3.7)$$

$$\|B_h u\|_{M'_h} \leq \|B_h u\|_{M'} \quad \forall u \in X, \quad (3.8)$$

$$\|B'_h \mu\|_{X'_h} \leq \|B_h \mu\|_{X'} \quad \forall \mu \in M. \quad (3.9)$$

For each $\chi \in M'$, we define $V_h(\chi)$ as the finite-dimensional counterpart of $V(\chi)$:

$$V_h(\chi) = \{u_h \in X_h \mid b(u_h, \mu_h) = \langle \chi, \mu_h \rangle \quad \forall \mu_h \in M_h\}, \quad (3.10)$$

and we denote

$$V_h = V_h(0) = \ker(B_h) \cap X_h = \{u_h \in X_h \mid b(u_h, \mu_h) = 0 \quad \forall \mu_h \in M_h\}. \quad (3.11)$$

We remark that in general $V_h \not\subset V$ and $V_h(\chi) \not\subset V(\chi)$ because M_h is a proper subspace of M .

Now, in order to approximate Problem Q_l , we define *Problem $Q_{l,h}$* :

Problem 11. (*Problem $Q_{l,h}$*). Given $l \in X'$ and $\chi \in M'$, find a pair $(u_h, \lambda_h) \in X_h \times M_h$ satisfying

$$a(u_h, v_h) + b(v_h, \lambda_h) = \langle l, v_h \rangle \quad \forall v_h \in X_h, \quad (3.12)$$

$$b(u_h, \mu_h) = \langle \chi, \mu_h \rangle \quad \forall \mu_h \in M_h, \quad (3.13)$$

and we associate with Problem $Q_{l,h}$ the following Problem $P_{l,h}$:

Problem 12. (Problem $P_{l,h}$). Find $u_h \in V_h(\chi)$ satisfying

$$a(u_h, v_h) = \langle l, v_h \rangle \quad \forall v_h \in V_h. \quad (3.14)$$

As $V_h \not\subset V$ and $V_h(\chi) \not\subset V(\chi)$, Problem $P_{l,h}$ may be viewed as an *external* approximation of Problem P_l in Section 1.3.

The operator form for the discrete mixed problem is

$$A_h u_h + B_h' p_h = l \quad \text{in } X_h', \quad (3.15)$$

$$B_h u_h = \chi \quad \text{in } M_h'. \quad (3.16)$$

It is clear that the approximation scheme here defined is *consistent*, in the sense that the solution (u, p) of the continuous problem satisfies the discretized scheme:

$$A_h u + B_h' p = l \quad \text{in } X_h', \quad (3.17)$$

$$B_h u = \chi \quad \text{in } M_h'. \quad (3.18)$$

Now, the question arises whether a solution of the discrete problem exists and whether it is *convergent* to the infinite-dimensional solution. The first answer is given by the following existence theorem. We observe that, also in the finite-dimensional case, if (u_h, λ_h) is a solution of Problem $Q_{l,h}$, then u_h is a solution of Problem $P_{l,h}$. The next theorem yields the condition for the converse to hold.

Theorem 24. 1) Assume that the following conditions hold:

(i) the set $V_h(\chi)$ is not empty;

(ii) there exists a constant $\alpha^* > 0$ such that

$$a(v_h, v_h) \geq \alpha^* \|v_h\|_X^2 \quad \forall v_h \in V_h. \quad (3.19)$$

Then Problem $P_{l,h}$ has a unique solution $u_h \in V_h(\chi)$.

2) Assume that hypothesis (ii) holds and, in addition, that

(iii) there exists a constant $\beta^* > 0$ such that

$$\sup_{v_h \in X_h} \frac{b(v_h, \mu_h)}{\|v_h\|_X} \geq \beta^* \|\mu_h\|_M \quad \forall \mu_h \in M_h. \quad (3.20)$$

Then $V_h(\chi)$ is not empty and there exists a unique $\lambda_h \in M_h$ such that (u_h, λ_h) is the only solution of Problem $Q_{l,h}$.

We observe that, since $V_h \not\subset V$, the coercivity condition (3.19) is not a direct restriction to the finite-dimensional subsets of the coercivity property on the continuous spaces given by (1.57). However, in most practical situations the discrete coercivity is not a major obstacle. Also the discrete inf-sup condition (3.20) cannot be directly deduced from its continuous counterpart (1.55). In this case, the proof is much more delicate and it is often of rather technical nature.

The following theorem is the most important step towards a convergence result, providing upper bounds for the difference between the solutions of the associated continuous and discrete problems.

Theorem 25. 1) Assume that the hypotheses (i) and (ii) in Theorem 24 hold. Then there exists a positive constant C_1 depending only upon α^* , $\|a\|$ and $\|b\|$ such that the following error bound holds:

$$\|u - u_h\|_X \leq C_1 \left\{ \inf_{v_h \in V_h(X)} \|u - v_h\|_X + \inf_{\mu_h \in M_h} \|\lambda - \mu_h\|_M \right\}, \quad (3.21)$$

where (u, λ) is the solution of the corresponding Problem Q_l .

2) Assume that hypotheses (ii) and (iii) in Theorem 24 hold. Then there exists a positive constant C_2 depending only upon α^* , β^* , $\|a\|$ and $\|b\|$ such that

$$\|u - u_h\|_X + \|\lambda - \lambda_h\|_M \leq C_2 \left\{ \inf_{v_h \in X_h} \|u - v_h\|_X + \inf_{\mu_h \in M_h} \|\lambda - \mu_h\|_M \right\}. \quad (3.22)$$

The estimates given by (3.21) and (3.22) are a generalization for mixed problems of the well-known *Céa's lemma* concerning elliptic forms. They express the relationship between the *solutions* of the continuous and discrete problems. Eq. (3.22) claims that the error between (u, λ) and (u_h, λ_h) is *quasi-optimal*, in the sense that it is bounded by a constant times the sum of the *best approximation errors* for u and λ in the spaces X_h and M_h . Note that the best approximation errors do not depend on the operators of the problem, but they are an intrinsic property of the respective continuous and discrete *spaces*.

Now it is evident that *convergence* of the finite element approximation to Problem Q_l given by Problem $Q_{l,h}$ occurs whenever one has

$$\inf_{v_h \in X_h} \|u - v_h\|_X \rightarrow 0 \quad \text{as } h \rightarrow 0, \quad (3.23)$$

$$\inf_{\mu_h \in M_h} \|\lambda - \mu_h\|_M \rightarrow 0 \quad \text{as } h \rightarrow 0, \quad (3.24)$$

and the constant C_2 does not depend on h . To this purpose, the following general convergence result can be stated.

Theorem 26. Let u and u_h be the solutions of Problems P_l and $P_{l,h}$, respectively. Assume that the following hypotheses hold:

1. the form $a(\cdot, \cdot)$ satisfies (3.19) with a constant $\alpha^* > 0$ independent of h ;
2. there exist a dense submanifold $\mathcal{V}(\chi)$ of $V(\chi)$, a dense subspace \mathcal{M} of M and two mappings $r_h : \mathcal{V}(\chi) \rightarrow V_h(\chi)$ and $\rho_h : \mathcal{M} \rightarrow M_h$ with:

$$\lim_{h \rightarrow 0} \|r_h v - v\|_X = 0 \quad \forall v \in \mathcal{V}(\chi), \quad (3.25)$$

$$\lim_{h \rightarrow 0} \|\rho_h \mu - \mu\|_M = 0 \quad \forall \mu \in \mathcal{M}. \quad (3.26)$$

Then

$$\lim_{h \rightarrow 0} \|u - u_h\|_X = 0. \quad (3.27)$$

As a corollary, we have the convergence of the solution of Problem $Q_{l,h}$ to Problem Q_l .

Corollary 2. *Let (u, λ) and (u_h, λ_h) be the solutions of Problems Q_l and $Q_{l,h}$, respectively. We retain the same hypotheses as Theorem 26 on $a(\cdot, \cdot)$ and M and we assume that $b(\cdot, \cdot)$ satisfies the discrete uniform inf-sup condition (3.20). If there exist a dense subspace \mathcal{X} of X and a mapping $r_h : \mathcal{X} \rightarrow X_h$ satisfying:*

$$\lim_{h \rightarrow 0} \|r_h v - v\|_X = 0 \quad \forall v \in \mathcal{X}, \quad (3.28)$$

then

$$\lim_{h \rightarrow 0} (\|u - u_h\|_X + \|\lambda - \lambda_h\|_M) = 0. \quad (3.29)$$

From Theorem (26) and Corollary (2) we observe that the determination of convergent finite element approximations for the abstract linear mixed problem here considered ultimately consists in appropriate choices for the spaces X_h and M_h , so that suitable operators r_h and ρ_h can be constructed.

It may be interesting for the error analysis to determine error estimates in other norms than (3.21). These estimates may turn out to be sharper. To this purpose, let us extend to the case of Problems P_l and $P_{l,h}$ the classical *duality argument* of Aubin and Nitsche. We introduce a Hilbert space H with scalar product (\cdot, \cdot) and associated norm $|\cdot|$ such that $X \subset H$ with continuous and dense embedding. By the Riesz representation theorem, we identify H with its dual space H^* for the scalar product (\cdot, \cdot) . Therefore, H can be considered as a subspace of X^* , $H \subset X^*$, with a continuous and dense embedding.

In order to evaluate $|u - u_h|$, we introduce for each $g \in H$ the unique solution (ϕ_g, ξ_g) of the following problem, which can be interpreted as a *dual* problem of the original linear mixed Problem Q_l :

$$a(v, \phi_g) + b(v, \xi_g) = (g, v) \quad \forall v \in X, \quad (3.30)$$

$$b(\phi_g, \mu) = 0 \quad \forall \mu \in M. \quad (3.31)$$

The estimate in the norm of the space H is given by the following result.

Theorem 27. *Assume that Problem $P_{l,h}$ has a unique solution $u_h \in V_h(\chi)$. Then there exists a constant C , depending only on $\|a\|$ and $\|b\|$, such that:*

$$|u - u_h| \leq C \left\{ \|u - u_h\|_X + \inf_{\mu_h \in M_h} \|\lambda - \mu_h\|_M \right\} \\ \times \sup_{g \in H} \left\{ \frac{1}{g} \left(\inf_{\phi_h \in V_h} \|\phi_g - \phi_h\|_X + \inf_{\xi_h \in M_h} \|\xi_g - \xi_h\|_M \right) \right\}. \quad (3.32)$$

Let us consider as an example the finite element approximation of the Stokes equations given by Problem 3 in Section 1.3. We set the finite-dimensional spaces

$$W_h \subset \mathbf{H}^1(\Omega), \quad Q_h \subset L^2(\Omega). \quad (3.33)$$

Then we define

$$X_h = W_h \cap \mathbf{H}_0^1(\Omega) = \{\mathbf{v}_h \in W_h : \mathbf{v}_h|_{\Gamma} = \mathbf{0}\}, \quad (3.34)$$

$$M_h = Q_h \cap L_0^2(\Omega) = \{q_h \in Q_h \mid \int_{\Omega} q_h \, d\Omega = 0\}. \quad (3.35)$$

With these spaces, the discrete Stokes problem in $Q_{l,h}$ form reads as follows:

Problem 13. *Find a pair $(\mathbf{u}_h, p_h) \in X_h \times M_h$ such that*

$$a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in X_h, \quad (3.36)$$

$$b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in M_h. \quad (3.37)$$

Since it can be shown that the range space of the divergence operator is $L_0^2(\Omega)$, we have $\operatorname{div} \mathbf{u}_h \in L_0^2(\Omega)$. Hence, (3.37) can be written as

$$b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h. \quad (3.38)$$

In view of this remark, the space V_h for the $P_{l,h}$ form of the discrete Stokes problem becomes

$$V_h = \{\mathbf{v}_h \in X_h : b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h\}. \quad (3.39)$$

Hence, the discrete Stokes problem in $P_{l,h}$ form is

Problem 14. *Find $\mathbf{u}_h \in V_h$ such that*

$$a(\mathbf{u}_h, \mathbf{v}_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in V_h. \quad (3.40)$$

In order to construct the spaces X_h and M_h and the operators r_h and ρ_h as required by Theorem 26, for the Stokes problem it is preferable to work with the following set of hypotheses.

Hypothesis (H1). (Approximation property for X_h). There exists an operator $R_h \in \mathcal{L}(\mathbf{H}^2(\Omega); W_h) \cap \mathcal{L}(\mathbf{H}^2(\Omega) \cap \mathbf{H}_0^1(\Omega); X_h)$ and an integer l such that, for a constant C independent of h :

$$\|\mathbf{v} - R_h \mathbf{v}\|_1 \leq Ch^m \|\mathbf{v}\|_{m+1} \quad \forall \mathbf{v} \in \mathbf{H}^{m+1}(\Omega), \quad 1 \leq m \leq l. \quad (3.41)$$

Hypothesis (H2). (Approximation property for Q_h). There exists an operator $S_h \in \mathcal{L}(L^2(\Omega); Q_h)$ and an integer l such that:

$$\|q - S_h q\|_0 \leq Ch^m \|q\|_m \quad \forall q \in H^m(\Omega), \quad 0 \leq m \leq l. \quad (3.42)$$

Hypothesis (H3). (Discrete uniform inf-sup condition). There exists a constant $C' > 0$, independent of h , such that:

$$\sup_{0 \neq \mathbf{u}_h \in X_h} \frac{b(\mathbf{u}_h, q_h)}{\|\mathbf{u}_h\|_1} \geq C' \|q_h\|_0 \quad \forall q_h \in M_h.$$

Hence, we see from these hypotheses that the role of the operators r_h and ρ_h in Theorem 26 is replaced by the operators R_h and S_h . We also remark that Hypotheses (H1) and (H2) are standard approximation properties for the study of elliptic problems, while Hypothesis (H3) arises specifically in the analysis of mixed finite element discretizations.

Then, the convergence Theorem 26 for the Stokes problem can be formulated as follows.

Theorem 28. *Under the hypotheses (H1), (H2) and (H3), Problem 13 has a unique solution $(\mathbf{u}_h, p_h) \in V_h \times M_h$ and \mathbf{u}_h is also the only solution of Problem 14. In addition, (\mathbf{u}_h, p_h) tends to the solution (\mathbf{u}, p) of Problem 3:*

$$\lim_{h \rightarrow 0} (\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0) = 0. \quad (3.43)$$

Furthermore, when (\mathbf{u}, p) belongs to $\mathbf{H}^{m+1}(\Omega) \times (H^m(\Omega) \cap L_0^2(\Omega))$ for some integer $1 \leq m \leq l$, we have the error bound:

$$\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0 \leq Ch^m \{\|\mathbf{u}\|_{m+1} + \|p\|_m\}. \quad (3.44)$$

Various pairs of spaces (X_h, M_h) have been studied in literature in order to satisfy the requirements of Hypotheses (H1), (H2) and (H3). A thorough discussion on different choices of finite element spaces for mixed problems can be found in [13, 5]. In particular, it is well-known that not all choices of finite element spaces are acceptable for guaranteeing that the discrete inf-sup condition in Hypothesis (H3) is fulfilled. Among these pairs of spaces, let us outline the definition of the so-called *Taylor-Hood* family of finite element spaces. The distinguishing feature

of a finite element method lies upon the fact that the construction of the finite-dimensional spaces starts from a partition or triangulation of the domain into *finite elements*. To be more precise, a *triangulation* \mathcal{T}_h of the domain $\bar{\Omega}$ is defined by a set of element domains $\{\kappa_i\}$ (e.g. triangles and/or quadrilaterals in two dimensions, tetrahedra and/or hexahedra in three dimensions) such that the following hold:

- $\bar{\Omega} = \bigcup_{\kappa \in \mathcal{T}_h} \kappa$;
- $\text{int}(\kappa_i) \cap \text{int}(\kappa_j) = \emptyset$ if $i \neq j$, i.e., the intersection of the interiors of distinct elements is empty;
- every κ_i is a bounded closed subset of $\bar{\Omega}$ with nonempty interior and piecewise smooth boundary;
- the triangulation is *geometrically conforming*, i.e. any two non-disjoint elements share exactly either one face, or one side, or one vertex.

In order to measure the regularity of a triangulation, for every element κ we define

$$h_\kappa = \text{diameter of } \kappa, \quad (3.45)$$

$$\rho_\kappa = \sup\{\text{diameter of } B; B \text{ is a ball contained in } \kappa\}. \quad (3.46)$$

We characterize every element by the ratio

$$\sigma_\kappa = \frac{h_\kappa}{\rho_\kappa}. \quad (3.47)$$

The *mesh size* h of the triangulation \mathcal{T}_h is defined as

$$h := \max_{\kappa \in \mathcal{T}_h} h_\kappa. \quad (3.48)$$

A family of triangulations \mathcal{T}_h with respect to the mesh size h is said to be *regular* as h tends to zero if there exists a $\sigma > 0$, independent of h and κ , such that

$$\sigma_\kappa \leq \sigma \quad \forall \kappa \in \mathcal{T}_h. \quad (3.49)$$

Moreover, the family of triangulations is *uniformly regular* or *quasi-uniform* if there exists a constant $\tau > 0$ such that

$$\tau h \leq h_\kappa \leq \sigma \rho_\kappa \quad \forall \kappa \in \mathcal{T}_h. \quad (3.50)$$

A set of basis functions for the finite-dimensional spaces, often referred to as *shape functions*, is constructed starting from a domain triangulation.

Let us consider the Taylor-Hood pair of finite element spaces for the discrete Stokes problem. For the sake of conciseness, let us assume a triangulation $\mathcal{T}_{s,h}$ of

$\bar{\Omega}$ made of N -simplices, namely triangles (2-simplices) or tetrahedra (3-simplices), and let us denote by P_k the set of polynomials on \mathbb{R}^N of degree less than or equal to k . For the pair of spaces (W_h, Q_h) in (3.33) and the associated pair (X_h, M_h) in (3.34) and (3.35) we set

$$W_h = \{\mathbf{v} \in \mathcal{C}^0(\bar{\Omega})^N : \mathbf{v}|_{\kappa} \in P_2^N \quad \forall \kappa \in \mathcal{T}_{s,h}\}, \quad (3.51)$$

$$Q_h = \{q \in \mathcal{C}^0(\bar{\Omega}) : q|_{\kappa} \in P_1 \quad \forall \kappa \in \mathcal{T}_{s,h}\}, \quad (3.52)$$

$$X_h = W_h \cap \mathbf{H}_0^1(\Omega) = \{\mathbf{v} \in \mathcal{C}^0(\bar{\Omega})^N : \mathbf{v}|_{\kappa} \in P_2^N \quad \forall \kappa \in \mathcal{T}_{s,h}, \mathbf{v}|_{\Gamma} = 0\}, \quad (3.53)$$

$$M_h = Q_h \cap L_0^2(\Omega), \quad (3.54)$$

where $\mathcal{C}^0(\bar{\Omega})$ denotes the space of continuous functions on $\bar{\Omega}$. A similar definition of Taylor-Hood spaces for quadrilateral or hexahedral meshes can be given. The space W_h is the well-known Lagrange finite element space of *continuous piecewise-quadratic* functions and it is devoted to the approximation of the velocity field, while Q_h is the space of *continuous piecewise-linear* functions for pressure.

The operators R_h and S_h can be constructed in such a way that all the hypotheses are satisfied. Therefore, we have the following convergence result for the Taylor-Hood family of finite element spaces applied to the Stokes problem. Prior to that, we must make a special hypothesis on the triangulation $\mathcal{T}_{s,h}$.

Hypothesis (T). The triangulation $\mathcal{T}_{s,h}$ has a set of interior nodes $\{a_r\}_{r=1}^R$ such that the set of associated macroelements $\{\Omega_r\}_{r=1}^R$, where each macroelement Ω_r is the union of all elements $\kappa \in \mathcal{T}_{s,h}$ that have a_r as one vertex, is a partition of $\bar{\Omega}$.

Theorem 29. *Let Ω be a bounded, plane polygon and let the solution (\mathbf{u}, p) of the Stokes Problem 3 satisfy*

$$\mathbf{u} \in [\mathbf{H}^k(\Omega) \cap \mathbf{H}_0^1(\Omega)]^N, \quad p \in H^k(\Omega) \cap L_0^2(\Omega), \quad k = 1 \text{ or } 2. \quad (3.55)$$

If the triangulation $\mathcal{T}_{s,h}$ is regular and satisfies Hypothesis (T), the solution (\mathbf{u}_h, p_h) of Problem 13 with the spaces X_h and M_h defined by (3.53) and (3.54) satisfies the estimate

$$\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0 \leq C_1 h^k \{|\mathbf{u}|_{k+1} + |p|_k\}, \quad k = 1 \text{ or } 2. \quad (3.56)$$

When Ω is convex, a refined estimate holds:

$$\|\mathbf{u} - \mathbf{u}_h\|_0 \leq C_2 h^{k+1} \{|\mathbf{u}|_{k+1} + |p|_k\}, \quad k = 1 \text{ or } 2. \quad (3.57)$$

Furthermore, if $\mathcal{T}_{s,h}$ is uniformly regular (but Ω not necessarily convex) we also have:

$$\|p - p_h\|_1 \leq C_3 h^{k-1} \{|\mathbf{u}|_{k+1} + |p|_k\}, \quad k = 1 \text{ or } 2. \quad (3.58)$$

We remark the presence of the sharper estimate (3.57) which is determined as an application of the duality arguments used for Theorem 27. We refer the reader to [4, 8] for a rigorous definition of the concept of *finite element* and for detailed discussions on the construction of finite element spaces.

3.2 Finite element multigrid algorithm

The multigrid algorithm is among the most efficient solvers for the numerical solution of elliptic problems. It has found a considerable interest in literature due to various appealing features. In particular, it proves to be very attractive due to a computational work that is only proportional to the number of degrees of freedom of the discretized system [4, 37, 38]. Here we discuss the fundamentals of the multigrid algorithm in a finite element framework. This algorithm has been implemented for the numerical solution of the equations of the optimality system considered in this work.

For the sake of simplicity, we describe the algorithm for the following model problem. Let $\Omega \subset \mathbb{R}^2$ be a convex polygon; we seek for $u \in V = H_0^1(\Omega)$ such that

$$a(u, v) = (f, v) \quad \forall v \in V, \quad (3.59)$$

where $a(u, v)$ is a bilinear elliptic form on $V \times V$ and $f \in L^2(\Omega)$. By elliptic regularity, we have that $u \in H^2(\Omega) \cap H_0^1(\Omega)$.

In order to formulate the multigrid algorithm we introduce some notations. First, let us consider a sequence of triangulations \mathcal{T}_k of the domain Ω obtained via classical midpoint subdivision starting from the coarser triangulation \mathcal{T}_1 . Let $V_k \subset V$ be the space of continuous piecewise linear functions with respect to \mathcal{T}_k that vanish on $\partial\Omega$. Note that

$$\mathcal{T}_{k-1} \subset \mathcal{T}_k \Rightarrow V_{k-1} \subset V_k \quad \forall k \geq 1. \quad (3.60)$$

Then a finite element discretized form of Problem (3.59) is as follows.

Find $u_k \in V_k$ such that

$$a(u_k, v_k) = (f, v_k) \quad \forall v_k \in V_k. \quad (3.61)$$

Let h_k be the mesh size of \mathcal{T}_k , i.e. $h_k = \max_{T \in \mathcal{T}_k} \text{diam}T$, and let $n_k = \dim V_k$. The goal of a multigrid algorithm is to find a $\hat{u}_k \in V_k$ in $\mathcal{O}(n_k)$ operations, such that it satisfies

$$\|u_k - \hat{u}_k\|_{H^s(\Omega)} < Ch_k^{2-s} \|u\|_{H^2(\Omega)}, \quad s = 0, 1, k = 0, 1, 2, \dots \quad (3.62)$$

Hence, this algorithm is expected to be convergent as the mesh size goes to zero and with a computational complexity that is only proportional to the number of degrees of freedom n_k of the space V_k . Let us now introduce the *mesh-dependent inner product* $(\cdot, \cdot)_k$ on V_k as

$$(v, w)_k = h_k^2 \sum_{i=1}^{n_k} v(p_i) w(p_i), \quad (3.63)$$

where p_i is the set of n_k internal vertices of \mathcal{T}_k . Then, we can define the operator $A_k : V_k \rightarrow V_k$ and the function $f_k \in V_k$ as

$$\begin{aligned} (A_k v, w)_k &= a(v, w) & \forall v, w \in V_k, \\ (f_k, w)_k &= (f, v) & \forall v, w \in V_k. \end{aligned}$$

so that the discretized problem (3.61) can be written in operator form as

$$A_k u_k = f_k. \quad (3.64)$$

Now, the main ingredients to formulate a multigrid algorithm are the intergrid transfer operators, in particular the *prolongation* and *restriction* operators. The prolongation operator $I_{k-1}^k : V_{k-1} \rightarrow V_k$ is defined as the natural injection, i.e.

$$I_{k-1}^k v = v \quad \forall v \in V_{k-1}. \quad (3.65)$$

The restriction operator $I_k^{k-1} : V_k \rightarrow V_{k-1}$ is defined to be the transpose operator of I_{k-1}^k with respect to the mesh-dependent inner product $(\cdot, \cdot)_k$, i.e.

$$(I_k^{k-1} v, w)_{k-1} = (v, I_{k-1}^k w)_k \quad \forall v \in V_k, w \in V_{k-1}. \quad (3.66)$$

We are now in a position to formulate the so-called *k-th level iteration*. The *k-th* level iteration algorithm is denoted as $MG(k, z_0, g)$ and defined as the approximate solution of $A_k z = g$ with initial guess z_0 . In the following we denote with m_1 and m_2 some positive integers and with Λ_k some upper bound for the spectral radius of A_k satisfying $\Lambda_k < Ch_k^{-2}$. This approximate solution is obtained in the following steps

Algorithm. *k-th* level iteration $MG(k, z_0, g)$

- 1: **if** ($k = 1$) **then**
- 2: $MG(1, z_0, g) = A^{-1}g$.
- 3: **else if** ($k > 1$) **then**
- 4: **for** $l = 1, \dots, m_1$ **do**
- 5:

$$z_l = z_{l-1} + \frac{1}{\Lambda_k}(g - A_k z_{l-1}) \quad (\text{pre-smoothing})$$

- 6: **end for**
- 7: $\bar{g} = I_k^{k-1}(g - A_k z_{m_1}), q_0 = 0$
- 8: **for** $i = 1, \dots, p$ **do**
- 9:

$$q_i = MG(k-1, q_{i-1}, \bar{g}). \quad (\text{error correction})$$

- 10: **end for**
- 11: $z_{m_1+1} = z_{m_1} + I_{k-1}^k q_p$.

The analysis of the algorithm convergence and the estimate of the computational complexity can be performed both for the \mathcal{V} - and \mathcal{W} -cycle methods, by considering the so-called *approximation* and *smoothing* properties. We refer the reader to [4] for details.

3.3 Approximation of the optimality system

In this section we approximate the equations of the optimality system derived in Section 2.5.3 as a first-order necessary condition for the optimal control problem considered in this thesis. We use a conforming finite element method as described in Section 3.1. Let $\mathbf{X}_h \subset \mathbf{H}^1(\Omega)$ and $Q_h \subset L^2(\Omega)$ be two families of finite dimensional subspaces parametrized by h that tends to zero. We denote $\mathbf{X}_{h,0} = \mathbf{X}_h \cap \mathbf{H}_0^1(\Omega)$, $Q_{h,0} = Q_h \cap L_0^2(\Omega)$ and, for any subset $\Gamma_s \subset \Gamma$, $\mathbf{X}_{h,\Gamma_s} = \mathbf{X}_h \cap \mathbf{H}_{\Gamma_s}^1(\Omega)$. We choose for \mathbf{X}_h and Q_h the Lagrange finite element spaces of continuous piecewise-quadratic and continuous piecewise-linear functions given by (3.51) and (3.52), respectively. With these finite-dimensional spaces the approximation properties and the inf-sup condition illustrated in Section 3.1 are satisfied. In order to solve the optimal control problem we must solve the optimality system in the variables $(\mathbf{u}_h, r_h, \mathbf{b}_h, \sigma_h, \mathbf{B}_{eh}, \boldsymbol{\lambda}_h, \pi_{1h}, \boldsymbol{\xi}_h, \pi_{2h}, \pi_{3h})$. As in the infinite-dimensional case, we can divide the discrete optimality system into three parts: the state system, the adjoint system and the control equation. The discrete form of the state system (2.46) for the variables $(\mathbf{u}_h, r_h, \mathbf{b}_h, \sigma_h) \in \mathbf{X}_h \times S_h \times \mathbf{X}_{h,0} \times Q_{h,0}$ can be written as

$$\begin{aligned}
& \frac{1}{Re} a(\mathbf{u}_h, \mathbf{v}_{1h}) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_{1h}) - S_1 c(\mathbf{b}_h + \mathbf{B}_{eh}, \mathbf{b}_h + \mathbf{B}_{eh}, \mathbf{v}_{1h}) + \\
& \quad d(\mathbf{v}_{1h}, r_h) = \langle \mathbf{f}_h, \mathbf{v}_{1h} \rangle - \langle \mathbf{t}_h, \mathbf{v}_{1h} \rangle_{\Gamma \setminus \Gamma_1}, \\
& d(\mathbf{u}_h, q_{1h}) = 0, \\
& \frac{1}{Re_m} a_m(\mathbf{b}_h + \mathbf{B}_{eh}, \mathbf{v}_{2h}) + c(\mathbf{u}_h, \mathbf{b}_h + \mathbf{B}_{eh}, \mathbf{v}_{2h}) - \\
& \quad c(\mathbf{b}_h + \mathbf{B}_{eh}, \mathbf{u}_h, \mathbf{v}_{2h}) + d(\mathbf{v}_{2h}, \sigma_h) = 0, \\
& d(\mathbf{b}_h + \mathbf{B}_{eh}, q_{2h}) = 0
\end{aligned} \tag{3.67}$$

for all test functions $(\mathbf{v}_{1h}, q_{1h}, \mathbf{v}_{2h}, q_{2h}) \in \mathbf{X}_{h,\Gamma_1} \times S_h \times \mathbf{X}_{h,0} \times Q_{h,0}$ and boundary conditions $\mathbf{u}_h = \mathbf{g}_h$ on Γ_1 and $\mathbf{b}_h = \mathbf{0}$ on Γ .

The discrete adjoint system corresponding to (2.69) and (2.70), in the variables

$(\boldsymbol{\lambda}_h, \pi_{1h}, \boldsymbol{\xi}, \pi_{2h}) \in \mathbf{X}_{h,\Gamma_1} \times S_h \times \mathbf{X}_{h,0} \times Q_{h,0}$, can be written as

$$\begin{aligned}
& \frac{1}{Re} a(\delta \mathbf{u}_h, \boldsymbol{\lambda}_h) + c(\delta \mathbf{u}_h, \mathbf{u}_h, \boldsymbol{\lambda}_h) + c(\mathbf{u}_h, \delta \mathbf{u}_h, \boldsymbol{\lambda}_h) + d(\delta \mathbf{u}_h, \pi_{1h}) \\
& \quad + c(\delta \mathbf{u}_h, \mathbf{b}_h + \mathbf{B}_{eh}, \boldsymbol{\xi}_h) - c(\mathbf{b}_h + \mathbf{B}_{eh}, \delta \mathbf{u}_h, \boldsymbol{\xi}_h) + \alpha(\mathbf{u}_h - \mathbf{u}_d, \delta \mathbf{u}_h) = 0, \\
& d(\boldsymbol{\lambda}_h, \delta r_h) = 0, \\
& \frac{1}{Re_m} a_m(\delta \mathbf{b}_h, \boldsymbol{\xi}_h) + c(\mathbf{u}_h, \delta \mathbf{b}_h, \boldsymbol{\xi}_h) - c(\delta \mathbf{b}_h, \mathbf{u}_h, \boldsymbol{\xi}_h) \\
& \quad - S_1 c(\delta \mathbf{b}_h, \mathbf{b}_h + \mathbf{B}_{eh}, \boldsymbol{\lambda}_h) - S_1 c(\mathbf{b}_h + \mathbf{B}_{eh}, \delta \mathbf{b}_h, \boldsymbol{\lambda}_h) + d(\delta \mathbf{b}_h, \pi_{2h}) = 0, \\
& d(\boldsymbol{\xi}_h, \delta \sigma_h) = 0
\end{aligned} \tag{3.68}$$

for all test functions $(\delta \mathbf{u}_h, \delta r_h, \delta \mathbf{b}_h, \delta \sigma_h) \in \mathbf{X}_{h,\Gamma_1} \times S_h \times \mathbf{X}_{h,0} \times Q_{h,0}$ with boundary conditions $\boldsymbol{\lambda}_h = \mathbf{0}$ on Γ_1 and $\boldsymbol{\xi}_h = \mathbf{0}$ on Γ .

The discrete form of the optimality condition (2.68), for the variables $(\mathbf{B}_{eh}, \pi_{3h}) \in \mathbf{X}_h \times Q_{h,0}$, takes the form

$$\begin{aligned}
& \beta(\mathbf{B}_{eh}, \delta \mathbf{B}_{eh}) + \gamma a(\mathbf{B}_{eh}, \delta \mathbf{B}_{eh}) + \frac{1}{Re_m} a_m(\boldsymbol{\xi}_h, \delta \mathbf{B}_{eh}) + c(\mathbf{u}_h, \delta \mathbf{B}_{eh}, \boldsymbol{\xi}_h) \\
& \quad - c(\delta \mathbf{B}_{eh}, \mathbf{u}_h, \boldsymbol{\xi}_h) - S_1 c(\delta \mathbf{B}_{eh}, \mathbf{b}_h + \mathbf{B}_{eh}, \boldsymbol{\lambda}_h) \\
& \quad - S_1 c(\mathbf{b}_h + \mathbf{B}_{eh}, \delta \mathbf{B}_{eh}, \boldsymbol{\lambda}_h) + d(\delta \mathbf{B}_{eh}, \pi_{3h}) = 0, \\
& d(\mathbf{B}_{eh}, \delta \pi_{3h}) = 0
\end{aligned} \tag{3.69}$$

for all test functions $(\delta \mathbf{B}_{eh}, \delta \pi_{3h}) \in \mathbf{X}_{h,\Gamma_d} \times Q_{h,0}$ and boundary conditions $\mathbf{B}_{eh} = \Phi_{0d,h}$ on Γ_d . The optimal boundary control Φ_{0h} for the magnetic field is then extracted directly as

$$\Phi_{0h} = \gamma_0(\mathbf{B}_{eh}) \tag{3.70}$$

where γ_0 is the trace operator.

Remark 8. In order to obtain the discrete version of the optimality system, one could also choose to start with the finite element approximation of the state system and introduce discrete constraint operators M_h and N_h corresponding to M and N but acting on finite-dimensional spaces. Then, one can compute the Fréchet differentials of such operators on the finite-dimensional spaces. This possibility would bring some theoretical differences; for instance, some proofs would become trivial in the finite-dimensional framework. Nevertheless, this issue is mainly a matter of taste from a theoretical point of view and the numerical approximations are expected to be similar. Our choice follows the so-called *differentiate-then-discretize* approach [15].

3.4 Gradient algorithm for the optimality system

It is worth noting that the state, adjoint and control variables in the optimality system are tightly coupled. A simultaneous, fully coupled solution of the optimality system on the whole computational domain is unrealistic for the computational load and the memory requirement. Thus, a decoupled solution of these equations is required, even though it brings some numerical oscillations. An accurate *gradient algorithm* can decrease the magnitude of these oscillations and increase the robustness of the solution of the optimality system.

The outline of the gradient algorithm that we consider here reads as follows:

0. Set $\mathbf{B}_{eh}^0, \mathbf{B}_{eh}^{-1}, \omega^0 = 1$ and J^0 to some high value.

For a given optimization step k :

1. Compute $\mathbf{B}_{eh}^k = \mathbf{B}_{eh}^{k-2} + \omega^k(\mathbf{B}_{eh}^{k-1} - \mathbf{B}_{eh}^{k-2})$.
2. Solve the state system (3.67) for $(\mathbf{u}_h^k, r_h^k, \mathbf{b}_h^k, \sigma_h^k)$.
3. Compute the objective functional \mathcal{J}^k .
 - 3.1 **if** $\mathcal{J}^k < \mathcal{J}^{k-1}$, solve the adjoint system (3.68) for $(\boldsymbol{\lambda}_h^k, \pi_{1h}^k, \boldsymbol{\xi}_h^k, \pi_{2h}^k)$ and the control equation (3.69) for $(\mathbf{B}_{eh}^k, \pi_{3h}^k)$, then set $\omega^{k+1} = 1.5 \omega^k$.
 - 3.2 **else if** $\mathcal{J}^k > \mathcal{J}^{k-1}$, set $\omega^{k+1} = 0.5 \omega^k$.
4. Increase k and go to step 1 as far as $|\mathcal{J}^k - \mathcal{J}^{k-1}| > \epsilon$

The control space for the variable \mathbf{B}_{eh} is explored according to a line search strategy with direction given by the difference between two subsequent controls, $(\mathbf{B}_{eh}^{k-1} - \mathbf{B}_{eh}^{k-2})$. The intensity of the search direction is adjusted by the adaptive step size ω^k . This is a relaxation parameter that is employed to either over-relax or under-relax the new control solution. In case the value of the functional \mathcal{J}^k decreases, a new adjoint and control solution is computed and ω^k is increased. On the other hand, when the value of the functional increases the control search direction is not modified and the intensity of ω^k is reduced. With this gradient algorithm the separation between the equations takes into account the couplings typically occurring in an optimality system. In fact, the adjoint variables do not appear directly in the state system and, for this reason, the solution of the state and adjoint equations can be carried out separately. Nonlinear iterations are required for the state equations, thus increasing the complexity of the algorithm. These nonlinear iterations can be performed with classical procedures such as Newton

or Picard methods [13, 34]. The adjoint system and the optimality condition are instead linear in their unknowns, as they are derived from first-order differentiation and the cost functional is quadratic.

We now describe the gradient algorithm in further details. The outer iteration of the optimization loop is denoted as k and the value ϵ^K is the associated convergence tolerance. Superscript d refers to the state (direct) system of the Navier-Stokes and MHD equations; ϵ_1^D and ϵ_2^D are the respective convergence values. The nonlinear iterations for the Navier-Stokes equations are denoted as n with tolerance ϵ^N . For the adjoint and control system we use the index a . The values ϵ_1^A , ϵ_2^A and ϵ_3^A denote the tolerances on the adjoint Navier-Stokes, adjoint MHD and control variables, respectively. The gradient algorithm reads:

Optimization loop (k index):

- Initialize $\mathbf{B}_e^{k=0}$, $\mathbf{B}_e^{k=-1}$, $\omega^{k=0}$, $\mathbf{b}_h^{k=0}$, $\sigma_h^{k=0}$, $\mathbf{u}_h^{k=0}$, $r_h^{k=0}$.
- For $k \geq 1$, solve the optimization loop.

Relaxation of the control:

- Compute $\mathbf{B}_e^k = \mathbf{B}_e^{k-2} + \omega^{k-1}(\mathbf{B}_e^{k-1} - \mathbf{B}_e^{k-2})$

State system (d index):

- Set $(\mathbf{b}_h^{d=0}, \sigma_h^{d=0}) = (\mathbf{b}_h^{k-1}, \sigma_h^{k-1})$.
- For $d \geq 1$, solve the state system.

Navier-Stokes system (n index):

- Set $(\mathbf{u}_h^{n=0}, r_h^{n=0}) = (\mathbf{u}_h^{k-1}, r_h^{k-1})$.
- For $n \geq 1$, solve the Navier-Stokes system with a nonlinear algorithm (here we consider Picard linearization) for the variables $(\mathbf{u}_h^n, r_h^n) \in \mathbf{X}_h \times S_h$

$$\begin{aligned} & \frac{1}{Re} a(\mathbf{u}_h^n, \mathbf{v}_{1h}) + c(\mathbf{u}_h^{n-1}, \mathbf{u}_h^n, \mathbf{v}_{1h}) - S_1 c(\mathbf{b}_h^{d-1} + \mathbf{B}_{eh}^k, \mathbf{b}_h^{d-1} + \mathbf{B}_{eh}^k, \mathbf{v}_{1h}) + \\ & d(\mathbf{v}_{1h}, r_h^n) = \langle \mathbf{f}_h, \mathbf{v}_{1h} \rangle - \langle \mathbf{t}_h, \mathbf{v}_{1h} \rangle_{\Gamma \setminus \Gamma_1}, \\ & d(\mathbf{u}_h^n, q_{1h}) = 0, \end{aligned} \tag{3.71}$$

with $\mathbf{u}_h = \mathbf{g}_h$ on Γ_1 , for all test functions $(\mathbf{v}_{1h}, q_{1h}) \in \mathbf{X}_{h,\Gamma_1} \times S_h$.

- Increase $n = n + 1$ until $\|\mathbf{u}_h^n - \mathbf{u}_h^{n-1}\| + \|r_h^n - r_h^{n-1}\| < \epsilon^N$ for some $n = \bar{n}$, so that convergence of the Navier-Stokes loop is achieved, then set $\mathbf{u}_h^d = \mathbf{u}_h^{\bar{n}}$.

MHD system:

- Solve the linear MHD system for the variables $(\mathbf{b}_h^d, \sigma_h^d) \in \mathbf{X}_{h,0} \times Q_{h,0}$

$$\begin{aligned} & \frac{1}{Re_m} a_m(\mathbf{b}_h^d + \mathbf{B}_{eh}^k, \mathbf{v}_{2h}) + c(\mathbf{u}_h^d, \mathbf{b}_h^d + \mathbf{B}_{eh}^k, \mathbf{v}_{2h}) - \\ & c(\mathbf{b}_h^d + \mathbf{B}_{eh}^k, \mathbf{u}_h^d, \mathbf{v}_{2h}) + d(\mathbf{v}_{2h}, \sigma_h^d) = 0, \\ & d(\mathbf{b}_h^d, q_{2h}) = 0. \end{aligned} \quad (3.72)$$

with $\mathbf{b}_h = \mathbf{0}$ on Γ , for all test functions $(\mathbf{v}_{2h}, q_{2h}) \in \mathbf{X}_{h,0} \times Q_{h,0}$.

- Increase $d = d + 1$ until $\|\mathbf{u}_h^d - \mathbf{u}_h^{d-1}\| + \|r_h^d - r_h^{d-1}\| < \epsilon_1^D$ and $\|\mathbf{b}_h^d - \mathbf{b}_h^{d-1}\| + \|\sigma_h^d - \sigma_h^{d-1}\| < \epsilon_2^D$ for some iteration $d = \bar{d}$. Then, convergence of the overall state system is achieved and one can set $(\mathbf{u}_h^k, r_h^k, \mathbf{b}_h^k, \sigma_h^k) = (\mathbf{u}_h^{\bar{d}}, r_h^{\bar{d}}, \mathbf{b}_h^{\bar{d}}, \sigma_h^{\bar{d}})$.

Computation of the objective functional:

- Compute the functional

$$\mathcal{J}^k(\mathbf{u}_h^k, \mathbf{B}_{eh}^k) = \frac{\alpha}{2} \|\mathbf{u}_h^k - \mathbf{u}_d\|_0^2 + \frac{\beta}{2} \|\mathbf{B}_{eh}^k\|_0^2 + \frac{\gamma}{2} a(\mathbf{B}_{eh}^k, \mathbf{B}_{eh}^k). \quad (3.73)$$

Adjoint and control system (a index):

- If $\mathcal{J}^k < \mathcal{J}^{k-1}$, set $\mathbf{B}_{eh}^{a=0} = \mathbf{B}_{eh}^k$ and $\boldsymbol{\xi}_h^{a=0} = \boldsymbol{\xi}_h^{k-1}$ and solve the adjoint and control equations for $a \geq 1$.

- Solve the linear *adjoint Navier-Stokes system* for $(\boldsymbol{\lambda}_h^a, \pi_{1h}^a) \in \mathbf{X}_{h,\Gamma_1} \times S_h$

$$\begin{aligned} & \frac{1}{Re} a(\delta \mathbf{u}_h, \boldsymbol{\lambda}_h^a) + c(\delta \mathbf{u}_h, \mathbf{u}_h^k, \boldsymbol{\lambda}_h^a) + c(\mathbf{u}_h^k, \delta \mathbf{u}_h, \boldsymbol{\lambda}_h^a) + d(\delta \mathbf{u}_h, \pi_{1h}^a) \\ & + c(\delta \mathbf{u}_h, \mathbf{b}_h^k + \mathbf{B}_{eh}^{a-1}, \boldsymbol{\xi}_h^{a-1}) - c(\mathbf{b}_h^k + \mathbf{B}_{eh}^{a-1}, \delta \mathbf{u}_h, \boldsymbol{\xi}_h^{a-1}) + \alpha(\mathbf{u}_h^k - \mathbf{u}_d, \delta \mathbf{u}_h) = 0, \\ & d(\boldsymbol{\lambda}_h^a, \delta r_h) = 0, \end{aligned} \quad (3.74)$$

with $\boldsymbol{\lambda}_h = \mathbf{0}$ on Γ_1 , for all test functions $(\delta \mathbf{u}_h, \delta r_h) \in \mathbf{X}_{h,\Gamma_1} \times S_h$.

- Solve the linear *adjoint MHD system* for $(\boldsymbol{\xi}_h^a, \pi_{2h}^a) \in \mathbf{X}_{h,0} \times Q_{h,0}$

$$\begin{aligned} & \frac{1}{Re_m} a_m(\delta \mathbf{b}_h, \boldsymbol{\xi}_h^a) + c(\mathbf{u}_h^k, \delta \mathbf{b}_h, \boldsymbol{\xi}_h^a) - c(\delta \mathbf{b}_h, \mathbf{u}_h^k, \boldsymbol{\xi}_h^a) \\ & - S_1 c(\delta \mathbf{b}_h, \mathbf{b}_h^k + \mathbf{B}_{eh}^{a-1}, \boldsymbol{\lambda}_h^a) - S_1 c(\mathbf{b}_h^k + \mathbf{B}_{eh}^{a-1}, \delta \mathbf{b}_h, \boldsymbol{\lambda}_h^a) + d(\delta \mathbf{b}_h, \pi_{2h}^a) = 0, \\ & d(\boldsymbol{\xi}_h^a, \delta \sigma_h) = 0 \end{aligned} \quad (3.75)$$

with $\boldsymbol{\xi}_h = \mathbf{0}$ on Γ , for all test functions $(\delta \mathbf{b}_h, \delta \sigma_h) \in \mathbf{X}_{h,0} \times Q_{h,0}$.

- Then, solve the *control equation* for the variables $(\mathbf{B}_{eh}^a, \pi_{3h}^a) \in \mathbf{X}_h \times Q_{h,0}$

$$\begin{aligned} & \beta(\mathbf{B}_{eh}^a, \delta \mathbf{B}_{eh}) + \gamma a(\mathbf{B}_{eh}^a, \delta \mathbf{B}_{eh}) + \frac{1}{Re_m} a_m(\boldsymbol{\xi}_h^a, \delta \mathbf{B}_{eh}) + c(\mathbf{u}_h^k, \delta \mathbf{B}_{eh}, \boldsymbol{\xi}_h^a) \\ & - c(\delta \mathbf{B}_{eh}, \mathbf{u}_h^k, \boldsymbol{\xi}_h^a) - S_1 c(\delta \mathbf{B}_{eh}, \mathbf{b}_h^k + \mathbf{B}_{eh}^a, \boldsymbol{\lambda}_h^a) \\ & - S_1 c(\mathbf{b}_h^k + \mathbf{B}_{eh}^a, \delta \mathbf{B}_{eh}, \boldsymbol{\lambda}_h^a) + d(\delta \mathbf{B}_{eh}, \pi_{3h}^a) = 0, \\ & d(\mathbf{B}_{eh}^a, \delta \pi_{3h}) = 0 \end{aligned} \quad (3.76)$$

with $\mathbf{B}_{eh} = \Phi_{0d,h}$ on Γ_d , for all test functions $(\delta \mathbf{B}_{eh}, \delta \pi_{3h}) \in \mathbf{X}_{h,\Gamma_d} \times Q_{h,0}$.

- Increase $a = a + 1$ until $\|\boldsymbol{\lambda}_h^a - \boldsymbol{\lambda}_h^{a-1}\| + \|\pi_{1h}^a - \pi_{1h}^{a-1}\| < \epsilon_1^A$ and $\|\boldsymbol{\xi}_h^a - \boldsymbol{\xi}_h^{a-1}\| + \|\pi_{2h}^a - \pi_{2h}^{a-1}\| < \epsilon_2^A$ and $\|\mathbf{B}_{eh}^a - \mathbf{B}_{eh}^{a-1}\| + \|\pi_{3h}^a - \pi_{3h}^{a-1}\| < \epsilon_3^A$ for some $a = \bar{a}$.

- Set $(\boldsymbol{\lambda}_h^k, \pi_{1h}^k, \boldsymbol{\xi}_h^k, \pi_{2h}^k, \mathbf{B}_{eh}^k, \pi_{3h}^k) = (\boldsymbol{\lambda}_h^{\bar{a}}, \pi_{1h}^{\bar{a}}, \boldsymbol{\xi}_h^{\bar{a}}, \pi_{2h}^{\bar{a}}, \mathbf{B}_{eh}^{\bar{a}}, \pi_{3h}^{\bar{a}})$. Also, set $\omega^{k+1} = 1.5 \omega^k$ and $\mathcal{J}^{k+1} = \mathcal{J}^k$.

Under-relaxation:

- If $\mathcal{J}^k > \mathcal{J}^{k-1}$, then set $\omega^{k+1} = 0.5 \omega^k$ and do not update the value of the functional ($\mathcal{J}^{k+1} = \mathcal{J}^{k-1}$).

New optimization loop:

- Increase $k = k + 1$ until $|\mathcal{J}^k - \mathcal{J}^{k-1}| < \epsilon^K$.

We remark that the update of the lifting function must not be performed on all the nodes of the computational domain. In fact, the lifting function extends both the fixed and the variable boundary conditions. Thus the lifting function must not be modified on the boundary nodes of Γ_d , where a fixed boundary condition for the magnetic field is enforced.

Chapter 4

Computational implementation of the optimality system

In order to solve the discretized optimality system described in Chapter 3, the *FEMuS* finite element multiphysics library has been developed. In Section 4.1 we illustrate the general structure and purpose of the library. The configuration and the link with external libraries are discussed in Section 4.2. Section 4.3 is devoted to the description of the most important library classes. In Section 4.4 we discuss how these classes are used for the implementation of a multiphysics application such as the optimality system here considered.

4.1 The *FEMuS* library

The numerical implementation of the optimality system has been performed as an application of the finite element *FEMuS* library (*Finite Element Multiphysics Solver*) developed at the Laboratory of Montecuccolino at the University of Bologna. In its early stages, the *FEMuS* project was born as an example application of the *LibMesh* open source finite element library [28]. At present, *FEMuS* is a stand-alone library that links against various advanced external libraries such as *LibMesh* itself, PETSc, MPI and HDF5.

The *FEMuS* library is devoted to the finite element solution of partial differential equations. Special attention is devoted to multi-physics problems of various kind such as two-phase flows, thermo-fluidynamics, MHD and fluid-structure interaction. In these applications a set of coupled equations has to be solved and a common framework of physical data and computational structures has to be set. The optimality system that we consider in this thesis belongs to this class of problems.

The purpose of the *FEMuS* library is to provide a common set of classes and

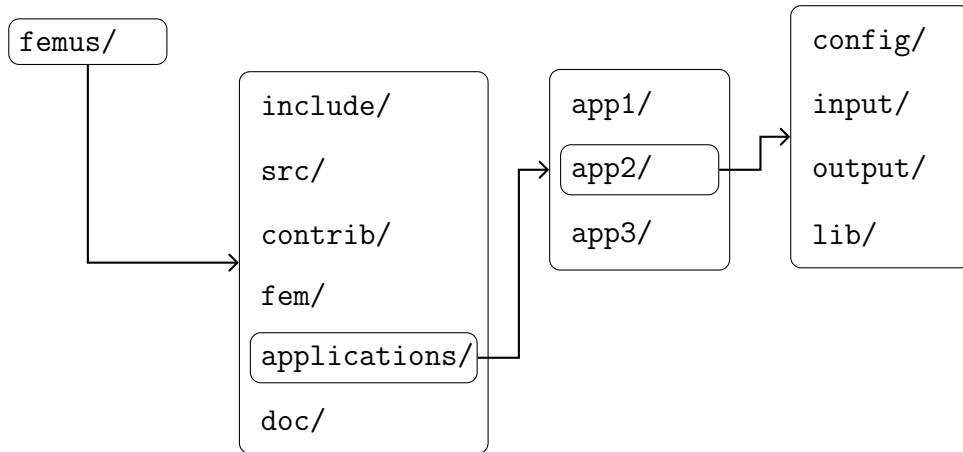


Figure 4.1: *FEMuS* directory structure.

functions that can be used for the implementation of a generic finite element multiphysics application. In this way the code development for a specific application is restricted to a number of functions which should be as small as possible (functions strictly related to a specific domain geometry, implementation of specific boundary conditions for a certain equation, etc.). All the other functionalities should be provided by the library.

Given the complexity of the problems at hand, the library has been developed with particular attention to the modularity and the optimization of memory requirements and execution time. In order to meet these conditions the C++ language has been chosen, which is characterized by an object-oriented paradigm and by several forms of polymorphism such as type inheritance, generic programming and operator overloading. These features are particularly appealing as they improve code readability, allow code reusability and do not engender considerable runtime overhead at code execution.

In order to understand the main principles underlying the development of the *FEMuS* library, we briefly describe its directory structure. This is reported in Fig. 4.1. The files are divided into the following sub-directories:

- *include*, containing the header files;
- *src*, containing the source files;
- *contrib*, containing modified code of external open-source libraries;
- *applications*, containing the programs using the library;
- *fem*, containing the shape function values and weight values for the supported Gauss quadrature rules and canonical elements;

- *doc*, containing the library documentation.

The *applications* folder contains all the programs that make use of the library: finite element simulations and utility applications. Each of them is basically characterized by a *main()* function and specific source files. The folder of every finite element simulation has the following sub-directories:

- *config*, containing the configuration files;
- *input*, for the mesh and multigrid files;
- *output*, for the output files of a simulation;
- *lib*, for the library object file as compiled for the specific application.

Let us now give a brief description of the content of the various *FEMuS* directories.

The *include* and *src* directories

In this folder one can find the declarations and implementations of the library classes and of the other data structures (namespaces, enumerations, etc.).

The *include* directory contains the header files (*.h* suffix) where the declarations of the library classes are given. They also contain the implementations of inline functions and templates whose object code is generated only by implicit instantiation. A separate header file is usually created for every class.

The *src* directory has the source files (*.C* suffix) that define the functions declared by the headers. For example, the *Mesh.h* file contains the declaration of the data members and member functions of the *Mesh* class, while in the *Mesh.C* file one can find the implementations of these functions.

The *contrib* and *fem* directories

The *contrib* directory contains the source code that was taken from external open-source libraries and that was modified in order to meet some specific purposes. It also contains few other auxiliary open source external contributions. For instance, in the *laspack* subfolder one can find a slightly modified version of the Laspack library. This is a linear algebra package for the solution of sparse linear systems in scalar architectures. Some solvers have been added to this library.

The *fem* directory consists of plain text files containing the values of the shape functions and derivatives and the weight values at the Gaussian points of the supported Gauss quadrature rule and for the supported finite element families. The Lagrange quadratic and linear finite elements in all space dimensions are supported

at present, for hexahedral/quadrilateral (HEX27, HEX8, QUAD9, QUAD4), tetrahedral/ triangular (TET10, TET4, TRI6, TRI3) and edge (EDGE3, EDGE2) geometric element shapes. The fifth-order Gauss quadrature rule is implemented, which is exact for the given finite element families.

The names of the *fem* files have the form *shape(S)-(N)D-(FF).in*. In the name of each file, *S* is a number with values *0* for hexahedral/quadrilateral elements or *1* for the tetrahedral/triangular ones. The digit *N* is the space dimension. The digits denoted as *(FF)* indicate the number of element shape functions. Hence, for instance, the file *shape0-3D-8.in* contains the values for the Lagrange HEX8 finite element (hexahedral shape in 3 dimensions with 8 shape functions).

These files can be generated by the *genfegauss* program provided with the library. They are used by a generic application of the library for the computation of integrals which characterize the variational form of a finite element problem.

The *applications* directory

The classes and functions defined and implemented in the files of the *include* and *src* directories can be used for writing different applications. Each application can be implemented in a subfolder in the *applications* directory. One must provide a *main()* function and possibly other specific routines that characterize the particular finite element simulation.

The *applications* directory also contains the *gencase* and *genfegauss* utility applications. The *gencase* program is used for the generation of the mesh and multi-grid operator files of every finite element simulation in the pre-processing stage. It makes use of the LibMesh mesh generation functions. The *genfegauss* directory contains the *genfegauss* program that generates the files containing the Gauss quadrature values provided in the *fem* directory. These files are read by every finite element application at run-time.

4.2 Configuration of the *FEMuS* library

4.2.1 External libraries

Various external libraries are used by the *FEMuS* library in order to perform operations such as linear algebra solution, mesh generation, input/output.

The following external libraries are required for the use of the *FEMuS* library, as can be seen in Figure 4.2:

- an MPI implementation (such as OpenMPI), for parallel execution;
- PETSc, for parallel linear algebra solvers;

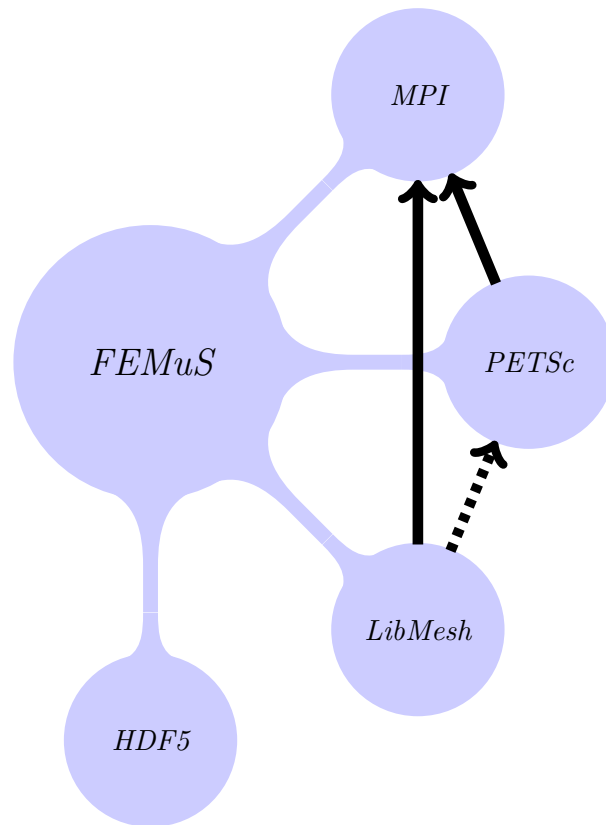


Figure 4.2: External libraries required by *FEMuS*. The dashed arrow from *LibMesh* to *PETSc* indicates that the *PETSc* libraries are not called by *FEMuS* through the interface given by *LibMesh* but they are used directly.

- *LibMesh*, for the generation of mesh files and multigrid operator files;
- *HDF5*, for file input/output tasks.

An *MPI* implementation is required for the installation of the parallel communication features of both *PETSc* and *LibMesh*, and it is also used by *FEMuS* directly. In order to avoid incompatibility issues, it is recommended to use the same *MPI* library for *FEMuS*, *PETSc* and *LibMesh*.

The *PETSc* library performs all the linear algebra operations for parallel architectures. It provides the implementation of numerous linear and nonlinear solvers and preconditioners. The *LibMesh* package is an open-source C++ finite element adaptive library. It provides a very rich list of functionalities for finite element simulations, with support for various finite element families, adaptive mesh refinement, various mesh file formats and mesh geometric elements. Some of the features of the present version of the *FEMuS* library have been developed by tak-

ing inspiration from the LibMesh implementations, such as a unified interface for the handling of different linear algebra packages. At the present stage, the *FEMuS* library requires the installation of LibMesh only for the mesh generation features. The rest of the *FEMuS* code does not perform LibMesh calls. Therefore, the link with PETSc libraries is direct and does not pass through LibMesh calls (see Figure 4.2).

We also remark that LibMesh comes bundled with a list of contribution libraries, most of which are not required for a proper use of the *FEMuS* library. In particular, only the Metis and Parmetis contribution libraries need to be enabled in the LibMesh installation. Concerning the file input/output (I/O) operations in *FEMuS*, these are performed with the use of the HDF5 library. This library defines a binary data format with hierarchical structure that permits a flexible and modular handling of data structures as well as a much faster I/O communication than plain text ASCII file format. All the mesh files and solution outputs in the *FEMuS* routines are generated in the HDF5 format, together with an associated plain text XDMF wrapper file. The XDMF files are required for visualizing the data contained in the HDF5 files in a scientific visualization software. The XDMF format permits to define the association between a mesh file and the corresponding field values. Both the mesh and the field data are contained in HDF5 files. In HDF5 files the values are stored in a hierarchical structure and a compressed storage format. The XDMF format can be interpreted by some scientific visualization software such as the open source *ParaView* application. This application is based on the VTK library that provides various tools for data visualization. *ParaView* is designed for viewing data obtained from parallel architectures and distributed or shared memory computers, but it can also be used for serial platforms.

The *FEMuS* library has been installed so far on both Linux and MAC/OS platforms and it has been successfully compiled with the compilers provided with the *Gnu Compiler Collection* (GCC) with the use of the GNU Make utilities. The Git version control software has been used for tracking the code development stages. We suggest the use of the Valgrind memory check utility for tracking memory leaks, segmentation faults and other memory related issues. Also, the raw data and hierarchical structure of the binary HDF5 files can be viewed with the useful HDFVIEW software. These use of these tools is recommended for helping the development of *FEMuS* classes and applications.

4.2.2 Internal configuration

In order to install the *FEMuS* library, a first configuration step is needed. At the present status, the configuration consists in setting a list of shell environment variables. Some of these variables must be defined in a shell terminal and the others are set afterwards by sourcing a configuration script.

The configuration script *configure_femus.sh* in the main *femus* directory must be edited by the user according to the specific machine where the *FEMuS* library is being installed. The purpose of this script is to define the locations of the external libraries in the current machine, so that a proper installation of the *FEMuS* library can be performed. After running the configuration script, all the informations related to the installation paths of the external libraries are provided to the *FEMuS* Makefiles, so that the library and its applications can be successfully compiled and run. In the future we plan to use advanced tools such as *Cmake* or the *GNU Autotools* in order to generate automatically the configuration scripts and the Makefiles for each specific machine.

Therefore, the configuration consists of the following steps:

- set some environment variables in a shell terminal;
- edit the configuration script *configure_femus.sh*;
- run the configuration script.

First of all, one must open a shell environment and set the environment variables `FM_FEMUS_METHOD`, `FM_PETSC_METHOD`, `FM_LIBMESH_METHOD` and the variable `FM_MYMACHINE`. The first three variables define the compiling mode for the *FEMuS*, PETSc and LibMesh libraries respectively. The values that can be assumed by each of them are *opt* (optimized mode) or *dbg* (debug mode). The use of debugging mode is recommended for the development of the code as it helps the developer in finding errors in the implementation of the program. The optimized mode is much faster as many debugging function calls are excluded and it is used for performing the simulations. In order to avoid possible incompatibility issues, it is suggested that the same compiling mode be used both for *FEMuS* and for all the involved external libraries. However, different compile modes can be used in some cases. For instance, no incompatibility issues have been encountered so far when the *FEMuS* library has been used in debug mode and the LibMesh and PETSc libraries in optimized mode. This is a favorable condition in terms of library development as it allows one to perform the debugging of *FEMuS* functions without adding a considerable time overhead for the debug execution of the PETSc libraries.

The `FM_MYMACHINE` variable is used to identify a particular list of variables in the configuration script. This list of variables defines the installation paths and other information concerning the external libraries for the specific machine where *FEMuS* is being installed. The names of the respective folders are set in the variables `FM_HDF5_FOLDER`, `FM_LM_FOLDER`, `FM_PETSC_FOLDER` and `FM_MPI_FOLDER`.

When the configuration script has been modified, the user must launch the command

```
source configure_femus.sh
```

With the *source* command the variables defined in the configuration script after the *export* command are correctly exported to the shell environment.

After the configuration script has been run, one has to choose the application he intends to work on by setting the `FM_MYAPP` environment variable. For instance, the command

```
export FM_MYAPP=optsys
```

is employed to choose the application of the *FEMuS* library that is contained in the *optsys* folder under the *applications* subdirectory. If one wants to change the application, one simply has to set the environment variable with the name of the new application. No further run of the configure script is required.

After choosing the application with the previous command, one can move to the folder and work in it. The *FEMuS* library will be compiled for the specific application. By typing “make” one can compile the library and also create the application executable. The compilation of the library is related to the specific application, in order to remove pieces of code that are not relevant for that application itself.

4.3 Library main classes

We give here a description of the main classes of the *FEMuS* library in terms of their content and structure. In subsequent paragraphs, we show how these classes are used and configured.

We list here the most important classes:

- *EquationsMap*, that defines a common interface for handling a map of equations together with all the objects necessary to complete them;
- *EqnBase*, for the definition of the basic structures that characterize an equation;
- *Quantity*, for the basic definition of a physical quantity entering in the equations as an unknown or as a given function;
- *QuantityMap*, for handling the quantities involved in each application;
- *Mesh*, for storing the necessary mesh information such as coordinates and connectivity;
- *GeomEl*, for the definition of the geometric element composing the mesh;

- *Domain*, in case the domain has a well-defined shape;
- *FEElemBase* and its child classes, for the definition of canonical finite elements;
- *FEGauss*, for holding values of shape functions and derivatives at Gauss points;
- *Utils*, that provides common utilities;
- *Files*, for the handling of input/output file paths;
- *Physics*, for specifying physical data;
- *GenCase*, for the generation of mesh files and multigrid operator files;
- *EqnQL*, a child class of the *EqnBase* equation class that provides various functionalities for handling quadratic and linear unknowns;
- *Vect*, for handling values of various operators at Gauss points;
- *RunTimeMap*, for class runtime configuration.

4.3.1 The *EquationsMap* class

A major goal of the *FEMuS* library is to provide an interface for managing an arbitrary number of *equations* involving an arbitrary number of *physical quantities*. In the numerical solution of a multi-physics problem, where numerous data and informations have to be shared among different equations, it may be very convenient to set up a computational structure that enables a flexible definition of equations and quantities and provides an interface to retrieve all the data required by each equation.

To this purpose, the *EquationsMap* class has been defined. It contains a map to hold all the equations of each specific application and it also gathers all the class objects required to implement a finite element discretization. As shown in Figure 4.3, the objects that compose the *EquationsMap* class are:

- *Physics*, for specific physical data;
- *QuantityMap*, for the internal and external involved quantities;
- *Mesh*, for the domain triangulation;
- *TimeLoop*, for time discretization;

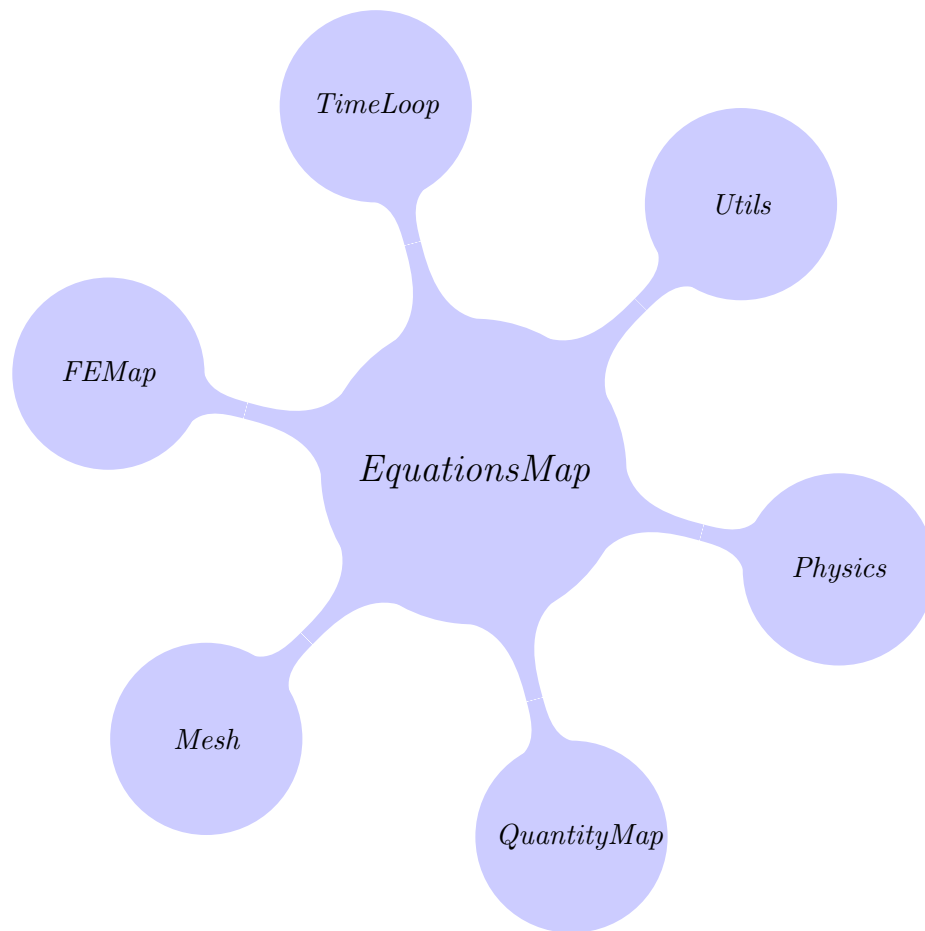


Figure 4.3: Overview of the *EquationsMap* class and the classes that compose it.

- *FEMap*, for space discretization;
- *Utils*, for various utilities.

An instantiation of each of these objects is created and then passed to the constructor of the class. Each specific equation implemented by the user has a reference to the *EquationsMap* object, so that all the equations can have access to the required objects.

The arguments of the constructor of the *EquationsMap* class are intended to be as general as possible, while retaining the possibility for specific characterizations. Through the use of the *std::map* template class from the C++ Standard Template Library, a map of pointers to base *EqnBase* objects and a map of pointers to base *Quantity* objects are defined. Then, the user that defines specific child classes of the *EqnBase* and *Quantity* types can easily add these child instantiations to

the corresponding maps. We remark that, when child pointers are added to a map of father pointers, they apparently lose their property of being child pointers. Nevertheless, a conversion back to the child type can be made with C++ cast functionalities such as `static_cast()`. Similar considerations can be made for the *Physics* class which is intended to be again a base class from which particular child classes can be implemented.

4.3.2 The *EqnBase* class

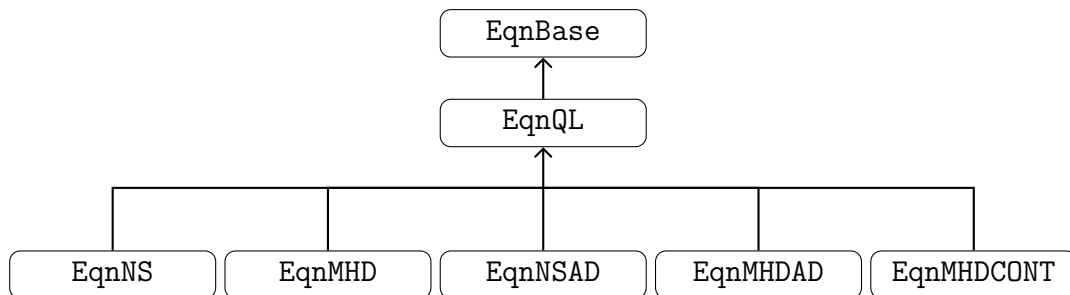


Figure 4.4: An example of inheritance diagram with the *EqnBase* class as the father class.

The *EqnBase* class is intended to be a base class for the construction of specific equations. It provides the essential data structures and routines for the definition of an equation. These include:

- basic equation attributes;
- map of the unknown quantities of the equation;
- linear algebra structures, i.e. vectors, matrices, prolongation and restriction operators for the multigrid algorithm;
- the map of the degrees of freedom;
- initial and boundary conditions.

The constructor of every child equation class must call the *EqnBase* constructor. The base part of every equation is constructed by passing a vector that contains the pointers to the unknown quantities (child classes of the *Quantity* base class). Also, a reference to the *EquationsMap* object is passed to the constructor, so that each equation has access to all the necessary objects like mesh and physical data. From the vector of unknown quantities, the number of scalar unknowns of the system as well as their finite element order (e.g. Lagrange quadratic or linear) is determined, together with the dimension of vectors, matrices and multigrid operators. The so-called *DoF map*, an array which defines the association between mesh nodes

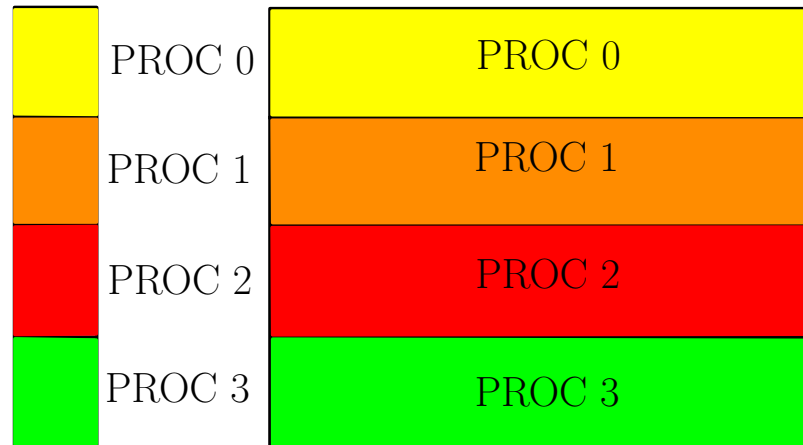


Figure 4.5: The subdivision of vectors and matrices as required by PETSc. Vectors must be divided into blocks of contiguous elements, while matrices must be split into blocks of contiguous rows. Each block is assigned to a different processor.

and degrees of freedom (DoF) of the linear system can then be constructed for all multigrid levels. This map is based on the mesh domain decomposition for parallel computing obtained in the preprocessing stage with the *gencase* application.

In the multigrid framework, each level has its own set of vectors, matrices and multigrid operators. For each level, these structures are divided in P parts, where P is the number of processors that is chosen for a parallel run. In a domain decomposition approach, each part is associated to a single portion of the computational domain. As required by the PETSc libraries, the subdivision of vectors is then performed in such a way that degrees of freedom associated to a given subdomain are contiguous in the DoF indexing. Accordingly, matrices are subdivided so that the DoF's associated to a given subdomain correspond to a contiguous block of rows (see Figure 4.5). Once the DoF map is filled, all the structures at all multigrid levels can be initialized.

The *EqnBase* class also allows the implementation of two different methods for enforcing the boundary conditions: the *nodal flag* method or the *element flag* method. With the nodal flag method, a flag is associated to each degree of freedom of the system, both on the boundary and in the interior of the domain. The *_nodal_bc* integer array in the *EqnBase* class is devoted to hold these nodal flag values. This array can be filled in various ways according to the specific requirements of each equation. Let us consider for instance the case of the Navier-Stokes equation. The flag is set to 1 for all the degrees of freedom in the interior of the domain. For the velocity degrees of freedom, the flag can assume a value equal to either 0 or 1 for Dirichlet or Neumann boundary conditions, respectively. For pressure, a flag equal to 0 indicates that the pressure boundary integral must be

computed, and a value 1 is used otherwise. The pure virtual function *bc_read* of the *EqnBase* class must be implemented in each child equation depending on the particular boundary mesh geometry and is devoted to the assignment of the nodal flags in the *_nodal_bc* array. The *bc_read* function receives the coordinates that are used to identify the node to which the flags must be associated. We remark that in the case of mixed saddle point equations, such as the Navier-Stokes equations involving velocity and pressure, not all the choices of boundary conditions are consistent with the variational formulation of the equation. It is up to the user to check the consistency of the boundary condition flags for the involved unknowns.

So far we have considered the flags that determine the *kind* of boundary condition to be enforced. The corresponding values can be retrieved from the implementation of the pure virtual function *ic_read*, which must be implemented by the user for setting the initial conditions. Clearly, the initial condition values on the boundary correspond to the boundary condition values (time-dependent boundary conditions are not considered here).

The *element flag* method for the boundary conditions associates instead some flags to the mesh boundary elements. An *_elem_bc* array and a corresponding virtual function called *elem_bc_read* are used. In the case of vector valued unknowns, it may be more convenient to use an element flag method rather than a nodal method. A boundary element has in fact a natural definition of normal and tangential directions, while ambiguities can arise for nodes, especially at the intersection of two boundary surfaces. The user can decide what type of boundary flag implementation to use according to his needs.

4.3.3 The *Quantity* class

This class is the base class for the definition of a physical quantity to be introduced in each application. In order for a *Quantity* object to be fully defined, one simply has to provide:

- a name (in the *_name* string), so that the quantity can be easily identified;
- the number of scalar components (*_dim*);
- the finite element family (denoted by an integer index, *_FEord*);
- the reference values for all its scalar components (*_refvalue* pointer);
- the pointer to the associated equation (*_eqn*);
- a reference *_qtypmap* to the map to which the *Quantity* belongs.

The finite element family determines the approximation order of the *Quantity*. The support for Lagrange quadratic and linear families has been implemented so far in the library.

The main goal of the *Quantity* base class is to define a physical quantity that participates in the system of equations and to determine a flexible way to retrieve its DoF values for performing the necessary Gauss point interpolations. The DoF values for a *Quantity* can be retrieved in either of the following ways:

- with an *associated equation*;
- with a *user provided function*.

The associated equation of a *Quantity* is the equation to which that quantity is an unknown. The *_eqn* pointer is used to store this association, which however is not mandatory. In the case where no equation is associated to the quantity, the pointer is simply set to NULL. Then, the quantity does not correspond to any equation unknown and it represents a *given external field*. In that case, the user must provide an implementation of the *FunctionDof* function, in which an arbitrary function of time and space variables can be inserted. In that case, the degrees of freedom are not taken from an unknown vector of an equation but they are simply evaluated with the given function. This possibility of switching the method for retrieving the degrees of freedom of a quantity based on the content of the *_eqn* pointer allows the user to set up a flexible implementation of his equations, where the involved quantities can be either given a priori or computed according to some equation model. Numerical simulations of physical phenomena with different modeling complexities can then be easily performed.

4.3.4 The *Vect* class

The *Vect* class is a small class that is used for performing Gauss point interpolations. The values of the degrees of freedom to be interpolated are stored in the *_val.dofs* array. The interpolations are performed with the functions of the *EqnBase* class

```
void gradVect_g(const uint vbflag, Vect& myVect ) const;
void funcVect_g(const uint vbflag, Vect& myVect )const;
void curlVect_g(const uint vbflag, Vect& myVect ) const;
```

According to the function that is called, a corresponding Gauss array in the *Vect* class is filled. Therefore, *funcVect_g* fills the *val_g* array, *gradVect_g* is used for *grad_g* and *curlVect_g* computes the curl, that goes in the *curl_g3D* array. We remark that arrays whose names end with *3D* are used whenever a cross-product type operation is needed, as it happens for the computation of the curl.

The *Vect* class was naturally conceived to be a “Gauss point” counterpart of the *Quantity* class. In fact, it contains a pointer for the quantity association. Taking the informations from the *Quantity* association, the variables *FEord*, *dim* and *ndof* can be filled. They hold the finite element family, the number of scalar components and the number of volume and boundary degrees of freedom, respectively. If no *Quantity* is associated, then these values must be filled independently of it.

We remark that the implementations of the Gauss interpolation functions are such that they can perform in a unique routine the interpolation of both quadratic and linear quantities defined either on the volume or on the boundary, for any number of scalar components.

4.3.5 The *Mesh*, *GeomEl* and *Domain* classes

The *Mesh* class holds the data and functions to handle the domain triangulation in two or three dimensions for the given number of multigrid levels and processors. The most important data members of the *Mesh* class are:

- a reference to a *GeomEl* object;
- a pointer to a *Domain* object;
- an array to hold the node coordinates;
- an array that contains the element connectivities, for both volume and boundary meshes, for all levels.

The *GeomEl* object is used to define the type of geometric element that constitutes the mesh. At present, meshes with only one type of geometric element are supported. The support is for the quadratic types HEX27, QUAD9, TET10 and TRI6. In the future we plan to support hybrid meshes as well, i.e. meshes with more than one geometric element type.

The *Domain* object can be optionally used to hold basic information on the domain shape in the case where the domain has a well-defined shape, such as a box, cylinder, etc. At present, the *Box* class is provided with the library as a child of the *Domain* class. In case the user wants to define a new domain shape, this may be simply implemented as a child of the *Domain* class. When the domain has no particular shape, one simply has to pass the NULL value to the *Domain* pointer in the *Mesh* constructor.

The arrays of the *Mesh* class for the node coordinates and element connectivity are filled by reading the mesh HDF5 file generated by the *gencase* preprocessing application. The connectivity and coordinates arrays are serial at present. We plan to extend them to a parallel structure. In a domain decomposition approach, each processor in fact only needs the mesh information for the associated subdomain.

The *Mesh* class also provides the functions for printing the XDMF wrapper file of the HDF5 mesh file and a function for converting the quadratic mesh element connectivity into their corresponding child linear connectivities. This last functionality is particularly useful when the postprocessing visualization software is not able to read the XDMF quadratic elements.

4.3.6 The *GenCase* class

The *GenCase* class is used in the *gencase* service application with the purpose of preparing a general finite element application of the *FEMuS* library. In particular the class is devoted to the two tasks of mesh generation and preparation of linear algebra structures (matrices and multigrid operators). These tasks are performed by taking into account a *parallel* and *multigrid* framework. All the necessary structures are then generated according to a given number of multigrid levels and a given number of processors. Clearly, a simple serial and single grid solution can also be performed as a special case. The mesh and linear algebra structures are all printed to files in HDF5 format that are placed in the *input* subdirectory of the considered application. Then, once the application is running, they are read by the *Mesh* and *EqnBase* classes.

Mesh generation

Mesh generation in the *GenCase* class starts from a coarse level and then creates the desired number of finer levels. The coarse mesh generation can be obtained either by using LibMesh mesh generation functions, or by reading an external mesh file.

The LibMesh library provides some mesh generation functions for some simple domain geometries. For instance the function *MeshTools::Generation::build_cube* can build a box domain of dimensions $[a, b] \times [c, d] \times [e, f]$. Other parameters such as *nintervx*, *nintervy* and *nintervz* are used to define the number of mesh subdivisions along the *x*, *y* and *z* directions. These data can be retrieved from the runtime *femus.conf.in* configuration file of each specific application (see Section 4.4). All the other configuration parameters for the *GenCase* class and the *gencase* application (e.g., the number of multigrid levels) can be obtained from the application-specific file. In fact, the *gencase* application is not a stand-alone executable, but a utility application at the service of each finite element simulation.

Since only few simple geometries can be obtained with the LibMesh calls, the need arises to interact with CAD-based mesh software. At present, the *FEMuS* library can read mesh files obtained with the Gambit and Salome mesh generators. The auxiliary *GambitIO* and *SalomeIO* classes are used for parsing the respective file formats.

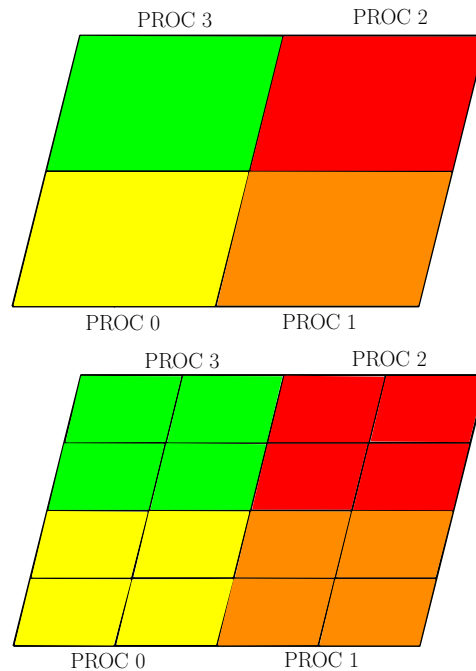


Figure 4.6: The Gencase class must take into account both the subdomain and the level subdivisions of the mesh geometrical entities (elements and nodes).

After a coarse mesh has been obtained, another LibMesh function call generates all the mesh levels. A LibMesh *MeshRefinement* object takes a coarse mesh object and refines it for the requested number of levels with the *uniformly_refine* function. This performs a standard midpoint refinement.

Linear algebra parallel multigrid structures

The task of preparing the linear algebra structures in a parallel and multigrid framework has the goal of generating a mesh *node* numbering and an *element* numbering that consider both the *level* and the *subdomain* divisions (see Figure 4.6). When performing a parallel simulation, the mesh is divided into subdomains, i.e. separate portions of the domain, each of which is associated to one processor. A subdomain division can be equivalently referred to as a *processor* division. Therefore, the possibility of identifying the geometrical mesh entities (elements and nodes) for all levels allows each application to appropriately construct the multigrid linear algebra structures. Within each level, these structures can then be parallelized, due to the fact that the subdomain division is tracked at all levels by the generated element and node numberings.

The generation of the element and node numberings consists in a *reordering*

of the mesh element and node numberings given by the LibMesh classes. In the LibMesh classes, the numberings are generated according to adaptive mesh refinement rules, where both coarse and fine elements coexist and are considered in a unique array.

The element reordering performed by *GenCase* is as follows. To every element, a weight value w is associated, given by

$$w = \text{LEV} + \text{NLEVS} \times \text{PROC},$$

where LEV and PROC are the level and subdomain indices of the current LibMesh element and NLEVS is the total number of levels. In this way, the same weight w is associated only to the elements with equal level and equal subdomain. Then, the nodes with $w = 0$ are numbered starting from 0 and increasing the index, then nodes with $w = 1$ are numbered subsequently, and so on. A similar but more complicated weighting procedure is adopted for the node reordering, due to the fact that the additional distinction between *quadratic* and *linear* nodes must be obtained.

With the reordered element numbering, the element connectivity per level and subdomain can be generated. According to the newly constructed node reordering, one can print the coordinate array and build the DoF map for all levels, subdomains and for both quadratic and linear variables. Hence, the sparsity patterns for matrices, prolongation and restriction operators can be determined. While the matrix entries will be filled by the application depending on the equation to be implemented, the entries of the prolongation and restriction operators can be directly computed in the *GenCase* class since they only depend on geometric information.

4.3.7 The *FEElemBase* and *FEGauss* classes

The *FEElemBase* class is the base class for the definition of the canonical finite elements used for the discretization of the quantities. In Figure 4.7 we report its inheritance diagram. The class is defined based on the parameter VB . The preprocessor variable VB is equal to 2 and it is used throughout the library to define the range of an array index referring to both volume and boundary data. Index 0 is related to volume and index 1 is used for the boundary. Therefore, the `_ndof[VB]` array holds the number of degrees of freedom for the element and for its corresponding boundary element of the same order (for instance, the boundary element of a QUAD9 or TRI6 finite element is EDGE3 in both cases; the boundary element of TET10 is TRI6). The `_name[VB]` string array holds a string with the name that identifies the volume and boundary elements.

Depending on the space dimension, geometric element shape and finite element order, a different child class of *FEElemBase* must be instantiated, among

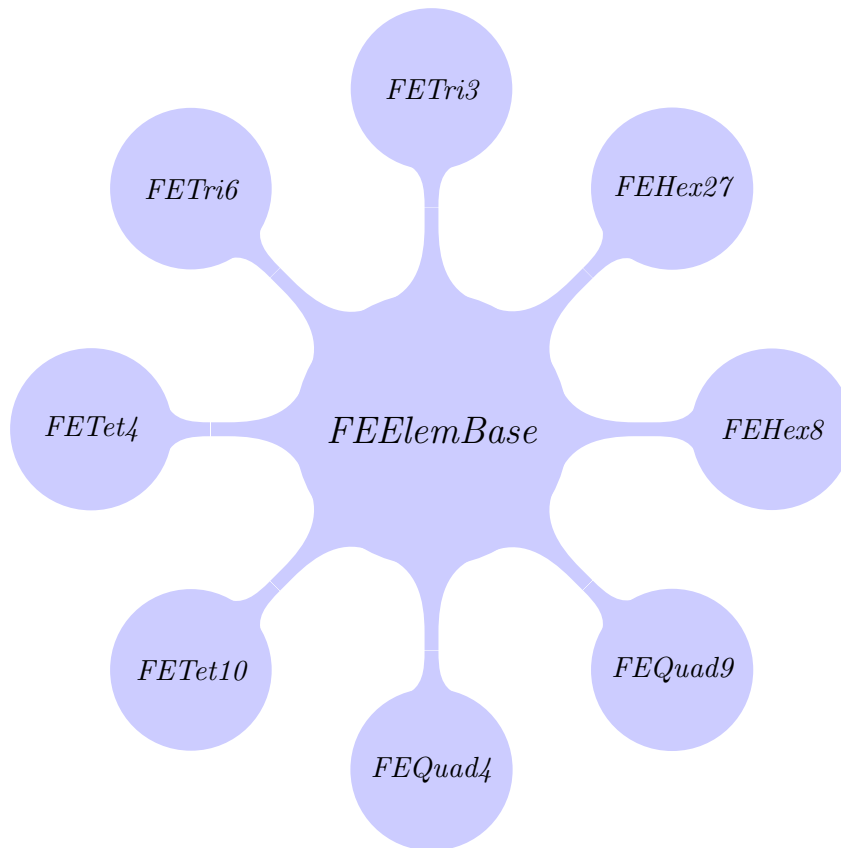


Figure 4.7: Inheritance diagram for the FEElemBase class.

those shown in Figure 4.7. In order to achieve this choice the static *build* member function has been defined. Being static, this member function is not related to a specific *FEElemBase* instantiation. This function simply consists of *switch* constructs. Based on three indices for the space dimension (2 or 3), the geometric element shape (triangular/tetrahedral or quadrilateral/hexahedral) and the finite element order (quadratic or linear), a pointer for an instantiation of the desired child finite element is returned. Since the return type of the function is actually a pointer to a father class (*FEElemBase**), pure virtual functions are required in order to retrieve child specific data from a father interface. In particular, the *get_embedding_matrix* function of the *FEElemBase* class is used for computing multigrid operators, while *get_prol* is defined for printing linear elements over a quadratic mesh.

The *FEGauss* class holds the values of the shape functions and derivatives at the Gauss points for the considered finite element. This class basically reads the data contained in the files in the *fem* directory of the library.

4.3.8 The *RunTimeMap* class

The purpose of the *RunTimeMap* class is to provide a class with the possibility of run-time configuration. In order to do that, one should simply add a *RunTimeMap* object as a data member of the class that requires runtime configuration. This can be very useful as it can allow one to vary some parameters without recompiling neither the library nor the specific application executable.

Let us explain how runtime configuration is obtained. Every specific application of the library has a runtime configuration file called *femus_conf.in* in its *config* subdirectory (see Section 4.4). This file can contain several groups of parameters. Each parameter is defined by a name and a value and each group is associated to a specific class. In order to express this association, the group is enclosed within begin and end tags similar to the XML language syntax.

The *RunTimeMap* constructor receives a string that corresponds to the name of the begin and end tags (e.g. `<Class1>` and `</Class1>`) enclosing the group of parameters of interest for the considered class. These parameters are expressed by a name and a value separated by spaces. It is generally convenient to define the tag name to be equal to the corresponding class name. When the *RunTimeMap* object of the class invokes the *read* function, all the parameters within the corresponding tags are parsed. The C++ datatype for the values must be chosen at compile time for each class. In fact, *RunTimeMap* is a class template whose template parameter is the datatype for the values. Typical instantiations are with primitive data types (`RunTimeMap<double>` or `RunTimeMap<int>`) or with STL types (`RunTimeMap<std::string>`). If a class needs to be configured with parameters of different types, it can simply be endowed with a *RunTimeMap* template instantiation for each type, provided that the tag names are distinct.

4.3.9 The *Physics* class

This class is intended as an interface provided to the *EquationsMap* class for the definition of the physical parameters that are shared by the set of equations to be solved. The user must provide an implementation of a child of this class, in which the specific physical data are defined.

For instance, the child class defined by the user can be useful for holding constant physical parameters or for setting common reference values to be used for the nondimensionalization of the equations. In this way one performs the computation of the nondimensional numbers (such as Reynolds, magnetic Reynolds, Hartmann numbers) in a unique place, avoiding multiple computations of the same quantity in various points of the code. In this way the likelihood of implementation errors is reduced. The required physical parameters can be easily provided in the *femus_conf.in* file and be read at runtime (see Section 4.4).

4.3.10 The *Utils* and *Files* classes

The *Utils* class is passed to the constructors of all the fundamental classes of the library. In fact, it contains a *RunTimeMap* object to hold parameters that are not specifically related to one class but are shared among various classes. Also, it provides a set of useful functions for HDF reading and printing as well as basic mathematical operations (dot product, cross product, vector normalization, etc.)

The *Files* class contains a *RunTimeMap* object instantiated with *std::string* datatype for the configuration of the file names and file paths related to the *FEMuS* library. These strings are read in the *femus.conf.in* file within the `<Files>` and `</Files>` tags. For instance, the variable denoted with `F_MESH_READ` corresponds to the name of the external mesh file to be used in case of external mesh generation. Also, one can set the names of the input mesh files, output solution files or log files. The *Files* class also gathers the functions that are used for I/O service purposes, such as file copy, stream redirection to file or directory name checking.

4.4 Multiphysics application structure

In this Section we illustrate the general structure of a multiphysics application of the *FEMuS* library. The optimality system studied in this work has been implemented within this framework. Then, we describe all the required steps for performing a simulation, from the configuration to the final run. We also discuss the implementation of the *main* function for an application and the matrix and right-hand side assembly for an equation.

4.4.1 Basic structure

Every application can be implemented in a subfolder of the *applications* directory. The minimal template structure of an application, as seen in Figure 4.8, is composed of

- the four folders *config*, *lib*, *input* and *output*;
- a source file with a *main()* function (e.g., *main.C*);
- a Makefile for the application.

The *lib*, *input* and *output* of a newly generated application are initially empty. They will be filled during the various steps needed to perform the finite element simulation. The *config* directory instead contains the required configuration files that we will describe later. The Makefile may be easily defined by taking a Makefile of another application as a template and by modifying it according to the specific application.

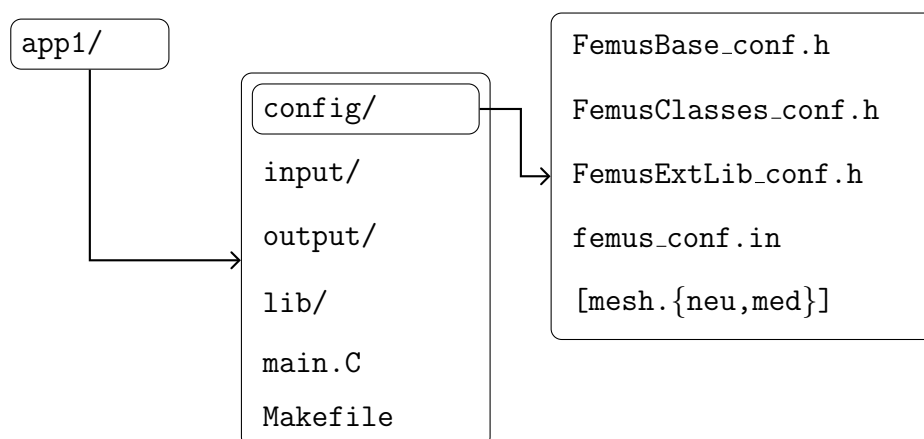


Figure 4.8: Basic structure of the files of a multiphysics application.

The steps required for the implementation of a generic finite element application can be identified with its subdirectories. Therefore, we can consider the following stages:

- Configuration (*config* directory);
- Library compilation (*lib* directory);
- Run preparation (*input* directory);
- Simulation (*output* directory).

Let us now illustrate these single steps in conjunction with a description of the files related to each subdirectory.

Configuration

The *config* directory contains the files for the configuration of the *FEMuS* library. The configuration files constitute the interface through which the classes are tailored to the user needs. The user only has to modify these files in order to get the required library setup. Two kinds of configuration can be distinguished:

- *compile-time* configuration;
- *run-time* configuration.

Compile-time configuration consists in setting various preprocessor directives as defined by the library. These preprocessor directives (mostly of the *#define* type) affect the compilation of the library and the application executable so that the library fits the requirements of the current simulation. For a better understanding the directives are grouped within the following header files:

- *FemusBase_conf.h*;
- *FemusClasses_conf.h*;
- *FemusExtLib_conf.h*.

The *FemusBase_conf.h* file contains some basic macros for the library configuration; for instance one can set there the space dimension of the simulation (dimensions 2 and 3 are supported). The compile-time variables that are associated to a single library class are set in the *FemusClasses_conf.h* header file. The *FemusExtLib_conf.h* file contains the macros related to the activation of the function calls of the required external libraries (e.g. PETSc, MPI). Some examples of these preprocessor directives and their meanings are reported in Table 4.1.

Directive	Value	Description
#define DIMENSION	2	2D simulation
#define DIMENSION	3	3D simulation
#define BDRY_DIM	DIMENSION-1	Boundary dimension
#define LSOLVER	LASPACK	Laspack serial solver
#define LSOLVER	PETSC	PETSc parallel solver
#define HAVE_MPI	1	Enable MPI calls
#define PRINT_INFO	1	Enable log info printing

Table 4.1: Some options in the configuration header files.

All the library classes endowed with a *RunTimeMap* object can be configured at run-time (for a description of the *RunTimeMap* class, see Section 4.3.8). Run-time configuration can be adopted whenever the informations to be passed to the application do not need a library recompilation. The *femus_conf.in* file is split into parts, delimited by XML-like begin and end tags. Each part is devoted to the runtime parameters of a specific *RunTimeMap* instantiation. As it happens for library classes, user-defined classes for the current application can also be endowed with a *RunTimeMap* object and their parameters be defined similarly in the *femus_conf.in* file.

We observe that runtime configuration is one of the various file input/output (I/O) operations performed by the *FEMuS* functions. While very useful, the occurrence of I/O operations should be considered carefully as it may in principle deteriorate code performance. The *FEMuS* classes are conceived in such a way that run-time configuration occurs once and for all at class construction. The configuration parameters are then loaded in memory and they are deleted only at class destruction. The meaning of each parameter and the values it may assume are described in the class specific documentation. Table 4.2 gives a brief description of the most important parameters. We remark that the use of an XML-like

structure for the *femus_conf.in* file can turn out to be useful for an association of the configuration file with a Graphic User Interface (GUI). This could allow a more user-friendly way of setting parameters in case of several classes.

Parameter	value (ex.)	description
libmesh_gen	1	use LibMesh mesh generator
mgops_gen	1	generate multigrid operators files
dt	0.01	time step increment
itime	0.	initial time value
nsteps	100	number of time steps to be run
printstep	5	printing step interval
restart	0	restart step index
lxb	0	box x begin coordinate
lxe	1.	box x end coordinate
nintervx	20	number of x coarse subdivisions
nolevels	4	number of levels
mesh_ord	0	quadratic mesh order
geomel_type	0	hexahedral mesh element
F_MESH_READ	mymesh.neu	external mesh file
BASESOL	sol	output solution filename prefix
RUN_LOG	run_log.txt	simulation log filename
Uref	1	reference velocity
Lref	1	reference length
Tref	1	reference temperature
rho0	1.	density
mu0	1.	viscosity
kappa0	1.	thermal conductivity
cp0	1.	heat capacity
komp0	0.	compressibility factor
qheat	0.	heat density power
dirgx	0	x gravity
mumhd	1.e-6	magnetic permeability
sigmhd	1.e+6	electrical conductivity
Bref	6.e-3	reference magnetic field
alphaVel	1.e3	control penalty value

Table 4.2: Some configuration parameters in the *femus_conf.in* file.

The *config* directory is also devoted to hold a mesh file generated from an external mesh generator (e.g. Gambit). The *gencase* utility application can parse this file through the *GambitIO* class and print the necessary input mesh and multigrid files in the *input* directory of the current application. When the external mesh file is needed, the *libmesh_gen* parameter within the $\langle Gencase \rangle$ tags must be set to 0. Also, care must be taken in setting some parameters in the *femus_conf.in* file consistently with respect to the external mesh file. For instance, the mesh order given by *mesh_ord* must be consistent with the external file, as well as the box dimensions *lxb*, *lxe*, *lyb*, *lye*, *lzb* and *lze* in case of box-shaped mesh.

Library compilation

Once the configuration header files are set according to the user needs, the *FEMuS* library can be compiled. This allows the user to have a version of the library that is tailored and optimized for the specific application (for instance, unneeded function calls can be excluded).

In order to compile the library one has to type the command “make libfemus” from the directory of the application. As a result, a shared library file named *libfemus.so* is generated. A different file is generated in a separate subdirectory for the chosen compiling mode, which is determined by the value of the `FM_FEMUS_METHOD` environment variable (see Section 4.2.2). Once the library is compiled, the application executable and the utility *gencase* executable can be compiled as well. Both of them will be linked against the generated library file.

In the future, we plan to have a unique compilation of the library and to convert compile time resolutions (such as setting the space dimension) into run time resolutions through template programming.

Run preparation

The goal of the *input* directory is to contain the mesh and multigrid files generated by the *gencase* utility for the current application.

Once the configuration files in the *config* directory are set and the *FEMuS* library is generated, the *gencase* application can be compiled by entering the *gencase* subdirectory under the *applications* library directory and then typing the *make* command. The executable is generated for the chosen compiling mode (e.g. *gencase-opt* or *gencase-dbg*). Depending on the number of processors N chosen for the related application, the executable can be launched with “`mpiexec -n N gencase-dbg/opt`”. After this execution, the *input* directory of the considered application is filled with the following generated files:

- *mesh.h5*, that contains information on the grid;

- *Matrix(l).h5* (one file for each level (l)), that contain the sparsity pattern for the system matrix on level (l) ;
- *Prol(l-1)-(l).h5* (one file for each level pair $(l-1)-(l)$), where the prolongation operator sparsity patterns and entries from level $(l-1)$ to level (l) are stored;
- *Rest(l)-(l-1).h5* (one file for each level pair $(l)-(l-1)$), that contain the restriction operator sparsity patterns and entries from level (l) to level $(l-1)$;
- *multimesh.xmf*, for a graphical visualization of the volume and boundary meshes at all levels as contained in the *mesh.h5* file.

See Section 3.2 for a description of the finite element multigrid algorithm and related operators.

The *gencase* program can be seen as a preprocessing stage for every finite element application that must be performed with the *FEMuS* library. Its purpose is to prepare a general *parallel* and *multigrid* simulation. To this aim, the *gencase* application makes use of the advanced classes and tools provided by the LibMesh library. The program collects all the informations and data related to the mesh and multigrid operators from the *FEMuS* configuration files and prints the previously given list of files. The hierarchical data structure of the generated HDF5 files allows to define a general and modular interface for the *FEMuS* library classes. Regardless of the way the mesh is generated (LibMesh function calls or external mesher), a mesh HDF5 file is produced whose fields are read in the same way in each *FEMuS* finite element application. These fields fully characterize the mesh features (connectivity, node maps, etc.) that are needed for the final simulation.

It is recommended that both the compile time and the run time configuration files, as well as the number of processors provided at command line (which is another type of run time configuration), are not modified before compiling and launching the corresponding finite element simulation. While the change of some parameters (such as physical non-geometrical properties) may not influence the *gencase* execution, a modification of other values (e.g., the number of processors or the number of levels) would necessarily require the *gencase* program to be run a second time. A change in the header files would even require to restart from the recompilation of the *FEMuS* library.

Simulation

Once the previous steps are concluded (namely, configuration, library compilation and run preparation with *gencase*), the executable for the finite element simulation can be generated and run. Its compilation can be done by simply typing *make* from the shell terminal in the current application directory. Then the executable

(e.g. *main-opt*) can be run with the same number of processors N as the previous *gencase* preparation run, with the command “`mpiexec -n N main-opt`”.

It is recommended that the *FEMuS* library and related executables be run in debug mode during the development stages, while the optimized mode should be used for production runs. The output files of an application run are printed in a subfolder of the *output* application directory. Each run is generated in a separate subfolder. In order to distinguish them in an unambiguous manner, the format of the run folder name is YYYY-MM-DD_hh-mm-ss, which corresponds to the instant of time (YYYY standing for the year, MM for the month, DD for the day, hh for the hour, mm for the minute and ss for the second) when the executable was launched (e.g., 2012-02-24_12-11-32).

For every solution step (the solution step being a time step, a nonlinear step or some general algorithmic step) the fields with the values computed by the program are printed in HDF5 files. For a solution step index M , a file *sol.M.h5* is printed. Each of these files has a corresponding XDMF file that defines the association between the fields and the corresponding mesh file for visualization. Other library routines can be called in order to print boundary and initial conditions, a log of the program execution or other informations. All these files are printed in the output directory of the specific run. Also, the mesh input files are automatically copied into the corresponding output directory, so that all the I/O files related to a given run are gathered in a unique folder and can be moved anywhere and still be readable.

An interesting *restart* feature is also available with the *FEMuS* library. In order to restart a simulation from the solution step M (*sol.M.h5* file) contained in the YYYY-MM-DD_hh-mm-ss folder, simply do as follows:

- create an empty text file with name *run_to_restart_from* in the *output* directory and write the name YYYY-MM-DD_hh-mm-ss of the previous run directory into it;
- set the *restart* parameter to M in the *femus.conf.in* file.

The subsequent run creates a new run folder; the initial condition arrays are not initialized with the initial condition *ic_read* function (see Section 4.3.2), but they are filled with the field values contained in the *sol.M.h5* file of the previous run.

While some configuration parameters may be modified without creating inconsistent simulation results, it is recommended that the configuration parameters remain unchanged, especially the number of levels and the number of processors. It is then clear that the *gencase* program need not be run a second time in case of restart.

4.4.2 Main function

Here we describe the most important phases for implementing the *main* function of a generic finite element simulation within the framework of the *FEMuS* library. The main steps are: *FEMuS* library setup, preparation of an *EquationsMap* object and solution of the equation system.

The *FEMuS* library setup basically consists in the initialization of the parallel libraries; other setup tasks may be performed, such as command line parsing, file path setting, directory checking, standard output redirection to file, etc. Each of these operations can be performed with specific library calls.

The preparation of an *EquationsMap* object is at the heart of each finite element simulation. In this step all the necessary classes for the construction of an *EquationsMap* object must be instantiated. The largest part of user code development for the specific application consists in implementing the necessary classes to complete this stage. Based on the scheme of Figure 4.3 in Section 4.3.1, the following substeps can be identified:

- *Utils* instantiation and parameter reading;
- *Mesh* initialization in terms of a *GeomEl* object and an optional *Domain* child object (e.g. *Box*);
- instantiation of an implemented *Physics* class;
- *TimeLoop* instantiation for time discretization;
- *FEMap* instantiation for space discretization and addition of the required finite element families;
- *QuantityMap* initialization and addition of quantities;
- *EquationsMap* initialization and addition of equations;
- possible association of equations to quantities.

The mandatory implementations the user must provide are: a *Physics* child class, one or more *Quantity* and one or more *EqnBase* child classes. The characterization of the other classes does not require any code implementation by the user, but it simply takes place by setting the corresponding parameters in the configuration files. Wherever possible, such as for the implementation of a *Domain* child object, the user can provide new classes. It is a good development practice to design classes that not only fit a specific application, but also may be eligible

for expanding the overall *FEMuS* library. The library is in fact conceived to provide base functionalities (*father class* level); each application is characterized by peculiar implementations (*child class* level).

Once all the basic ingredients are available, the *EquationsMap* object is constructed. Then, we need to instantiate the implemented quantities and equations and add them to the respective maps. In fact, the maps are supposed to be containers for all the involved physical quantities and equations in the application (see Sections 4.3.1 and 4.3.3). In particular, two general types of physical quantities for the quantity map can be distinguished: unknowns (which can be defined as *unknown quantities*); and provided functions (*external quantities*) of time, space or other variables. The distinction between the two basically consists in how their degrees of freedom are available in the system. In the first case they are *computed* by the solution of an equation, while in the second one they are *provided* explicitly by some user function.

Let us consider an example concerning the Navier-Stokes equations with a *given* magnetic field. The unknown quantities are clearly velocity and pressure, while the magnetic field is an external quantity. The equation constructor needs the definition of a vector of the unknown quantities (see Section 4.3.2). So, a vector for velocity and pressure is created and passed to the equation constructor. Clearly, the quantity associated to the external magnetic field does not belong to this vector. Once the equation is instantiated, it can be added to the equation map and be associated to each unknown quantity.

When the equation and quantity maps are filled, the DoF map of each equation can be prepared, together with the related multigrid operators, boundary and initial conditions. All the linear systems are ready for the solution. The user can implement a function that receives the *EquationsMap* object and calls the involved equations in the required order and with all the peculiar algorithmic procedures for different convergence criteria, update sequences, etc. This function performs the solution of the multiphysics problem.

The optimality system implemented in this thesis may be considered as an example of application of the *FEMuS* library. The equations and quantities implemented for the optimality system are schematically represented in Figure 4.9. The state, adjoint and control unknown quantities are defined, together with external quantities such as the desired velocity for the cost functional. The equations of the optimality system are implemented in the classes:

- *EqnNS*, for the Navier-Stokes equations, whose unknowns are given by velocity (*Velocity* class) and pressure (*Pressure*);
- *EqnMHD*, for the MHD equations, whose unknowns are the homogeneous magnetic field (*MagnFieldHom*) and the Lagrange multiplier associated to the divergence-free constraint (*MagnFieldHomLagMult*);

- *EqnNSAD* for the adjoint Navier-Stokes equations, the unknowns being the adjoint velocity (*VelocityAdj*) and adjoint pressure (*PressureAdj*);
- *EqnMHDAD*, for the adjoint MHD equations, whose unknowns are the adjoint homogeneous magnetic field (*MagnFieldHomAdj*) and the adjoint associated Lagrange multiplier (*MagnFieldHomLagMultAdj*);
- *EqnMHDCONT*, for the control equation, the unknowns being the lifting function of the magnetic field boundary conditions (*MagnFieldLift*) and its divergence-free constraint Lagrange multiplier (*MagnFieldLiftLagMult*).

4.4.3 Equation implementation

The implementation of the equation classes for an application is the part in the user code development that requires greatest attention. The routines that must be implemented in this class typically concern: boundary conditions, initial conditions and assembly of the linear system.

The implementation of the boundary and initial conditions was already illustrated in Section 4.3.2. Here, we describe the assembly of the matrix and right-hand side. A similar procedure is used for the assembly of all the equations involved in the optimality system.

The assembly routine of an equation receives two input parameters *vb* and *Level*. In fact it performs the assembly of both the volume (*vb*= 0) and the boundary (*vb*= 1) integrals involved in the weak form of the equation, for the multigrid level *Level*. The subdomain dependency is passed to the routine from a variable in the equation class through the implicit *this* class pointer in the function interface.

The assembly of an equation consists in the following steps:

- definition of the involved *Vect* quantities;
- element loop for the given level and subdomain;
- Gauss points loop for the current element;
- DoF loop for shape and test functions.

First of all, a definition of the involved *Vect* objects is in order. As discussed in Section 4.3.4, these objects are the Gauss point counterparts of the quantity objects. We can distinguish between internal and external *Vect* instantiations: the internal objects are associated to the equation unknowns, while the external objects correspond to other quantities, whose degrees of freedom are either provided by function or computed in *another* equation. The *Vect* objects associated to the unknowns of the equation hold the DoF and gauss values of the previous iteration step (clearly, the *current* DoF values still have to be computed).

The quantity pointer for a *Vect* object is taken from the vector of unknown quantities of the equation (see Section 4.3.2). Through that pointer the informa-

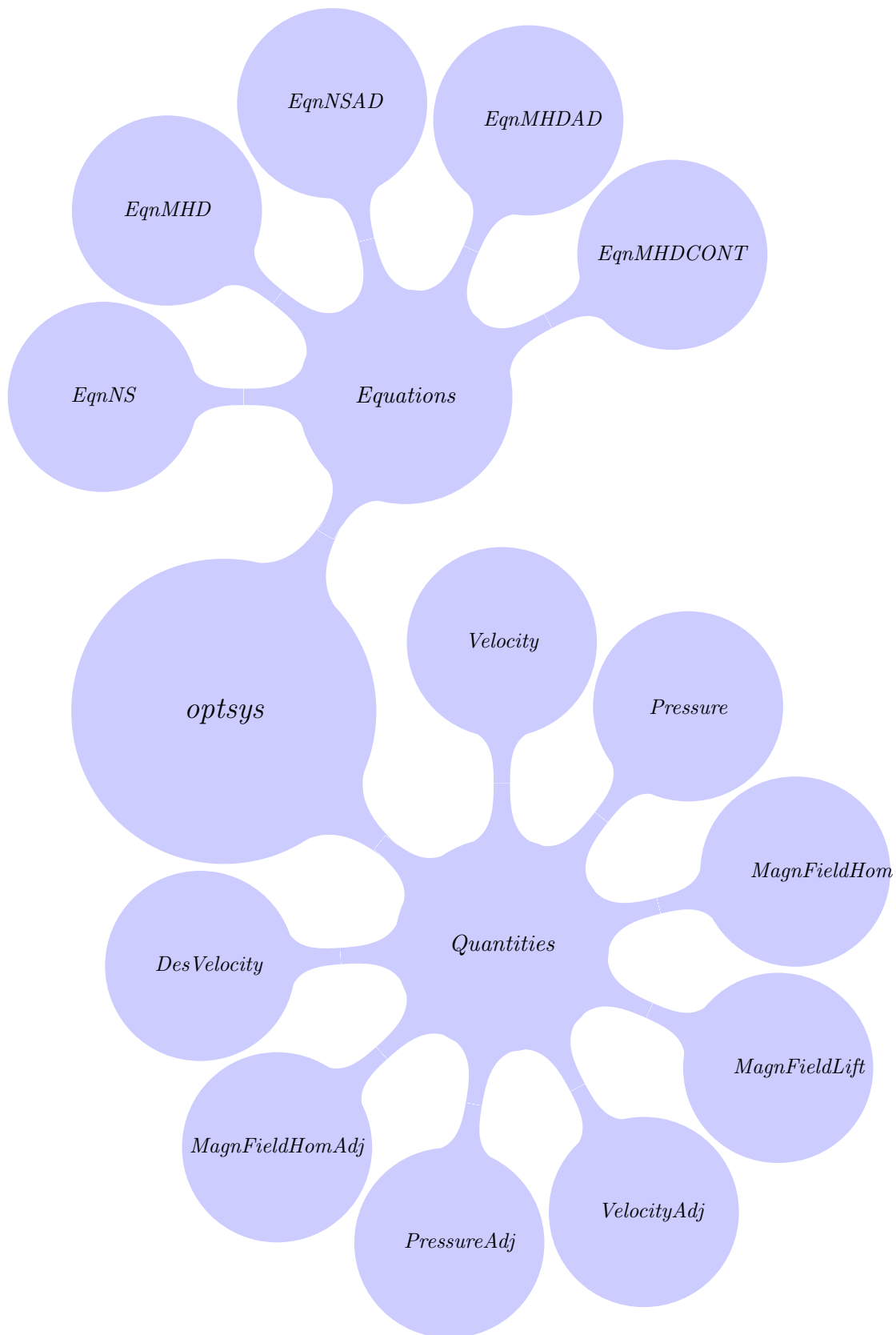


Figure 4.9: The equations and quantities for the optimality system (some quantities are not reported here for the sake of conciseness).

tions to allocate the *Vect* arrays can be obtained. These arrays must be deallocated at the end of the assembly routine.

Once all the *Vect* arrays are allocated, the element loop is performed. For each element in the current level and subdomain, one must retrieve the relevant geometric informations from the mesh such as connectivity and node coordinates and then the DoFs for the involved quantities. The information related to the element degrees of freedom for both the internal and the external *Vect* objects must be obtained. With the call the function *get_el_dofs_bc_ic* one can get the DoF indices, DoF boundary condition flags and DoF old values for the internal quantities. For the external quantities two ways of obtaining the DoF's of a *Vect* object are possible: either from the associated equation or from a predefined function.

After all the element DoFs are collected, the loop over the element Gauss points can be performed. First, one has to retrieve the values of all the shape functions, canonical derivatives, Jacobian transformation and derivatives in physical space for the given Gauss point. As it happens with mixed equations like for the optimality system, both quadratic and linear finite element functions are needed. Gauss point interpolations can be done by using the functions described in Section 4.3.4. For instance, to obtain the old velocity value and old velocity gradient components at the current Gauss point, one can call the *funcVect_g* and *gradVect_g* functions; the magnetic field values and its curl, needed for the Lorentz force, can be computed by *funcVect_g* and *curlVect_g*.

Finally, the element matrix and right hand side can be filled with the operators that characterize the equation (Laplacian, advection, etc.) computed at the Gauss points. We remark that the *row* matrix indices of the element matrix correspond to the *test* functions, while the *column* indices are associated to the *shape* functions of the unknowns. A boundary condition array is used as a *row* flag for the nodal implementation of the Dirichlet boundary conditions (see Section 4.3.2). In case of a Dirichlet boundary node, only the matrix diagonal and the element right hand side are set to a nonzero value, according to the Dirichlet value to be enforced.

The contributions of the element matrix *KeM* and right-hand side *FeM* are added to the global counterparts *A* and *b* of the current level by using for each multigrid level the corresponding *add_matrix* and *add_vector* functions. An analogous procedure holds for the boundary element assembly loop (*vb*= 1), which is devoted to the enforcement of boundary integrals for Neumann boundary conditions.

We finally observe that all the equation terms must be implemented in nondimensional form on a nondimensional domain. The library takes care of multiplying by the reference values (given in the configuration *femus_conf.in* file) for printing the physical results.

Chapter 5

Numerical results

In this chapter we show the results of two and three-dimensional numerical solutions of the optimality system considered in this thesis. In Section 5.1 we describe a simple verification test for a correct implementation of the lifting function method for enforcing boundary conditions. The results of a two-dimensional solution of the optimality system are reported in Section 5.2. In Section 5.3 we illustrate the computations for three-dimensional optimal control cases.

5.1 Code verification test for the lifting function

This simple code verification test is performed to verify the correctness of the implementation of the lifting function method for enforcing the magnetic field boundary conditions. In particular we intend to verify that, for given values of the Dirichlet boundary conditions, different lifting functions of these values yield the same solution in terms of the flow state variables, provided that the lifting functions are divergence-free. Therefore, the divergence-free constraint must be necessarily enforced on the lifting function to guarantee that the same solution for given boundary values is obtained, regardless of the particular lifting function chosen. Note that in this test we are not solving for the whole optimality system, where the lifting function of the magnetic field boundary conditions is the unknown control variable that is computed by the optimization algorithm. Here we simply consider to give a lifting function as a datum for the state equations.

In order to perform this test we refer to the well-known MHD *Hartmann flow* as a basic configuration [10]. This represents the steady and fully developed flow of an electrically conductive fluid between two parallel plates in the presence of an external magnetic field orthogonal to them. It is known to be one of the few cases for which an analytical solution to the Navier-Stokes and MHD equations (2.1) can be determined. Let us consider a two-dimensional infinite channel simulated by a

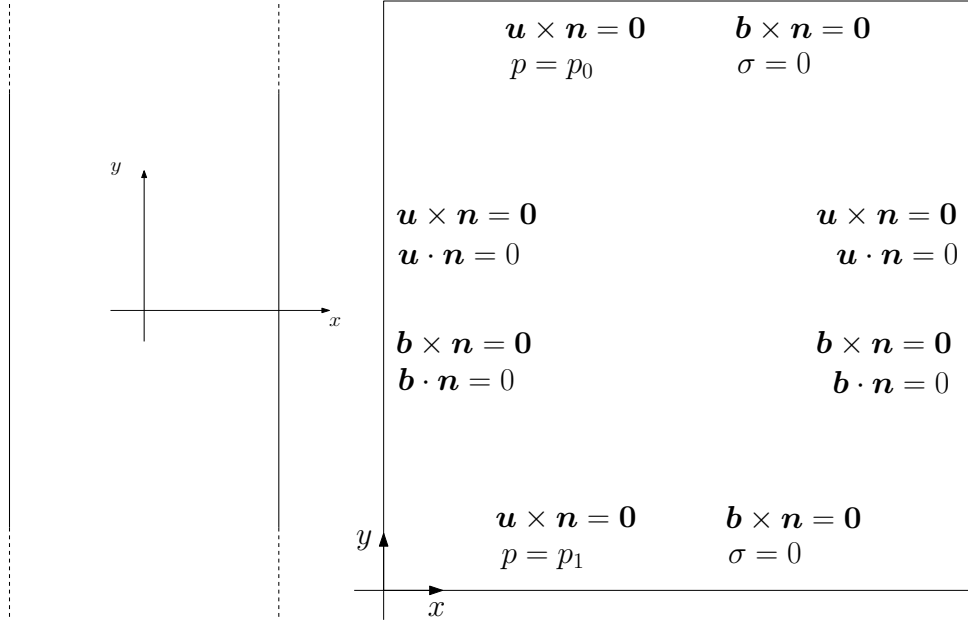


Figure 5.1: Lifting function verification test. Domain and boundary conditions for the Hartmann flow.

square computational domain $[0, 1] \times [0, 1]$ as in Figure 5.1. Let Γ_w be the union of the left and right sides of the square which correspond to the wall plates. We denote as Γ_f the union of the bottom and top sides which correspond to the inlet and outlet sections of the channel. On Γ_w we specify no slip boundary conditions $\mathbf{u} = \mathbf{0}$ for the velocity field and $\mathbf{B} \cdot \mathbf{n} = B_{0x}$, $\mathbf{B} \cdot \boldsymbol{\tau} = 0$ for the magnetic field, where B_{0x} is a constant and the normal and tangential unit vectors at the boundary are denoted as \mathbf{n} and $\boldsymbol{\tau}$, respectively. On Γ_f we assume a vanishing tangential velocity, $u_x = 0$, and uniform pressure on each side, with p_1 on the bottom face and p_0 on the top one, so that the pressure drop along the flow is $P = p_1 - p_0 > 0$. Since u_x is uniformly zero along the inlet and outlet sections, then on Γ_f one has $\partial u_x / \partial x = 0$ and hence $\partial u_y / \partial y = 0$ from the divergence-free constraint. Therefore, the longitudinal velocity has a vanishing normal derivative at the inlet and outlet sections. In a similar way we set $\sigma = 0$ and $\mathbf{B} \cdot \boldsymbol{\tau} = B_{0x}$ uniformly on the bottom and top of the domain, which implies again $\partial B_x / \partial x = 0$ and $\partial B_y / \partial y = 0$ on Γ_f .

The total magnetic field \mathbf{B} can be split as the sum of two contributions $\mathbf{B} = \mathbf{b} + \mathbf{B}_e$, where \mathbf{B}_e is a divergence-free lifting function that extends the magnetic field Dirichlet boundary conditions. Hence, one can write the magnetic field equation

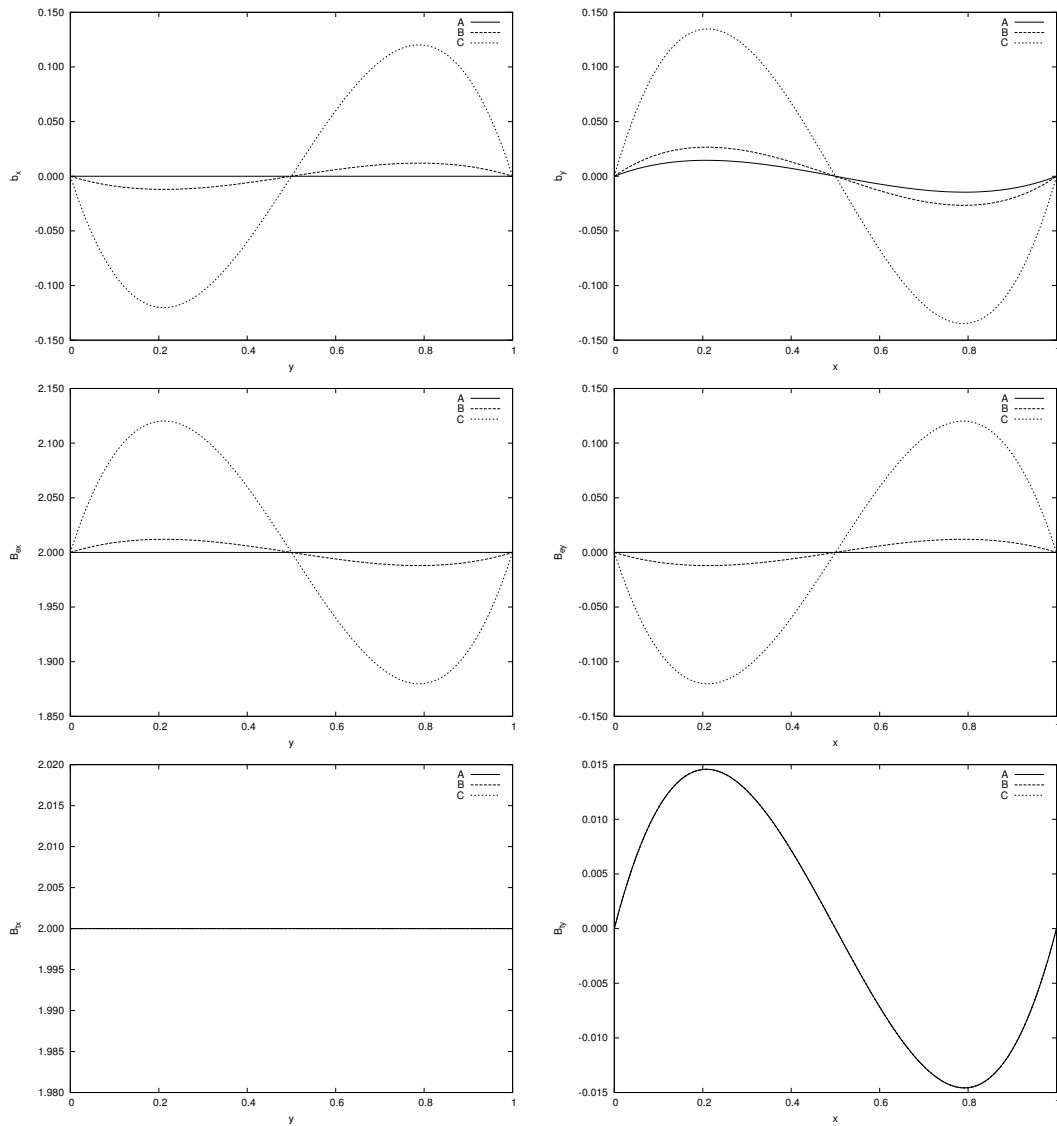


Figure 5.2: Lifting function verification test. The magnetic fields b_x , B_{ex} , B_x along the line $x = 0.5$ (left column) and the magnetic fields b_x , B_{ex} , B_y along the line $y = 0.5$ (right column) for $K_1 = 0$ (A), 1 (B), 10 (C) with divergence-free magnetic field lifting functions.

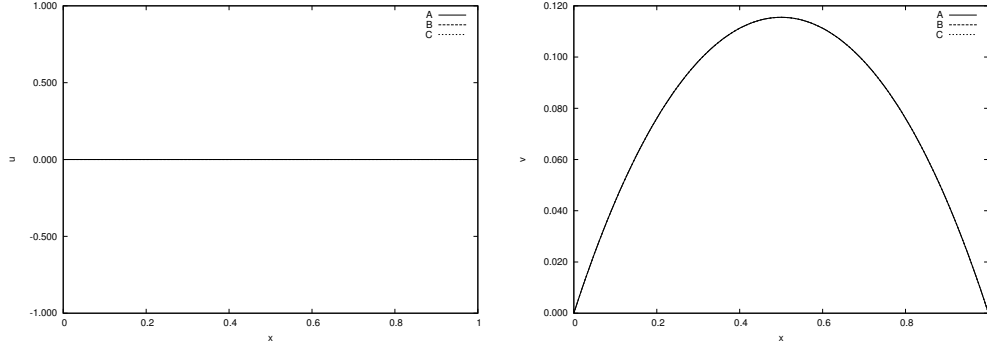


Figure 5.3: Lifting function verification test. The velocity components u_x (left) and u_y (right) along the line $y = 0.5$ for $K_1 = 0$ (A), 1 (B), 10 (C) with divergence-free magnetic field lifting functions.

from (2.1) as

$$\begin{aligned} \frac{1}{Re_m} \nabla \times (\nabla \times \mathbf{b}) - \nabla \times (\mathbf{u} \times \mathbf{b}) + \nabla \sigma = \\ \nabla \times (\mathbf{u} \times \mathbf{B}_e) - \frac{1}{Re_m} \nabla \times (\nabla \times \mathbf{B}_e), \end{aligned} \quad (5.1)$$

$$\nabla \cdot \mathbf{b} = -\nabla \cdot \mathbf{B}_e = 0,$$

together with the boundary conditions

$$\begin{aligned} \mathbf{b} \cdot \boldsymbol{\tau} = 0 \quad \mathbf{B}_e \cdot \boldsymbol{\tau} = 0 \quad \text{on } \Gamma_w, \\ \mathbf{b} \cdot \mathbf{n} = 0 \quad \mathbf{B}_e \cdot \mathbf{n} = B_{0x} \quad \text{on } \Gamma_w, \\ \mathbf{b} \cdot \boldsymbol{\tau} = 0 \quad \mathbf{B}_e \cdot \boldsymbol{\tau} = B_{0x} \quad \text{on } \Gamma_f, \\ \sigma = 0 \quad \text{on } \Gamma_f. \end{aligned} \quad (5.2)$$

For any given function \mathbf{B}_e and any velocity field \mathbf{u} , we can solve for (5.1) with the boundary conditions given by (5.2) and obtain $\mathbf{b} = (b_x, b_y)$. Among all the possible choices of divergence-free lifting functions, if we choose the function $\mathbf{B}_e = (B_{0x}, 0)$, which is constant over the domain Ω , it is possible to find the analytical solution of the Hartmann flow for the velocity and magnetic fields.

Following the definitions of reference and nondimensional values given in Section 2.2, we choose B_{0x} as the reference value for the magnetic field, so that the nondimensional Hartmann number is $H_m = B_{0x} L \sqrt{\sigma_c / \mu}$. Due to the hypotheses assumed for the Hartmann flow, we have that the pressure drop P is a constant; we denote its nondimensional value as $P^* = (PL) / (\rho U^2)$.

From the Navier-Stokes and MHD equations (2.1) one obtains the following equations for the nondimensional longitudinal velocity u_y and nondimensional lon-

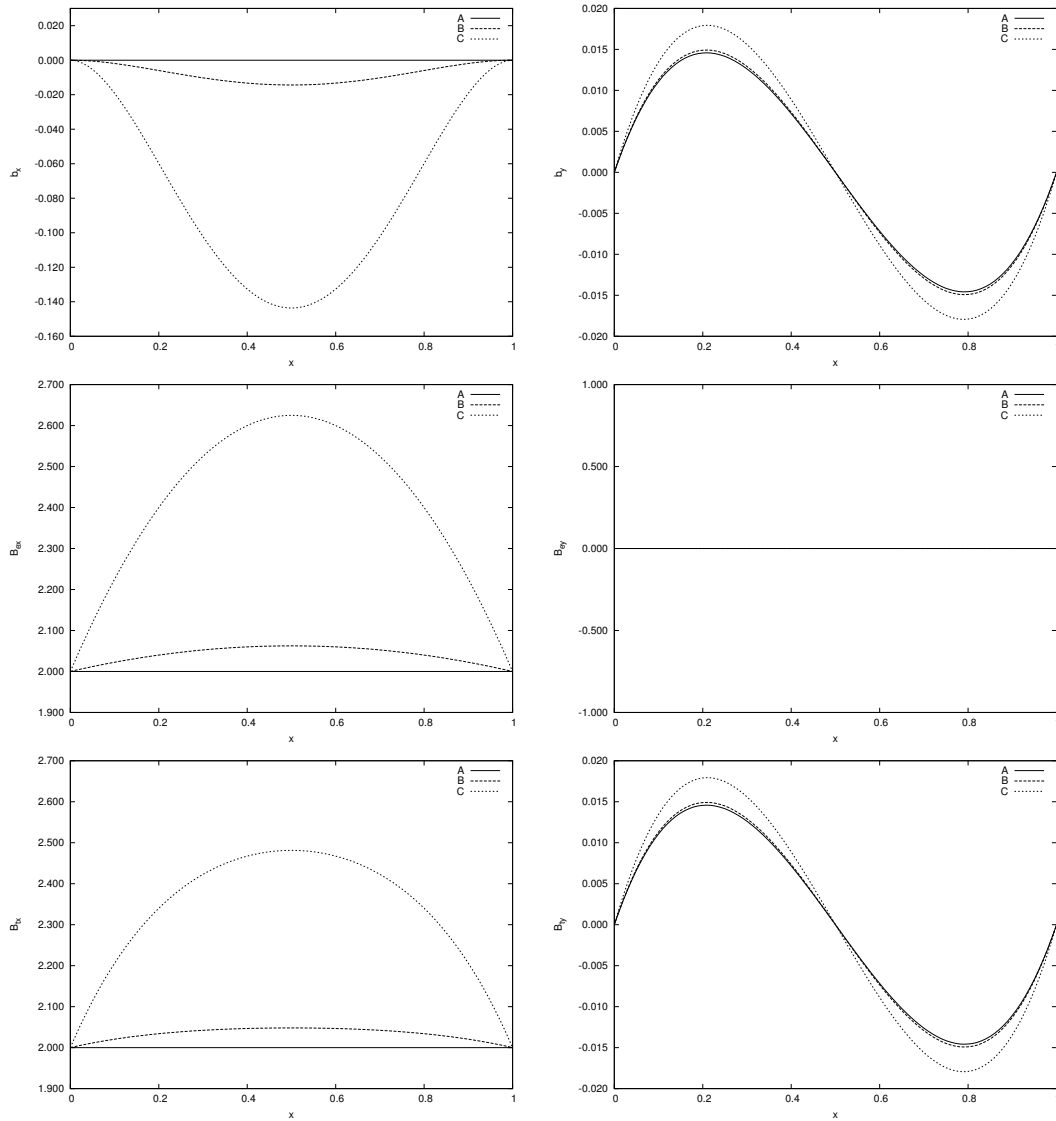


Figure 5.4: Lifting function verification test. The x -components b_x , B_{ex} , B_x (left column) and the y -components b_y , B_{ey} , B_y (right column) along the line $y = 0.5$ for $K_2 = 0$ (A), 1 (B) and 10 (C) with non divergence-free magnetic field lifting functions.

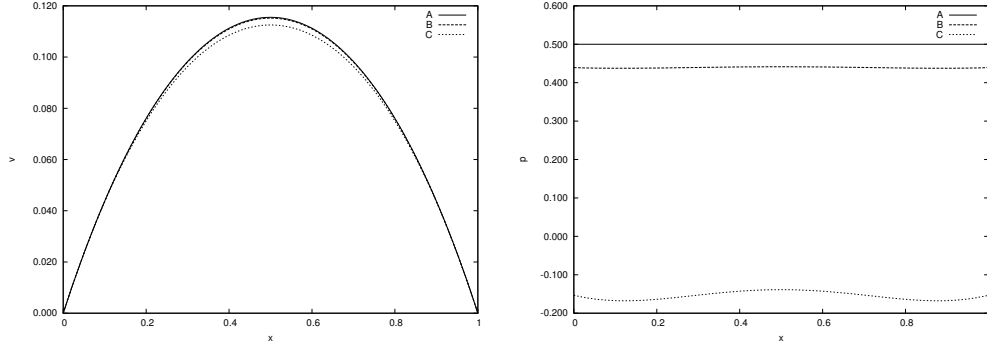


Figure 5.5: Lifting function verification test. The y -component u_y of the velocity field (left) and the pressure p (right) along the line $y = 0.5$ for $K_2 = 0$ (A), 1 (B) and 10 (C) with non divergence-free magnetic field lifting functions.

itudinal magnetic field b_y [10]:

$$\frac{d^2 u_y}{dx^2} - H_m^2 u_y = -P^* Re H_m \coth H_m, \quad (5.3)$$

$$\frac{db_y}{dx} = Re_m (P^* \frac{Re}{H_m^2} (H_m \coth H_m - 1) - u_y), \quad (5.4)$$

with boundary conditions as stated above. We remark that, thanks to the Hartmann flow assumptions, the original nonlinearities in the flow equations have been removed. The solutions read

$$u_y(x) = P^* \frac{Re}{H_m} \frac{\cosh(H_m) - \cosh(H_m x)}{\sinh(H_m)}, \quad (5.5)$$

$$b_y(x) = P^* Re_m \frac{Re}{H_m^2} \frac{\sinh(H_m x) - x \sinh(H_m)}{\sinh(H_m)}. \quad (5.6)$$

We observe that the choice of the constant lifting function $\mathbf{B}_e = (B_{0x}, 0)$ results in a special case in which \mathbf{B}_e and \mathbf{b} are orthogonal to each other.

For any choice of \mathbf{B}_e satisfying the same boundary conditions, the MHD equations yield the same solution, provided that the lifting function is divergence-free. With the purpose of verifying this condition we numerically solve the state MHD equations for two different cases: a divergence-free and a non divergence-free lifting function.

Let us first consider the case of divergence-free lifting functions. Since we have a two-dimensional domain, a divergence-free function can be derived from a potential $\phi(x, y)$. For instance, let us consider

$$\phi(x, y) = K_1 [x(x-1)]^2 [y(y-1)]^2 \quad (5.7)$$

and choose the lifting function as

$$B_{ex} = B_{0x} + \frac{\partial\phi}{\partial y} = B_{0x} + K_1 2y(y-1)(2y-1)[x(x-1)]^2, \quad (5.8)$$

$$B_{ey} = -\frac{\partial\phi}{\partial x} = -K_1 2x(x-1)(2x-1)[y(y-1)]^2. \quad (5.9)$$

In Figure 5.2 the magnetic fields B_{ex}, b_x and B_x along the line $x = 0.5$ and the magnetic fields B_{ey}, b_y and B_y along the line $y = 0.5$ for the three cases $K_1 = 0$ (A), 1 (B), 10 (C) and with $B_{0x} = 2$ in (5.8) and (5.9) are shown on the left and right columns, respectively. It can be seen that for all these cases the magnetic field given by the sum $\mathbf{b} + \mathbf{B}_e$ is the same. Also, the velocity components u_x and u_y remain the same for the three cases, as shown in Figure 5.3. Therefore, the solution of the MHD equations is independent of the choice of a particular divergence-free lifting function of the same boundary conditions. Now we consider a non divergence-free lifting function. We plot the results for

$$B_{ex} = B_{0x} + K_2 x(1-x)y(1-y), \quad B_{ey} = 0. \quad (5.10)$$

In Figure 5.4 the x and y components of the magnetic field \mathbf{B} and its contributions \mathbf{b} and \mathbf{B}_e are shown along the line $y = 0.5$ for $K_2 = 0$ (A), 1 (B) and 10 (C) and with $B_{0x} = 2$ in (5.10). We see that the components of the magnetic field \mathbf{B} do depend on the choice of the lifting function \mathbf{B}_e . Since the divergence of the lifting \mathbf{B}_e is not zero, the total resulting magnetic field is different for each case. Therefore, the velocity component u_y and the pressure p change as well, as shown in Figure 5.5.

5.2 Two-dimensional optimal control case

In this section we report the results about the numerical solution of the optimality system derived in Chapter 2 and discretized in Chapter 3 for a two-dimensional case. The obtained numerical solution is a candidate solution for the optimal control problem. Let us consider a two-dimensional square channel $\Omega = [0, 1] \times [0, 1]$, discretized by quadrilateral elements. The values for the fluid density, dynamic viscosity, electrical conductivity and magnetic permeability are taken as $\rho = 10^3 \text{ kg/m}^3$, $\mu = 1 \text{ Pa}\cdot\text{s}$, $\sigma_c = 10^6 \text{ S/m}$ and $\mu_0 = 10^{-6} \text{ H/m}$, respectively. The reference values for the length, velocity and magnetic field are $L = 1 \text{ m}$, $U = 1 \text{ m/s}$ and $B = 10^{-3} \text{ T}$, so that we have $Re = 10^3$, $Re_m = 1$ and $H_m = 1$. From now on and unless otherwise stated, we refer to nondimensional quantities. The boundary conditions we consider are intended to reproduce a channel flow

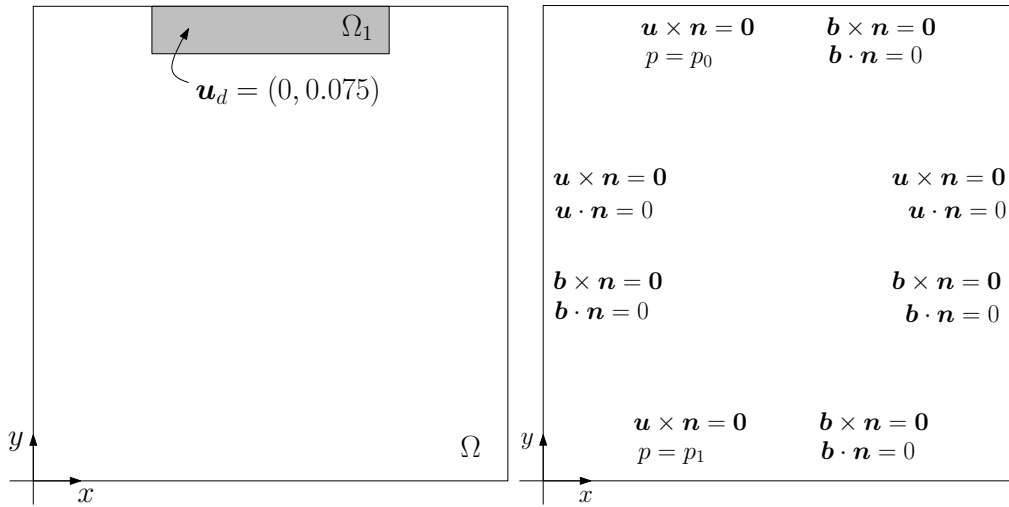


Figure 5.6: Two-dimensional optimal control case. Computational domain Ω with target velocity region Ω_1 (left) and boundary conditions for the state variables of the optimality system (right).

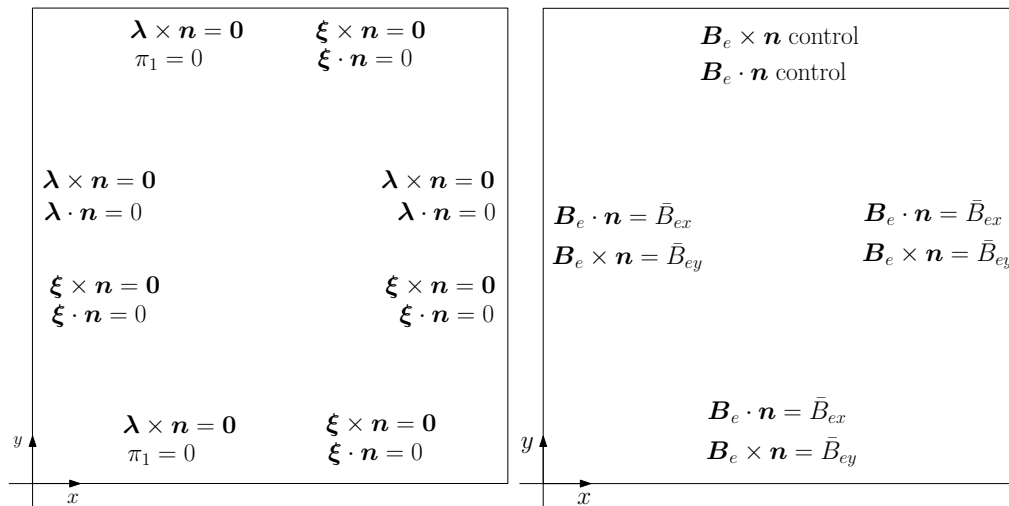


Figure 5.7: Two-dimensional optimal control case. Boundary conditions for the adjoint variables (left) and the control variable (right) of the optimality system.

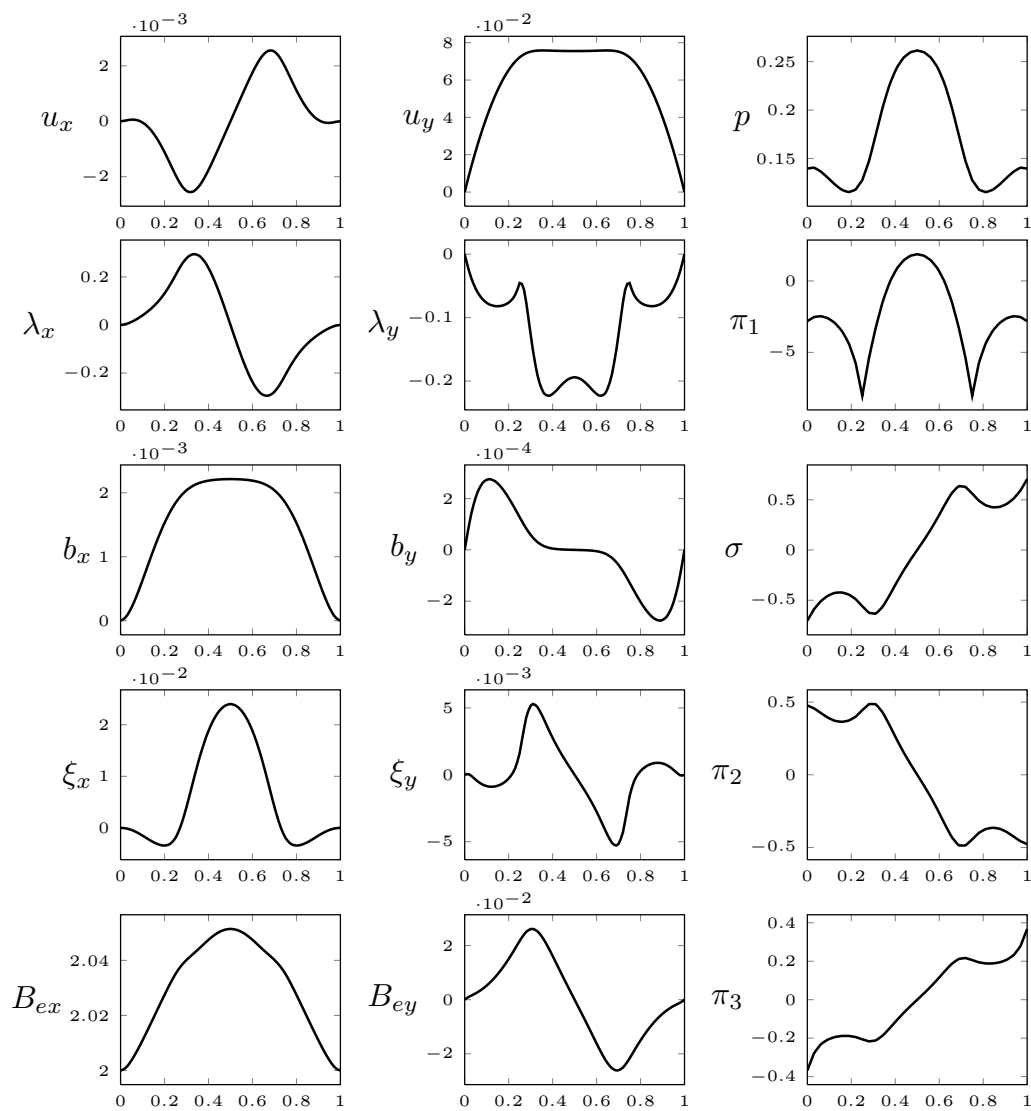


Figure 5.8: Two-dimensional optimal control case. State, adjoint and control variables of the optimality system along the line $y = 0.95$ with $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 10^5$.

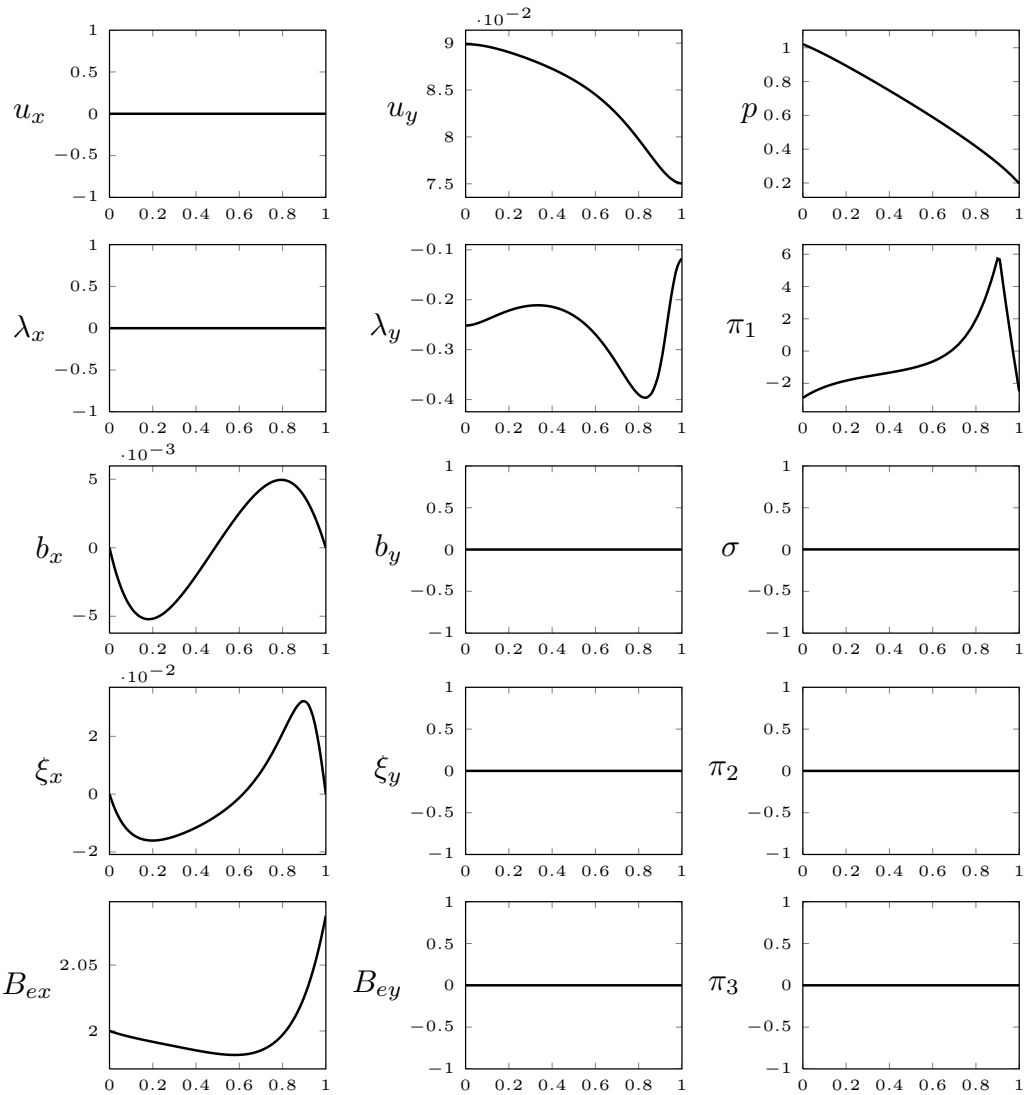


Figure 5.9: Two-dimensional optimal control case. State, adjoint and control variables of the optimality system along the line $x = 0.5$ with $\bar{B}_e = (2, 0)$ and $\alpha = 10^5$.

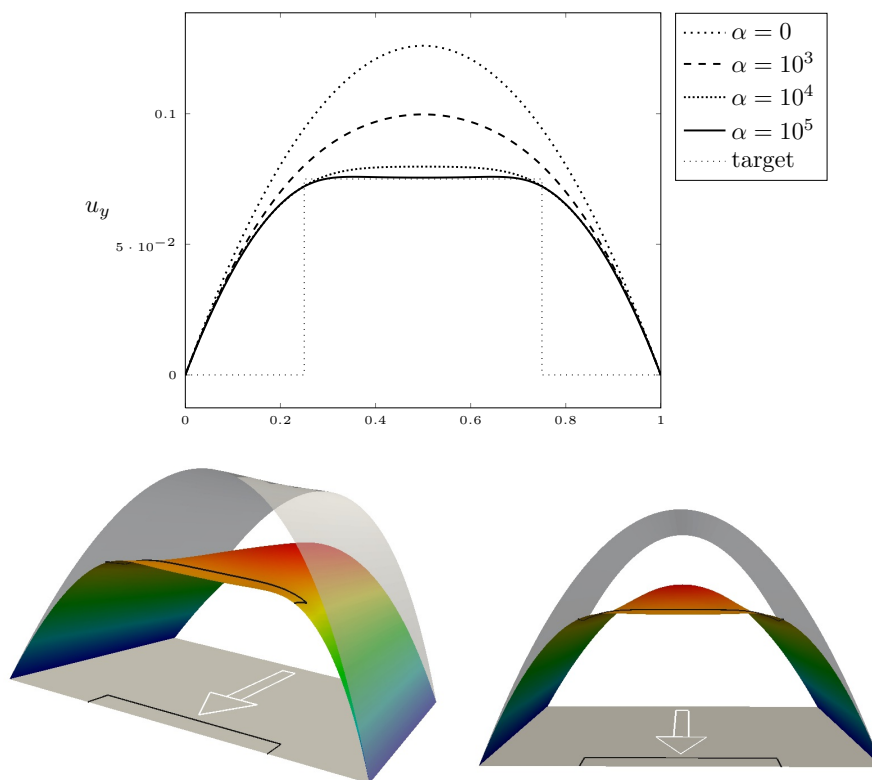


Figure 5.10: Two-dimensional optimal control case. Above: longitudinal velocity profiles along the line $y = 0.95$ with $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 0, 10^3, 10^4, 10^5$ and comparison with the target. Below: two views of the y component of the velocity field for the case $\alpha = 10^5$. The grey curved surface represents the velocity profile that would be achieved in the absence of magnetic field control ($\alpha = 0$).

α	F_0
0	$7.89865 \cdot 10^{-5}$
10^3	$1.34407 \cdot 10^{-6}$
10^4	$6.86789 \cdot 10^{-7}$
$5 \cdot 10^4$	$2.08805 \cdot 10^{-7}$
$7.5 \cdot 10^4$	$2.03182 \cdot 10^{-7}$
10^5	$2.01474 \cdot 10^{-7}$

Table 5.1: Two-dimensional optimal control case. Some values of the velocity error over the target region, $F_0 = \int_{\Omega_1} \|\mathbf{u} - \mathbf{u}_d\|^2 d\mathbf{x}$, for different values of α .

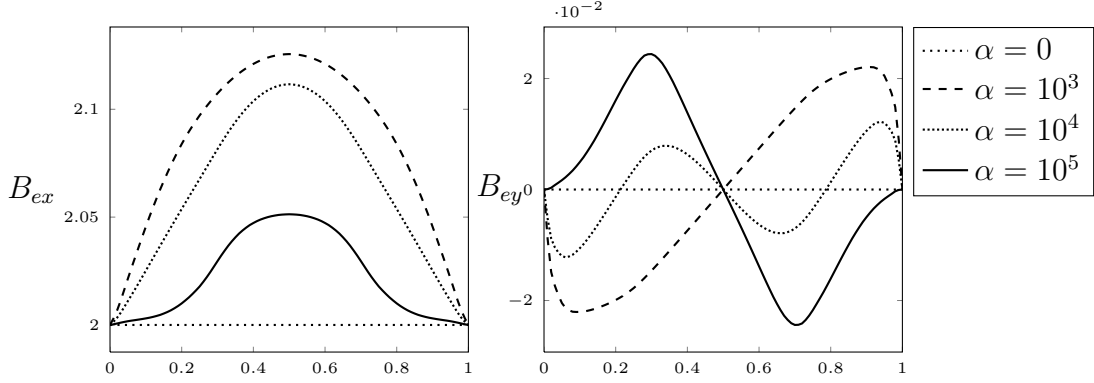


Figure 5.11: Two-dimensional optimal control case. Boundary control as restriction of the lifting function \mathbf{B}_e on the side $y = 1$, for the penalty values $\alpha = 0, 10^3, 10^4, 10^5$.

configuration. For the inlet and outlet sides $y = 0$ and $y = 1$ we enforce

$$\mathbf{u} \times \mathbf{n} = 0, \quad p = p_i \quad \text{on } y = 0, \quad (5.11)$$

$$\mathbf{u} \times \mathbf{n} = 0, \quad p = p_0 \quad \text{on } y = 1, \quad (5.12)$$

where $p_i - p_0 = 1$ is the pressure jump driving the flow. We enforce no-slip boundary conditions on the wall sides $x = 0$ and $x = 1$,

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{u} \times \mathbf{n} = 0 \quad \text{on } x = 0, x = 1. \quad (5.13)$$

For the magnetic field conditions, we subdivide the boundary into the control part $\Gamma_c \subset \Gamma$ and $\Gamma_d = \Gamma \setminus \Gamma_c$. In the portion Γ_c the values of the lifting function \mathbf{B}_e are allowed to vary during the solution of the optimality system in order to minimize the objective functional. On the other hand, the boundary conditions for the lifting function on Γ_d are fixed. The target flow is a desired constant velocity in the y -direction given by $\mathbf{u}_d = (0, 0.075)$ on the target region $\Omega_1 = \{(x, y) \in \mathbb{R}^2 \mid x \in [0.25, 0.75], y \in [0.90625, 1]\}$. The subregion Ω_1 for the desired velocity is shown in Figure 5.6, together with the boundary conditions associated with the state variables. The boundary conditions for the adjoint variables $(\boldsymbol{\lambda}, \pi_1, \boldsymbol{\xi}, \pi_2)$ and the control variable \mathbf{B}_e are summarized in Figure 5.7. We study the minimization of the functional

$$\mathcal{J}(\mathbf{u}, \mathbf{B}_e) = \frac{\alpha}{2} \int_{\Omega_1} (\mathbf{u} - \mathbf{u}_d)^2 d\mathbf{x} + \frac{1}{2} \int_{\Omega} (\nabla \mathbf{B}_e)^2 d\mathbf{x}. \quad (5.14)$$

Indeed, we have chosen $\beta = 0$ and $\gamma = 1$ for the objective functional (2.19) of the control problem. Also, the integral of the velocity error has been limited to the

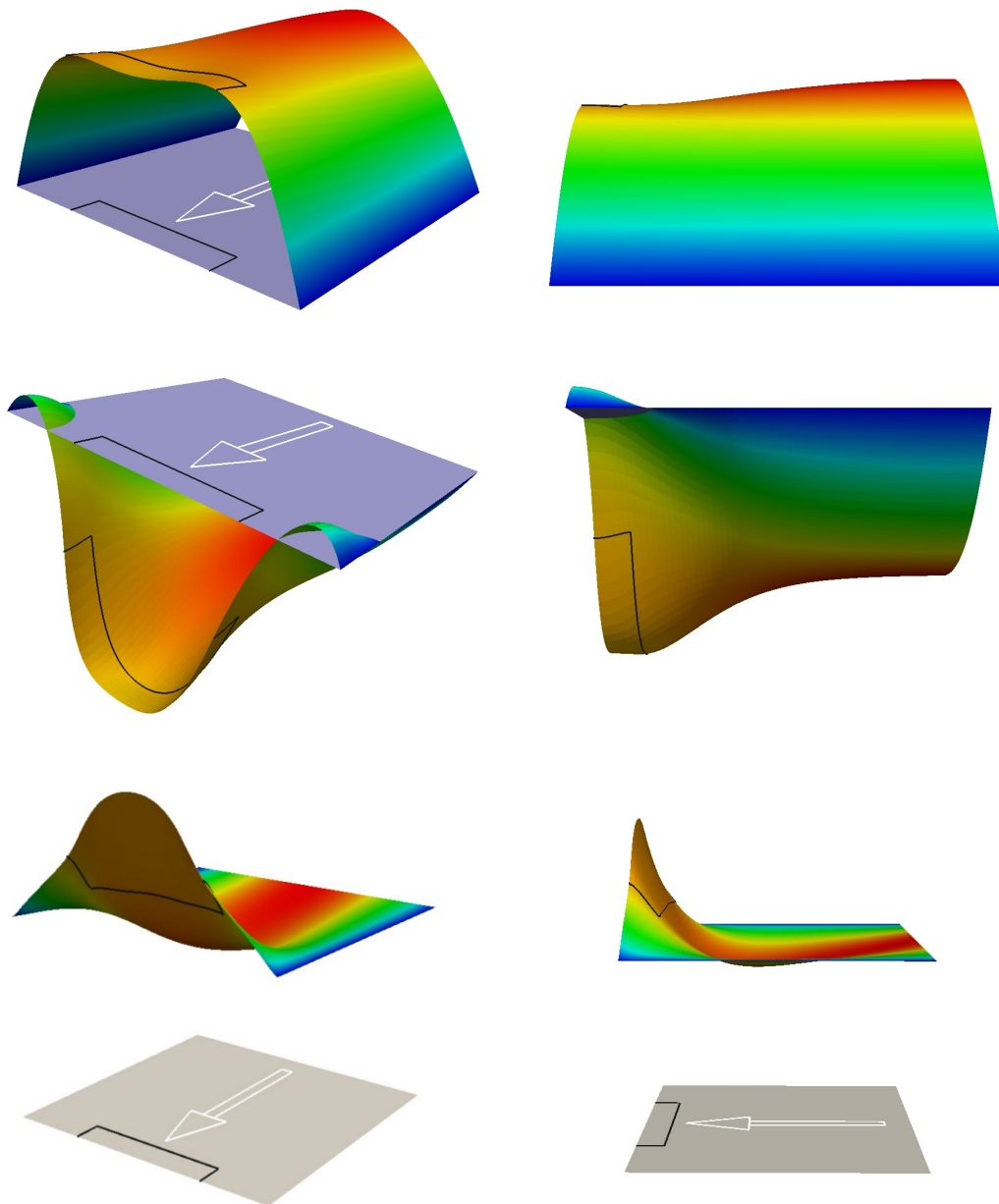


Figure 5.12: Two-dimensional optimal control case. Plots of the y component of the velocity (top row), the y component of the adjoint velocity (middle row) and the x component of the magnetic field lifting function (bottom row) over the domain Ω . The arrow indicates the direction of the flow.

target subregion Ω_1 . This is reasonable, inasmuch as one intends to minimize the error only in the region where a target Ω_1 is given.

It is clear that various choices of the sides for the boundary control portion Γ_c can be considered, which may depend on issues like the position of the target region Ω_1 . Given our choice of the target region, we consider the control portion $\Gamma_c = \{(x, y) \in \Omega \mid y = 1\}$ for a numerical investigation. The control variable is therefore the magnetic field on the outflow side. On the portion $\Gamma_d = \Gamma \setminus \Gamma_c$ a fixed value $\mathbf{B}_e = \bar{\mathbf{B}}_e$ is given. In Figures 5.8 and 5.9 we show the plots along the lines $y = 0.95$ and $x = 0.5$, respectively, of the state, adjoint and control variables obtained from the numerical solution of the optimality system, for $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 10^5$.

It is clear that the choice of the parameter α affects the accuracy in the velocity matching towards the desired profile. In order to study the effect of the parameter α on the minimization of the functional, we plot on the top of Figure 5.10 the velocity profile along the line $y = 0.95$ for the values $\alpha = 0, 10^3, 10^4, 10^5$. Notice that the case $\alpha = 0$ corresponds to the flow profile that would be achieved in the absence of magnetic field. Two different views of the longitudinal velocity field u_y on the domain Ω are shown at the bottom of Figure 5.10 together with the velocity solution without control. In Table 5.1 we also report the values of the velocity error over the target region, $\int_{\Omega_1} \|\mathbf{u} - \mathbf{u}_d\|^2 d\mathbf{x}$, for various values of the penalty parameter α . It is evident that a higher value of the parameter α yields a more and more accurate control, as the velocity profile gets closer to the desired velocity \mathbf{u}_d and the error norm is smaller. In Figure 5.11 we show the boundary control obtained from the restriction of the lifting function \mathbf{B}_e on the line $y = 1$ for $\alpha = 0, 10^3, 10^4, 10^5$. This represents the control solution of the original boundary optimal control problem. Figure 5.12 shows for two different views the plots over the domain Ω of the longitudinal velocity u_y (top row), longitudinal adjoint velocity λ_y (middle row) and the component B_{ex} of the magnetic field lifting function. It is evident how the velocity profile approaches the desired profile in the target region. In that region the adjoint velocity is negative, in order to push the velocity value from the uncontrolled case towards the desired objective. The lifting function of the magnetic field adjust its values on the boundary control portion in order to obtain an optimal solution.

5.3 Three-dimensional results

5.3.1 Hartmann flow optimal control case

We consider here a first three-dimensional solution of the optimality system. The domain region is given by the parallelepiped $\Omega = \{(x, y, z) \in \mathbb{R}^3 \mid x \in [0, 1], y \in$

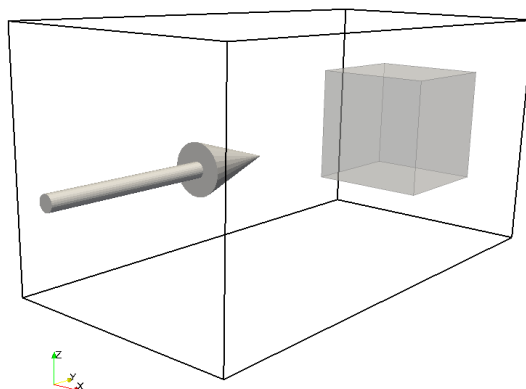


Figure 5.13: Hartmann flow optimal control case. Computational domain Ω and target region Ω_c (highlighted in gray). The arrow indicates the direction of the flow.

k	\mathcal{J}^k	ω^k	$\Delta\mathcal{J}^k$	k	\mathcal{J}^k	ω^k	$\Delta\mathcal{J}^k$
1	3.32491e-05	1	-	16	1.23179e-06	0.0667419	+
2	0.00500133	1.5	+	17	1.22854e-06	0.033371	-
3	0.00287462	0.75	+	18	1.28434e-06	0.0500565	+
4	0.000326773	0.375	+	19	1.22905e-06	0.0250282	+
5	3.59377e-06	0.1875	-	20	1.22845e-06	0.0125141	-
6	2.18803e-06	0.28125	-	21	1.2365e-06	0.0187712	+
7	2.86024e-05	0.421875	+	22	1.22865e-06	0.00938559	+
8	1.9643e-06	0.210938	-	23	1.22844e-06	0.00469279	-
9	1.26166e-06	0.316406	-	24	1.22965e-06	0.00703919	+
10	1.15135e-05	0.474609	+	25	1.22851e-06	0.00351959	+
11	1.63692e-06	0.237305	+	26	1.22844e-06	0.0017598	-
12	1.23698e-06	0.118652	-	27	1.22863e-06	0.0026397	+
13	2.16142e-06	0.177979	+	28	1.22847e-06	0.00131985	+
14	1.23072e-06	0.0889893	-	29	1.22844e-06	0.000659924	
15	1.6191e-06	0.133484	+				

Table 5.2: Hartmann flow optimal control case. Values of the objective functional \mathcal{J}^k , the relaxation factor ω^k and sign of the difference $\Delta\mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$ for each optimization step k for $\alpha = 200$ and $u_{d,y} = 1.4u_{y,Hm}(x)$. A + sign means that the functional has increased from the previous step.

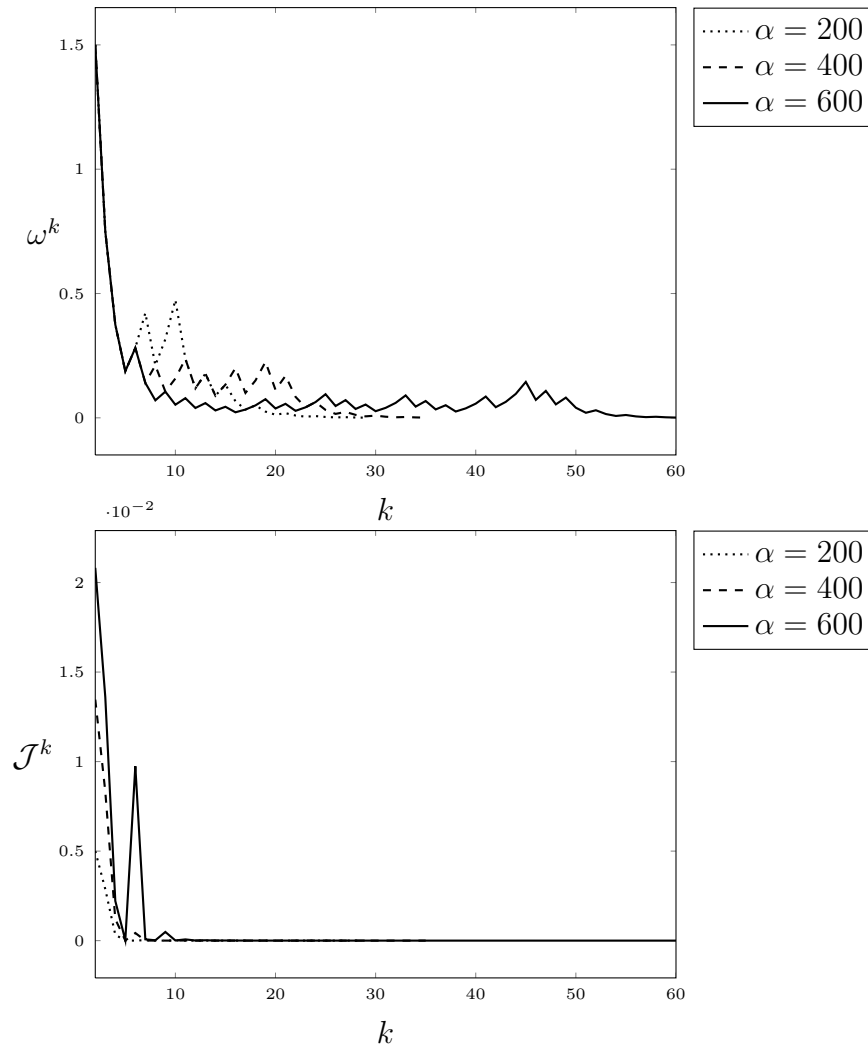


Figure 5.14: Hartmann flow optimal control case. Values of the relaxation parameter ω^k and of the objective functional \mathcal{J}^k for $\alpha = 200, 400, 600$ as a function of the optimization steps.

α	k_{end}	$\mathcal{J}^{k_{end}}$
200	29	1.22844e-06
400	35	7.38943e-07
600	60	5.13533e-07

Table 5.3: Hartmann flow optimal control case. Number of optimization steps k_{end} and final value of the cost functional $\mathcal{J}^{k_{end}}$ for increasing α values.

$[0, 2]$, $z \in [0, 1]$). The base flow, i.e. in the absence of control, is an equivalent three-dimensional Hartmann flow in the y -direction, namely a Hartmann flow in the y -direction with symmetry boundary conditions on the z -direction. It is useful to consider this case due to the fact that an analytical solution for the base flow is available (see Section 5.1). In this way, one can easily use an analytical expression as a reference for building desired velocity profiles, in such a way that one does not choose a target velocity that is too distant from the flow solution without control. This is particularly advantageous for the optimization algorithm, in order to avoid the possibility of converging towards unwanted spurious minima of the objective functional [15].

We retain the same values for the fluid physical properties and for the reference quantities as in Section 5.2. The target region where we want to steer the velocity \mathbf{u} to the desired profile \mathbf{u}_d is $\Omega_c = \{(x, y, z) \in \mathbb{R}^3 \mid x \in [0.25, 0.75], y \in [1.5, 2], z \in [0.25, 0.75]\}$, as depicted in Figure 5.13. The boundary portion Γ_c for the boundary magnetic field control is the face $y = 2$. On the rest of the boundary surface the nondimensional magnetic field is taken as $B_{ex} = 5$ in the x -direction.

In order to show the performance of the gradient algorithm described in Section 3.4 for the numerical solution of the discrete optimality system of Section 3.3, we report in Table 5.2 the results of an optimization loop. The integral of the velocity error in the objective functional (2.19) is assumed to be limited to the target subregion Ω_c , as considered in Section 5.2. The penalty values of the functional are equal to $\alpha = 200$, $\beta = 0$, $\gamma = 1$ and we consider a desired velocity with only a y component given by $u_{d,y} = 1.4 u_{y,Hm}(x)$, where $u_{y,Hm}(x)$ denotes the Hartmann velocity profile in (5.5). Hence, the desired profile is a multiplication by a factor 1.4 of the flow solution without control. It is clear that the relaxation parameter ω^k diminishes whenever the difference between the objective functional values of two subsequent optimization steps, $\Delta \mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$, is positive, in order to under-relax the action of the control. The converse happens for the over-relaxation. In order to study the influence of the parameter α , in Figure 5.14 we also plot the parameter ω^k and the cost functional \mathcal{J}^k as a function of the optimization steps for three different runs with $\alpha = 200, 400, 600$. As we can see in Table 5.3, the value of the cost functional at the end of each optimization loop is smaller for higher α . Clearly, the tracking term $\alpha(\mathbf{u} - \mathbf{u}_d)$ in the adjoint Navier-Stokes equation becomes more and more important, so that the velocity eventually gets closer to the desired profile. On the other hand, more optimization steps are required. In fact, a higher penalty α brings more oscillations in the optimality system because of the higher “thrust” given by the adjoint velocity $\boldsymbol{\lambda}$. If the parameter α is increased too much, one may experience lack of convergence of the linear solvers.

k	\mathcal{J}^k	ω^k	$\Delta\mathcal{J}^k$	k	\mathcal{J}^k	ω^k	$\Delta\mathcal{J}^k$
1	0.000975954	1	–	21	0.000441077	0.0563135	–
2	0.0223241	1.5	+	22	0.000441556	0.0844703	+
3	0.00974474	0.75	+	23	0.000441068	0.0422351	–
4	0.000774439	0.375	–	24	0.000441327	0.0633527	+
5	0.000818477	0.5625	+	25	0.000441064	0.0316764	–
6	0.00051892	0.28125	–	26	0.0004412	0.0475145	+
7	0.00050708	0.421875	–	27	0.000441062	0.0237573	–
8	0.00046856	0.632812	–	28	0.00044113	0.0356359	+
9	0.000883594	0.949219	+	29	0.000441061	0.0178179	–
10	0.000500301	0.474609	+	30	0.000441092	0.0267269	+
11	0.000445635	0.237305	–	31	0.000441061	0.0133635	–
12	0.000458215	0.355957	+	32	0.000441072	0.0200452	+
13	0.000442116	0.177979	–	33	0.00044106	0.0100226	–
14	0.000447673	0.266968	+	34	0.000441062	0.0150339	+
15	0.000441366	0.133484	–	35	0.00044106	0.00751695	–
16	0.000444232	0.200226	+	36	0.000441058	0.0112754	–
17	0.000441163	0.100113	–	37	0.000441096	0.0169131	+
18	0.000442729	0.150169	+	38	0.000441058	0.00845657	+
19	0.000441099	0.0750847	–	39	0.000441058	0.00422828	
20	0.000441968	0.112627	+				

Table 5.4: Flow inversion optimal control case. Values of the objective functional \mathcal{J}^k , the relaxation factor ω^k and sign of the difference $\Delta\mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$ for each optimization step k . A + sign means that the functional has increased from the previous step.

5.3.2 Flow inversion optimal control case

We now consider a full three-dimensional case. The computational domain and the target region are the same as in Section 5.3.1 (see Figure 5.13), as well as the portions of the boundary for fixed (Γ_d) and variable (Γ_c) boundary conditions of the magnetic field lifting function. Also, the integral of the velocity error in the objective functional is again limited to the target region Ω_c . Unlike Section 5.3.1, in this case the cross section of the channel is finite, thus representing a real three-dimensional flow. Hence, the uncontrolled flow is again in the y direction and no-slip boundary conditions are enforced for the velocity on the faces orthogonal to the z and x directions, which represent the channel walls. Concerning the boundary conditions of the Navier-Stokes equations, we enforce an inlet velocity profile driving the flow on the face $y = 0$ as $u_{in} = x(1-x)z(1-z)$, while on the

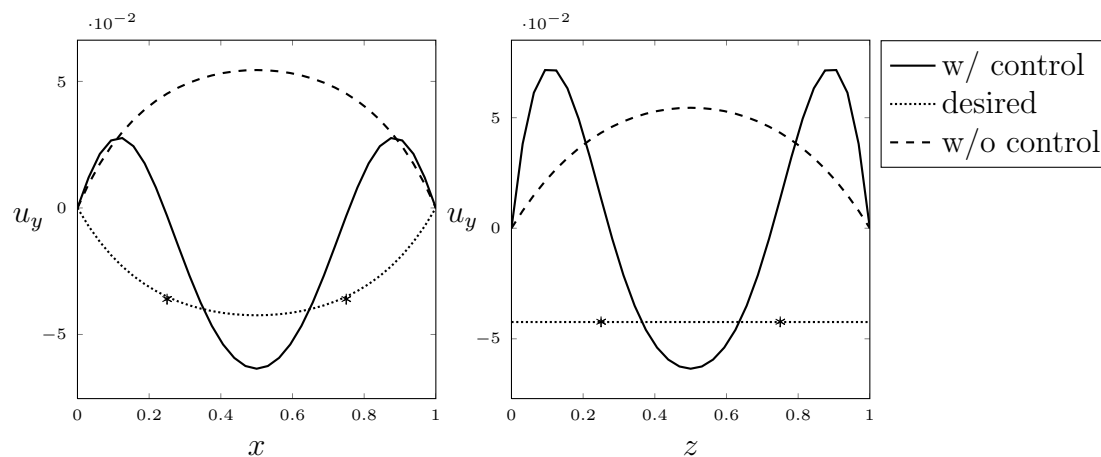


Figure 5.15: Flow inversion optimal control case. Plot along the line $\{x \in [0, 1], y = 1.9, z = 0.5\}$ (left) and the line $\{x = 0.5, y = 1.9, z \in [0, 1]\}$ (right) of the computed optimal velocity (solid), the desired velocity (dashed) and the velocity of the state problem with no control (dotted). The asterisks indicate the limits of the target region Ω_c .

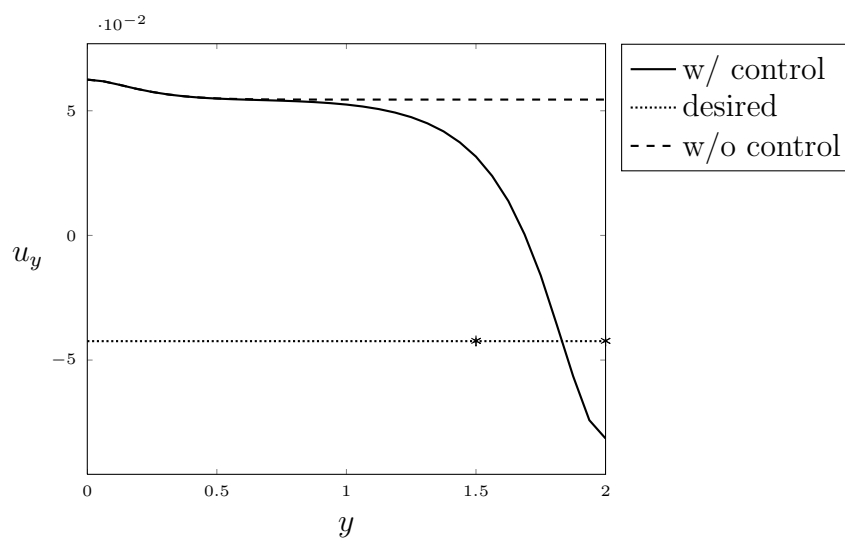


Figure 5.16: Flow inversion optimal control case. Plot along the line $\{x = 0.5, y = [0, 2], z = 0.5\}$ of the computed optimal velocity (solid), the desired velocity (dashed) and the velocity of the state problem without control (dotted). The asterisks indicate the limits of the target region Ω_c .

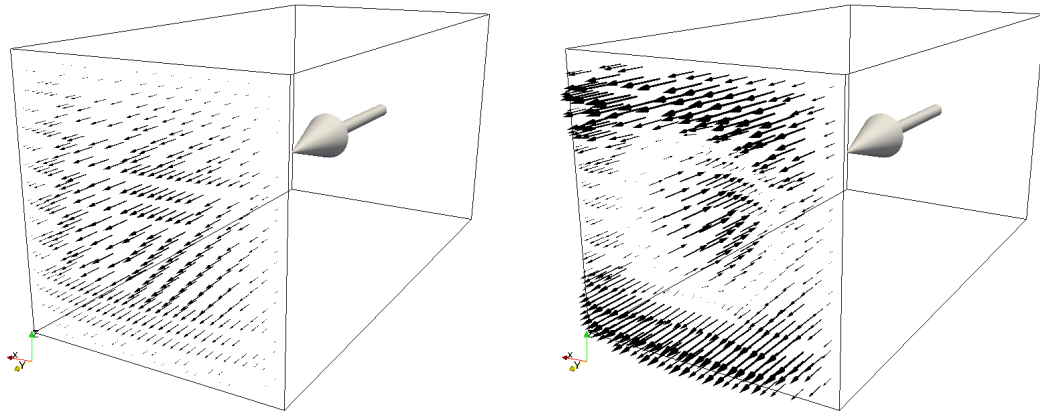


Figure 5.17: Flow inversion optimal control case. The velocity vector field on the outlet face of the MHD channel without control (left) and with control, as computed at the end of the optimization algorithm (right). The gray arrow indicates the prescribed direction of the flow.

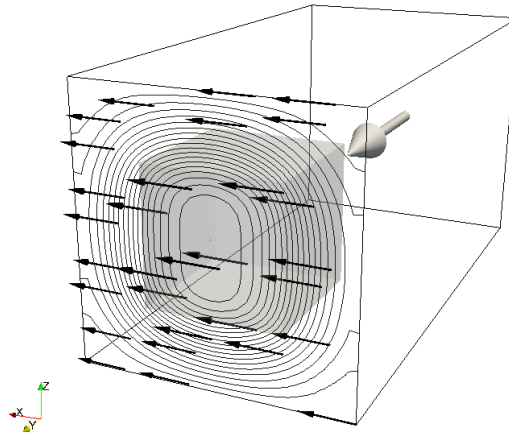


Figure 5.18: Flow inversion optimal control case. Contour lines of the magnitude of the lifting function \mathbf{B}_e , together with the representation of some lifting function vectors on the outlet face of the channel, as computed at the end of the optimization algorithm.

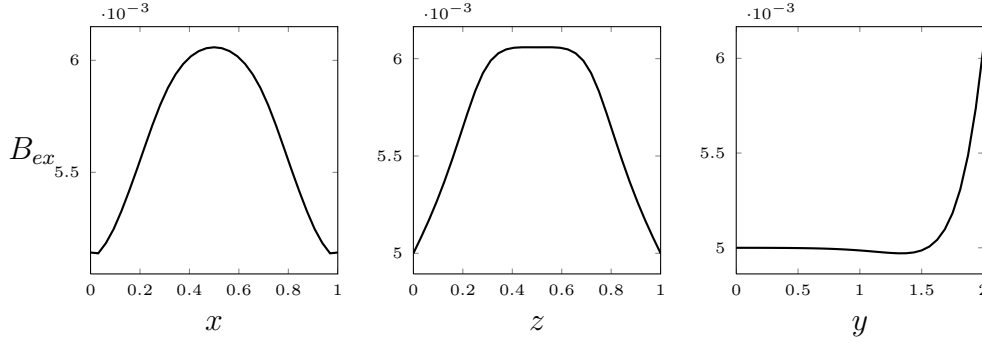


Figure 5.19: Flow inversion optimal control case. Plot along the lines $\{x \in [0, 1], y = 1.9, z = 0.5\}$ (left), $\{x = 0.5, y = 1.9, z \in [0, 1]\}$ (center) and $\{x = 0.5, y = [0, 2], z = 0.5\}$ (right) of the x -component of the magnetic field lifting function \mathbf{B}_e , as a final result of the gradient algorithm.

outlet face $y = 2$ we set a zero pressure outflow condition. Of course, the inlet velocity profile u_{in} is zero on the channel walls, consistently with the enforced no-slip boundary conditions.

For this case we report the results about the numerical solution of the optimality system as obtained with the gradient algorithm of Section 3.4, with the penalty parameters in the objective functional given by $\alpha = 100$, $\beta = 0$ and $\gamma = 1$. As a desired velocity, we choose a longitudinal y component given by $u_{d,y} = -u_{y,Hm}(x)$ and zero for the other components, where $u_{y,Hm}(x)$ denotes again the Hartmann velocity profile given by (5.5). Notice the presence of the minus sign, which means that we aim to obtain a *flow inversion* in the target region Ω_c .

In Table 5.4 we list for each optimization step k the most important values that characterize the gradient algorithm, namely the values of the objective functional \mathcal{J}^k , the relaxation factor ω^k and the sign of the difference of the objective functional values between two subsequent steps, $\Delta\mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$. For this run, the gradient algorithm stopped after 39 optimization steps.

As a result of the final step of the gradient algorithm, in Figure 5.15 we plot along two lines parallel to the x -axis (left) and to the z -axis (right) the computed optimal longitudinal velocity (solid), the desired one (dashed) and the velocity of the state problem without control, i.e., with $\alpha = 0$ (dotted). Figure 5.16 shows a plot of the same quantities along the longitudinal line $\{x = 0.5, y = [0, 2], z = 0.5\}$. It is clear from these figures that the candidate optimal velocity field differs from the solution without control and that it becomes negative in the target region, in order to reach the negative desired profile. Figure 5.17 shows the difference between the velocity vector field on the outlet of the MHD channel in the absence of control (left) and the obtained velocity solution at the end of the optimization loop

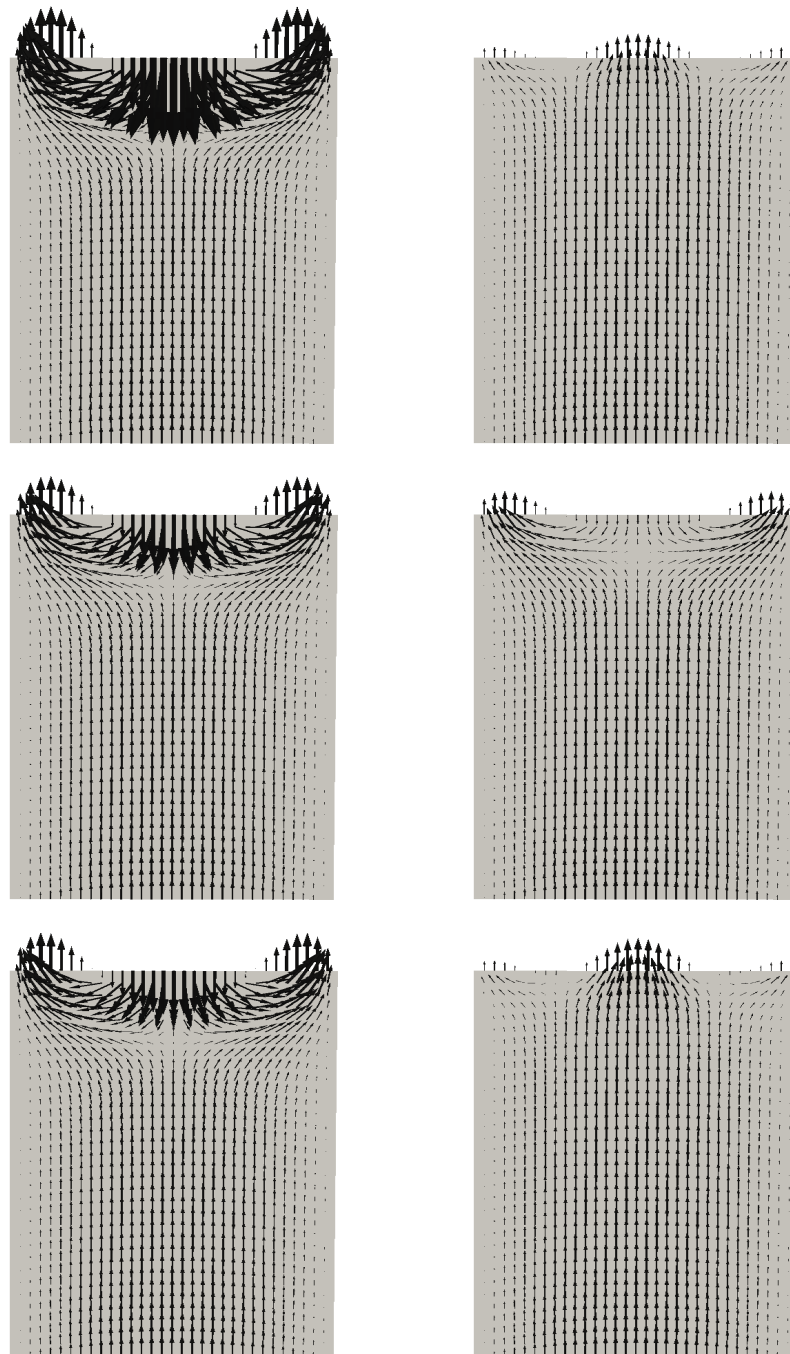


Figure 5.20: Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

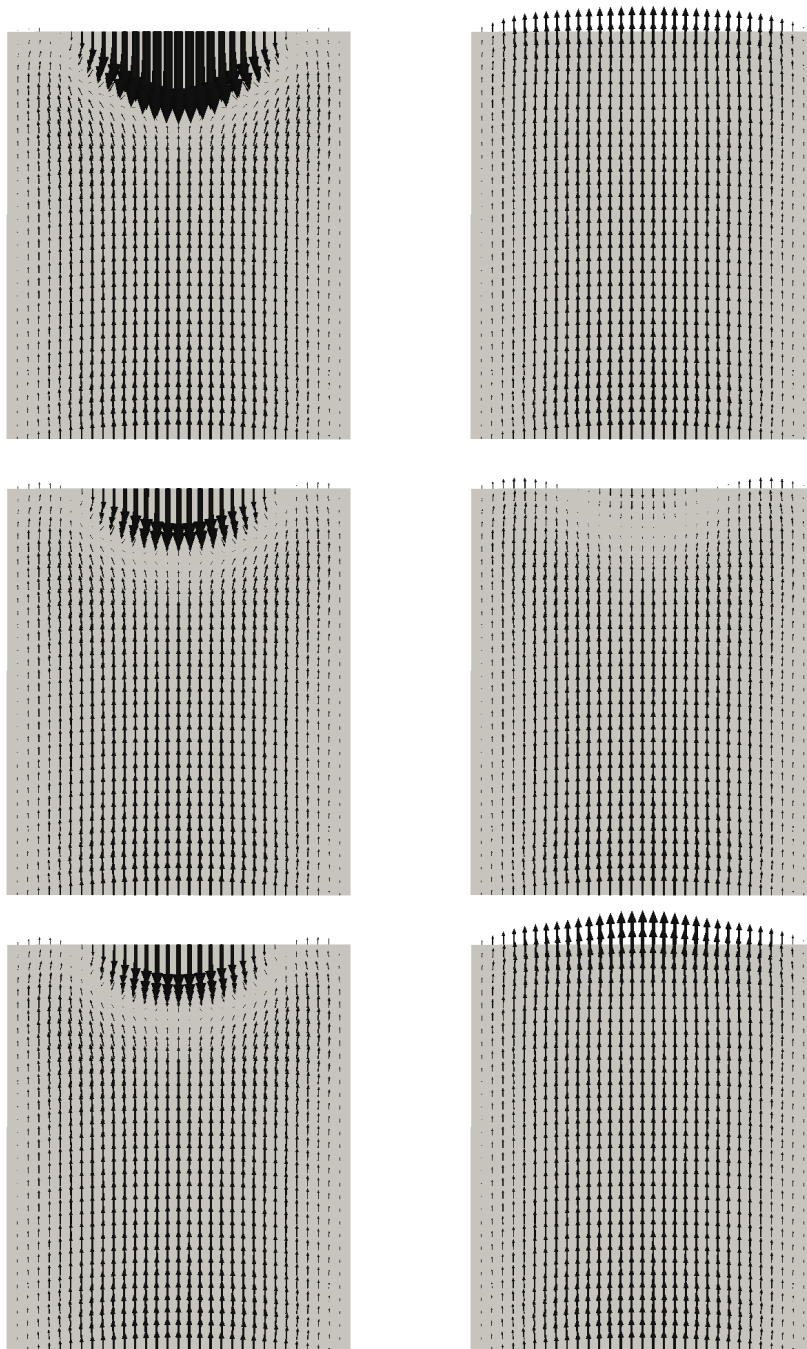


Figure 5.21: Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

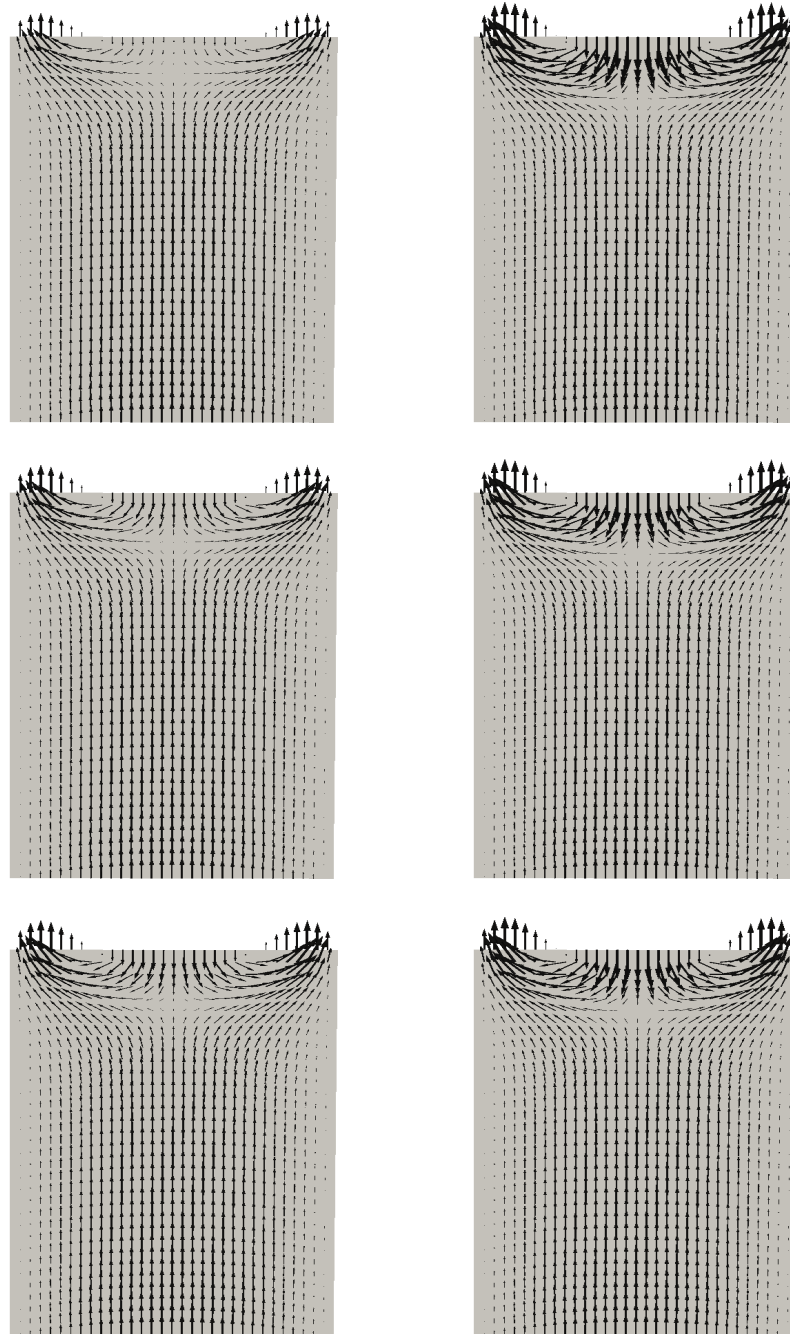


Figure 5.22: Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

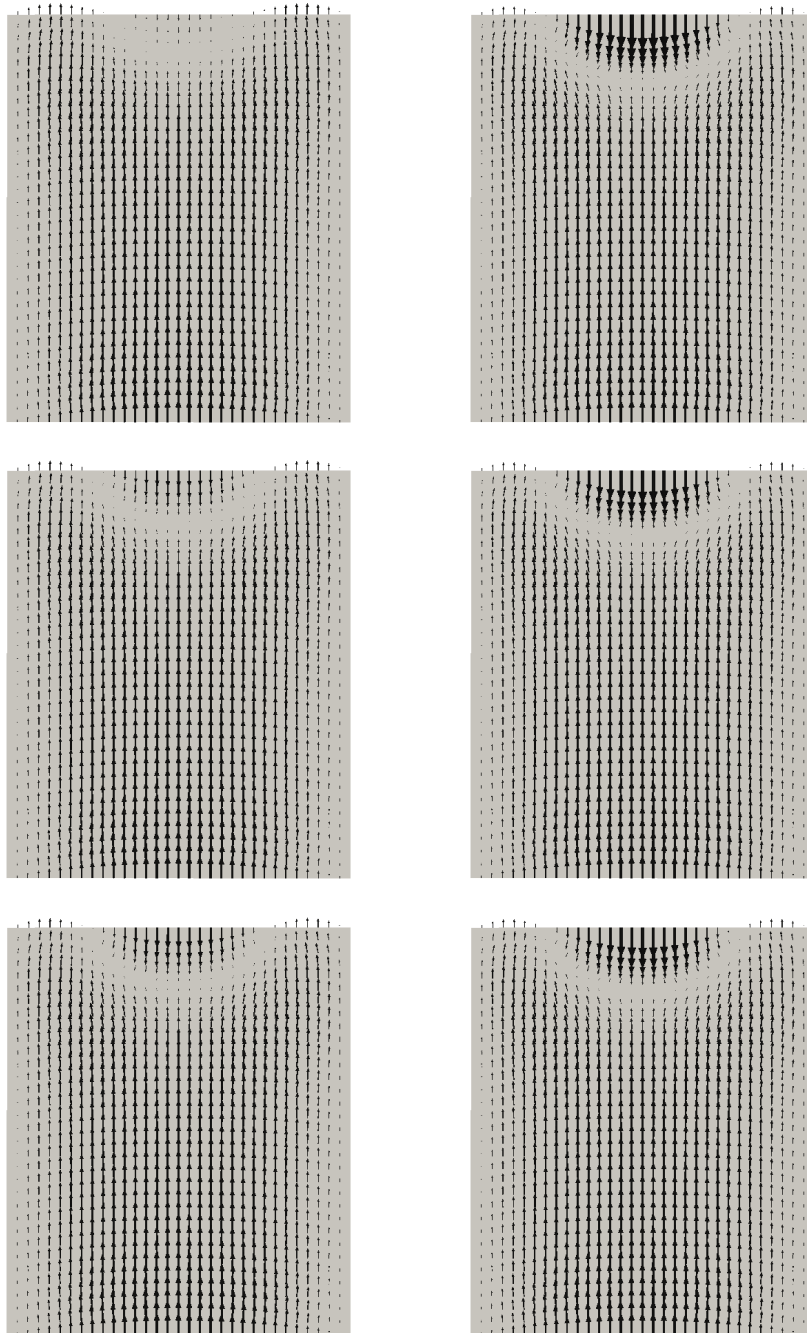


Figure 5.23: Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

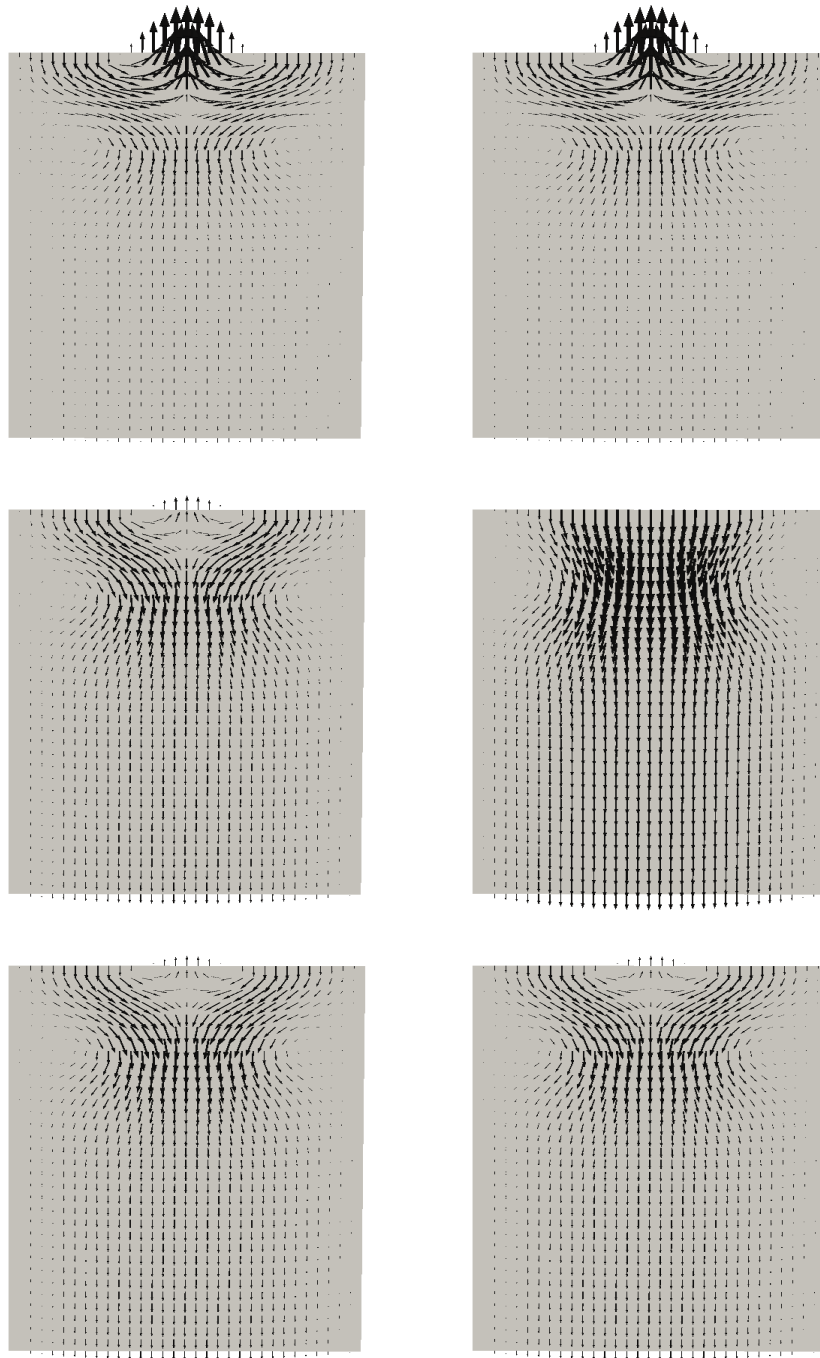


Figure 5.24: Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

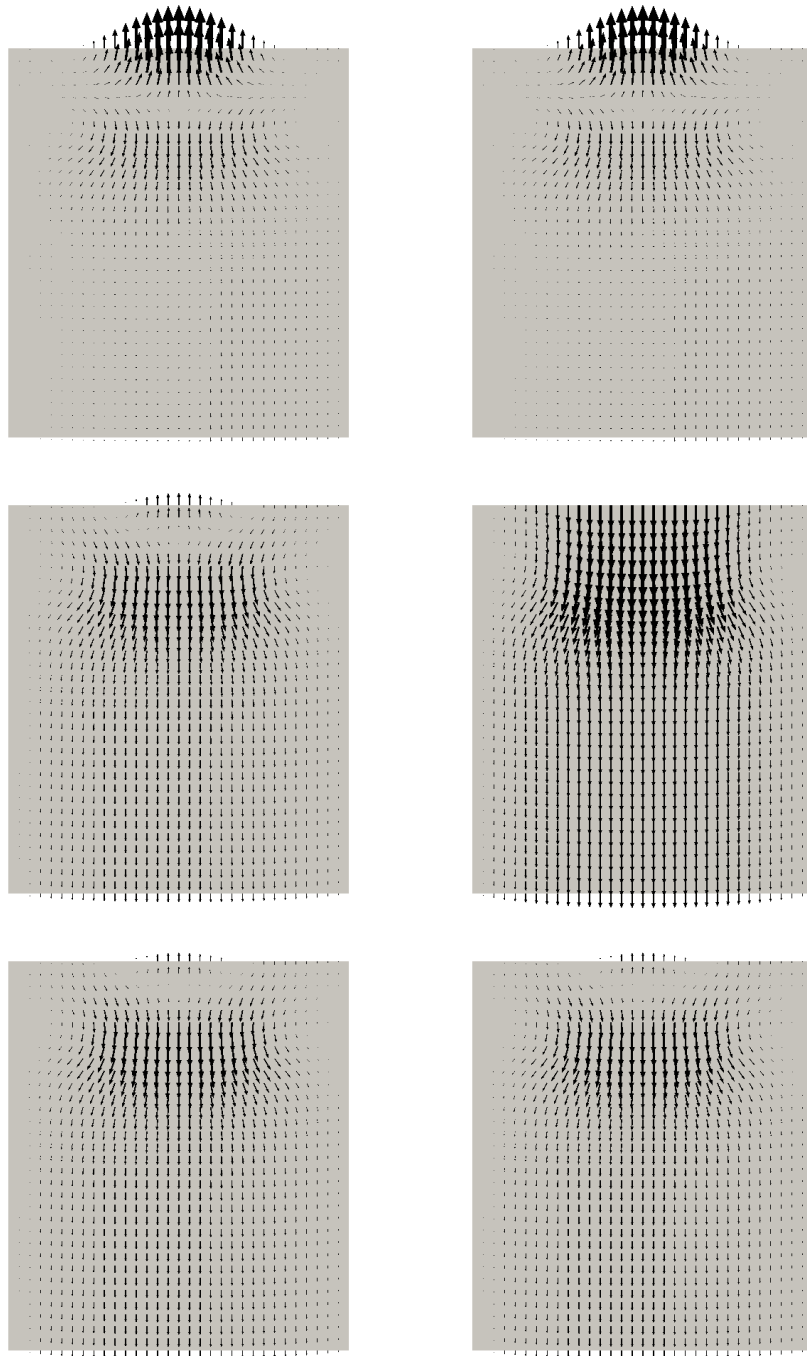


Figure 5.25: Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

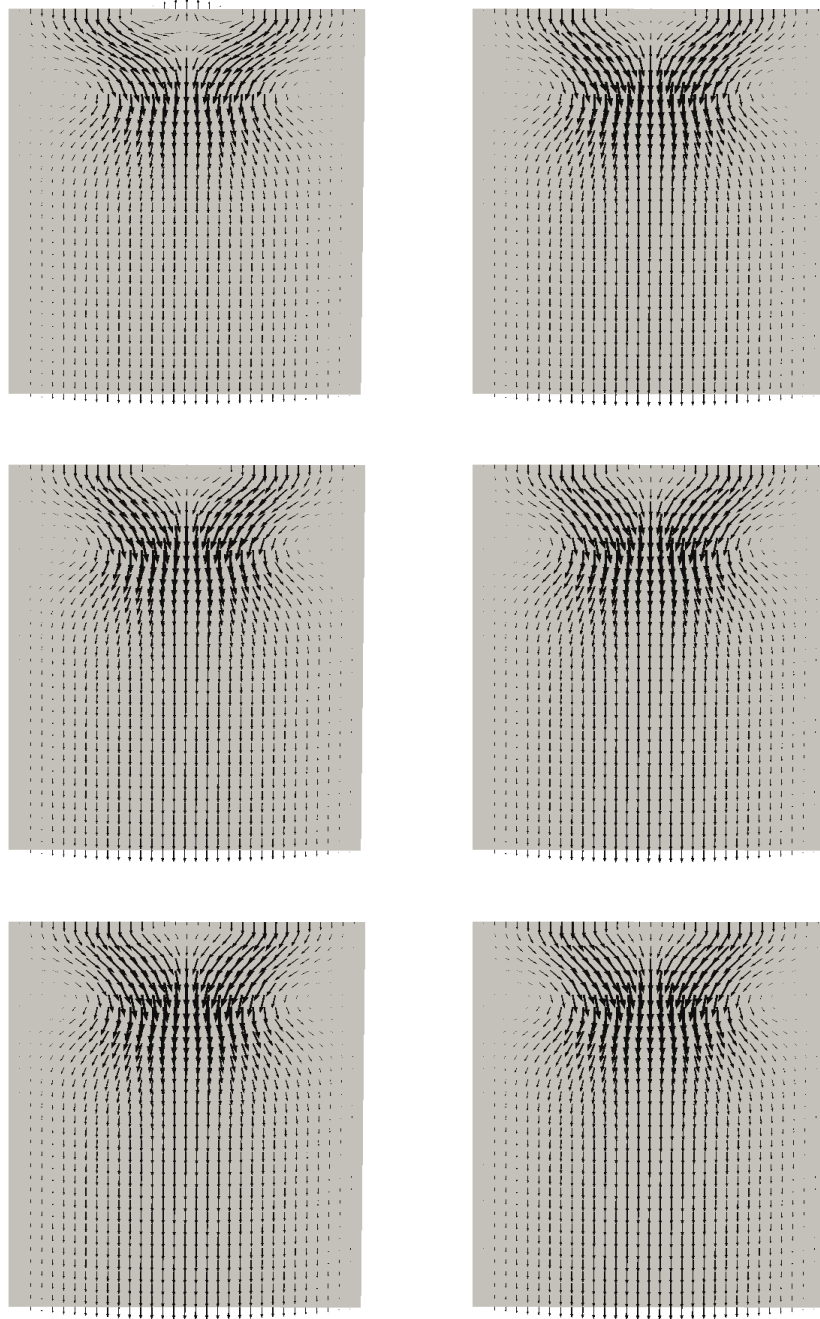


Figure 5.26: Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

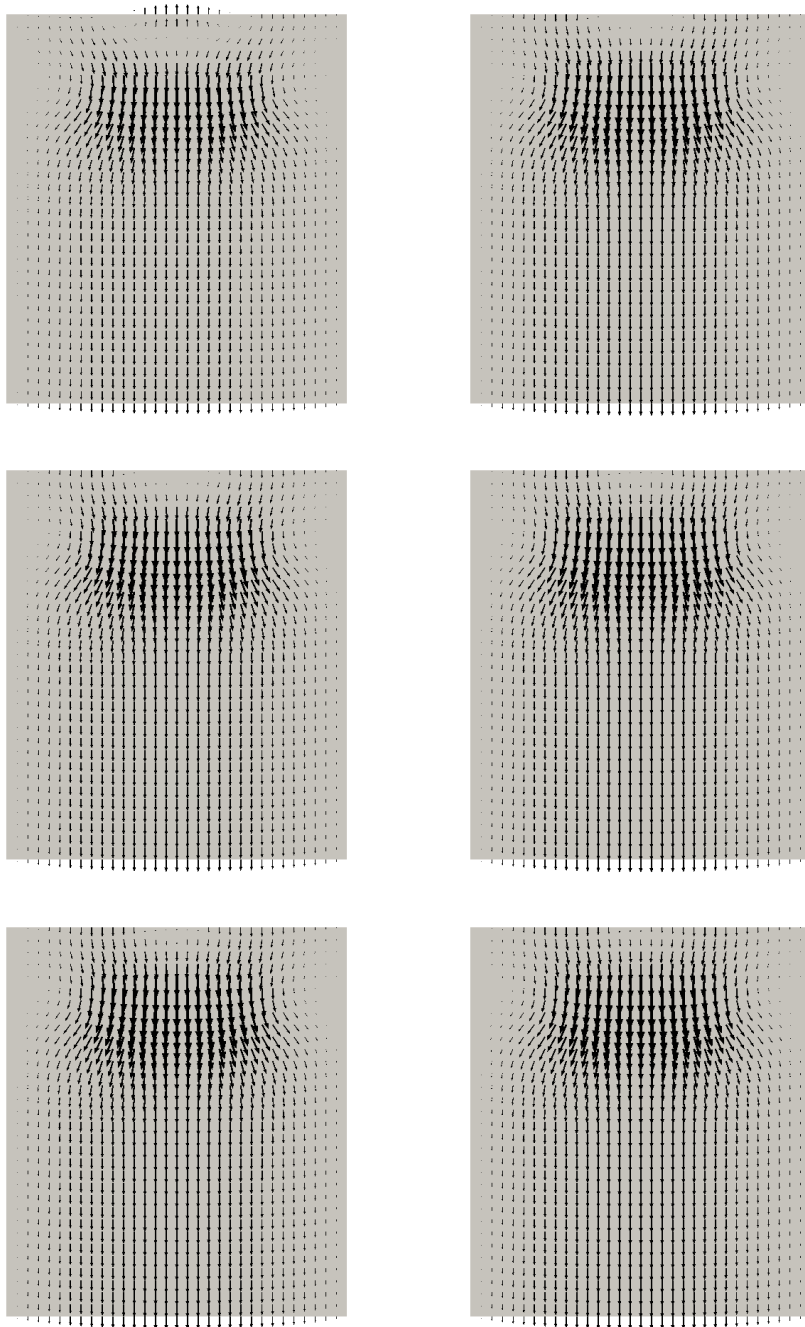


Figure 5.27: Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.

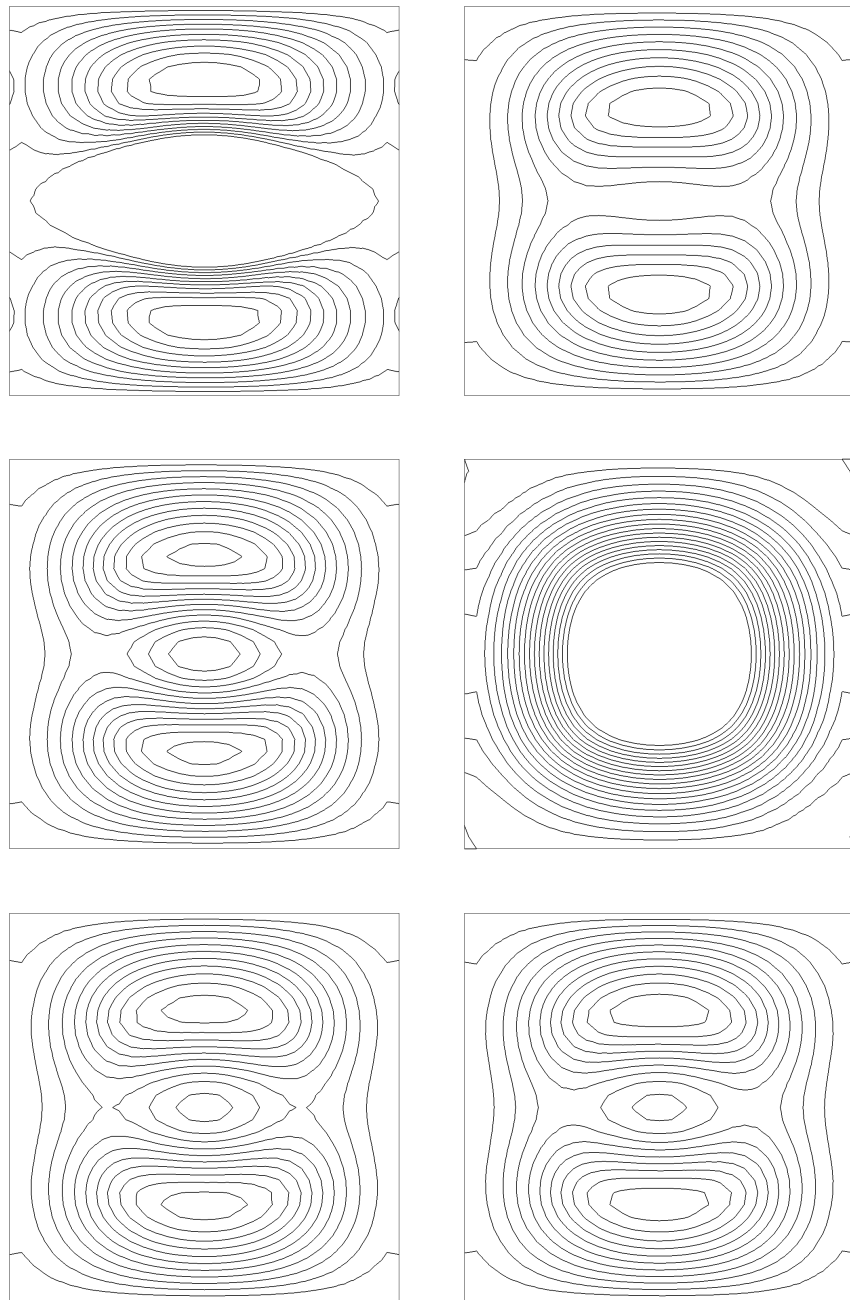


Figure 5.28: Flow inversion optimal control case. Contours of the magnitude of the lifting function B_e on the control surface $y = 2$ for the optimization steps 5-10 (top to bottom, left to right).

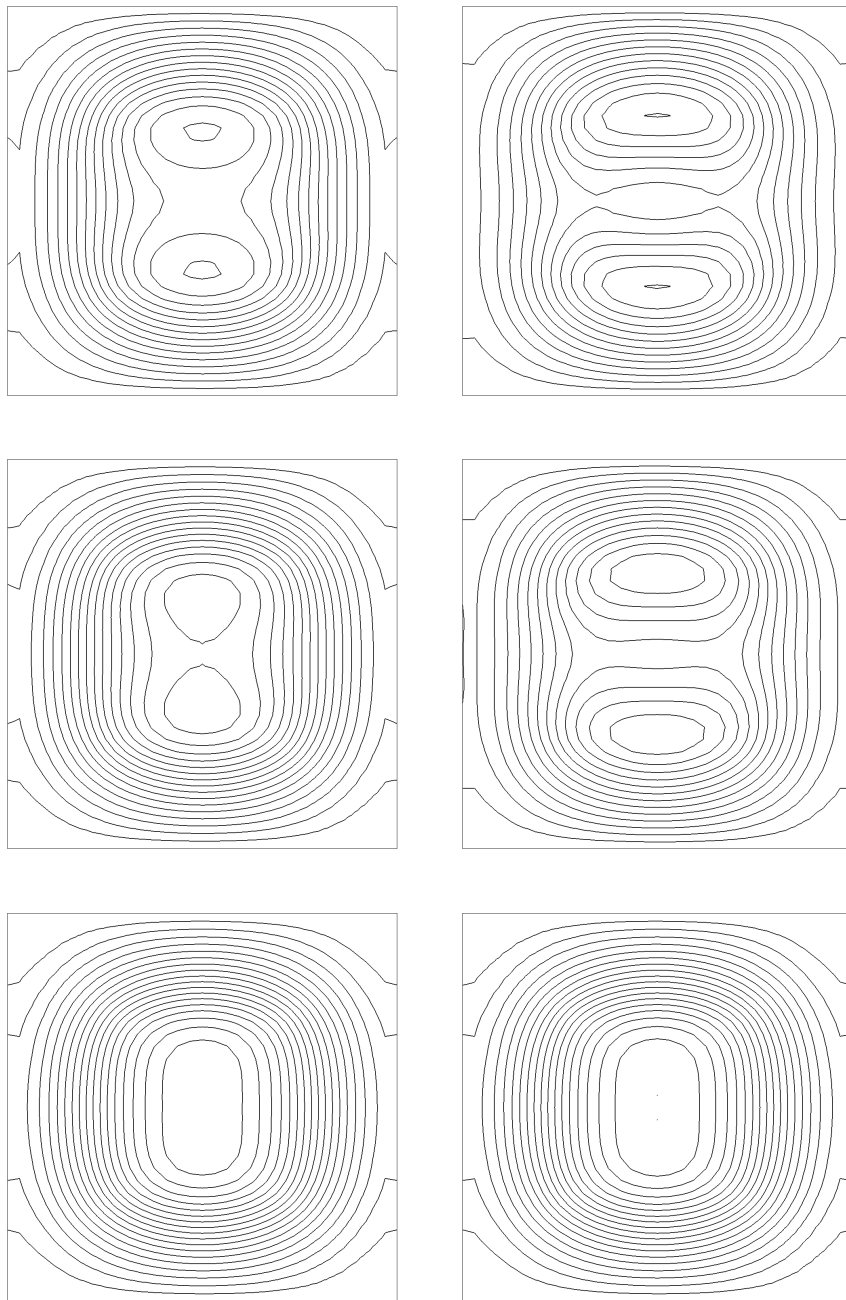


Figure 5.29: Flow inversion optimal control case. Contours of the magnitude of the lifting function B_e on the control surface $y = 2$ for the optimization steps 11-14 and for the last two steps 38 and 39 (top to bottom, left to right).

(right). The inversion of the flow due the magnetic control can be clearly noticed. In Figures 5.18 and 5.19 we show the computed lifting function at the final step of the gradient algorithm. Figure 5.18 shows the contour lines of the magnitude of the lifting function together with a sampling of lifting function vectors. The results of the simulation tell us that the z and y components of the lifting function are negligible with respect to the x component. In Figure 5.19 we also report the plots of B_{ex} along three lines parallel to the x (left), z (center) and y (right) axes, respectively.

In order to show the evolution of the gradient algorithm, in Figures 5.20 and 5.21 we report the velocity vector field for the steps 5-10 of the algorithm on the midplane surfaces $x = 0.5$ and $z = 0.5$, respectively. For the sake of compactness, the domain is scaled by a factor 2 in the y direction. We can clearly observe that during these first steps of the gradient algorithm the velocity field has to be adjusted and therefore it undergoes rapid changes between two adjacent steps. Then, as can be seen in the following Figures 5.22 and 5.23, in the subsequent steps the oscillations are smaller and the algorithm eventually reaches convergence. The evolution of the adjoint velocity field follows closely the corresponding state field, as can be noticed in Figures 5.24, 5.25, 5.26 and 5.27. Finally, in Figures 5.28 and 5.29 we plot the contours of the magnetic field lifting function on the outlet control surface $y = 2$. The restriction of the lifting function to this control surface corresponds to the boundary control solution of the original boundary optimal control problem.

Conclusions

In this thesis a new approach to the boundary optimal control of the incompressible steady MHD equations has been presented. With the introduction of the lifting function for the boundary conditions on the magnetic field, boundary control problems can be formulated as extended distributed problems, bringing theoretical and computational benefits. We have provided a systematic mathematical formulation of the boundary optimal control problem in terms of the minimization of a cost functional constrained by the steady incompressible MHD equations. The existence of a solution to the state equations and to the optimal control problem has been shown. The Lagrange multiplier technique has been used to derive an optimality system, whose solutions are candidate solutions for the optimal control problem. In order to achieve the numerical solution of the optimality system, a finite element approximation has been considered for the discretization together with an appropriate gradient-type algorithm, with the purpose of diminishing the numerical oscillations induced by the decoupling of the equations. A finite element object-oriented library has been developed to obtain a parallel and multigrid computational implementation of the optimality system. Numerical results of both two- and three-dimensional computations have shown that a possible minimum for the optimal control problem can be computed in a robust and accurate manner. Thanks to the generality of the proposed lifting function approach, we expect it to be suitable for the solution of a broad class of boundary optimal control problems constrained by partial differential equations.

List of Figures

3.1	The full multigrid algorithm.	66
4.1	<i>FEMuS</i> directory structure.	74
4.2	External libraries required by FEMuS. The dashed arrow from LibMesh to PETSc indicates that the PETSc libraries are not called by <i>FEMuS</i> through the interface given by LibMesh but they are used directly.	77
4.3	Overview of the <i>EquationsMap</i> class and the classes that compose it.	82
4.4	An example of inheritance diagram with the <i>EqnBase</i> class as the father class.	83
4.5	The subdivision of vectors and matrices as required by PETSc. Vectors must be divided into blocks of contiguous elements, while matrices must be split into blocks of contiguous rows. Each block is assigned to a different processor.	84
4.6	The Gencase class must take into account both the subdomain and the level subdivisions of the mesh geometrical entities (elements and nodes).	89
4.7	Inheritance diagram for the <i>FEElemBase</i> class.	91
4.8	Basic structure of the files of a multiphysics application.	94
4.9	The equations and quantities for the optimality system (some quantities are not reported here for the sake of conciseness).	103
5.1	Lifting function verification test. Domain and boundary conditions for the Hartmann flow.	106
5.2	Lifting function verification test. The magnetic fields b_x, B_{ex}, B_x along the line $x = 0.5$ (left column) and the magnetic fields b_x, B_{ex}, B_y along the line $y = 0.5$ (right column) for $K_1 = 0$ (A), 1 (B), 10 (C) with divergence-free magnetic field lifting functions.	107
5.3	Lifting function verification test. The velocity components u_x (left) and u_y (right) along the line $y = 0.5$ for $K_1 = 0$ (A), 1 (B), 10 (C) with divergence-free magnetic field lifting functions.	108

5.4	Lifting function verification test. The x -components b_x, B_{ex}, B_x (left column) and the y -components b_y, B_{ey}, B_y (right column) along the line $y = 0.5$ for $K_2 = 0$ (A), 1 (B) and 10 (C) with non divergence-free magnetic field lifting functions.	109
5.5	Lifting function verification test. The y -component u_y of the velocity field (left) and the pressure p (right) along the line $y = 0.5$ for $K_2 = 0$ (A), 1 (B) and 10 (C) with non divergence-free magnetic field lifting functions.	110
5.6	Two-dimensional optimal control case. Computational domain Ω with target velocity region Ω_1 (left) and boundary conditions for the state variables of the optimality system (right).	112
5.7	Two-dimensional optimal control case. Boundary conditions for the adjoint variables (left) and the control variable (right) of the optimality system.	112
5.8	Two-dimensional optimal control case. State, adjoint and control variables of the optimality system along the line $y = 0.95$ with $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 10^5$	113
5.9	Two-dimensional optimal control case. State, adjoint and control variables of the optimality system along the line $x = 0.5$ with $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 10^5$	114
5.10	Two-dimensional optimal control case. Above: longitudinal velocity profiles along the line $y = 0.95$ with $\bar{\mathbf{B}}_e = (2, 0)$ and $\alpha = 0, 10^3, 10^4, 10^5$ and comparison with the target. Below: two views of the y component of the velocity field for the case $\alpha = 10^5$. The grey curved surface represents the velocity profile that would be achieved in the absence of magnetic field control ($\alpha = 0$).	115
5.11	Two-dimensional optimal control case. Boundary control as restriction of the lifting function \mathbf{B}_e on the side $y = 1$, for the penalty values $\alpha = 0, 10^3, 10^4, 10^5$	116
5.12	Two-dimensional optimal control case. Plots of the y component of the velocity (top row), the y component of the adjoint velocity (middle row) and the x component of the magnetic field lifting function (bottom row) over the domain Ω . The arrow indicates the direction of the flow.	117
5.13	Hartmann flow optimal control case. Computational domain Ω and target region Ω_c (highlighted in gray). The arrow indicates the direction of the flow.	119
5.14	Hartmann flow optimal control case. Values of the relaxation parameter ω^k and of the objective functional \mathcal{J}^k for $\alpha = 200, 400, 600$ as a function of the optimization steps.	120

- 5.15 Flow inversion optimal control case. Plot along the line $\{x \in [0, 1], y = 1.9, z = 0.5\}$ (left) and the line $\{x = 0.5, y = 1.9, z \in [0, 1]\}$ (right) of the computed optimal velocity (solid), the desired velocity (dashed) and the velocity of the state problem with no control (dotted). The asterisks indicate the limits of the target region Ω_c 123
- 5.16 Flow inversion optimal control case. Plot along the line $\{x = 0.5, y = [0, 2], z = 0.5\}$ of the computed optimal velocity (solid), the desired velocity (dashed) and the velocity of the state problem without control (dotted). The asterisks indicate the limits of the target region Ω_c 123
- 5.17 Flow inversion optimal control case. The velocity vector field on the outlet face of the MHD channel without control (left) and with control, as computed at the end of the optimization algorithm (right). The gray arrow indicates the prescribed direction of the flow. . . . 124
- 5.18 Flow inversion optimal control case. Contour lines of the magnitude of the lifting function \mathbf{B}_e , together with the representation of some lifting function vectors on the outlet face of the channel, as computed at the end of the optimization algorithm. 124
- 5.19 Flow inversion optimal control case. Plot along the lines $\{x \in [0, 1], y = 1.9, z = 0.5\}$ (left), $\{x = 0.5, y = 1.9, z \in [0, 1]\}$ (center) and $\{x = 0.5, y = [0, 2], z = 0.5\}$ (right) of the x -component of the magnetic field lifting function \mathbf{B}_e , as a final result of the gradient algorithm. 125
- 5.20 Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction. 126
- 5.21 Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction. 127
- 5.22 Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction. . . 128
- 5.23 Flow inversion optimal control case. Velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction. . . 129

5.24	Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.	130
5.25	Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 5-10 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.	131
5.26	Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $x = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.	132
5.27	Flow inversion optimal control case. Adjoint velocity vector field on the longitudinal midplane surface $z = 0.5$ for the optimization steps 11-14 and the last two steps 38 and 39 (top to bottom, left to right). The domain is scaled by a factor 2 in the longitudinal direction.	133
5.28	Flow inversion optimal control case. Contours of the magnitude of the lifting function \mathbf{B}_e on the control surface $y = 2$ for the optimization steps 5-10 (top to bottom, left to right).	134
5.29	Flow inversion optimal control case. Contours of the magnitude of the lifting function \mathbf{B}_e on the control surface $y = 2$ for the optimization steps 11-14 and for the last two steps 38 and 39 (top to bottom, left to right).	135

List of Tables

4.1	Some options in the configuration header files.	95
4.2	Some configuration parameters in the <i>femus_conf.in</i> file.	96
5.1	Two-dimensional optimal control case. Some values of the velocity error over the target region, $F_0 = \int_{\Omega_1} \ \mathbf{u} - \mathbf{u}_d\ ^2 d\mathbf{x}$, for different values of α	115
5.2	Hartmann flow optimal control case. Values of the objective functional \mathcal{J}^k , the relaxation factor ω^k and sign of the difference $\Delta\mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$ for each optimization step k for $\alpha = 200$ and $u_{d,y} = 1.4 u_{y,Hm}(x)$. A + sign means that the functional has increased from the previous step.	119
5.3	Hartmann flow optimal control case. Number of optimization steps k_{end} and final value of the cost functional $\mathcal{J}^{k_{end}}$ for increasing α values.	120
5.4	Flow inversion optimal control case. Values of the objective functional \mathcal{J}^k , the relaxation factor ω^k and sign of the difference $\Delta\mathcal{J}^k = \mathcal{J}^k - \mathcal{J}^{k-1}$ for each optimization step k . A + sign means that the functional has increased from the previous step.	122

Bibliography

- [1] R. ADAMS, *Sobolev Spaces*, Academic Press, New York, 1975.
- [2] G. V. ALEKSEEV AND A. B. SMISHLIAEV, *Solvability of the boundary-value problems for the Boussinesq equations with inhomogeneous boundary conditions*, *Journal of Mathematical Fluid Mechanics*, 3, pp. 18–39.
- [3] G. BORNIA, A. CERVONE, AND S. MANSERVISI, *Optimal control for incompressible steady MHD flows via constrained extended boundary approach*, in *Proceedings of the V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, Lisbon, 2010.
- [4] S. C. BRENNER AND L. R. SCOTT, *The mathematical theory of finite element methods*, Springer-Verlag, 1994.
- [5] F. BREZZI AND M. FORTIN, *Mixed and hybrid finite element methods*, Springer-Verlag, 1991.
- [6] F. BREZZI, J. RAPPAZ, AND P. RAVIART, *Finite dimensional approximation of nonlinear problems*, *Numerische Mathematik*, 36 (1980), pp. 1–25.
- [7] R. BRIZITSKII AND D. TERESHKO, *On the solvability of boundary value problems for the stationary magnetohydrodynamic equations with inhomogeneous mixed boundary conditions*, *Differential Equations*, 43, pp. 246–258.
- [8] P. CIARLET, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
- [9] P. A. DAVIDSON, *An Introduction to Magnetohydrodynamics*, Cambridge University Press, Cambridge, UK, 2001.
- [10] V. C. A. FERRARO AND C. PLUMPTON, *An introduction to magneto-fluid mechanics*, Oxford University Press, Oxford, 1966.

- [11] J. F. GERBEAU, *A stabilized finite element method for the incompressible magnetohydrodynamic equations*, Numerische Mathematik, 87 (2000), pp. 83–111.
- [12] V. GIRAULT AND P. RAVIART, *The Finite Element Approximation of the Navier-Stokes Equations*, Springer-Verlag, Berlin, 1979.
- [13] —, *The Finite Element Method for Navier-Stokes Equations: Theory and Algorithms*, Springer-Verlag, New York, 1986.
- [14] R. GRIESSE AND K. KUNISCH, *Optimal control for a stationary MHD system in velocity-current formulation*, Siam J. Control Optim., 45 (2006), pp. 1822–1845.
- [15] M. D. GUNZBURGER, *Perspectives in flow control and optimization*, SIAM, Philadelphia, 2003.
- [16] M. D. GUNZBURGER AND L. S. HOU, *Finite-dimensional approximation of a class of constrained nonlinear optimal control problems*, Siam J. Control and Optimization, 34 (1996), pp. 1001–1043.
- [17] M. D. GUNZBURGER AND L. S. HOU, *Finite-dimensional approximation of a class of constrained nonlinear optimal control problems*, SIAM journal on control and optimization, 34 (1996), p. 1001.
- [18] M. D. GUNZBURGER AND S. MANSERVISI, *Analysis and approximation of the velocity tracking problem for Navier-Stokes flows with distributed control*, SIAM J. Numer. Anal., 37 (2000), pp. 1481–1512.
- [19] —, *The velocity tracking problem for Navier-Stokes flow with boundary control*, SIAM Journal on Control and Optimization, 39 (2000), pp. 594–634.
- [20] M. D. GUNZBURGER, A. J. MEIR, AND J. S. PETERSON, *On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible Magnetohydrodynamics*, Mathematics of Computations, 56 (1991), pp. 523–563.
- [21] M. D. GUNZBURGER AND C. TRENCHIA, *Analysis and discretization of an optimal control problem for the time-periodic MHD equations*, Journal of mathematical analysis and applications, 308 (2005), pp. 440–466.
- [22] M. D. GUNZBURGER AND C. TRENCHIA, *Analysis of an optimal control problem for the three-dimensional coupled modified Navier-Stokes and Maxwell equations*, J. Math. Anal. Appl., 333 (2007), pp. 295–310.

- [23] U. HASLER, A. SCHNEEBELI, AND D. SCHÖTZAU, *Mixed finite element approximation of incompressible MHD problems based on weighted regularization*, Applied Numerical Mathematics, 51 (2004), pp. 19 – 45.
- [24] L. S. HOU AND A. J. MEIR, *Boundary optimal control of MHD flows*, Appl. Math. Optim, 32 (1995), pp. 143–162.
- [25] L. S. HOU AND J. S. PETERSON, *Boundary optimal control for an electrically conducting fluid using boundary electrical potential controls*, Nonlinear Analysis, Theory, Methods and Applications, 24 (1995), pp. 857–874.
- [26] L. S. HOU AND S. S. RAVINDRAN, *Computations of boundary optimal control problems for an electrically conducting fluid*, J. Comput. Phys., 128 (1996), pp. 319–330.
- [27] J. JACKSON, *Classical electrodynamics*, Wiley, New York, 1975.
- [28] B. S. KIRK, J. W. PETERSON, R. H. STOGNER, AND G. F. CAREY, *LibMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations*, Engineering with Computers, 22 (2006), pp. 237–254.
- [29] L. LANDAU AND E. LIFCHITZ, *Électrodynamique des milieux continus*, MIR, Moscow, 1969.
- [30] A. J. MEIR AND P. G. SCHMIDT, *Variational methods for stationary MHD flow under natural interface conditions*, Nonlinear Analysis, Theory, Methods and Applications, 26 (1996), pp. 659–689.
- [31] ———, *Analysis and numerical approximation of a stationary MHD flow problem with nonideal boundary*, SIAM J. Numer. Anal., 36 (1999), pp. 1304–1332.
- [32] S. S. RAVINDRAN, *Real-time computational algorithm for optimal control of an MHD flow system*, SIAM J. Sci. Comput., 26 (2005), pp. 1369–1388.
- [33] P. H. ROBERTS, *An introduction to Magnetohydrodynamics*, Longmans, London, 1967.
- [34] D. SCHÖTZAU, *Mixed finite element methods for stationary incompressible magneto-hydrodynamics*, Numer. Math., 96 (2004), pp. 771–800.
- [35] R. TEMAM, *Navier-Stokes Equations*, North-Holland, Amsterdam, 1979.
- [36] V. M. TIKHOMIROV, *Fundamental Principles of the Theory of Extremal Problems*, Wiley, Chichester, 1982.

-
- [37] U. TROTTEBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Pr, 2001.
 - [38] P. WESSELING, *Introduction to multigrid methods*, Institute For Computer Applications In Science And Engineering Hampton Va, 1995.
 - [39] M. WIEDMER, *Finite element approximation for equations of Magnetohydrodynamics*, Mathematics of Computation, 69 (1999), pp. 83–101.
 - [40] K. YOSIDA, *Functional analysis*, Springer–Verlag, Berlin, 1980.